OpenStreetMap Sample Project Data Wrangling with MongoDB

BanksJ

Map Area: Shenzhen China
https://mapzen.com/data/metro-extracts/metro/shenzhen_china/

一、 在地图中遇到的问题

对下载的深圳地区数据集进行取样观察，发现存在以下几点问题：

- 街道名称，节点名称 name 不规范，有的缩写，有的全称，不利于聚合分析，采用课件中的代码已修改。例如，‘Rd’-→’Road’
- 邮政编码出现如：DD78 1878 之类的，不是中国标准邮编。暂不清楚意义，未做处理。
- Tag 标签 k 属性可能包括‘：’，统一作分级处理，添加至内层循环。忽略所有出现异常字符的标签。
- 分级处理过程中，可能已经存在对应的键值，将该键值暂存，赋值给内层字典键” value ”
- 某些字段中英文混杂，且层级过多，不便聚合或索引

二、 数据概述

```
shenzhen_china.osm ----> 146 MB
shenzhen_china.osm.json ----> 222 MB
```

```
# number of documents
db.getCollection('osm_shenzhen').find({}).count()
# >804813

# Number of node
db.getCollection('osm_shenzhen').find({'type':'node'}).count()
# >726087

# number of ways
db.getCollection('osm_shenzhen').find({'type':'way'}).count()
# >78726

# number of unique uid
db.getCollection('osm_shenzhen').distinct('created.uid').length
# >841

# top 1 contributing uid
db.getCollection('osm_shenzhen').aggregate([{'$group':{'_id':'$created.uid',
    'count':{'$sum':1}}},{'$sort':{'count':-1}},{'$limit': 1}])
# [{'_id':'44514','count':'176094'}]

# number of cafe
```

```
db.getCollection('osm_shenzhen').find({'amenity':'cafe'}).count()
# >74
# number of General store, department store, mall
db.getCollection('osm_shenzhen').find({'shop': {'$in':['department_store','general','kiosk','mall','supermarket']}}).count()
# >231

# top 10 amenity
db.getCollection('osm_shenzhen').aggregate([
{'$match':{'amenity':{'$exists':true}}},
{'$group':{'_id':'$amenity', 'count':{'$sum':1}}},
{'$sort':{'count':-1}},
{'$limit':10}])
'''
[{"_id" : "parking","count" : 630.0},{"_id" : "toilets","count" : 423.0},
{"_id" : "school","count" : 370.0},{"_id" : "restaurant","count" : 311.0},
{"_id" : "shelter","count" : 235.0},{"_id" : "post_box","count" : 166.0},
{"_id":"place_of_worship","count" : 151.0},{"_id" : "bank","count" : 144.0},
{"_id":"bus_station","count":132.0},{"_id" : "fuel","count" : 121.0}]
'''
```

三、 关于数据集的其他想法

文档中的 name 字段，存在多种语言，分别使用字典进行存储。部分节点本身有 name 属性，则存为 value 键值，且该值为中英文混写。此时，对 name 进行统计分析比较麻烦。可以考虑对 name 字典内的值统一清洗，只保留唯一中文名。如果存在英文名称则，另建立字段 name_en 存储。其余语言的名称，均忽略。

但是，在清洗过程中如果只存在其他语言名称，则导致部分信息丢失。当信息过于完整，则层级过多，不利于分析。当简化信息，可能会丢失部分信息。此时，需要根据业务需求权衡，选择哪种方案进行处理。

对数据集中的烹饪、宗教类型感兴趣，进行了聚合分析。该分析不会修改数据集，结果如下：

```
# top 5 cuisine
db.getCollection('osm_shenzhen').aggregate([
{'$match':{'amenity':{'$exists':true},'cuisine':{'$exists':true},'amenity':'restaurant'}},
{'$group':{'_id':'$cuisine', 'count':{'$sum':1}}},
{'$sort':{'count':-1}},
{'$limit':5}])
'''
[{"_id" : "chinese","count" : 47.0},{"_id" : "noodle","count" : 6.0},
{"_id" : "japanese","count" : 5.0},{"_id" : "regional","count" : 5.0},
{"_id" : "local","count" : 5.0}]
'''

```

```
# biggest religion
db.getCollection('osm_shenzhen').aggregate([
{'$match':{'amenity':'place_of_worship', 'religion':{'$exists':true}}},
{'$group':{'_id':'$religion', 'count':{'$sum':1}}},
{'$sort':{'count':-1}}])
'''
[{"_id" : "christian","count" : 42.0},{"_id" : "buddhist","count" : 34.0},
{"_id" : "taoist","count" : 29.0},{"_id" : "muslim","count" : 1.0}]
'''
```

四、 总结

数据的审核、清洗、分析实际是循环进行，在分析过程中，发现问题，考虑解决方案及其风险，再次审核、清洗、分析，循环进行。