OpenStreetMap Data Case Study

Roger Duong October 2017 (Rev. 2)

Outline

Map Area

The map used is the city I currently live in: Singapore

- https://www.openstreetmap.org/export#map=12/1.3450/103.8500

Sources

Singapore Postal Code system

- http://eresources.nlb.gov.sg/infopedia/articles/SIP_1006_2010-05-27.html

Workflow

This project is organized with the following workflow:

1. Launch the audit scripts to explore the dataset (`audit.py`)
2. Identify problems encountered in the map, develop programmatic ways to solve those issues and include them into `transform.py`
3. Extract the data from the osm dataset (`load.py`)
4. Transform the extracted dataset by calling (`transform.py`)
5. Load the transformed dataset into csv files (`load.py`)
6. Load the csv files into a sqlite3 database (`schema.sql`)
7. Run SQL queries to explore the map dataset

Problems Encountered in the Map

This section we discuss the findings of the step 2 of the workflow. After extracting a sample of the entire dataset, and running it through an audit.py file, we can notice the following issues:

1. Incorrect street types.
2. Incorrect postal codes.

The following elaborates on the issues encountered.

Street Types

The audit script revealed several types of incorrect street types:

1. Overabbreviated street names: for example 'St' instead of 'Street'.
2. Incorrectly capitalized street names: for example 'road' instead of 'Road'.
3. Typo errors: for example 'aenue' instead of 'Avenue'.

It should be noted that in Singapore street types may appear:

- At the end of the street name, like in the U.S. For example 'Orchard *Boulevard*'.
- At the beginning of the street name. For example '*Jalan* Besar'.
- At the second to last position in the street name. For example 'Ang Mo Kio *Avenue* 1' (note: the '1' is not the house number, it is the street number, as there are 10 street named 'Ang Mo Kio Avenue 1' to 'Ang Mo Kio Avenue 10').

The helper function `audit_street_type` in audit.py takes into account those locale specifics.

Postal Codes

Singapore uses a postal code system where each postal delivery point –usually each building– is identified with a unique 6-digit postal code. Therefore entries with anything other than 6 digits are incorrect.

Initial audit of the data revealed the following types of incorrect entries:

1. Entries where the last characters are the postal code, and where the leading character strings are any kind of text, usually 'Singapore 123456'. Solving these incorrect entries simply consist in removing all space characters and selecting the sub-string of the last 6 characters.
2. Entries of 5 characters, where the leading zero of the correct entry has been stripped

down, probably by casting the postal code to an integer type. Solving these incorrect entries require only to add the leading zero.
3. Other incorrect entries not falling into the categories above. Solving the incorrect entries can be done by querying the Singapore Post website, which has a page to search for postal codes for a given house number and street. This required a helper function to scrape the webpage and to retrieve the data. The helper function to get the postal code incorporate some time delay and user-agent parameters to avoid being blocked by the Singapore Post website after repeated queries. Because of the time delay, it was important to treat the highest number of incorrect entries by the methods 1 and 2 above, to avoid slow script execution.

Data Overview

This section discusses the findings of the setp 7 of the project workflow. It describes basic statistics about the dataset.

File sizes

The summary of file size is returned by a helper function `summarize_dataset` in the load.py file.

```
singapore.osm   211.41 MB
nodes.csv       73.31 MB
nodes_tags.csv  3.54 MB
ways.csv        8.66 MB
ways_nodes.csv  26.80 MB
ways_tags.csv   13.55 MB
```

Number of nodes

```
SELECT COUNT(*) FROM nodes;
```

```
887137
```

Number of ways

```
SELECT COUNT(*) FROM ways;
```

```
144159
```

Number of unique users

```
SELECT COUNT(DISTINCT(e.uid))
FROM (SELECT uid FROM nodes UNION ALL SELECT uid FROM ways) e;
```

```
1520
```

Top 10 contributing users

```
SELECT e.user, COUNT(e.user) as contrib
FROM (SELECT user FROM nodes UNION ALL SELECT user FROM ways) as e
GROUP BY e.user
ORDER BY contrib DESC
LIMIT 10;
```

```
JaLooNz,275589
cboothroyd,50194
Luis36995,38471
ridixcr,38004
calfarome,32845
rene78,29926
nikhilprabhakar,22755
yurasi,20454
jaredc,19039
dmastin82,16963
```

Data Exploration

This section further drills down on the dataset. It discusses some findings of step 7 of the project workflow.

Total number of restaurants

In Singapore, we love to eat out. So let's show some statistics about our restaurants.

```
SELECT COUNT(*) as count
FROM nodes_tags
    JOIN (SELECT DISTINCT(id)
    FROM nodes_tags
    WHERE nodes_tags.value = "restaurant") as nt
    ON nt.id = nodes_tags.id
WHERE nodes_tags.key = "cuisine";
```

```
636
```

List top 10 cuisines of restaurants

Let's further drill down by returning the top most frequent cuisines of restaurants.

```
SELECT nodes_tags.value, COUNT(*) as count
FROM nodes_tags
    JOIN (SELECT DISTINCT(id)
    FROM nodes_tags
    WHERE nodes_tags.value = "restaurant") as nt
    ON nt.id = nodes_tags.id
WHERE nodes_tags.key = "cuisine"
GROUP BY nodes_tags.value
ORDER BY count DESC
LIMIT 10;
```

```
chinese,135
japanese,72
korean,44
pizza,43
italian,37
indian,35
asian,31
thai,29
french,15
seafood,13
```

As we can see, although Chinese cuisine –unsurprisingly– dominates the types of cuisines, there is a very wide variety of cuisines. This truly makes Singapore a cosmopolitain city!

Total number of cafes

Here we take a look at the cafes in Singapore.

```
SELECT COUNT(*) as count
FROM nodes_tags
    JOIN (SELECT DISTINCT(id)
    FROM nodes_tags
    WHERE nodes_tags.value = "cafe") as nt
    ON nt.id = nodes_tags.id
WHERE nodes_tags.key = "cuisine";
```

```
91
```

List top 10 styles of cafe

We further drill down with the style of cafes.

```
SELECT nodes_tags.value, COUNT(*) as count
FROM nodes_tags
    JOIN (SELECT DISTINCT(id)
    FROM nodes_tags
    WHERE nodes_tags.value = "cafe") as nt
    ON nt.id = nodes_tags.id
WHERE nodes_tags.key = "cuisine"
GROUP BY nodes_tags.value
ORDER BY count DESC
LIMIT 10;
```

```
coffee_shop,43
international,6
regional,4
sandwich,4
italian,3
Western,2
asian,2
coffee_shop;regional,2
french,2
"Hawker or Foodcourt, Chinese",1
```

Looking at the list of styles for cafes, it appears that the entries should be further cleaned:

1. some entries have inconsistent capitalization like: `Western` and `asian`. These values can be easily cleaned through regular expression operations, similarly to what was done for the street names.
2. some entries contain multiple values separated by a semi-colon or a slash like: `coffee_shop;regional` or `Western/Italian`.
3. some entries are registered as strings within quotation marks like `"Hawker or Foodcourt, Chinese"`. Those are special cases of multivalued keys.

List top 10 styles of fast-foods

We run the same type of analysis for the fast-foods.

```
SELECT nodes_tags.value, COUNT(*) as count
FROM nodes_tags
JOIN (SELECT DISTINCT(id)
FROM nodes_tags
WHERE nodes_tags.value = "fast_food") as nt
ON nt.id = nodes_tags.id
WHERE nodes_tags.key = "cuisine"
GROUP BY nodes_tags.value
ORDER BY count DESC
LIMIT 10;
```

```
burger,59
chicken,27
sandwich,13
pizza,12
chinese,7
fast_food,5
ice_cream,5
asian,4
american,2
japanese,2
```

Running this query for more items would show the same data quality issues as the nodes tags on cafes. The entries can be cleaned using the an identical script.

Conclusion

This data wrangling exercice on Singapore Open Street Map data highlights that there is a great wealth of information available for further analysis.

Additional Ideas

The data on restaurant, cafe and fast-foods can be used for marketing and consulting analyses.

Problem Statement and Value Proposition

Existing and future F&B managers frequently face the following problems:

- When setting up a new restaurant:
    - If it is a new oulet of an existing franchise: Where to locate the restaurant? Where is the competition? What is the accessibility (travel time, car park ,public transportation)?
    - If it is the first outlet: Same questions and in addition: Which cuisine? Which clientele (casual, fine dining etc.)?
- Existing restaurant makeover: Which cuisine? Which clientele (casual, fine dining etc.)?

We can imagine to develop a web-accessible restaurant data analytics, using various map data sources to overlay data on restaurants businesses. And what about even incorporating some recommendation features in the analytics. The ultimate goal would be to help decision-makers to determine the characteristics of restaurant to open that would be likely to be more successful. This platform would make the analytical process less time consuming, and more accurate.

Access to this platform could be monetized to restaurant consultants. Those restaurant consultants would then sell their services to F&B management companies, F&B managers, or individuals looking at starting a restaurant business.

The datasets to include would be:

- OpenStreetMap datasets for location of outlets, car parks, public transportation facilities.
- Singapore open data sets https://data.gov.sg/ for demographics, finance, transportation among others.
- Yelp datasets https://www.yelp.com.sg/dataset for restaurant reviews notably.
- some dataset to measure travel time from certain points

Limitations

One limitation is that some information of the micro-environment of a specific location cannot

be captured by those macroscopic datasets: for example it would not be possible to take into account the location of an outlet within a shopping mall, and what would be the effect on the business of some location-specific features (eg. the effect of a corner unit versus a row unit in a shopping mall would not be accurately modelled). This could be overcome by allowing for some user specific entries.

Anticipated Problems

Cleaning

In the restaurant, cafe and fast-foods tags, many entries have the usual user-entered issues of capitalisation, and typos errors. Those can be cleaned through scripting similar to what was done for street names.

Data Integration

The analytics platform would integrate multiple datasets, with their own schemas and conventions. Integrating them into a meaningful larger dataset would require a careful work on:

- adapting the schemas,
- managing the refresh of dataset.

Multivalue keys

The OSM dataset contains many entries with multiple value for the key `cuisine`, separated by a colon. Putting the analytics platform together would require to define a strategy to interpret the multivalue key. We can use the same method employed to split street names. Finding a way to treat multivalue keys is important to provide sufficient richness to the data.

Possible options are already discussed here:

- http://wiki.openstreetmap.org/wiki/Multiple_values
- http://wiki.openstreetmap.org/wiki/Proposed_features/Multivalued_Keys.