OpenStreetMap Data Case Study

Map Area

Hong Kong

http://overpass-api.de/query_form.html
query: (node(22.142,113.803,22.534,114.420);<;);out meta;

I have chosen the Hong Kong dataset as this is my hometown and I would like to understand the place where I live from a data perspective. The dataset downloaded also include some of the area in China, which present additional challenges in parsing the data as convention of naming China's street is significantly different from that of Hong Kong.

Problems Encountered in the Map

In order to quickly understand the dataset and spot the issue with the data, I have used the code as provided in the project preparation section and extracted a small sample of data for examination. I have spotted X problems as listed below:

1. Inconsistent street name with abbreviations
2. Inconsistent street name with Chinese text and unrelated symbols
3. Inconsistent house numbers
4. Irregular website name
5. Irregular phone number

Inconsistent Street Name – Abbreviation

One common inconsistency of street name in the dataset is similar to that of the problem set, that the ending of the words are abbreviated. (e.g. Kafia Blvd, Longzhu 4th Rd). I would use the function already provided in the audit.py to correct the data.

Inconsistent Street Name – Chinese Text and other unrelated symbols

Street name contains a number of irregularities besides abbreviation. For example, Chinese texts or irregular symbols are included in the street name (e.g. "青山公路 – 深井段 Castle Peak Road - Sham Tseng", "Kwai Shing Circuit\").

```
street_name = re.sub(ur'[\u2013-\u9fff]+',"", unicode(street_name))
street_name = re.sub(r'(-?  )', "", street_name)
street_name = re.sub(r'([\\])', "", street_name)
street_name = street_name.lstrip()
```

Inconsistent House Number

House number in Hong Kong should only contain 1 – 3 numbers, one alphabet and/or related symbols (e.g. "-"). A function will be written in order to clear unrelated words.

My assumption is that the house number be a number containing one to three digits and a character (e.g. 1, 12, 123, 123A), or a range of number comprising of one to three digits, with occasional character at the end (1-1, 312 – 333, 22A – 22B).

I have designed a function which utilize regex to match and return the common pattern of house numbers. If I cannot search the mentioned pattern, I will set the housenumber as blank.

```
housenumber_re = re.compile(r'(^[0-9]{0,3}[A-Z]? ?-? ?[0-9]{1,3}[A-Z]?)|([0-9]{0,3}[A-Z]? ?-? ?[0-9]{1,3}[A-Z]?$)')

def clean_housenumber(housenumber):
    m = housenumber_re.search(housenumber)
    if m:
        correct_housenumber = m.group()
        return correct_housenumber
    else:
        housenumber = ""
        return housenumber
```

Irregular Phone Number

With a brief glance, I can spot that the phone number are not stored with the same format. I would like to fix the problem and ensure user of the file can easily compare the phone numbers. I will assume that all of the numbers are to be stored with the below format:

Hong Kong Number: +852 XXXX XXXX
Shenzhen Number: +86 0775 XXX XXX (Although the data is supposed to be Hong Kong data, some of the area in China is also covered from the MapZen dataset)

I would first use a simple regex format to strip all the phone numbers except the digit, then standardize the format of the numbers according to the length of the string.

```
def clean_phone(phone):
    phone = re.sub("\D", "", phone)
    if len(phone) == 8:
        phone = "+852 " + phone[0:4] + " " + phone[4:8]
    elif len(phone) == 11:
        phone = "+" + phone[0:3] + " " + phone[3:7] + " " + phone[7:11]
    elif len(phone) == 12:
        phone = "+86 0775 " + phone[4:8] + " " + phone[8:12]
    elif len(phone) == 13:
        phone = "+86 0775 " + phone[5:9] + " " + phone[9:13]
    else:
        phone = ""
    return phone
```

Irregular Website Format

Just like the phone numbers, the website are not in regular format. Some of which contains “http” at the beginning and some not. I would like to standardize the format to “http(s)://www.xxxxxxxxxxxxx”. Again,

I will first use regex to match the assumed pattern. If such pattern is not present, I will use several if lines to deal with the common errors.

```
website_re = re.compile(r'(http)(s?):[//][//]www[.].+')

def clean_website(website):
    m = website_re.search(website)
    if m is None:
        if website[0:3] =="www":
            website = "http://" + website[0:len(website)]
        elif website[0:4] == "http":
            website = "http://www." + website[7:len(website)]
        else:
            website = "http://www." + website[0:len(website)]
    else:
        website = m.group()
    return website
```

Data Overview and Additional Ideas

This section contains the basic statistics concerning the dataset, the SQL queries used to gather them, and some additional ideas about the data in context.

File sizes:

```
Hong_Kong.osm..... 225 MB
Hong_Kong.db..... 162 MB
Nodes.csv..... 83 MB
Nodes_tags.csv..... 4.6 MB
Ways.csv..... 6.5 MB
Ways_nodes.csv..... 30 MB
Ways_tags.csv..... 13 MB
```

Number of nodes:

```
sqlite> SELECT COUNT(*) FROM nodes;
```

1033451

Number of ways:

```
sqlite> SELECT COUNT(*) FROM ways;
```

113142

Number of banks in nodes:

```
sqlite> SELECT count(value) FROM nodes_tag WHERE value = 'bank' GROUP BY value;
```

331

Number of bridge in ways:

```
sqlite> SELECT count(key) FROM ways_tags WHERE key = 'bridge' GROUP BY key;
```

5928

Number of unique users:

```
sqlite> SELECT COUNT(DISTINCT(e.uid)) FROM (SELECT uid FROM nodes UNION ALL SELE
CT uid FROM ways) e;
```

1134

Top ten contributing users

```
sqlite> SELECT e.user, COUNT(*) as num
FROM (SELECT user FROM nodes UNION ALL SELECT user FROM ways) e
GROUP BY e.user
ORDER BY num dESC
LIMIT 10;
```

hlaw	491759
KX675	78948
MarsmanRom	40234
bTonyB	40006
jc86035	38651
"Philip C"	36643
OmniBusSun	31627
R17466	30792
ystsoi	27088
Mapier	24292

Additional ideas

Include population related data in the map

Include Hong Kong census data to the Openstreetmap dataset to allow additional analysis by relating these demographic data different nodes/ways indicating physical infrastructure. The investigations can draw relationship between population and local regional environment. Example analysis will be to check the number of number of banks to population and see if the population is sufficiently served financially.

Benefit

- Additional layer of analysis to understand relationship between cityscape and population

Anticipated problems

- Data from government not user friendly and require additional processing
- Openstreetmap data is incomplete and may not facilitate the desired analysis

Clear data submitted by non-frequent users

By running the below query, there are 918 users who are contributing less than 100 data. From a common sense perspective, these infrequent users can be perceived as unprofessional data contributor and may add bad quality data to the dataset. We can clear these data or extract these data for extra auditing.

```
sqlite> SELECT COUNT(*) FROM
(SELECT e.user, COUNT(*) as num
FROM (SELECT user FROM nodes UNION ALL SELECT user FROM ways) e
GROUP BY e.user
HAVING num<100) u;
```

Benefit

- Improve overall data accuracy by removing data from amateur contributor
- Remove nodes which are potentially created out of vandalism

Anticipated Problems

- Infrequent users may also be contributing useful data. Cleaning these data may affect the completeness of the dataset

Discard nodes with low accuracy and de-duplicate nodes with similar location

According to the information I have gathered form different sources, 6 decimal place GPS location (accuracy ~ 0.11m) is a good enough measure for creating maps. As a result, I would discard all nodes with less than six decimal place GPS locations and the remaining to six decimal place. If nodes are found to be of the same location, deduplication will be conducted and a more experienced contributor (based on number of contributions) will be adopted.

Benefit

- Standardize accuracy of every nodes or ways and ensure all data are referring to the same scale
- Remove duplicating data and prevent confusing analysis

Anticipated Problems:

- Frequent contributors may not always be correct. Current rules of believing frequent contributors may remove actual correct data
- GPS location with six decimal place may not be accurate enough to describe specific nodes. De-duplication process may be based on false assumption and useful data are eliminated.

Conclusion

After having an understanding over the Hong Kong Openstreetmap dataset, I am quite certain that the data is not complete. However, I think the dataset still serve as a useful base in understanding the city from a data perspective. More insights can be drawn if we can enrich the data by attracting more contributor and overlay other data on the Openstreetmap dataset for analysis purpose.

Reference

http://stackoverflow.com/questions/2718196/find-all-chinese-text-in-a-string-using-python-and-regex

http://stackoverflow.com/questions/1450897/python-removing-characters-except-digits-from-string

http://stackoverflow.com/questions/1240504/regular-expression-to-match-string-starting-with-stop

https://discussions.udacity.com/t/workflow-for-project/182916/12

https://discussions.udacity.com/t/cleaning-street-names/187411/2

https://discussions.udacity.com/t/upload-my-csv-files-sqlite/190795/2

http://gis.stackexchange.com/questions/8650/measuring-accuracy-of-latitude-and-longitude