

案例研究：OpenStreetMap 数据

在地图中遇到的问题

1. 地址英文名称翻译不规范（“Lu”，“Dao”）
2. 节点和路径标签英文名称输入错误（“Hospita”，“Road”）
3. 街道英文名称使用缩写（“RD”）
4. 城市名不一致（“Tianjin”，“tianjin”，“天津市”，“天津”）
5. 邮编超出范围

英文名称问题

```
def update_name(name, mapping):  
    road_cn = ['lu', 'Lu', 'Dao', 'dao']  
    n = street_type_re.search(name).group()  
    #1  
    if n in mapping.keys(): #1  
        name = name.replace(n, mapping[n])  
    #2  
    for e in road_cn:  
        if name.find(e) == len(name) - len(e):  
            name = name.replace(e, ' Road')  
    #3  
    if name.find('qiao') == len(name) - len('qiao'):  
        name = name.replace('qiao', ' Bridge')  
    return name
```

在使用 `audit.py` 对数据审查发现，节点和路径英文名，即 `key=en:name` 的标签值中，存在两类问题：一是使用英文名缩写，如“RD”，“ST”，二是使用了音译，如“lu”，“dao”，在 `update_name` 函数中统一替换为 `Road`。此外，有部分使用音译的名称单字间没有空格，无法用函数中#1 部分代码替换，因此增加代码#2、#3 进行替换。

城市名不一致

在审查中发现城市名词存在格式不统一问题,如“Tianjin”,“tianjin”,“天津市”,“天津”。在清理过程中,将所有不标准加入 `mapping_city` 字典并用以下代码清理:

```
mapping_city = {"tianjin": "Tianjin",
                "Tianjin/China": "Tianjin",
                u"天津": u"天津市",
                u"北京": u"北京市"}

def update_city_name(name, mapping_city):
    if name in mapping_city.keys():
        name = mapping_city[name]
    return name
```

邮编问题

```
sqlite> SELECT value FROM nodes_tags WHERE key =
"postcode" AND (value<300000 or value>=310000)
UNION ALL
SELECT value FROM ways_tags WHERE key = "postcode" AND
(value<300000 or value>=310000);
```

063000

100176

天津市邮编为 30 开头,使用以上语句查询,发现两条记录不在范围内。其中 063000 为河北省唐山市邮编,而 100176 为北京市邮编。

```
sqlite> SELECT value FROM nodes_tags
WHERE key = 'city'
AND id IN (SELECT id FROM nodes_tags WHERE key =
"postcode" AND (value<300000 or value>=310000))
UNION
```

```
SELECT value FROM ways_tags WHERE key = 'city' AND id IN
(SELECT id FROM ways_tags WHERE key = "postcode" AND
(value<300000 or value>=310000));
```

Tangshan

北京市

继续查询这两条记录对应的城市名称，发现确实为唐山市和北京市，因此邮编没有错误。由于北京市和唐山市都在天津市附近，因此推断应为数据集包含了这两座城市的数据信息。

```
sqlite> SELECT MIN(lat), MAX(lat), MIN(lon), MAX(lon)
FROM nodes;
```

38.3780122|39.7789921|116.5280002|118.5309772

通过查询节点的坐标范围，发现数据集坐标覆盖率天津市、北京市东南部地区及河北省唐山市、廊坊市。

```
sqlite> SELECT value FROM nodes_tags WHERE key = 'city'
UNION
SELECT value FROM ways_tags WHERE key = 'city';
```

Tangshan

Tianjin

北京市

天津市

永清县

继续查询数据集中所有城市，发现包含了唐山市和北京市。因此验证了之前的推断，两条超出范围的邮编数据并不属于错误记录。

数据概述

文件大小

```
tianjin_china.osm ..... 74 MB
```

```
tianjin_china.db ..... 41 MB
nodes.csv ..... 29 MB
nodes_tags.csv ..... 0.8 MB
ways.csv ..... 2.6 MB
ways_tags.csv ..... 3.4 MB
ways_nodes.csv ..... 10 MB
```

节点数量

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

357426

途径数量

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

45072

唯一用户的数量

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

312

其他数据探索

大学校园数:

```
sqlite> SELECT count(*) FROM ways_tags WHERE value =
"university";
```

这里的数量为所选地图范围内所有大学的所有校区数量总计。

贡献最多的前十名用户：

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

```
422128715|Chen Jia|74797
188964751|Xbear |50752
456885606|zhenghy76|33764
112527030|uk1967|32435
228094967|trekki|23232
334601111|zhongguo|19569
367040209|u_kubota|15494
452075727|katpatuka|13874
454846926|pingsler|11818
398545163|Oberaffe|10767
```

只出现一次的用户：

```
sqlite> SELECT COUNT(*)
FROM
(SELECT nw.id, nw.user, COUNT(*) as num
FROM (SELECT id, user FROM nodes UNION ALL SELECT id,
user FROM ways) nw
GROUP BY nw.user
```

```
HAVING num=1) u;
```

```
422128715|Chen Jia|74797  
188964751|Xbear |50752  
456885606|zhenghy76|33764  
112527030|uk1967|32435  
228094967|trekki|23232  
334601111|zhongguo|19569  
367040209|u_kubota|15494  
452075727|katpatuka|13874  
454846926|pingsler|11818  
398545163|Oberaffe|10767
```

额外改进建议：

在数据审查过程中发现的最为突出的问题是英文翻译方法混乱，不同贡献者没有统一的翻译标准，导致街道名、城市名英文译名格式不一致，甚至中英文夹杂。建议 [OpenStreetMap.org](https://openstreetmap.org) 可以为贡献者提供统一的名称翻译格式标准，增加一个自动纠正格式的插件或一个官方译名库来对用户对用户贡献的信息进行验证和预处理，从而避免这类问题的出现。

在本案例的数据清理中，通过编程方式对不规范的英文翻译批量清理，格式统一、规范的名称翻译可以让数据变得干净，方便后续的查询和分析工作，减少因格式不同统一可能导致的重复或多次统计等错误。然而，通过批量处理清理数据也存在局限性，具体翻译问题仍需要手动逐一处理。因此，如果在用户贡献信息时，就对数据进行验证和预处理，可以提升数据有效性和准确性、降低数据清理的难度。当然，对于系统无法自动纠错的数据，可能给数据贡献这提出了更高的要求，需要用户提供更加准确的数据。