Case study - Open Street Map data

Introduction

This is a case study of some map data from the openstreetmap.org. We'll parse the data from an XML file exported from their website, clean up two problems we have found in the data, import it into an SQL database, and explore the data.

In this case study we'll work with the map data of Hong Kong. http://www.openstreetmap.org/node/2833125787 (http://www.openstreetmap.org/node/2833125787)

Specifically the map data spans latitudes 22.1543 to 22.5620, longitudes 113.7964 to 114.4281, for an area of around 2880 square km. A small proportion of it will be Shenzhen, China, that lies right across the border to the north.

Hong Kong has two official langauges - English and Chinese, and is a separate jurisdiction from China.

These characteristics will present interesting challenges in cleaning the data, as we shall see.

Problems found in the data

Street names and languages

The tagging conventions of Open Street Map (OSM) for Hong Kong is to have three tags. name= *for the name in Traditional Chinese followed by a space and then the name in English; name:en*= for the English name; and name:zh=* for the Chinese name.

https://wiki.openstreetmap.org/wiki/Multilingual_names#Hong_Kong (https://wiki.openstreetmap.org/wiki/Multilingual_names#Hong_Kong)

Many streets in the OSM data set have the following problems:

- have either the name:en= *or name:zh*= missing
- the name=* tag contain only the name in one language
- have typos
- have the wrong translation or transliteration for the name in either language
- have variants of Chinese characters rather than the official ones (government named - afterall it's the government that names all the streets), which are still correct language-wise, but are considered separate characters in unicode. In plain string comparisons they will be different from the official version, while in smarter systems like google, they may or may not get a match.

The solution

The Lands Department of the Hong Kong government publishes data of the official names of all the roads, streets, lanes, etc. in both Chinese and English. http://www.landsd.gov.hk/mapping/en/download/psi/opendata.htm (http://www.landsd.gov.hk/mapping/en/download/psi/opendata.htm)

We will refer this to the 'official list' from now on. It will be used as a reference to audit and clean the street names in the OSM data.

Audit

With the script audit_bilingual_street_names.py, we can see the entries with the problems mentioned above.

Below is a part of the results from the audit.

- en_only and zh_only are the values in the name:en=* and name:zh=* tags in the OSM file.
- reg_eng and reg_chi are segments of the value in the name=* tags, splitted into English and Chinese parts.
- look_up_results is the matching entry in the Lands Department street name data

	id	key	type	value
13	1168103126	phone	regular	27898521
27	1298704683	phone	regular	+852 25290901
36	1694252796	phone	regular	+8613590258862
56	2102316713	fax	regular	+852 25258355
58	2155650086	fax	regular	+852 26686643
83	2438453080	phone	regular	+852 3982 9999
289	4264504091	phone	regular	+85222923000

Problems with the official list

The official list isn't perfectly clean either. There are entries with typos and trailing spaces, as well as duplicate entries (same Chinese and English names, referring to a single street)

	en_only	reg_eng	zh_only	reg_chi	look_up_results
0	Aberdeen Tunnel	Aberdeen Tunnel	香港仔隧道	香港仔隧道	[香港仔隧道, Aberdeent Tuntntel]
71	Wan Chai Interchange	Wan Chai Interchange	灣仔交匯處	灣仔交匯處	[灣仔交匯處, Wan Chai Interchantge]

A handful of streets in Hong Kong have identical names in one langauge but not the other.

Consider this trio (first two have the same Chinese names): Central Road 中間道 Middle Road 中間道 Middle Road 中間路

They refer to three different roads. If an OSM road has only the tag name="Middle Road" or name="中間道", there is no way we can tell which road the tag is refering to, within the scope of this study.

In addition, a few streets in Hong Kong share Chinese names with streets in the part of Shenzhen the map includes, again within the scope of this study we cannot audit streets with these names.

Cleaning

We are going to modify the official list as follows before using it as a reference to audit and clean our OSM data:

- delete all extra copies of perfect duplicates
- correct the typos
- discard all entries that share names in any language with other entrie(s)
- discard all entries that share names with streets in the part of Shenzhen that our map inevitably includes

We then update the street names with the following function. It updates a list of dicts, with each dict being a tag and the list being a way parsed from the OSM XML file.

We call this function in the process_map() function, update the way_tags we have parsed with it, before writing to the csv files. We'll also record the update in a csv file, with the id, element type (way in this case), and the field updated (name in this case).

```
def fix_street_names(way_tags):
    '''
    Takes a list of way_tags and update the names if it's a street
    if possible, and add tags if any of the name tags is missing
    '''
    if is_street(way_tags):
        osm_names = get_street_names(way_tags)
        look_up_result_index, not_found_count = name_look_up(osm_names)
    else:
        return way_tags

    # if there is no match, or multiple(contradicting) matches, we can't
    # decide which so we'll skip those
    if len(look_up_result_index) != 1:
        return way_tags

    index = look_up_result_index.pop()
    way_id = way_tags[0]['id']
    eng_name = index_to_name[index]['eng']
    chi_name = unicode(index_to_name[index]['chi'])
    reg_name = chi_name + ' ' + eng_name
    eng_missing, chi_missing, reg_missing = True, True, True

    # overwrite with the official name if the tag exist
    for tag in way_tags:
        if tag['type'] == 'name' and tag['key'] == 'en':
            tag['value'] = eng_name
            eng_missing = False
        if tag['type'] == 'name' and tag['key'] == 'zh':
            tag['value'] = chi_name
            chi_missing = False
        if tag['type'] == 'regular' and tag['key'] == 'name':
            tag['value'] = reg_name
            reg_missing = False

    # if they don't, we add the tag
    if eng_missing:
        way_tags.append({
            'id': way_id, 'type': 'name', 'key': 'en', 'value': eng_name
        })
    if chi_missing:
        way_tags.append({
            'id': way_id, 'type': 'name', 'key': 'zh', 'value': chi_name
        })
    if reg_missing:
        way_tags.append({
            'id': way_id, 'type': 'regular', 'key': 'name',
            'value': reg_name
        })
    return way_tags
```

Phone number formats

The second problem we have noticed with the OSM data is in the formats of the phone number data.

In OSM's tagging conventions, phone numbers should have the format of "+<country code> <area code> <number>" with the options of using the hypen (-) instead of the space as separators, and segmenting the number with the same separators according to very well established national standards. Multiple values should be delimited by semicolons.

https://wiki.openstreetmap.org/wiki/Key:phone (https://wiki.openstreetmap.org/wiki/Key:phone)

The country code of Hong Kong is 852 and all numbers, cellular or landline (barring a few government hotlines) have 8 digits. The country code of the PRC is 86, and the area code of Shenzhen in 755 In the People's Republic of China (PRC), all cell phones have 11 digits and starts with 13 to 19 (as of January 2018) and are nation wide and does not have an area code. Landlines, on the other hand, have an area code, followed by a local number of up to 8 digits. Those in Shenzhen may have anywhere from 6 to 8 digits.

In our OSM data set, phone numbers can:

- have parantheses around the country code and/or the area code, instead of or in addition to the plus sign
- have the country code missing
- have commas as delimiters between values
- have unconventional and meaningless segmentation
- contain non-ASCII characters. In one entry, the Chinese full width cross sign ＋ (unicode 65291 in Dec) is used instead of the ASCII plus sign (+)
- be either a Hong Kong number, a PRC cell number or a Shenzhen landline number, all of which we plan to reformat
- have a redundant 0 in front of the area code for Shenzhen. It is a signal for making an inter-area call within the PRC, just like the + sign we dial before the country code when calling long distance.
- be a number of yet another country, whose formatting is out of the scope of this study

Audit

We'll first audit the data by using regex to look for any way tags and node tags that have a value that looks like a phone number(s) to take a look at how the values look like, as well as what keys do these tags have. We'll also look at any tags that have the key "phone" to pick up phone number format patterns we may have missed with the regex.

The following are some examples from the audit results.

	id	key	type	value
1	278953259	phone	regular	2196 8170
0	26412171	phone	regular	+ 852 2522 0922
5	629118678	phone	regular	+852 28937565
11	1080024761	phone	regular	+85227822682
52	2089466239	phone	regular	(+852) 2529 9280
53	2089466240	phone	regular	(852) 2526 9215
63	2247751944	source	regular	852-27892280
216	3902902464	phone	regular	+85 22 19 21222
129	2850358549	phone	regular	852-2522-1184
134	2937037495	phone	regular	+852 31051830, +852 31041831, +852 31051832
150	3174799316	phone	regular	+86 755 961200
154	3233923021	phone	regular	+86 0755-86378888
266	4236943490	phone	regular	0755 8618 4166
324	4332469220	phone	regular	(86 755) 8298-9888
3	418074992	fax	regular	86-755-82318399
84	2466257098	phone	regular	+86 0755 8295 3332
89	2526452599	phone	regular	+86 755 28011122
634	239837956	phone	regular	＋8675526719666
498	4894691821	phone	regular	13714841831
195	3724774689	phone	regular	+8613554775581
174	3565049095	phone	contact	+41 44 586 00 04
75	2411879109	phone	source	survey

Cleaning

After the audit we have a list of keys whose corresponding value may contain phone numbers, and we know all the possible formats the phone numbers can be in.

We then use following function to reformat the phone numbers in the OSM conventional format. It will be called by another function fix_phones_in_tags(tags) which passes it the phone number string to be updated. As with names, we'll record the update in update_history.csv

```
HK_PHONE_STRIPPED_RE = re.compile(r'^(852)?(\d{8})$')
PRC_CELL_STRIPPED_RE = re.compile(r'^(86)?(1[3-9]\d{9})$')
SZ_LAND_STRIPPED_RE = re.compile(r'^(86)?0?(755)(\d{6,8})$')
NON_DIGIT_CHAR_RE = re.compile(u'[- +)(＋]+')
DELIMITERS_RE = re.compile(',|;')

def fix_phone_value(in_string):
    '''
    Takes a phone number value that may contain multiple numbers
    separated by commas or semicolons, change them to the format
    recommended by OSM and return.
    '''
    phone_list = []
    out_string = ''
    for value in DELIMITERS_RE.split(in_string):
        stripped = re.sub(NON_DIGIT_CHAR_RE, '', value)
        m = HK_PHONE_STRIPPED_RE.search(stripped)
        if m:
            phone_list.append('+852 ' + m.group(2))
            continue

        m = PRC_CELL_STRIPPED_RE.search(stripped)
        if m:
            phone_list.append('+86 ' + m.group(2))
            continue
        m = SZ_LAND_STRIPPED_RE.search(stripped)
        if m:
            phone_list.append('+86 755 ' + m.group(3))
    if phone_list:
        for phone_number in phone_list:
            out_string = out_string + phone_number + ';'
        out_string = out_string[:-1]
    else:
        out_string = in_string
    return out_string
```

Data overview

File sizes

Hong_Kong.osm - 306MB

Hong_Kong.db - 216MB

nodes.csv - 112MB

nodes_tags.csv - 7MB

ways.csv - 9MB

ways_nodes.csv - 38MB

ways_tags.csv - 17MB

update_history.csv - 19KB

Number of ways

```
query = '''
SELECT COUNT(*)
FROM ways;
'''
display(query)
```

```
161676
```

Number of nodes

```
query = '''
SELECT COUNT(*)
FROM nodes;
'''
display(query)
```

```
1419739
```

Number of unique users

```
query = '''
SELECT COUNT(DISTINCT uid)
FROM (SELECT uid FROM nodes UNION ALL SELECT uid FROM ways);
'''
display(query)
```

```
1652
```

Number of roads / streets with the name(s) updated

```
query = '''
SELECT COUNT(*)
FROM update_history
WHERE field_updated = 'name';
'''
display(query)
```

```
484
```

Number of phone numbers updated

```
query = '''
SELECT COUNT(*)
FROM update_history
WHERE field_updated = 'phone';
'''
display(query)
```

```
439
```

Additional exploration

Users and the number of updates we've made to their features, by their total number of contributions to OSM in the data set

```
query = '''
SELECT subq_a.uid, updates, contributions
FROM (
    SELECT uid, COUNT(*) AS updates
    FROM (
        SELECT w.uid as uid
        FROM update_history AS uh
        JOIN ways AS w
        ON w.id = uh.id
        WHERE uh.element_type = 'way' and uh.field_updated = 'name'
        UNION ALL
        SELECT n.uid AS uid
        FROM update_history AS uh
        JOIN nodes AS n
        ON n.id = uh.id
        WHERE uh.element_type = 'node' and uh.field_updated = 'name'
    ) AS subq_c
    GROUP BY uid
) AS subq_b
LEFT JOIN (
    SELECT uid, COUNT(*) as contributions
    FROM (
        SELECT w.uid AS uid
        FROM ways as w
        UNION ALL
        SELECT n.uid as uid
        FROM nodes as n
    )
    GROUP BY uid
) AS subq_a
ON subq_b.uid = subq_a.uid
ORDER BY contributions DESC
;
'''
display(query)
```

169827	24	320589
1787380	14	149037
3306940	17	90942
6172648	56	73880
31685	5	67398
6095805	25	43700
6346054	9	42846
4618964	8	36937
4892967	1	35776
1242214	40	33930
2238851	7	30814
6095772	20	30259
261189	4	28033
1265238	2	21429
541214	2	17069
6946532	5	12892
6172587	11	12471
1870014	10	10085
6095754	5	8634
1832122	15	7910
6346034	2	6904
460264	3	6840
6808315	1	4790
6172866	1	4230
4547543	19	2780
4725981	1	2205
3778777	8	1986
3317733	13	1896
3569951	1	1597
3742941	22	1567
5722166	5	1299
3319178	16	1242
5027347	17	1142
3474216	31	1013
678519	1	976
3544143	21	946
126508	7	772
4471944	4	760
4240913	2	683
3275816	1	648
3321532	1	552
3321091	2	459
4816761	1	380
4498020	2	364
1931010	1	265
5725727	2	237
675538	7	224
307415	1	186
711265	2	179
2226712	1	178
101184	1	82
145231	1	82
3587625	1	30
3248699	1	26
44457	3	20
2385962	1	5

We can see that many of the features we have updated are contributed by very frequent users, most likely with automation.

Names of other features

After our audit and updates, it's natural to wonder if the same problem exists in the names of other ways. Here we'll look at buildings and amenities.

Number of buildings or amenities with at least one name:

```
query = '''
SELECT COUNT(*)
FROM ways_tags
WHERE (key = 'amenity' OR key = 'building') AND ways_tags.id IN (
    SELECT distinct id
    FROM ways_tags
    WHERE (key = 'en' AND type = 'name') OR
    (key = 'zh' AND type = 'name') OR
    (key = 'name' AND type = 'regular')
);
'''
display(query)
```

```
17201
```

Number of buildings or amenities with either the keys name, name:en or name:zh missing:

```
query = '''
SELECT COUNT(*)
FROM ways_tags
WHERE (key = 'amenity' OR key = 'building') AND ways_tags.id NOT IN(
    SELECT a.id
    FROM ways_tags AS a
    JOIN ways_tags AS b
    ON a.id = b.id
    JOIN ways_tags AS c
    ON a.id = c.id
    WHERE a.key = 'en' AND a.type = 'name'
        AND b.key = 'zh' AND b.type = 'name'
        AND c.key = 'name' AND c.type = 'regular'
) AND ways_tags.id IN (
    SELECT distinct id
    FROM ways_tags
    WHERE (key = 'en' AND type = 'name') OR
    (key = 'zh' AND type = 'name') OR
    (key = 'name' AND type = 'regular')
);
'''
display(query)
```

```
3224
```

Quite a portion of the features - which should have names in both languages in Hong Kong, are not tagged according to OSM's conventions. Sometimes it is simply that the name in either language is not available to the user, but often they are both there, just not put into the right tags. This seems to happen even with frequent users doing automated edits.

Recommendations

Audits and updates for the consistency of multilingual names for buildings and amenities can be done by obtaining data from other government bodies.

To further improve the data, we suggest that data from the Rating and Valuation Department used. They provide lists of names of all buildings, in both Chinese and English. The data can be scraped and used in a similar fashion as above.
http://www.rvd.gov.hk/en/public_services/property_information.html
(http://www.rvd.gov.hk/en/public_services/property_information.html)

The advantages of this approach include:

- They come with the address so there is an easy way to tell which building an entry is, even in cases where there are buildings sharing the same name or other ambiguities.
- Contents from these sources usually come with less restrictive terms of use.

Possible drawbacks are:

- The data comes in pdf and word document files, not the most machine-friendly formats. Some coding may be required to scrape the data from the files.
- Despite being data from a government body, the names in it are not guaranteed official as they are voluntarily submitted by developers.

Conclusion

We can see that OSM has acquired a wealth of data contributed by users, and at the same time much of this data is not perfectly clean.

Our phone number format update should have cleaned up all the Hong Kong phone numbers, while the street names update is limited to streets that can be matched to an official name.

The OSM data in Hong Kong is certainly not complete, and our cleaning has its limitations. But we hope we have helped making it better than when we started.