OpenStreetMap Project: Data Wrangling with MongoDB

Rinlapat Pusanasurapant

Map Area: Chiang Mai, Thailand

https://s3.amazonaws.com/metro-extracts.mapzen.com/chiang-mai_thailand.osm.bz2https://www.openstreetmap.org/relation/1908771

I've chosen Chiang Mai, Thailand dataset because I visited this province last year and it's a very popular province that everyone including Thais and foreigners would like to visit. So, this dataset should contain a lot of shops, cafes, hotels, and places that are benefit for doing data analysis.

1. Problems Encountered in the Map

After reviewing the dataset, I found the following main problems:

- Some records do not contain "user" and "uid" attributes. I've decided to set `None` to these fields instead. So, the number of unique users that I'm going to calculate may not be accurate because we have no clue about creators of those records. Below is an example record that doesn't have "user" and "uid" attributes.

```
<node id="206881696" lat="18.7883551" lon="98.9853163" version="5"
timestamp="2008-04-15T13:29:39Z" changeset="157228"/>
```

- Some places do not have the default name (name), English name (name:en), or any name specified. If we would like to list the name of shops or places nearby a specified location, we can't list all of them. In a case that a record doesn't contain the default name but it has the alternative names. I'll make sure that at least "name:en" is set to the "name" attribute by using audit_default_name() in the data_wrangler.py file.

- From mapzen website, "Chiang Mai, Thailand" dataset contains the data of other provinces such as Lamphun province (ลำพูน). So, those data should be filtered out so that the basic statistics about the dataset will be more accurate. However, this requires many steps and the third-party API to find the province of each record using latitude and longitude, so I'll skip it for now and assume that this dataset contains only Chiang Mai province data.

- In some nodes, they contain only the default address attributes such as `addr:city` and `addr:street` attributes without any alternative language.

  Some of them are in Thai:

  ```
  # (1) City attribute contains the address, not only the city. <tag
  k="addr:city" v="หมู่ 10 ตำบลแม่เหียะ อำเภอเมืองเชียงใหม่" />

  # (2) City attribute contains the village name only. <tag
  k="addr:city" v="หมู่บ้านเฮ้าส์แอนด์วิว" /> <tag k="addr:street" v="ซอย 5" />
  ```

  Some of them are in English:

  ```
  <tag k="addr:city" v="Chiang Mai" /> <tag k="addr:street" v="Sermsuk
  Road" />
  ```

  So, it's very hard to clean these data automatically.

- In a case that `addr:street` attribute is written in English, I found that some of them contain "Rd", "Rd.", or "Road". So, I've decided to rename "Rd" and "Rd." to "Road" by using audit_streetname() in the data_wrangler.py file. For example, "Arak Rd. Soi 3" is changed to "Arak Road Soi 3".

2. Data Overview

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

File sizes

- chiang-mai_thailand.osm ..... 118.9 MB
- chiang-mai_thailand.osm.json ..... 154.9 MB

Number of documents

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

632174
```

Number of nodes

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

575150
```

Number of ways

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

57024
```

Number of unique users

```
> db.chiangmai.aggregate([     { "$group": { "_id": "users",
"unique_users": { "$addToSet": "$created.user" } } },     { "$unwind":
"$unique_users" },     { "$group": { "_id": "$_id", "count": { "$sum": 1 }
} } ])

[{u'count': 357, u'_id': u'users'}]
```

Number of shops in Chiang Mai

```
> db.chiangmai.find({ "shop": { "$exists": True } }).count()

1710
```

Number of hospitals in Chiang Mai

```
> db.chiangmai.find({ "amenity": "hospital" }).count()

76
```

Top 1 contributing user

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

[{u'count': 231419, u'_id': u'Johnny Carlsen'}]
```

3. Additional Ideas

As a tourist, I would like to find hotels near Chiang Mai international airport. Latitude and longitude of the airport are 18.767749 and 98.9640088 respectively.

To make sure that I can use geospatial in MongoDB, I've decided to restructure the way latitude and longitude are stored to follow the GeoJSON format.

List of hotels near Chiang Mai international airport within 2 km.

```
> db.chiangmai.find({ "tourism": "hotel", "loc": { "$near": { "$geometry":
{ "type": "Point" , "coordinates": [AIRPORT_LNG, AIRPORT_LAT] },
"$maxDistance": 2000 } } }, { "_id": 0, "name": 1, "loc.coordinates": 1} )

[{u'loc': {u'coordinates': [98.974149, 18.763768]}, u'name': u'3b'},
{u'loc': {u'coordinates': [98.9780929, 18.7695194]}, u'name': u'Airport
Greenery Hotel'}]
```

At first, I've tried to find hotels near the airport within 1 km but it seems there is no hotel around it. So, I've decided to change from 1 km to 2 km instead and it found only 2 hotels. From this example, it means that we can use the geospatial in MongoDB to find places nearby our current location.

In my opinion, it may be a good idea if OpenStreetMap automatically fills in the missing data such as postal code, area, etc based on the latitude and longitude specified by users so that it guarantees that the postal code, area, etc. will be available in the dataset. However, it's also possible that those filled in data may contain the incorrect values due to the inaccurate latitude and longitude. So, it still requires contributors or reviewers to review these values manually even though OpenStreetMap may already had the form validation that automatically validates filled in data before storing the data in the database.

Conclusion

After working with this dataset, I found that the data is incomplete such as name, address, etc. So, I've cleaned some data to make sure it's ready to be analyzed. However, some of data can't be automatically cleaned. In some default attributes, some of them are in That but some of them are in English as discuss in section 1. It requires more contributors to review and audit the

data. The contributors can be tourists or people who live in Chiang Mai so that the data is more accurate. From the above exercises, even though this dataset is incomplete but it states that this dataset still provides the valuable data about Chiang Mai province that can be used in the real world projects, for example, we can create an application that allows users to find nearby restaurants or places in Chiang Mai.

Files & Folder

- data_wrangle_osm/lesson6 : Exercises in lesson 6
- data_wrangle_osm/data_wrangler.py : Data wrangler
- data_wrangle_osm/analyzer.py : Data analyzer
- data_wrangle_osm/data/chiang-mai_thailand.osm.zip : Raw data
- data_wrangle_osm/data/chiang-mai_thailand.osm.json.zip : Cleaned data
- data_wrangle_osm/data/sample.osm : Sample file of chiang-mai_thailand.osm

References

http://docs.mongodb.org/manual/
http://docs.mongodb.org/manual/reference/operator/query/near/#op._S_near
http://wiki.openstreetmap.org/wiki/Map_Features#References