OpenStreetMap Sample Project
Data Wrangling with MongoDB

Steven Ko

1. Problems Encountered in the Map

(1) Most of the street name are write in Chinese however some of them are in English. Since there are only 19 data write in English. These are pretty few compare to Chinese 5983 data. For data uniformity, I simply exclude those datas

(2) Inconsistent name tag
Most of the name tag key are simply 'name', however there are some in different formats

ex:
    name:zh-classical',
    'name:zh-min-nan',
    'name:zh-py',
    'name:zh-simplified',
    'name:zh-tradition',
    'name:zh-traditional',
    'name:zh-yue',
    'name:zh_TW'

I update name tag with 'zh' to 'name' if the node don't contain 'name' tag, and discard it if the node already has name tag (since it's duplicated)

(3) Some street name are not normal street name in Chinese, like: 三界公坑, 桃園縣新屋鄉東明村2鄰東勢

I remove those data

2. Data Overview

File sizes

taipei_taiwan.osm .... 220 MB
taipei_taiwan.osm.json .... 315 MB

Number of documents
        > db.map.find().count()
        1173771

Number of nodes
        > db.map.find({"type":"node"}).count()
        1063215

Number of ways
        > db.map.find({"type":"way"}).count()
        110528

Number of unique users
        > len(db.map.distinct("created.user"))
        1376

Top 1 contributing user
        > db.map.aggregate([{"$group":{"_id":"$created.user", "count":{"$sum":1}}},
        {"$sort":{"count":-1}}, {"$limit":1}])
        {u'count': 208886, u'_id': u'Supaplex'}

Number of users appearing only once (having 1 post)
        > db.map.aggregate([{"$group":{"_id":"$created.user", "count":{"$sum":1}}},
{"$group":{"_id":"$count", "num_users":{"$sum":1}}}, {"$sort":{"_id":1}}, {"$limit":1}])
        {u'num_users': 209, u'_id': 1}

Number of restaurants
        > db.map.aggregate([{"$match" : {"amenity" : "restaurant"} },
        {"$group":{"_id":"$name"}},
        {"$group":{"_id":1, "count" : { "$sum":1}}}])
        {u'count': 2764, u'_id': 1}

Number of kinds of amenity

```
> db.map.aggregate([
{"$match" : {"amenity" : {"$exists": "true"} } },
{"$group":{"_id":"$amenity","count" : { "$sum":1}}},
{"$group":{"_id":1, "count" : { "$sum":1}}}
])
{u'count': 146, u'_id': 1}
```

3. Additional Ideas

Contributor statistics

- Top user contribution percentage ("Supaplex") - 18.8%
- Top 10 users contribution: 62.5% (733652/1173771)

Most of the data are contributed by top 10 users

Tag uniformity

Since there are too many kinds of tags in data, it's hard to calibrate and use it. It would be great if Openstreet can define some most used tags and the structure clearly. In this way people who submit data can follow it, and data analyst can know how to use the data efficiently

"Position" data consistency

Some of the data has position info, like: longitude, latitude, and address. So the address can be used to cross check data consistency of [longitude, latitude] by using Google Maps api

Some interesting result in Taipei

Most popular cuisines

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

```
        {"$limit":10}])
```
鍋物, 水餃、麵、小菜, 義法料理, noodles, chinese,chinese_breakfast, mexican, tibetan 咖啡, peruvian, 燒臘

mexican food & tibetan food are not very popular in Taipei, I think the result is biased Maybe it's because local Taiwanese are not likely to upload data to OpenStreet

Biggest religion

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

None:496
taoist:271
buddhist:168

It can be find that most of Taiwaneses have no religion

Conclusion

There many data from OpenStreetMap are not used or parsed, like: relation, member. It's not intuitive to know what does those data mean. I think I should read the document to know it.

By wrangling those data, I know there are many infos stored on OpenStreepMap. But it is really rely on data analyst's survey, observation extract insights. For me, at first, I really have no idea how I can use those data, although I've already wrangle it and import it to MongoDB. But after watching the sample report, there are more and more ideas came to my mind. Practice and read other example is helpful