Udacity Data Analyst Nanodegree

Project No. 2: OpenStreetMap Project
Data Wrangling with MongoDB

Min Lai

Map Area: New Orleans, LA, United States

Source OSM XML was downloaded from
https://mapzen.com/metro-extracts

Zip file size: 69.3MB
Unzipped OSM XML file size: 1073.8MB

1. Problems Encountered in the Map

The unzipped OSM XML file is quite large which is over 1G. I split the file into a set of smaller files so that I can open a file in a text edit to have a flavor of the dataset. Then I coded several audit program to audit different data elements, in audit process, I found problems in following fields:

1.1 Lack of Address information

I count the total number of record got inserted into Mongo DB and the number of records actually has address:

Record Total	5,678,593	
Record having Address	89155	

So this tells me that this dataset collects very small portion of addresses in New Orleans metropolitan area. Address is the one of the basic information for a MAP especially for search purpose. I am surprised that so few addresses were collected, which make analysis based on address related information not very reliable.

1.2 **postal code malformed and out of geographic range**

First issue that I noticed is postal code format. Some postal code entered like 'LA 70116', which looks like a user entry error. And I googled New Orleans postal codes

which normally begin with “701”, and found lots of postal code in this file is not really New Orleans postal code. Some of them are Mississippi postal codes not even Louisiana postal codes. Here is list of zip code out of range:

['70357', '70002', '39576', '70403', '70039', '39501', '70433', '39520', '39560', '70471',
 '70083', '70001', '39571', '70460', '70458', '70053', '70394', '70454', '70421', '70062',
 '70032', '70043', '70401', '70448', '39525', '39503', '70135', '39556', '70446', '70012',
 '39574', '39466', '39573', '39572', '70037', '70354', '70003', '70345']

It looks like this dataset includes lots of nodes beyond in New Orleans metropolitan area. But it may not be a problem in data cleaning, since those address can be easily filtered by query after data being loaded to Mongo DB.

To deal with malformed issue, I create a method to cleaning the data

1.3 Street type has different abbreviations and misspelled type

For the same street type, the data set has different abbreviations to present it. Here is some examples:

Street Type	**Abbreviations in file**
Street	ST, St, st, St.
Road	Rd, rd, Rd., RD
Parkway	Pky, Pkwy, PKWY, PKY

Some of them are misspelled. For example: Circle is misspelled as Cirlce

I enhanced code in problem set 6 to map different abbreviations to the same street type name and correct the misspelled street type name.

1.4 Phone number in various format

I found following formats (not a complete list, I need more work on the regular express to programmatically find all pattern):

1-504-838-3800
504-866-0984
+1 504 4881946
(504) 595-3101
800- 535-9603
(504)3469783

+1 (504) 392 4562

I create a method to format all phone number to unified format xxx- xxx-xxxx like 504-866-0984.

2. Data Overview

File sizes

new_orleans_louisiana.osm..... 1073.8MB MB
new_orleans_louisiana.osm.json .... 1701.2 MB

Number of documents

```
> db.neworleans.count()
5678593
```

Count for all node type

```
> > db.neworleans.aggregate([
...                 {"$group" : {"_id" :  "$type", "count" : {"$sum" : 1}}},
...                 {"$sort" : {"count" : -1}}
...                 ])
{ "_id" : "node", "count" : 5403036 }
{ "_id" : "way", "count" : 275557 }
```

Document Type	Document Count
node	5403036
way	275557

Top 5 amenity

```
> db.neworleans.aggregate([ {"$match":{"amenity":{"$exists":1}}},
...                 {"$group" : {"_id" :  "$amenity", "count" : {"$sum" : 1}}},
...                 {"$sort" : {"count" : -1}},
...                 {"$limit" : 5}])
{ "_id" : "place_of_worship", "count" : 1120 }
{ "_id" : "school", "count" : 850 }
{ "_id" : "parking", "count" : 287 }
```

```
{ "_id" : "restaurant", "count" : 192 }
{ "_id" : "grave_yard", "count" : 188 }
```

Surprisingly, Place of worship is No.1 instead of restaurant. I lived in New Orleans for one year. I saw restaurant everywhere, especially in French Quarter. I guess lots of restaurant are not added to this dataset. From following query, I found that total number of documents having amenity is only 3793, which tells me that lot of businesses and other facilities are not entered into this dataset.

```
db.neworleans.find({"amenity":{"$exists":1}}).count()
```

Top five contributor

```
> db.neworleans.aggregate([
...                 {"$group" : {"_id" :  "$created.user", "count" : {"$sum"
 : 1}}},
...                 {"$sort" : {"count" : -1}},
...                 {"$limit" : 5}])
{ "_id" : "Matt Toups", "count" : 3264420 }
{ "_id" : "ELadner", "count" : 1012422 }
{ "_id" : "wvdp", "count" : 525318 }
{ "_id" : "woodpeck_fixbot", "count" : 254068 }
{ "_id" : "ELadnerImp", "count" : 252367 }
```

3. Additional Ideas

Contribution community is very small and not very active

521 users made contribution to this dataset, while top 5 contributors entered more than 93% of the documents in this data set. So most of the data were entered by very small number of users. For an open source map, more contributors will make map data richer. The project need to find a way to attract more contributors. And 10 latest entries to this data set were made about 2 weeks before I downloaded the file(I download the file on 02/28/15). So it looks like that people are still contributing to this dataset but not very actively.

Last 10 entry time

```
db.neworleans.aggregate([
...                 {"$group" : {"_id" :  "$created.user", "last_update_time" : {"$max" :
"$created.timestamp"}}},
...                 {"$sort" : {"last_update_time" : -1}},
...                 {"$limit" : 10}])


{ "_id" : "PaintManNO", "last_update_time" : "2015-02-14T12:49:19Z" }
{ "_id" : "abel801", "last_update_time" : "2015-02-13T22:03:04Z" }
{ "_id" : "ELadner", "last_update_time" : "2015-02-13T21:28:45Z" }
```

{ "_id" : "Matt Toups_nolaimport", "last_update_time" : "2015-02-12T23:51:29Z" }
{ "_id" : "Middendorf's Restaurant", "last_update_time" : "2015-02-12T17:52:17Z" }
{ "_id" : "Luis36995", "last_update_time" : "2015-02-11T16:40:51Z" }
{ "_id" : "wvdp", "last_update_time" : "2015-02-11T16:21:03Z" }
{ "_id" : "Pnrrth", "last_update_time" : "2015-02-10T16:06:31Z" }
{ "_id" : "samely", "last_update_time" : "2015-02-09T15:21:53Z" }
{ "_id" : "thevirginian", "last_update_time" : "2015-02-06T20:07:06Z" }

Audit consistence of OSM dataset

Open street map is an open source project, anyone can contribute it. How to make sure the data entered by different user are actually consistent with each other and dataset consistently have correct mapping of buildings, streets, highways etc. Maybe, randomly select some data points like some address, to cross check with other map system like Google map. Google map provides several APIs which can be leverage for this purpose.

Conclusion

Obviously, incompleteness is the biggest problem here in New Orleans open street map dataset, which can't be addressed in this project. However for practice purpose, it is not really an issue since basic analysis can still be done. For consistency, in this project, I didn't explore too much on it, but I think there may be a way to implemented by using Google map API. I also audited the tag types in the osm file, and didn't find any polemic tag. For the problems that list in section 2, I tried to clean the data to improve accuracy by correcting wrong zip code, uniformity by unifying the format of phone number, street type name. From the cleaning procedure, to improve data quality, open street map project needs to have better data quality control by having clearer definition on some data element like street type name, format of phone number.