OpenStreetMap Data Case Study

Map Area

ShangHai, China

- http://www.openstreetmap.org/node/778910398
- http://metro.teczno.com/#charlotte

This map is of the place where I live and work, so I'm interested in whether the data of shanghai is accurate or not and find some interesting data points in the map datasets.

1 Problems Encountered in the Map

Before running the data.py to write the data into the specific csv files, we need to audit the data quality to see where we need to modify the data to let it be consistence, completion and accuracy. Here, Audit.py is run to see the general quality of data, and the problems are shown below:

- As the version of Python is 3.5, all the codes used in the lessons ought to be modified. When use the code to get different size of osm file, ET.tostring() return a bytes object that can't be write into the osm file. Hence, this byte should be decode into a string from bytes in python3.5.

  ET.tostring(element,encoding = 'utf-8') *# code in python 2.7*

  ET.tostring(element,encoding = 'utf-8').decode('utf-8') *# code in python 3.5*

- See the key types distribution:

  {'lower': 1157746, 'lower_colon': 124829, 'other': 2552, 'problemchars': 2}

- As this map is for China, there are many values of roads are in chinese. Therefore, we didnt change them. But we synonym the English abbrevation of the values, like Rd. is changed to Road. The mapping dictionary is shown below:

  mapping = {

  "St.": "Street",
  "Ave": "Avenue",
  "Rd.": "Road",
  'Hwy.': "Highway",
  "Lu": "Road",
  "lu": "Road",
  "Rd.": "Road",
  "Rd)": "Road",
  "Rd.": "Road",
  "Rode" : "Road",
  "rd": "Road",
  "road": "Road",

```
"Rd," : "Road,",
"\n": "",
"\r\n":""
}
```

- As the difference of encoding between Python 2.7 and Python 3.5 is very big, the UnicodeDictWriter class in the data.ipynb should be rewritten shown as below:

```
class UnicodeDictWriter(csv.DictWriter, object):
    def writerow(self, row):
        super(UnicodeDictWriter, self).writerow({
            k: (v if isinstance(v, str) else v) for k, v in row.items()
        })

    def writerows(self, rows):
        for row in rows:
            self.writerow(row)
```

1.1 Postal Codes

Firstly, use the following query to see whether the postal codes have some problems.

```
SELECT tags.value, COUNT(*) as count FROM (

    SELECT * FROM nodes_tags UNION ALL

    SELECT * FROM ways_tags) tags

    WHERE tags.key = 'postcode'

    GROUP BY tags.value

    ORDER BY count DESC;
```

Here are the top ten results, beginning with the highest count:

value	count
200240	245
201620	63
201203	48
212003	48
201315	34
314211	31
214121	28
200231	26
215600	26
215000	23

We can find some problems in these top data, the explanation of some postal codes is shown below. The common feature these postal codes have is that they are not the postal codes of Shanghai which is the city we

need to research. This circumstance also proves that the data we get is extract from a rectangle area so that the data set contains the information of the cities around Shanghai.

212003: JiangSu Province, ZhenJiang City
314211: ZheJiang Province, PingHu City
214121: JiangSu Province, WuXi City
215600, 215000: JiangSu Province, SuZhou City

From the whole results execute from the query shown above, we also find an obvious wrong postal code that contains Chinese:

201315 上海	2

Since the volume of the wrong data point is just two, we can modify them in SQLite by using query.

2 Sort Cities by Count, Descending

```
SELECT tags.value, COUNT(*) as count FROM (
SELECT * FROM nodes_tags UNION ALL
SELECT * FROM ways_tags) tags
WHERE tags.key like '%city'
GROUP BY tags.value
ORDER BY count DESC;
```

Execute the query shown above, the top 15 results are shown below, beginning with the highest count:

value	count
上海市	559
上海	528
松江区	271
Shanghai	143
20	115
Wuxi	66
张家港市	55
24	54
30	53
杭州	51
Hangzhou	39
杭州市	36
28	31
新埭镇	30
26	26

Although the results shown are not the whole executing outcome, we still can confirm the suspension that this metro would perhaps be more aptly named “Shanghai City”, “上海市” or “shanghai” for its inclusion of surrounding cities in the sprawl. Therefore, there are two aspects should be done to let the data has the consistency. Firstly, the format of city value should be unified. Then, screen out the data which city value is “Shanghai”.

3 Data Overview and Additional Ideas

This section contains basic statistics about the dataset, the Sqlite queries are used to gather them, and some additional ideas about the idea in context.

3.1 File sizes

File name	File size
shanghai_china.osm	780Mb
shanghai_osm.db	411Mb
nodes.csv	304Mb
nodes_tags.csv	9.53Mb
ways.csv	26.6Mb
ways_tags.csv	31.5Mb
ways_nodes.csv	106Mb

3.2 Number of nodes

SELECT count (*) FROM nodes;

3876987

3.3 Number of ways

SELECT count (*) FROM ways;

472978

3.4 Number of unique users

SELECT COUNT(DISTINCT(e.uid))

FROM (SELECT uid FROM nodes UNION ALL SELECT uid FROM ways) e;

2515

3.5 TOP 10 contributing users

SELECT e.user, COUNT (*) as num

FROM (SELECT user FROM nodes UNION ALL SELECT user FROM ways) e

GROUP BY e.user

ORDER BY num DESC

LIMIT 10;

user	num
Chen Jia	710157
Austin Zhu	242355
aighes	193236
xiaotu	176214
katpatuka	140715
XBear	125344
iberutan	120723
Peng-Chung	111449
yangfl	109060
Holywindon	102394

3.6 Numbers of users appearing only once

SELECT COUNT (*) FROM

(SELECT e.user, COUNT (*) as num

FROM (SELECT user FROM nodes UNION ALL SELECT user FROM ways) e

GROUP BY e.user

HAVING num =1);

543

3.7 Number of nodes types

SELECT COUNT (*) FROM

(SELECT DISTINCT type FROM nodes_tags);

48

4 Additional Ideas

4.1 Contributor statistics

Here, we do the statistics of contributors to explore some meaningful points.

- Total contributors: 4349855

- Top user (Chen Jia): 710157, Contribution Percentage: 16%
- TOP 10: 2031647, Contribution Percentage: 47%

Obviously, the top user is an individual contributor. For an individual user, the contribution percentage is so high that his/her contribution is one third of Top-10's contributions. The top 10 results are shown below:

user	num
Chen Jia	710157
Austin Zhu	242355
aighes	193236
xiaotu	176214
katpatuka	140715
XBear	125344
iberutan	120723
Peng-Chung	111449
yangfl	109060
Holywindon	102394

From the top 10 distributions, the proportion of individual users is very big. Perhaps, Openstreet Map web is not very popular in China so that there are less company update or edit the data in this port. If Openstreet Map can give rewards to companies, maybe more companies will participate into edit the data. There are three benefits of such suggestion. Firstly, organizations or companies can edit the data more accurately and consistently than individuals do. Secondly, the data of maps can be more easily managed. Finally, if the map data is accurate resulting in more accurate navigation or other mapping products, much more customers will choose this kind of map. However, in many no-English-speaking countries, there are many local mapping companies occupying a huge market share. Therefore, if Openstreet Map need to take a place locally, the up-front cost will be huge.

5 Additional Data Exploration

5.1 Top 10 appearing amenities

```
SELECT value, COUNT (*) AS num

FROM nodes_tags

WHERE key = 'amenity'

GROUP BY value

ORDER BY num DESC

LIMIT 10;
```

value	num
bicycle_rental	2617
restaurant	1338
bank	624
cafe	442
toilets	435
fast_food	399
parking	317
community_centre	273
fuel	258
school	227

5.2 Religion

```
SELECT nodes_tags.value, COUNT(*) AS num
FROM nodes_tags
JOIN (SELECT DISTINCT(id) FROM nodes_tags WHERE
value='place_of_worship') i
ON nodes_tags.id = i.id
WHERE nodes_tags.key='religion'
GROUP BY nodes_tags.VALUE
ORDER BY num DESC;
```

value	num
buddhist	68
christian	23
taoist	5
muslim	2
confucian	1

5.3 Most popular cuisines

```
SELECT nodes_tags.value, COUNT(*) AS num
FROM nodes_tags
JOIN (SELECT DISTINCT(id) FROM nodes_tags WHERE
value='restaurant') i
ON nodes_tags.id = i.id
WHERE nodes_tags.key='cuisine'
```

GROUP BY nodes_tags.VALUE

ORDER BY num DESC

LIMIT 10;

value	num
chinese	146
italian	18
japanese	18
burger	16
pizza	12
noodles	10
asian	9
regional	9
mexican	8
international	7

6 Conclusion

Obviously, this data of the Shanghai area is incomplete. And in this exercise, I do many works on the abbreviations of the places that data is cleaned for the purpose of this exercise. From the data base, I find two important points needed to be posted. Firstly, there are many data is repeated. Because China is no-English-speaking country, but Openstreet Map is mainly using English, which result in a circumstance that one place is edited in two or three languages (like English, Chinese and Pingyin) . Another problem is that the area of some values contains another area of some values, while their values are different. Hence, all the editing regulations should be consistence and the definition or the explanation for each key or values should be accurate. If we have a good provisions at the beginning, then the data will be high-quality and more useful.