OpenStreetMap Data Case Study

Author: Jianyu Gong

Date: June 1st, 2017

Map Area

Toronto, Ontario, Canada

I choose Toronto as my study area because I am currently living in Toronto and I am trying to find a job in Toronto. Therefore, it is good to know about the map information in Toronto. After auditing and cleaning data, we can query the information quickly in the future.

Problems Encountered in the Map

The street names, postal codes and phone are audited by using audit.py. Several problems are found and presented as below.

The correct street name format should be ‘John Street East’, ‘Dundas Street’, ‘Jarvis Street’ and ‘Winston Churchill Boulevard’. The street address format problems are shown below:

- Over-abbreviated street names: (e.g. ‘John St E’ to ‘John Street East’)
- Inconsistent street names: (e.g. ‘Dundas street’ to ‘Dundas Street’, ‘JARVIS STREET’ to ‘Jarvis Street’)
- Wrong spelling: (e.g. ‘Winston Churchill Boulevade to ‘Winston Churchill Boulevard’)

The correct Canadian postal codes format should be A1A 1A1, where A is a capital letter and 1 is an integer. The postal code format auditing results and problems are summarized below:

```
Last login: Wed May 31 13:21:25 on ttys000
MacBook:~ Jianyu$ cd Desktop
MacBook:Desktop Jianyu$ cd OpenStreet
MacBook:OpenStreet Jianyu$ python2 audit.py
The incorrect postal codes are: ['M4E1g1', 'M36 0H7', 'n3r 5l8', 'm4x 1a6', 'l6c2t2', 'm3j 2n7', '96734', 'l7a 3r9'
, 'M2N 6k7', 'M5E', 'L7M', 'L4K', 'L0R', '14174', 'ON L5G 4V6', 'l3Y 3J2', 'L6r Oj6', 'l3Y 3J2', 'M4E1g1', 'M4l1j1'
, 'M4E1g3', 'M1c 2z2', 'l8h 1t8', 'L4w4M6', 'L4w4M6', 'L1H', 'M2J', 'm9b1b6', 'L3Y', 'M2N 6k7', 'M4BS26', 'm3L 2h9'
, 'M9R', 'L5l2r4', 'L67 0A7', 'M9C', '33913', 'M6K', 'L9&5M3', 'M5J 2G', 'L4B', 'l6p 2r1', '14174-1003', '14174-100
3', 'L7J 2', 'm2l 2k4', 'M1k0a4', 'm1g2l6', 'L3X 1KI3', 'M1W', 'M1W3w5', 'M1W3w5', 'M1W3w5', 'M1W3w5', 'M1W3w5', 'M
1W3w5', 'M1W3w5', 'M1W', 'M1W', 'l8g 3p1', 'L6k2G3', 'M2N', 'M6E 28V']
```

- Incomplete postal codes: (e.g. ‘M5E’, ‘L3Y’)
- Some letters are lower case: (e.g. ‘l6p 2r1’, ‘m3j 2n7’)
- No blank space in middle: (e.g. ‘M4E1g1’, ‘M1k0a4’, ‘M5A2K7’)
- Wrong Canadian postal format: (e.g. ‘14174-1003’, ‘ON L5G 4V6’)

After updating postal code, the postal codes containing lower case are changed to upper case and a space is added for the postal codes without blank space in middle.

OpenStreetMap Results

1. Sizes of the files

‘get_file_size.py’ is used to check the size of the file and the results are shown below:

```
MacBook:OpenStreet Jianyu$ python2 get_file_size.py
File Sizes:
toronto_canada.osm: 1.13455281314 GB
toronto.db: 851.45703125 MB
nodes.csv: 393.182883263 MB
nodes_tags.csv: 84.5013751984 MB
ways.csv: 41.6892700195 MB
ways_nodes.csv: 129.061340332 MB
ways_tags.csv: 91.1999101639 MB
```

2. Count Tags

‘Count_tags’ in ‘audit,py’ is used to count the occurrences of each tag, and the results are shown below:

- ‘node’: 5065357
- ‘nd’: 5797115
- ‘bounds’: 1
- ‘member’: 147130
- ‘tag’: 4983486
- ‘relation’: 9539

- ‘way’: 752169
- ‘osm’: 1

3. Number of Unique Users

```
sqlite> SELECT COUNT(DISTINCT(uid)) FROM (SELECT uid FROM nodes UNION ALL SELECT uid FROM Ways);
2628
```

4. American Area

According to last section, some postcodes containing purely numbers. Therefore, it is considered as American area. The postcodes - ‘96734’, ‘14174’ – are checked:

```
sqlite> SELECT id FROM nodes WHERE id IN (SELECT DISTINCT(id) FROM NodesTags WHERE key = 'postcode' and value = '967 34');
1547753193
sqlite> SELECT id FROM nodes WHERE id IN (SELECT DISTINCT(id) FROM NodesTags WHERE key = 'postcode' and value = '141 74');
3443667462
```

```
sqlite> SELECT * FROM NodesTags WHERE id = "1547753193";
1547753193|city|Kailua|addr
1547753193|housenumber|1320 Aulepe|addr
1547753193|postcode|967 34|addr
1547753193|province|HI|addr
1547753193|street|St #4|addr
```

```
sqlite> SELECT * FROM NodesTags WHERE id = "3443667462";
3443667462|building|house|regular
3443667462|city|Youngstown|addr
3443667462|state|NY|addr
3443667462|street|Woodland Court|addr
3443667462|postcode|141 74|addr
3443667462|housenumber|451|addr
```

After checking the Google Maps, the 1320 Aulepe Street, Kailua, HI is located in Hawaiian Islands and 451 Woodland Court, Youngstwon, NY is on the Canada-US border. Therefore, those two data should be deleted from database.

```
sqlite> DELETE FROM nodes WHERE id = "3443667462";
sqlite> DELETE FROM nodes WHERE id = "1547753193";
sqlite> DELETE FROM NodesTags WHERE id = "3443667462";
sqlite> DELETE FROM NodesTags WHERE id = "1547753193";
```

5. Number of Nodes

sqlite> SELECT COUNT(*) FROM nodes;

5065357

6. Number of Ways

sqlite> SELECT COUNT(*) FROM Ways;

752171

7. Top Five Contributing Users

```
sqlite> SELECT user, count(*) as num
   ...> FROM (SELECT user FROM nodes UNION ALL SELECT user FROM Ways)
   ...> GROUP By user
   ...> ORDER By num DESC
   ...> LIMIT 5;
andrewpmk|3389940
Kevo|484991
MikeyCarter|474443
Bootprint|209105
Victor Bielawski|142924
```

8. Top Five Amenity

```
sqlite> SELECT value, count(*) as num
   ...> FROM NodesTags
   ...> WHERE key = "amenity"
   ...> GROUP By value
   ...> ORDER By num DESC
   ...> LIMIT 5;
fast_food|3124
restaurant|2961
bench|2409
post_box|2032
cafe|1459
```

9. How Many Tim Hortons in Toronto Area?

```
sqlite> SELECT COUNT(*) FROM NodesTags WHERE value = 'Tim Hortons';
461
```

10. Top Five Cuisines

```
sqlite> SELECT value, COUNT(*) as num
   ...> FROM NodesTags
   ...> JOIN (SELECT DISTINCT(id) FROM NodesTags WHERE value = 'restaurant')
   ...> i ON NodesTags.id = i.id
   ...> WHERE NodesTags.key = 'cuisine'
   ...> GROUP By NodesTags.value
   ...> ORDER BY num DESC
   ...> LIMIT 5;
chinese|163
indian|104
japanese|93
italian|90
pizza|65
```

Additional Suggetions

sqlite> SELECT COUNT(*) FROM Nodestags WHERE key = "amenity";

26947

sqlite> SELECT COUNT(*) FROM NodesTags WHERE key = "wheelchair";

3743

During the investigation, the wheelchair accessible amenity is not enough. According to the query, the total number of amenity is 26947 and the total number of wheelchair accessible amenity is 3743 which is only 12.5% of all amenities. The low percentage of wheelchair accessible amenity maybe caused by incomplete information. The missing information can be found on various websites such as www.accessto.ca. That information can be updated through Python or manually. After updating, the percentage should be increased.

Conclusion

The OpenStreetMap project is challenging. Data wrangling skill, data cleaning skill as well as SQL knowledge are practiced and applied. However, the data is still no 100% clean. I am trying to change all the ‘St’s into ‘Street’ in all the street names. However, St can also be a part of name such as ‘St Clair’ or ‘St George’. Therefore, some names, such as St Geogre St, are difficult to change. Further cleaning is still ongoing. After the project, I learn a lot of interesting facts about Toronto and I have strong confidence on my future project and work.