Project 3: Cleaning and Exploring the Open Street Data for San-Jose Metro Area

This project cleans and explores the Open Street Map (OSM) data at the San-Jose metro area where I currently live. I downloaded the OSM data from this link MapZen. The Open street map is the largest open source transportation dataset created by individual contributions. Despite of its resourcefulness, it contains quite many human-made errors.

In this project, I am focused on understanding the quality of open street map data from a lens of urban planner and transportation researcher. I also provide suggestions on cleaning some of the data and results of my exploring the OSM data at San Jose area.

Data Quality and Cleaning

I first examine the data quality of this project. I am focused on four areas: street activities, street system, cycle network and public transportation network. Below are the problems of data quality and my suggestions.

---

1. **Street Activity: Incompleteness of address information**

   Data quality description:
   My first finding is that the OSM's address data is very incomplete. For city planner and transportation professionals, the address data of most neighborhoods in San Jose Area at OSM is not good enough for research or data analysis.

   When I audit my sample file using audit_postcode.py to see how many addresses are there in each neighborhood/zipcode. I notice there is one neighborhood having data larger than all the other areas combined.

   ```
   {'95054': 4, '95050': 1, '95037': 6, '95035': 2, '95014':
   386, '95128': 3, '95124': 1, '95125': 3, '95126': 2,
   '95127': 6, '95070': 7, '95123': 1, '94085': 1, '94086': 1,
   '94087': 5, '95002': 1, '95131': 1, '95113': 1, '95135': 1,
   '95134': 1, '95140': 2, '95051': 3}
   ```

   As a result, I went to openstreetmap website and compared the data in neighborhood 95014 and other area. I found that the 95014 neighborhood is an outlier as most of residential properties and amenities are clearly marked, while other neighborhoods do not have this information.

   Usage suggestion:
   Be VERY cautious when using OSM's address data for data analysis. It might not be accurate.

2. **Streets: the tiger tag in the k value of county, postcode, uid, etc. of streets imported from Tiger dataset.**

   Data quality description:
   In contrary to street activity information, the OSM maintains quite a good set of data in terms of road system in San-Jose area. However, it is hard for me to get any query result using keywords like postcode or county. This is because the key value for street-related information is not clean.

The country, postcode and uid of most streets are “TIGER:country”, “TIGER: zip_left” or “TIGER: zip_right”, “TIGER:uuid”. Since many street data of the open street map were imported from the Tiger dataset[1], these data are marked with “tiger”[2] at the value “k”. For researchers who don’t know this but want to analyze the road system by simply querying the county, postcode and etc., this would cause some problem, as they cannot simply search the data by using “county” or “zipcode”.

Usage suggestion and my action:
Update the keys by eliminating TIGER in the value of k and make it more consistent with OSM map key standard. See cleanning_OSM_data_to_a_new_XML.py

The tiger tags

```
<tag k="tiger:cfcc" v="A41" />
<tag k="tiger:county" v="Santa Clara, CA" />
<tag k="tiger:zip_left" v="94087" />
<tag k="source:maxspeed" v="sign" />
<tag k="tiger:name_base" v="Mary" />
<tag k="tiger:name_type" v="Ave" />
<tag k="tiger:zip_right" v="94087" />
<tag k="tiger:name_direction_prefix" v="S" />
```

will be changed to

```
<tag k="cfcc" v="A41" />
<tag k="county" v="Santa Clara, CA" />
<tag k="postcode_left" v="94087" />
<tag k="name_base" v="Mary" />
<tag k="name_type" v="Ave" />
<tag k="postcode_right" v="94087" />
<tag k="name_direction_prefix" v="S" />
```

While wrangling these data, I also add these fields to a dictionary called highway and set node[‘highway’]=highway. See cleanning_OSM_data_to_a_new_XML.py

3. **Bicycle system: inconsistency between bicycle and cycleway system and the confusing “no” value in the cycleway field.**

Data quality description:
OSM provide quite a lot of data on the cycleway information. It is something that transportation researchers would be very excited about.

[1] https://www.census.gov/geo/maps-data/data/tiger.html
[2] http://wiki.openstreetmap.org/wiki/Key:tiger

However, the data needs further validation and the accuracy requires certain improvement.

#inconsistency
The bicycle information are tagged at two tags. One is called “bicycle” with a value of “yes” or “no” indicating whether a bicycle has a right of way[3]. The other is called “cycleway” with value such as “lane”, “track” or “share” indicating its format of bike infrastructure on the street[4].

To examine the accuracy of data, I did a cross validation (audit_bicycle_crosscheck.py) to see whether the data in bicycle field and cycleway field is consistent. The code returns the value for “bicycle”, “cycleway” and “element id” of the elements that have both tags. Some of the results are as follows.

yes no 8941958
yes lane 8942625
yes track 26274651
yes no 233802747
yes no 234183465
yes no 234318879…..

We can see many elements having the bicycle value as yes, but the cycleway value as no. This is confusing and require further examination and validation.

#The “no” value of the cycleway:
I also simple query on the tag cycleway (audit_cycleway.py) and found out the “no” group that indicates there is no cycleway in this road.
{'track': 13, 'lane': 66, 'no': 17}

This is redundant and causing confusing, as applications like opencyclemap adds cycleway data from OSM data tagged with the field of cycleway. The no value induces problem. For example, South Mary Avenue at Sunnyvale has no cycle lane but tagged as “k”=”cycleway”, “v”=”no”. The “no” value for cycleway makes the avenue appear at opencyclemap.

Solution and usage suggestion:
For research: take the cycleway into account only when the value is not “no”. check this http://mijndev.openstreetmap.nl/~ligfietser/fiets/index.html
Re-examine the node/way while the bicycle value is not yes.

[3] http://wiki.openstreetmap.org/wiki/Key:access
[4] http://wiki.openstreetmap.org/wiki/Key:cycleway

I would also suggest remove the cycleway tag if the k is cycleway and the value is no, as it does not provide any extra information. See cleanning_OSM_data_to_a_new_XML.py;

4. **Insufficient information related to Public Transportation**.
   There is very limited information on public transportation. There are no public_transportation tag[5] in this entire dataset (using audit_public_transport) and the amenity with the value of bus_stop is only 1 throughout the dataset(amenity.py). This part of data is incomplete and requires further contribution.

5. **Others: Some Incorrect Street name/postcode/cityname.**
   I found Street name end with abbreviation "St", "Blvd" or lower-case Street(see audit script at audit_street.py). Using similar methods, I found problematic value for the addr:postcode and addr:city fields.

   Result of some of street name audit of sample.osm

   'Ave': {'1425 E Dunne Ave'},
   'Blvd': {'McCarthy Blvd'},
   'Rd': {'Mt Hamilton Rd'},
   'ave': {'wilcox ave'}}

   Solution:
   Change abbreviation to full name like Street, Boulevard, and change the postcode to five digit standard digits.
   see cleanning_OSM_data_to_a_new_XML.py;

Overview of the data

The overall size for the

- san-jose_california.osm 271.1M
- san_jose_california.osm.json 305.5M (after creating_json_data.py)
- sample.osm 13.7M
- sample.osm.json 15.3M (after creating_json_data.py)

```
#import san-jose_california.osm.json file to database and use
test database
>mongoimport --db test --collection osm --file san-
jose_california.osm.json
>use test
```

[5] http://wiki.openstreetmap.org/wiki/Key:public_transport

1. **The size of the file**

```
db.osm.find().count()
```

1399875

2. **number of unique users**

```
db.runCommand ( { distinct: "osm", key: "created.user" }
)['values'].length
```

1087

3. **number of nodes and ways**

```
db.osm.find({"type":"node"}).count()
```

1241702

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

158173

4. **number of chosen type of nodes, like cafes, shops etc**

```
db.osm.find({"amenity":"cafe"}).count()
```

226

```
db.osm.find({"amenity":"bus_stop"}).count()
```

1

As I explain before, the addression information at openstreetmap San Jose is very incomplete. The result is not reliable for any geo-location analysis.

5. **Top Contributors**

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

```
{ "_id" : "nmixter", "count" : 281591 }
{ "_id" : "mk408", "count" : 153545 }
{ "_id" : "Bike Mapper", "count" : 80789 }
{ "_id" : "samely", "count" : 77435 }
{ "_id" : "RichRico", "count" : 69880 }
```

Additional thoughts:

1. **Understand the Data discrepancy by comparing the address information and Street information of the top five neighborhoods**

   I did a query to find out the top 5 neighborhoods with largest number of highway tags and top 5 neighborhoods with the address information. Comparing the variability between the two groups, we can see the address information is likely to be more incomplete.

   Top 5 Neighborhood with largest number of address information

```
db.osm.aggregate([{$match:{"address.postcode":{"$exists":1}
}},{$group:{_id:"$address.postcode",count:{$sum:1}}},{$sort
:{count:-1}},{$limit:5}])
```

```
{ "_id" : "95014", "count" : 9917 }
{ "_id" : "95070", "count" : 231 }
{ "_id" : "94087", "count" : 209 }
{ "_id" : "94086", "count" : 191 }
 { "_id" : "95051", "count" : 140 }
```

   Top 5 Neighborhood with largest number of highway tags

```
db.osm.aggregate([{$match:{"street.zip_left":{"$exists":1}}
},{$group:{_id:"$street.zip_left",count:{$sum:1}}},{$sort:{
count:-1}},{$limit:5}])
```

```
{ "_id" : "95035", "count" : 852 }
{ "_id" : "95070", "count" : 827 }
{ "_id" : "95123", "count" : 764 }
{ "_id" : "95120", "count" : 742 }
{ "_id" : "95051", "count" : 655 }
```

   As we can see, 95014 have much more address information than the rest of neighborhoods. In terms of highway information, the data variability between neighborhoods is much smaller.

   Neighborhoods with the most amenity

```
db.osm.aggregate([{$match:{"amenity":
{"$exists":1},'address.postcode':{"$exists":1}}},{$group:{_
id:"$address.postcode",count:{$sum:1}}},{$sort:{count:-
1}},{$limit:5}])
{ "_id" : "95014", "count" : 67 }
{ "_id" : "94086", "count" : 29 }
```

```
{ "_id" : "95035", "count" : 28 }
{ "_id" : "94087", "count" : 27 }
{ "_id" : "95051", "count" : 26 }
```

We can see 95014 is also the neighborhood with the most amenity tags. This is probably related to its data completeness.

2. **I also want to understand the # of highway tags (non-cycleway) and # of cycleway tags to get a general idea of the proportion of bicycle infrastructure in the road system.**

```
db.osm.find({"street.street_type": {"$exists":1,
"$ne":"cycleway"}}).count()
1842

db.osm.find({"street.cycleway": {"$exists":1,
"$ne":"no"}}).count()
58639
```

The proportion of bicycle infrastructure is quite small at around 3%

I use the tag number by neighborhood to examine which neighborhoods has better bicycle infrastructure. More cyleway tags might indicate better bicycle infrastructure.

```
db.osm.aggregate([{$match:{"street.cycleway":{"$exists":1},
"street.zip_left":{"$exists":1}}},{$group:{_id:"$street.zip
_left",count:{$sum:1}}},{$sort:{count:-1}},{$limit:5}])

   { "_id" : "94087", "count" : 123 }
   { "_id" : "95112", "count" : 92 }
   { "_id" : "95035", "count" : 62 }
   { "_id" : "95134", "count" : 58 }
   { "_id" : "95014", "count" : 56 }
```

The neighborhood 94087 seems to have some good bicycle infasturcture.

3. **I am curious at the contributors for the 95014 neighborhoods. Who make this dataset so complete?**

```
db.osm.aggregate([{$match:{"address.postcode":"95014"}},{$g
roup:{_id:"$created.user",count:{$sum:1}}},{$sort:{count:-
1}},{$limit:5}])

   { "_id" : "n76_cupertino_import", "count" : 5070 }
   { "_id" : "erjiang_imports", "count" : 2694 }
   { "_id" : "Eureka gold", "count" : 858 }
   { "_id" : "karitotp", "count" : 408 }
   { "_id" : "dannykath", "count" : 304 }
```

The contributors for this neighborhood come from various sources.

Conclusion

The exercise brings me the excitement of exploring an open-source data but also makes me realize the incompleteness and lack of accuracy of open data. For researchers and data analysts using open street map data at the San Jose area, the street data are at a better condition, while the address information and public transportation data requiresfurther work.

Additional thoughts: **It is worth exploring a way to automate import the public transportation stations information into the open street map.** The benefit is clear- it will enrich the current dataset and provide a more comprehensive picture of transportation system in the bay area. However, importing public transportation data into openstreet map requires several steps and it requires the data of transit stops to be under no copyright. This might be challenging. San Jose area (The Valley Transportation Authority) has a great documentation of GTFS data including the latitude and longitude of bus stop information, but it might not be appropriate to import it into openstreetmap that are subject to change. We need to understand the term of usage under VTA, the regulation of OSM and maybe additional communication before executing this process.

In addition, there is not yet an intuitive tool importing the GTFS data to OSM. Researchers explored this topic before such as University of South Florida's multimodal trip planner project and Queensland GTFS data import project. But it requires users to use a set of tools and conducted several conversions to complete the task. In addition, even the importing methods are ripe, challenges in terms of duplicated stops in the same location, how to merge stops and how to add stop, and multiple operator or bus line tags will still remain. It would be good for the GTFS and OSM community as well as researchers to establish a standard for this action.

Python file description

Please put the python file in the same folder as sample.osm when you test running it.

Lesson 6 practice file are all under the lesson6 folder

Sample.osm creation data file is start.py

Data audit files are under project_code folder's data_audit folder

There are two data cleaning file. One is cleaning the data and save it to a new xml file - cleanning_OSM_data_to_a_new_XML.py
The other the cleaning the data and save it to a json file
creating_json_data.py

Reference

How to change value in the xml file
http://stackoverflow.com/questions/9177360/updating-xml-elements-and-attribute-values-using-python-etree

How to remove a tag in the xml file
https://docs.python.org/2/library/xml.etree.elementtree.html

Openstreetmap tags:
http://wiki.openstreetmap.org/wiki/Key:tiger
http://wiki.openstreetmap.org/wiki/Key:access
http://wiki.openstreetmap.org/wiki/Key:cycleway
http://wiki.openstreetmap.org/wiki/Key:public_transport

Cycleway tag check
http://www.opencyclemap.org
http://mijndev.openstreetmap.nl/~ligfietser/fiets/index.html

Mongodb-installation,the mongoshell
https://docs.mongodb.org/manual/tutorial/install-mongodb-on-os-x/
https://docs.mongodb.org/manual/mongo/

Mongodb –exists
https://docs.mongodb.org/manual/reference/operator/query/exists/
Mongodb-distinct
https://docs.mongodb.org/manual/reference/command/distinct/