

OpenStreetMap Project
Data Wrangling with MongoDB
Zhi Li

Map Area: Hong Kong, China

<https://www.openstreetmap.org/node/2833125787>

https://mapzen.com/data/metro-extracts/metro/hong-kong_china/

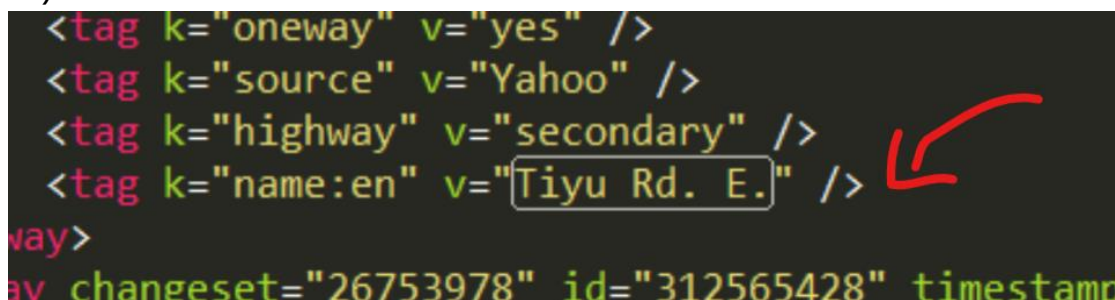
1. Problems Encountered in the Map

I noticed that there are some problems with the data after checking out sample.osm of Hong Kong` map osm. I would discuss them as the following order:

- ✂ Over-abbreviated street names
- ✂ Multi-language name data

Problem No.1 For tag which startswith "addr:"Over-abbreviated street names ("Tiyu Rd. E.")


```
<tag k="oneway" v="yes" />
<tag k="source" v="Yahoo" />
<tag k="highway" v="secondary" />
<tag k="name:en" v="Tiyu Rd. E." />
way>
ay changeset="26753978" id="312565428" timestamp
```



Data incompleteness is one of most common data problems. This kind of data need, obviously, to be fixed before import into my local MongoDB.

Problem No.2 For tag name , there are some versions of language name.

```
<tag k="phone" v="+ 852 2522 0922" />
<tag k="name:en" v="Victoria Peak" />
<tag k="name:ko" v="빅토리아 피크" />
<tag k="name:zh" v="扯旗山" />
<tag k="natural" v="peak" />
<tag k="wikidata" v="Q17541" />
```



I guess this kind of data redundancy is due to the fact openstreet maps data is created with many different persons with several speaking-language background or the maker want to provide map service to people from different regions.

One possible version:

```
{name: {
  "en": "Victoria Peak",
  "ko": "빅토리아 피크",
  "zh": "扯旗山"
}}
```

This would make the data format be more readable and organized.

Before importing cleaned json files into MongoDB, I've checked and modified problematic data in raw osm datafile and converted it into json format.

```
audit.py
data.py
```

2.Data Overview

This section contains basic statistics about Hong Kong dataset and the detailed MongoDB queries to gather them.

File sizes:

```
hongkong.osm -----643MB
hongkong.osm.json ----- 739MB
```

Number of documents

```
> db.hongkong.find().count()
```

3476607

Number of nodes

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

3141059

Number of ways

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

333069

Number of unique users

```
>db.hongkong.distinct({ "created.user" }).length
```

2542

Top 1 contributing user

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

```
{ "_id" : "hlaw", "count" : 438440 }
```

Number of users appearing only once (having 1 post)

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

```
{ "_id" : 1.0, "num_users" : 615.0 }
```

Number of distinct amenity

```
>db.hongkong.dinctinct("amenity")
```

```
123
```

3.Additional Ideas

Data Information Extraction and Cross-validation

As I have analysed above , data incompleteness is one of biggest problems. Well, we can use the technology —— information extraction from the Internet. In addition, some maps such as Bing Maps, Google Maps have open APIs, which may help us for data auditing and cleaning. (In Lesson 1-3 there are some secions about web scrapping. 😊)

Besides, we can also cooperated with other services providers such yelp, Dazhongdianping to help osm do better job in data maintainance which have massive data about a site especially pubic services sites data like café, resteraunts etc.

Pros and Cons (Think twice before we take actions)

Yet, this is an accessible and efficient method for data completeness in OSM. With machine helping us doing such repeated and boring stuff, it's absolutely more efficient and maneuvrable at some extent.

But here comes the questions too. What about the data validity and uniformity ? Different sources may have unique schema or namespaces. It can bring up a new kind of data consistency and redundancy.

From my personal stand, I suggest ontology technology might be a possible method to deal with these problems. I just found, for instance, there are some amenities such banks. Even several node are all ICBC banks, but they can be named into several versions. With ontology technology, ontology model has the ability to recognizing entities with its properties. This can make the difference in naming consistency, I think. There could be my part of my further works on this project.

Further tag cluster

There are some other tags which also has a lower_colon, but aren't "addr:street" or "name", like the tag "building" etc. Just for the project simplification, I didn't deal with other similar tags. A well-organized, readable osm should be designed delicately.

```
<nd ref="3230276629" />
<tag k="building:part" v="yes" />
<tag k="building:levels" v="18" />
<tag k="building:min_level" v="12" />
way>
way changeset="27407337" id="316843047" timestamp="2014-12-
```

However, I remind myself that maybe contributors have their own purpose of doing second colon namespaces. As the same time, it may also break the consistency of node/ways tags namespace. So I need to weight the pros and cons before I really before clean the data in Database.

Additional data exploration using MongoDB queries

Top 5 appearing amenities

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

```
{ "_id" : "restaurant", "count" : 2288.0 }
{ "_id" : "parking", "count" : 2136.0 }
```

```
{ "_id" : "school", "count" : 1881.0 }  
{ "_id" : "toilets", "count" : 1475.0 }  
{ "_id" : "bank", "count" : 997.0 }
```

Most popular cuisine(no surprise here)

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

```
{ "_id" : "chinese", "count" : 277.0 }
```

Conclusion

After this review of the Hong Kong OSM data, I'm sure the area is incomplete. I believe, I should say, it has been cleaned at some extent just for this project exercise like data reorganization, some fields' data completeness. It cannot be, however, recognized as a well-cleaned data because there are some other issuss which I haven't detected and fixed. Through this project, I've practiced how to assess data from accuracy, consistency etc? How to write some basic scripts to audit my chosen data and clean it? How to store and query data using MongoDB etc. This kind of practice learning is fantastic and useful.