Wrangling OpenStreetMap Data with MongoDB

By Armin Oliya, December 2015

In one of my other projects I need to make use of street names and business registration information available in OpenStreetMap. This is an attempt to clean up the crowd-sourced dataset.

Map Area: Singapore
https://www.openstreetmap.org/relation/536780

1. Problems Encountered in the Map

After initially downloading a small sample size of Singapore and doing some audits on different tag values, I noticed some problems with the data, which I will discuss in the following order:

- Data belonging to **other countries** appearing in the data set (from Malaysia and Indonesia)
- Untraditional **street name** abbreviations (such as Jln) which could appear in the beginning of the street name
- Inconsistent and incorrect **house numbers**
- Inconsistent **phone numbers**
- **Incorrect postal codes** (Singapore postal codes are all 6 digits)
- Names in **other languages** such as chinese and malay

Data from other countries

Due to its small size and close vicinity with neighbouring countries, you can easily spot nodes and ways which belong to Malaysia or Indonesia. It doesn't cause problem on its own, but the thing is that it affects the pattern/convention for street naming, postal codes, phone numbers, and other details.

Since this dataset is representing Singapore only, I excluded elements which didn't belong to Singapore. I checked this in my **inSingapore** function and ignored the element if the function returned false.

Untraditional **street name** abbreviations

Based on my knowledge and some quick audits, I found some singapore-specific street name abbreviations in the data set, such as Jl. and Jln (for Jalan which means street in Malay), Btk (for Butik which means hill in Malay) or Upp (for Upper).

I simply included these abbreviations in my mapping and made sure to search throughout the names, because Singaporean abbreviations normally appear in the beginning of street name.

Inconsistent House Numbers

House numbers are one of the nightmare-ish aspects of address in Singapore (not specific to this dataset). The thing is addresses in singapore are defined with:

- Block number or house number: for example, 31, 320A, 12
- (optional) Unit numbers defined as: #dd-dd. for example #03-21 means the unit with number 21 on the door which is located on the 3rd floor.

A full address example: **79** Anson Rd, **#01-03**, Singapore **079906**
79 is block number
#01-03 is unit number
079906 is post code

I noticed that the value of house number is used to represent block numbers, house numbers, or both. It sometimes also included other values such as:

- A
- Blk x
- NTU Admin Cluster

, all of which were ignored.

I decided to keep the value of this tag as long as it includes the block number or the optional unit number, as either alone or both together collectively could represent a single unit.

Inconsistent phone numbers

Phone numbers sometimes included non digits, or additional characters such as '-'. I standardised all to +65ddddddd (all singapore numbers are 8 digits).

Incorrect postal codes

Again, singapore postal codes are all 6 digits, so i discarded any value which didn't include 6 digits somewhere in the string. For example, these values were ignored:

- Malaysian postal codes with 5 digits
- random strings such as <different>

Alternate names

Due to its cosmopolitan nature, a lot of names have a chinese, malay, or indian origin. I kept these additional names under the **names** array.

2. Data Overview

This section contains basic statistics about the dataset and the MongoDB queries used to gather them.

* I created a 65MB version of the data set, as the original 200MB dataset threw memory exceptions repeatedly.

File sizes

singapore.osm 65MB
singapore.osm.json 92MB

Number of documents

```
> db.sg.find().count()
336797
```

Number of nodes

```
> db.sg.find({"type":"node"}).count()
294236
```

Number of ways

```
> db.sg.find({"type":"way"}).count()
42546
```

Number of unique users

```
> db.sg.distinct('created.user').length
937
```

Top 1 contributing user

```
>db.sg.aggregate([{$group:{_id:'$created.user',count:{$sum:1}}},{$sort:{_id:-1}},[{
$limit:1}])

{ "_id" : "????", "count" : 18 }
```

Number of users appearing only once (having 1 post)

```
> db.sg.aggregate([{"$group":{"_id":"$created.user", "count":{"$sum":1}}},
{"$group":{"_id":"$count", "num_users":{"$sum":1}}}, {"$sort":{"_id":1}},
{"$limit":1}])
{ "_id" : 1, "num_users" : 195 }
# "_id" represents postcount
```

Top 10 appearing amenities (singapore is famous for its free and ubiquitous public toilets!)

```
> db.sg.aggregate([{"$match":{"amenity":{"$exists":1}}},
{"$group":{"_id":"$amenity",
"count":{"$sum":1}}}, {"$sort":{"count":1}}, {"$limit":10}])


{ "_id" : "yes", "count" : 2 }
```

```
{ "_id" : "waste_disposal", "count" : 1 }
{ "_id" : "waste_basket", "count" : 10 }
{ "_id" : "veterinary", "count" : 1 }
{ "_id" : "vending_machine", "count" : 5 }
{ "_id" : "university", "count" : 16 }
{ "_id" : "townhall", "count" : 6 }
{ "_id" : "toilets", "count" : 51 }
{ "_id" : "theatre", "count" : 8 }
{ "_id" : "telephone", "count" : 5 }
```

Biggest religion (religious/cultural scene in singapore is highly diverse)

```
> db.sg.aggregate([{"$match":{"amenity":{"$exists":1},
"amenity":"place_of_worship"}},

{"$group":{"_id":"$religion", "count":{"$sum":1}}},

{"$sort":{"count":1}}])

{ "_id" : "taoist", "count" : 2 }
{ "_id" : "sikh", "count" : 1 }
{ "_id" : "muslim", "count" : 142 }
{ "_id" : "jewish", "count" : 2 }
{ "_id" : "hindu", "count" : 4 }
{ "_id" : "christian", "count" : 54 }
{ "_id" : "buddhist", "count" : 19 }
```

Most popular cuisines

```
> db.sg.aggregate([{"$match":{"amenity":{"$exists":1}, "amenity":"restaurant"}},
{"$group":{"_id":"$cuisine", "count":{"$sum":1}}},      {"$sort":{"count":1}}])

{ "_id" : "chinese", "count" : 33 }
{ "_id" : "italian", "count" : 11 }
{ "_id" : "korean", "count" : 11 }
{ "_id" : "pizza", "count" : 9 }
{ "_id" : "french", "count" : 8 }
{ "_id" : "japanese", "count" : 6 }
```

```
{ "_id" : "indian", "count" : 5 }
{ "_id" : "asian", "count" : 5 }
{ "_id" : "seafood", "count" : 5 }

{ "_id" : "regional", "count" : 3 }
{ "_id" : "pizza", "count" : 9 }
{ "_id" : "peranaken", "count" : 1 }
{ "_id" : "muslim", "count" : 1 }
```

3. Further Steps

I think after this cleaning round the dataset is quite clean to be used in other applications. There are a few other actions that can be taken to further clean the data or improve the accuracy.

One thing I noticed is that sometimes information that is needed for one field - for example phone number - could be included in a totally random field that no one expects. A more robust procedure would look up all “v” attribute values for all tags and use them (or even compare with an existing tag for consistency). This sounds to be quite tricky and some serious regex skills are required to avoid false positives or true negatives. But regex can handle cases that we can anticipate, and given the chaotic nature of data, I can imagine records for which another approach may be required.

Another way to improve the reliability of the data would be to verify lat, lon, street name, phone number, and post code with some existing and trusted database such as data.gov.sg or google maps api. The challenge would be how to resolve conflicts when there are contradictory information. this could be resolved - for example - by defining source priority for each information field or by looking at the timestamp of the data.