OpenStreetMap Data Case Study

7point5 (/u/27d144331d89)
2018.02.26 20:55 字数 855 阅读 0 评论 0 喜欢 0
(/u/27d144331d89)

编辑文章 (/writer#/notebooks/22666263/notes/24470518)

OpenStreetMap Data Case Study

Map Area

Isle of Wight, England

- Relation: https://www.openstreetmap.org/relation/154350 (https://link.jianshu.com?t=https%3A%2F%2Fwww.openstreetmap.org%2Frelation%2F154350)
- Download Link: https://download.geofabrik.de/europe/great-britain/england/isle-of-wight.html (https://link.jianshu.com?t=https%3A%2F%2Fdownload.geofabrik.de%2Feurope%2Fgreat-britain%2Fengland%2Fisle-of-wight.html)

The Isle of Wight is an island in south England I always want to visit. I am looking forward to digging out some interesting information as well as helping out improving OpenStreetMap.

1. Problems Encountered in the Map

- irregular street names (*Leeson Roadhttp://wightpaths.co.uk/rowmaptiles/ (https://link.jianshu.com?t=Roadhttp%3A%2F%2Fwightpaths.co.uk%2Frowmaptiles%2F){zoom}/{x}/{y}.png*)
- inconsistent street name spelling ('King Edwards Close', 'Littletown lane')
- inconsistent attribute spelling
- incomplete postcode

1.1 irregular street names:

When I ran an audit of the stree names, I used a defaultDict(int) to check the street types. If the name has a white space in it, i added the last part to the defaultDict. For those has only one word for their street names, I add the whole name into the defaultDict. I found an unusually long one, i.e. normally it is a short word like street or road. To correct this type of errors, I inserted a re function and split function into the shape_element function to separate the street name and kept the correct name before export the data.

```
'''re and split functions i used to correct the data;
item is a node of the document'''
pattern = re.compile(r'.*http')

if item.attrib['k'] and item.attrib['k'] == 'addr:street' and pattern.search(item.attrib['v'
    item.attrib['v'] = item.attrib['v'].split('http')[0]
```

1.2 inconsistent street name spelling

Some street names only has the first letter capitalized ('Littletown lane'). I unified the format by using split() and upper().

```
'''key is the value for attribute 'k' in <tag>;
value is the value for attribute 'v' in <tag> '''
if 'addr:street' in key and ' ' not in value:
    value = value[0].upper() + value[1:]
elif 'addr:street' in key and ' ' in value:
    value = ' '.join([i[0].upper() + i[1:] for i in value.split(' ')])
```

1.3 inconsistent attribute spelling

During auditing tag attributes, it turned out that some tags' **k** attributes are spelt as '**addr:Street**' whereas some others are spelt like '**addr:street**'. This could potentially cause trouble when analysing data with different types of spellings. To fix it, I used *.upper()* before sepeparate the **colons(:)** between certain **k** tags.

1.4 incomplete postcode

UK postcodes are alphanumeric. Each post code is divided into two parts separated by a single space: the outward code and the inward code respectively. 4 postcodes were found with only outward half, e.g. 'PO39'.

After chekcing the addresses with incomplete postcodes, it turned out that those addresses are neighbourhood in which they reside. Thus they should be updated with the same postcodes (PO39 0EX).

```
'''check the geo info of the locations with incomplete postcodes'''
sqlite> SELECT id, key, type, value FROM node_tags
WHERE id IN (SELECT id FROM node_tags WHERE key = 'postcode' AND value = 'PO39') ;
```

```
'''update postcodes'''
sqlite> UPDATE node_tags SET value = 'PO39 0EX'
WHERE id IN (select id from node_tags where key = 'postcode' and value = 'PO39')
AND key = 'postcode';
```

2. Data Overview

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

2.1 File sizes

isle-of-wight-latest --------- 127 MB
data.db --------------------- 138 MB
nodes.csv ------------------- 47.6 MB
nodes_tags.csv ------------- 19 MB
ways.csv -------------------- 5.9 MB
ways_tags.csv -------------- 20.7 MB
ways_nodes.csv ------------ 144.6 MB

2.2 Number of documents in the osm file

```
def file_check(osm_file):
    data = defaultdict(int)
    tree = ET.parse(osm_file)
    for item in tree.iter():
        data[item.tag]+=1
    return data
```

'osm': 1,
'bounds': 1,
'node': 588853,
'tag': 259899,
'way': 101359,
'nd': 759141,
'relation': 1691,
'member': 20227

2.3 Number of nodes

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

588853

2.4 Number of ways

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

101359

2.5 Number of unique users

```
select count(distinct(uid)) from (select uid from nodes union all select uid from ways);
```

507

2.6 Top 10 contributing users

```
select user, count(1) as num
from (select uid, user from nodes union all select uid, user from ways)
group by user
order by num desc
limit 10;
```

```
"Mappernerd"    374511
"iwhs"  122522
"dmgroom_ct"    81467
"dmgroom"   18411
"Andy Street"   9933
"jpennycook"    7831
"Rondon237" 6705
"DrMark"    6445
"UniEagle"  6395
"Gostman"   5028
```

2.7 Number of users only contributed 1 post

```
select count(*)
from (select user, count(1) as num
from (select uid, user from nodes union all select uid, user from ways)
group by user
having num = 1)
;
```

66

3. Additional Data Exploration

3.1 Number of tag types

```
select count(*)
from (select *
from (select key, value from node_tags union all select key, value from way_tags)
group by key)
;
```

571

3.2 Top 10 appearing tag types

```
select key, count(1) as num
from (select key, value from node_tags union all select key, value from way_tags)
group by key
order by num desc
limit 10
;
```

```
"building"  53454
"source"    28232
"highway"   21093
"natural"   14638
"barrier"   14490
"name"          12105
"city"          9622
"street"    9144
"access"    5825
"housenumber"   5129
```

3.3 Number of building types

```
select count(*)
from (select *
from (select key, value from node_tags union all select key, value from way_tags)
where key = 'building'
group by value)
;
```

46

3.4 Top 10 building types

```
select value, count(1) as num
from (select key, value from node_tags union all select key, value from way_tags)
where key = 'amenity'
group by value
order by num desc
limit 10
;
```

```
"yes"   22528
"residential"   18364
"house" 8861
"chalet"    2178
"apartments"    323
"garage"    291
"hut"   207
"commercial"    153
"terrace"   124
"barn"  78
```

3.5 Number of data provided by The National Public Transport Access Nodes (NaPTAN)

```
select count(*)
from (select *
from (select * from node_tags union all select * from way_tags)
where type = 'naptan'
group by id)
;
```

1335

3.6 General information of naptan locations

```
select key, count(1) as num
from (select *
from (select * from node_tags union all select * from way_tags)
where id in
(select id
from (select * from node_tags union all select * from way_tags)
where type = 'naptan'
group by id)
and type = 'regular')
group by key
order by num desc
;
```

```
"name"  1334
"source"    1328
"highway"   1321
"shelter"   82
"operator"  12
"note"  11
"bench" 9
"local_ref" 9
"alt_name"  8
"railway"   4
"network"   2
"physically_present"    2
"wheelchair"    2
"covered"   1
"fixme" 1
"seats" 1
```

3.7 The naptan location without a name attribute

```
select *
from (select * from node_tags union all select * from way_tags)
    where id in
        (select id
        from (select *
            from (select * from node_tags union all select * from way_tags)
            where id in
                (select id
                from (select * from node_tags union all select * from way_tags)
                where type = 'naptan'
                group by id
                )
        group by id
        )
    )
and id not in
(select id
from (select *
    from (select * from node_tags union all select * from way_tags)
    where id in
        (select id
            from (select * from node_tags union all select * from way_tags)
        where type = 'naptan'
        group by id
        )
    )
where key = 'name'
)
;
```

```
"533813619" "source"    "regular"   "naptan_import"
"533813619" "Street"    "naptan"    "East Street"
"533813619" "Bearing"   "naptan"    "SW"
"533813619" "AtcoCode"  "naptan"    "230000007750"
"533813619" "verified"  "naptan"    "no"
"533813619" "Indicator" "naptan"    "Adj"
"533813619" "CommonName"    "naptan"    "Cineworld"
"533813619" "NaptanCode"    "naptan"    "iowgjgt"
```

3.8 The naptan stop that needs to be fixed

`tag attributes:`

```
select key, value
from (select * from node_tags union all select * from way_tags)
where id in
    (select id
    from (select *
        from (select * from node_tags union all select * from way_tags)
        where id in
            (select id
            from (select * from node_tags union all select * from way_tags)
            where type = 'naptan'
            group by id
            )
        and key = 'fixme'
        )
    )
;
```

```
"name"  "Bannock Road South"
"fixme" ",ay not be in use"
"source"    "naptan_import"
"highway"   "bus_stop"
"Street"    "Bannock Road"
"Bearing"   "S"
"AtcoCode"  "230000005890"
"verified"  "yes"
"Indicator" "towards Blackgang"
"CommonName"    "Bannock Road South"
"NaptanCode"    "iowdtmg"
```

`lat and lon info:`

```
select lat, lon
from nodes
where id in
    (select id
    from (select *
        from (select * from node_tags union all select * from way_tags)
        where id in
            (select id
            from (select * from node_tags union all select * from way_tags)
            where type = 'naptan'
            group by id
            )
        and key = 'fixme'
        )
    )
;
```

lat: 50.597754
lon: -1.2654259

3.9 Number of locations required to be fixed with name or street information

```
select count(*)
from (select *
    from (select * from node_tags union all select * from way_tags)
    where id in
        (select id
        from (select * from node_tags union all select * from way_tags)
        where key = 'fixme' or (key = 'note' and value like '%fixme%'))
    and key in ('name', 'street')
    group by id
    )
;
```

139

3.10 Number of nodes with name or street information

`nodes with a name:`

```
select count(*)
from (select id, key, value from (select * from node_tags)
where key = 'name')
;
```

3899

`nodes with address (street information):`

```
select count(*)
from (select id, key, value from (select * from node_tags)
where key = 'street')
;
```

3613

`nodes with either a name or address:`

```
select count(*)
from (select id
    from (select id, key, value from (select * from node_tags)
    where key in ('name', 'street'))
    group by id
    )
;
```

6060

4. Facts and suggestions

- There are 507 unique users contributed to the area.
- 66 users (13%) have only submitted 1 posts.
- Top 3 users contributed 98.2% of overall entries.
- Top 3 user contributed 63.6%, 20.8%, 13.8% of overall entries respectively.

After viewing the data and posting some entries myself, it came to me that editing an entries could cost at least 5 minutes (including checking the right input). A fair amount of users might find the procedure to contribute time-consuming. I would suggest OpenStreetMap to breakdown the fields of data, highlight the priorities (e.g. name and address are prior to whether or not smoking is allowed inside the building), preload location information (e.g. city and province can be preloaded if locations nearby both have same values of city/province).

5. Conclusion

After wrangling and reviewing the dataset, I found out that the data was contributed by several users mostly. As a result, the data might not be all accurate as not all inputs can be reviewed by others. In addition, most locations have different levels of missing data. Although enlarging the size of data and enhancing the accuracy are both important, OSM should encourage more entires to be submitted before enhancing the accuracy as it is easy to get a large amount of data if they cooperate with other GPS processors or introduce more users to participate.

7point5 (/u/27d144331d89)
写了 855 字，被 0 人关注，获得了 0 个喜欢
(/u/27d144331d89)

01 两天前和朋友小张吃饭，小伙子原来在我公司做销售，半年后辞职自己创业，生意做的很不错，不到两年就买房买车了。 我很好奇地问他，“自己创业…

(/p/fdd4744dfeb4?
utm_campaign=maleskine&utm_content=note&utm

阿何 (/u/4389d5098b74?

utm_campaign=maleskine&utm_content=user&utm_medium=pc_all_hots&utm_source=recommendation)

为什么有人会不好意思上厕所？ (/p/b612a93eac15?ut…

有个很有趣的事情： 有的朋友在集体环境下，比如在公司或是学校的时候，会很少去厕所，实在憋不住了才会去洗手间，而且会尽量避开人群，避开他人的…

(/p/b612a93eac15?
utm_campaign=maleskine&utm_content=note&utm

不懂点心理 (/u/120b55beb9f9?
utm_campaign=maleskine&utm_content=user&utm_medium=pc_all_hots&utm_source=recommendation)

简书树洞开启|喜欢上一个没有结果的人，我到底要不要说？ (/p/f838e0cc06…

我是简书的简宝玉，白天我的正职是简书社区的大管家与代言人，关于简书有什么问题他们都会来问我。 但下了班之后，我也是一个和你们一样，是个有点烦恼的年轻人。我觉得走在大街上的每个人，都有烦恼困…

简宝玉 (/u/b52ff888fd17?
utm_campaign=maleskine&utm_content=user&utm_medium=pc_all_hots&utm_source=recommendation)