OpenStreetMap Project

Map Area

Chaozhou city, China

- https://www.openstreetmap.org/relation/3244431#map=9/23.8293/116.7767

This map is my hometown, I want to know more about it through wrangling its data and improve data to let others get the right information of my hometown.

Problems Encountered in the Map

After initially downloading chaozhoucity.osm HTML file,I used find_problems.py to grasp the value of some key and I noticed five main problems with the data, which I will discuss in the following order:

- Inconsistent phone numbers:

```
0592-6079482
+86 768-8351022
+86 753 2836666
17806732721
+865922022922
+86 592 511 2323
+86(0)5926888111
865923361666
```

- Inconsistent street names

```
龙塘大道 Longtang Avenue
饶平县三饶镇城基西路 Chengji W. Rd.
S222
民族路 Mínzú Road
Fuxing lu
黄冈大道
```

- Incorrect street name:

```
汤溪镇
锡西村五间过围
```

- Inconsistent city names:

```
Meizhou
Xiamen (Fujian)
潮州市
```

Inconsistent phone numbers

To make phone numbers consistent, I use regular function and python strings operation to wrangle it.

```
SUBTRACT = re.compile(r'\-')
WHITESPACE = re.compile(r'\s')
BRACKET = re.compile(r'\)')
```

```
def update_phone(phone):
    """check the phone which has problem pattern and update it"""

    if phone.startswith("0"):
        # remove initial "0" from string
        phone = phone.lstrip("0")
    if BRACKET.search(phone):
        # remove "(0)" in the middle of string
        phone = phone.replace("(0)", "")
    if SUBTRACT.search(phone):
        # remove "-" in the middle of string
        phone = phone.replace("-", "")
    if WHITESPACE.search(phone):
        # remove all whitespace in string
        phone = phone.replace(" ", "")
    if len(phone) == 11 or len(phone) == 10:
        # add "+86" for moilephone phone number(11-digit) and landline number(10-digit) which
without "86"
        phone = "+86" + phone
    if not phone.startswith("+"):
        # add "+" for string wfieldhich has "86" without "+"
        phone = "+" + phone
    if len(phone) == 13:
        # make landline number(13-digit) format consistent
        phone = phone[:3] + ' ' + phone[3:6] + ' ' + phone[6:]
    if len(phone) == 14:
        # make mobilephone number(14-digit) format consistent
        phone = phone[:3] + ' ' + phone[3:]
    return phone
```

This change all inconsistent phone numbers into two format:

```
mobilephone  '17806732721' >> '+86 17806732721',
landline  '865922022922' >> '+86 592 2022922',
```

Inconsistent and incorrect street names

I am interested in street names and find out all street names in chaozhoucity.osm:

```
def tag_key_value(input_file, k):
    """find the value and its amount correspond to key"""

    value = {}
    for event, elem in ET.iterparse(input_file):
        if event == 'end':
            if elem.get('k') == k:
                k_value = elem.get('v')
                try:
                    # if the key already exist, key's amount += 1
                    value[k_value] += 1
                except KeyError:
                    # if the key does not exist, key's amount = 1
                    value[k_value] = 1
        elem.clear() # discard element

    # make a dataframe consist of value and its amount
    value_df = pd.DataFrame(value, index=['amount']).T

    return value_df
```

```
street_value_df = tag_key_value(OSM_FILE, "addr:street")
```

After getting the street names, I notice most of the street names only include their simplified Chinese names, but some of them are mixed with Chiese(simplified and tranditional), Pinyin, and English.Some of their English are over-abbreviated.

```
黄冈大道
民族路 Mínzú Road
Fuxing lu
龙塘大道 Longtang Avenue
饶平县三饶镇城基西路 Chengji W. Rd.
```

Some highway names only have their corresponding codes, but others have their description before their corresponding codes.

```
县道X086
S222
```

Regardless of those faults, some street names are area names.

```
汤溪镇
锡西村五间过围
角美台商開發區文圃工業園
```

Inconsistent city names

Here is a sample of the results which indicates some city names only have their Pinyin names and miss their Chinese names.

```
Meizhou
Xiamen (Fujian)
潮州市
```

Data Overview

This section contains basic statistics about the dataset, the Sqlite queries used to gather them

File sizes

```
chaozhoucity.osm ......... 89.7 MB
chaozhoucity.db .......... 46 MB
nodes.csv ............. 36.1 MB
nodes_tags.csv ........ 0.60 MB
ways.csv .............. 1.7 MB
ways_nodes.cv ......... 11.9 MB
ways_tags.csv ......... 2 MB
```

Number of unique users

```
sqlite>
SELECT count(*)
FROM
```

```
    (SELECT nodes.uid
    FROM nodes LEFT JOIN
            (SELECT uid FROM ways GROUP BY uid) as ways_uid
    on nodes.uid = ways_uid.uid
    GROUP BY nodes.uid) as ways_and_nodes_uid;
```

```
267
```

Number of nodes

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

```
433300
```

Number of ways

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

```
28529
```

Top 15 appearing amenities

```
sqlite>
SELECT value, count(*) as num
FROM nodes_tags
WHERE key='amenity'
GROUP BY value
ORDER BY num DESC
LIMIT 15;
```

```
restaurant|42
bank|38
toilets|24
fuel|20
place_of_worship|20
hospital|16
pharmacy|15
ferry_terminal|14
bus_station|12
fast_food|12
school|12
parking|11
cafe|8
marketplace|8
police|7
```

Additional Data Exploration

Names of company

```
sqlite>
SELECT value
FROM
    (SELECT *
    FROM nodes_tags,
        (SELECT * FROM nodes_tags WHERE value = 'company' ) as company_id
    WHERE nodes_tags.id = company_id.id) as company_all
WHERE key = 'name';
```

```
潮州市三元陶瓷（集团）有限公司
广东永丰源陶瓷有限公司
潮州亚太集团有限公司
```

Additional Ideas

Set a word limit for the street names column

When I query the street names in nodes_tags, I notice some nodes that their very long street names are very long which are too ugly:

```
con = sqlite3.connect('chaozhoucity.db')
cur = con.cursor()
cur.execute("select * from nodes_tags where key = 'street';")
nodes_tags_street = cur.fetchall()

for tu in nodes_tags_street:
    if len(tu[2]) > 20:
        print tu[0], '|', tu[2]
```

```
4828099522 | Jixiang Mansion （Jianye Road）, 3 Jianye Rd, Siming Qu, Xiamen Shi, Fujian Province,
China, 361006
4860779222 | 208 Binhu North Road, Haicang District, Xiamen, Fujian province, China, 361026 海沧正元
逸林希尔顿
4860783425 | 15, North Bin Hu Road, Haicang District, Xiamen
```

I query the full information of one of them:

```
cur.execute("select value from nodes, nodes_tags where nodes.id = nodes_tags.id and nodes.id =
4828099522  ")
nodes_tags_street_long = cur.fetchall()
pretty_print(nodes_tags_street_long)
```

```
361006 |
Jixiang Mansion （Jianye Road）, 3 Jianye Rd, Siming Qu, Xiamen Shi, Fujian Province, China, 361006
|
Rosewood Lakeview Hotel |
+86 592 5112323 |
hotel |
https://www.tripadvisor.de/Hotel_Review-g297407-d610352-Reviews-Rosewood_Lakeview_Plaza-
Xiamen_Fujian.html |
```

Now I know some nodes' typers may type the full adress into their street names. Probably they just want others to know their adress explicitly and find them quickly, but they also make the data more dirty.

I suggest that the map should set a word limit for the street names column.

The benefits are:

- It can make the data more cleaned before we wrangle it.
- It can let people just type the significant thing into it without redundant information

Some anticipated problems in implementing the improvement:

- The word limit can not be perfect. If it is too small, some long street name will lose; if it is too big, it is no meaning for setting this limit. To decide the limit, we should get an amount of street names as sample. This procedure will cost lots of efforts and this also relate to machine learning to decide a tradeoff.

Conclusion

After this review of the data, I realized that different languages and different traditional ways of formatting is one of the most difficult things to deal with in a map, we can see that a quantity of information are mixed with Chiese(simplified and tranditional), Pinyin, and English. And probably it is not fair if we just use one format or one language as a standard for people who are from different languages countries.