OpenStreetMap Data Case Study- Data Wrangling with MongoDb

Study Area:

Releigh , NC , United States

I selected Raleigh for this case study data wrangling with mongodb to investigate the openstreetmap data and audit, clean and finally store to MongoDb .

Problems Encountered in the Map

Street Names

Most of the problem in this data set is name inconsistency of street addresses. They street names contains inconsistent abbreviations as listed below. That is different short names are used for the same street type.

```
'Ave'  : 'Avenue',
  'Blvd' : 'Boulevard',
  'Dr'   : 'Drive',
  'Dr.'  : 'Drive',
  'Blvd.': 'Boulevard',
  'Ln'   : 'Lane',
  'Pkwy' : 'Parkway',
  'Pky'  : 'Parkway',
  'Rd'   : 'Road',
  'Rd.'   : 'Road',
  'St'   : 'Street',
  'St,'  : 'Street',
  'street' :"Street",
  'Ct'   : "Court",
  'Cir'  : "Circle",
  'Cr'   : "Court",
  'ave'  : 'Avenue',
  'Hwg'  : 'Highway',
  'Hwy'  : 'Highway',
  'Sq'   : "Square"
```

A script written that uses regular expression to match the very last name of the street name as most street names end with street types like Avenue, Street, Road etc. That identifies street types that don’t have common name and updates the street names to the most commonly used street names.

Zip - Code

Like problems encountered in street naming, zipcodes also experience similar problems of naming or format inconsistency. A similar was used to check zipcodes for uniformity or consistency. There are zipcodes which are not in the area that start with '26' (normally zip code in the area should start with 27), some zip codes have only 4 digits, the other zip codes have formats like '27513-3507' and '275198404' which show inconsistency in the zip code formatting. Finally, the zip codes which are not consistent have been updated to consist of normal zip code formats

Data analysis

Size of Dataset

```
The size of the origanal dataset 483.230968 MB
The size of the json dataset 558.674601 MB
```

How many documents in the dataset ?

The result show that there are 2564349 documnets after processing the data

```
raleigh_data_json.find().count()
```

The number of nodes and ways in the dataset

```
print "Number of nodes:",raleigh_data_json.find({'type':'node'}).count()
print "Number of ways:",raleigh_data_json.find({'type':'way'}).count()
Number of nodes: 2325552
Number of ways: 238789
```

Who contributes the most?

Below is the list of the top ten contributors

```
:
result = raleigh_data_json.aggregate( [
                                        { "$group" : {"_id" : "$created.user",
                                        "count" : { "$sum" : 1} } },
                                        { "$sort" : {"count" : -1} },
                                        { "$limit" : 10 } ] )

print(list(result))
```

Top ten contributors

```
jumbanho  = 1557194
JMDeMai  = 202705
 bdiscoe  = 129883
 woodpeck_fixbot = 114022
bigal945   =  103432
yotann     =  66743
runbananas  = 32414
sandhill      = 32414
MikeInRaleigh  = 30731
```

```
Clay Hobbs      = 21942
```

List of top 10 post code in Raleigh

```
postcode = raleigh_data_json.aggregate( [
    { "$match" : { "address.postcode" : { "$exists" : 1} } },
    { "$group" : { "_id" : "$address.postcode", "count" : { "$sum" : 1} } },
    { "$sort" : { "count" : -1}},
      {"$limit":10}] )
print(list(postcode))

top ten list
27560 = 1612
27519 = 904
27609 = 721
27701 = 687
27705 = 526
27615 = 432
27510 = 328
27604 = 236
27513 = 190
27514 = 182
```

Other ideas

While auditing the street names and zipcodes in the xml dataset we have wittnessed that there are a lot of street naming inconsistncies , there are also problems in the zipcode formats ,some zipcodes are not in the area, some zipcodes contain only 4 digits and others have five and with extetnsions. This could be problems that arise due to the different bodys which have contributed to the openstreet dataset. So to alleviate this problems I would suggest the following:-

- To establish a standardized data inputting or reporting format that should be comminicated to to any contributor of the dataset, such as commenly used street names and zipcodes
- Use other data sources such as google API to augument with the data we accessed from openstreetmap which would solve missing data or also validates the present dataset.
- Ecouraging as many people or entities to contribute to the data, as we saw above in the top contributors list, only few people or entities are contributing, the top one contributes about 1557194 and the second one contibutes about 202705, which show a big difference in contibution to the dataset.

Conclusions¶

For this project I used data from www.openstreetmap.org for Raleigh, NC. The dataset was iteratelvely parsed to find out the total number of tags contained in the XML documents. Then I did auditing on streetname to find out if the names are consitent all over the whole document. I have found out that streetnames are not used consitently, for example 'St.', 'street'and 'St,' are used to for 'Street' , and 'Dr'/'Dr.' used for 'Drive' and soon. Later these names are upadated to the most common namings. This problem in naming inconsistency might be caused due to a number of contributor for the dataset. After auditig and updating of the dataset was converted from XML to JSON format and imported into MongoDB data base. I have faced challenge to import into the mongodb database using the 'insert()' method of pymongo until I able to import the json data using the 'mongoimport' method direclty in the command window. Finally some data analysis was made by connecting to MongoDB and accessing the stored json data from mongodb database.

Reference

1. http://wiki.openstreetmap.org/wiki/OSM_XML
2. https://www.w3schools.com/xml/xml_whatis.asp
3. https://stackoverflow.com/questions/15171622/mongoimport-of-json-file
4. https://stackoverflow.com/questions/9805451/how-to-find-names-of-all-collections-using-pymongo