OpenStreetMap Project Report

Data Wrangling with MongoDB

Darren Chiu

Map Area: Hong Kong, China
https://mapzen.com/data/metro-extracts/metro/hong-kong_china/

Table of Content

Code Structure

There are three code files included in this project submission.

1. data.py – reading data from xml file and store a json, no data processing is done in this file except converting the xml into json
2. cleaning_queries.py – cleaning queries to address the problems raised in part 2 of this report.
3. explore_queries.py – queries to explore the data set for part 3 and 4

Problem encountered in the map

1. Insufficient data in the area
   Since OpenStreetMap is not very well-known or famous in the area, there are very limited data in the region. While the original data have 3,091,401 nodes and 328,175 ways in it with a json file size of 667.9MB, the remaining size if I removed all the empty nodes and ways with no details(tags) would be 107,368 nodes and 323,804 ways with 97.9MB.

   And some of the nodes just got one tags with the 'created_by' key which is pretty meaningless as map data.

   While it might be better for us to remove all these empty data from further analysis, I kept those empty nodes for the purpose of analyzing the users and contributors in the next sections.

2. Inclusion of some mainland China data
   After taking a quick look on the smaller sample data set, it was obvious that some peripheral nodes from mainland China is also included. Those records can easily be identified due to the language differences of Hong Kong and mainland China, the nodes from mainland China are usually labeled with "name:zh_pinyin" tag while Hong Kong people are quite reluctant to use pinyin. Another clue would be making

use of the official data from Hong Kong government about latitude and longitude to remove those mainland China nodes.
I did a cleanup using the logic of “outside of Hong Kong latitude and longitude”. Surprisingly, 2,438,133 out of 3,091,401 records are outside of Hong Kong latitude and longitude.

```
mainland_china_node_query = {
    "type": "node",
    '$or': [
        {"lat": {"$lt": 22.08}},
        {"lat": {"$gt": 22.35}},
        {"lon": {"$lt": 113.49}},
        {"lon": {"$gt": 114.31}},
    ]
}
```

3. Data formatting and accuracy for street name
   I also took a look on the street names by doing a distinct street query in the dataset.

```
result = db.nodes.distinct("addr:street")
```

A number of issues was identified including shorthanded or typing errors with street names including “Rd”, “Road1”, “Street2”, “Road3”, “St.”, “St” etc. A search and replace was done to clean up all these issues.

Another issue was identified was excess information entered in the street field. One example is that people entered “503 Nathan Road, Yau Ma Tei, Kowloon” instead of “503 Nathan Road” into the street field. A cleanup was also done on the items that I identified. However, it is hard to develop a generic way to clean up all these as it is hard to intelligently identified excess information in the field.

The code for cleaning up street names are included in the cleaning_queries.py file.

Overview of the data

File Size

File	Size
hong-kong_china.osm	664.2MB
hong-kong_china.json	663MB

Statistics

1. Number of nodes

   pymongo query:

   ```
   all_nodes = db.hk.find({"type": "node"})
   print("Number of nodes: {}".format(all_nodes.count()))
   ```

   Result:

   3,091,401 (before cleaning)

   653,268 (after cleaning)

2. Raw Number of non-empty nodes (with at least one tag, before cleaning)

   pymongo query: no pymongo query is required but counted during converting xml into json

   Result: 107,368

3. Number of ways

   pymongo query:

   ```
   all_ways = db.hk.find({"type": "way"})
   print("Number of ways: {}".format(all_ways.count()))
   ```

   Result:

   328,175

4. Number of non-empty ways (with at least one tag, before cleaning)

   pymongo query: no pymongo query is required but counted during converting xml into json

   Result: 323,804

5. Number of unique users

   pymongo query:

   ```
   all_users = db.hk.distinct("user")
   print("Number of distinct users: {}".format(len(all_users)))
   ```

   Result: 2031 (after cleaning)

6. Number of natural scenarios

pymongo query:

```
pipeline = [{
    "$group": {
        "_id": "$natural",
        "count": {"$sum": 1}
    }
}]

results = db.hk.aggregate(pipeline)
for result in results:
    print(result)
```

Result:

```
{'count': 2, '_id': 'valley'}
{'count': 1, '_id': 'land'}
{'count': 101, '_id': 'sand'}
{'count': 1, '_id': 'heath'}
{'count': 27, '_id': 'tree_row'}
{'count': 1, '_id': 'forest'}
{'count': 22, '_id': 'saddle'}
{'count': 154, '_id': 'grassland'}
{'count': 24, '_id': 'scree'}
{'count': 1, '_id': 'ditch'}
{'count': 168, '_id': 'scrub'}
{'count': 1, '_id': 'cave'}
{'count': 1, '_id': 'hillock'}
{'count': 8, '_id': 'ridge'}
{'count': 24, '_id': 'bare_rock'}
{'count': 152, '_id': 'cliff'}
{'count': 17, '_id': 'fell'}
{'count': 7, '_id': 'yes'}
{'count': 106, '_id': 'stone'}
{'count': 913, '_id': 'tree'}
{'count': 3, '_id': 'spring'}
{'count': 8, '_id': 'rock'}
{'count': 7914, '_id': 'water'}
{'count': 4, '_id': 'mud'}
{'count': 1001, '_id': 'coastline'}
{'count': 17, '_id': 'cave_entrance'}
{'count': 808, '_id': 'wood'}
{'count': 135, '_id': 'bay'}
{'count': 81, '_id': 'wetland'}
{'count': 9, '_id': 'strait'}
{'count': 280, '_id': 'beach'}
{'count': 182, '_id': 'peak'}
```

Comment: I am particular interested in this because 75% of this tiny city is classified as country park area and at the same time Hong Kong are one of the highest density city in the world. With these data, one could track down each of the beautiful scenery like waterfall, cave, peak and beach one by one.

Other ideas about the dataset

Validating the coverage of dataset

While OpenStreetMap is not popular in the area, I am particularly interested to know the comprehensiveness of basic information of the area, including hospitals, schools, metro stations. Therefore, I did some queries based on several amenity types and compare with the official number of these amenities.

1. Government Home Affairs Department Offices (HAD Offices)
   Official number of offices: 20
   Pymongo Query:
   ```
   had_offices = db.hk.find({"name": {"$regex": '.*民政.*'}})
   for office in had_offices:
       pprint.pprint(office['name'])
   print("Number of railway stations: {}".format(office.count()))
   ```
   Result: 17 (but only one is matching what we are looking for)
   Comment: HAD offices is the entry point of a lot of government services but we could only find 1 out of 20 in the data set unfortunately.

2. Metro Stations
   Official number of stations: heavy rail: 93, light rail: 68
   Pymongo query:
   ```
   all_railway_stations = db.hk.find({"railway": 'station', "network": "MTR"})
   for station in all_railway_stations:
       pprint.pprint(station['name'])
   print("Number of hospitals: {}".format(all_railway_stations.count()))
   ```
   Result: 90
   Comment: From the perspective of railway stations, obviously OpenStreetMap is lacking a lot of information. It is highly unlikely someone could explore Hong Kong with the information from OpenStreetMap.

3. Apple Stores
   Official number of Apple Stores: 6
   Pymongo Query:
   ```
   apple_stores = db.hk.find({"name": {"$regex": '.*Apple Store.*'}})
   for store in apple_stores:
       pprint.pprint(store)
   print("Number of apple stores: {}".format(apple_stores.count()))
   ```
   Result: 3
   Comment: There are half coverage of Apple stores information in OpenStreetMap data. However, those seems to be the older Apple stores while the newer ones are not included.

Suggestions on improving the dataset

1. Work with Hong Kong government

   In fact, Hong Kong government have an initiative(https://data.gov.hk/en/) to provide open data to the public. OpenStreetMap might consider working with the Hong Kong government to add more information to OpenStreetMap dataset. Even that's not possible, we can still import the data that are already published to OpenStreetMap.

   Benefits:
   - High data accuracy and ensuring up-to-date information in long run
   - Free

   Anticipated Problems:
   - Slow or even no response from governments
   - Lack of incentives for government to collaborate

2. Work with corporations

   There are some corporations owning big dataset on mapping already. Examples are Niantic (the gaming company who made Ingress and Pokemon Go) and Google. While OpenStreetMap is an open platform, these companies might be willing to collaborate on creating the best mapping dataset in the world. Even if those corporations are not willing to collaborate, they might still have some api provided a limited level of information. In fact, some location information of OpenStreetMap were imported from Wikipedia.

   Benefits:
   - Quicker response that governmental organizations
   - Sometimes the data might be even more agile and up-to-date than government

   Anticipated Problems:
   - Might involve a cost in buying the data
   - Lack of incentives for corporations to open their data

Conclusions

To conclude this short investigation on the Hong Kong OpenStreetMap, we can see that OpenStreetMap is obviously lacking popularity in Hong Kong. Very limited location data are included in the dataset. A few extractions involving government facilities, railway and commercial stores showing that OpenStreetMap is definitely lacking information in all sectors instead of just a particular sector.

As a citizen of Hong Kong, it is well understood that Hong Kong is extremely slow pace on open data movement. Both government and commercial sector is lacking the understands of benefits of open data. While more and more startups are appearing in Hong Kong, I am hoping that there will be more and more open data of the city and allow data scientists to explore interesting trends and facts to help people exploring Hong Kong in unimaginable ways.