Data Wrangling – OpenStreetMap

In this project, I used python to audit and clean the OpenStreetMap dataset and import it into MongoDB for further analysis.

Area

I live and work in the San Jose now, so I just went to https://mapzen.com/data/metro-extracts/metro/san-jose_california/ and downloaded OSM XML file.

Sample Dataset

Since the original file is 350 MB, I extract a sample dataset around 11 MB.

```
OSM_FILE = "san-jose_california.osm"
SAMPLE_FILE = "sj_sample.osm"

k = 25

def get_element(osm_file, tags=('node', 'way', 'relation')):
    context = iter(ET.iterparse(osm_file, events=('start', 'end')))
    _, root = next(context)
    for event, elem in context:
        if event == 'end' and elem.tag in tags:
            yield elem
            root.clear()

with open(SAMPLE_FILE, 'wb') as output:
    output.write('<?xml version="1.0" encoding="UTF-8"?>\n')
    output.write('<osm>\n  ')

    # Write every kth top level element
    for i, element in enumerate(get_element(OSM_FILE)):
        if i % k == 0:
            output.write(ET.tostring(element, encoding='utf-8'))

    output.write('</osm>')
```

Audit & Clean Dataset

First, I showed how many tags in this dataset. we could see a large amount of things to be audited. In this project, I will further explore 'addr:street' and 'addr:postcode'.

Addr:street

During the audit, we could see the inconsistency of the street name, such as “Rd” and “Road”, “Ct” and “Court” and so on and so forth.

```
 Ave : 39,
'Avenue': 5520,
'Barcelona': 17,
'Bascom': 1,
'Bellomy': 1,
'Blvd': 20,
'Boulevard': 571,
'Boulvevard': 1,
'CA': 2,
'Cir': 1,
'Circle': 235,
'Court': 2280,
'Ct': 1,
'Dr': 9,
'Drive': 5559,
'East': 21,
'Esquela': 1,
'Expressway': 90,
'Flores': 1,
'Franklin': 1,
'Hamilton': 1,
'Highway': 4,
'Hill': 2,
'Hwy': 2,
'Julian': 1,
'Lane': 1062,
```

I only selected the expected street name and do mapping for them.

```
expected = ["Street", "Avenue", "Boulevard", "Drive", "Court", "Place", "Square", "Lane", "Road",
            "Trail", "Parkway", "Commons", "Cove", "Alley", "Park", "Way", "Walk" "Circle", "Highway",
            "Plaza", "Path", "Center", "Mission"]

mapping = { "Ave": "Avenue",
            "Ave.": "Avenue",
            "avenue": "Avenue",
            "ave": "Avenue",
            "Blvd": "Boulevard",
            "Blvd.": "Boulevard",
            "Blvd,": "Boulevard",
            "Boulavard": "Boulevard",
            "Boulvard": "Boulevard",
            "Ct": "Court",
            "Dr": "Drive",
            "Dr.": "Drive",
            "E": "East",
            "Hwy": "Highway",
            "Ln": "Lane",
            "Ln.": "Lane",
            "Pl": "Place",
            "Plz": "Plaza",
            "Rd": "Road",
            "Rd.": "Road",
            "St": "Street",
            "St.": "Street",
```

Addr:postcode

During the audit, I found the postcode are also inconsistent. Sometimes, it has “CA” or “CU” inside.

Sometimes they are 5 digits instead of 9 digits.

```
'95014-4667', '95014-0549', '95014-0548', '95014-4662', '95014-4663', '95014-0545', '95014-0544', '95014-0547', '95
014-0546', '95014-0541', '95014-0540', '95014-0543', '95014-0542', '95014-3012', '95014-3010', '95014-3011', '95014
-3014', '95014-3015', '95014-3018', '95037-4222', '95037-4221', '95037-4220', '95037-4225', '95037-4224', '95014-29
47', '95014-4449', '95014-2945', '95014-2944', '95014-2943', '95014-2942', '95014-2941', '95014-2940', '95014-444
0', '95014-4441', '95014-4442', '95014-4443', '95014-4444', '95014-4445', '95014-3490', '95014-4447', 'CA 95113',
'95014-3702', '95014-2911', 'CA 95116', '95014-3428', '95014-3029', '95014-0619', '95014-4431', '95014-3124', '9501
4-3125', '95014-3122', '95014-3123', '95014-3120', '95014-4430', '95014-0615', '95014-0614', '95014-0617', '95014-0
616', '95014-4344', '95014-3423', '95014-4434', '95014-3838', '95014-3421', '95014-3834', '95014-3835', '95014-254
1', '95014-2540', '95014-3830', '95014-3831', '95014-3832', '95014-3833', '95014-3234', '95014-2722', '95014-2055',
'95014-2054', '95014-2057', '95014-2056', '95014-2058', '95014-366', '95014-1627', '95014-1626', '95014-1625', '950
14-1624', '95014-1623', '95014-1622', '95014-1621', '95014-1620', '95013', '95014', '95014-1629', '95014-1628', '95
014-218', '95014-5712', '95037-4531', '95014-1967', '95037-4125', '95014-1801', '95014-1800', '95014-1807', '95037-
4121', '95014-1805', '95037-4123', '95014-3547', '95037-4326', '95014-5099', '95014-4273', '95014-3192', '95014-197
2', '95037-4032', '95037-4030', '95037-4031', '95014-5249', '95014-5248', '95014-4672', '95014-4671', '95014-4670',
'95014-4660', '95014-4661', '95014-3001', '95014-3000', '95014-3003', '95014-3002', '95014-3005', '95014-3004', '95
014-3007', '95014-3009', '95014-3008', '95014-5241', '95014-0599', '95014-5240', '95014-5243', '95014-5242', '95014
-3489', '95014-3488', '95014-4459', '95014-4457', '95014-5713', '95014-3487', '95014-4453', '95014-4452', '95014-44
51', '95014-4450', '95014-5247', '95014-5246', '95014-2442', '95014-2443', '95014-2440', '95014-2441', '95014-244
6', 'CA 95054', '95014-2444', '95014-2445', '95014-2448', '95014-2449', '95014-3134', '95014-3136', '95014-3131',
```

I removed CA, change all the zipcode to five digits and removed "CU" as well.

```
CA 95116 => 95116
CA 94085 => 94085
CA 94086 => 94086
95014-1899 => 95014
95014-3456 => 95014
95014-3457 => 95014
95014-1968 => 95014
95014-1960 => 95014
95014-1961 => 95014
95014-1962 => 95014
95014-1963 => 95014
95014-1964 => 95014
```

Parse Data to JSON

After auditing and cleaning the data, I parse the data to JSON file.

Import Dataset to MongoDB

In command, mongoimport --db OpenStreetMap --collection SanJose --file san-jose_california.osm.json.

Explore Dataset

Number of unique users

```
len(collection.distinct("created.user"))
```

```
1345
```

Number of nodes and ways

```
collection.find({"type":"node"}).count()
```

```
1346165
```

```
collection.find({"type":"way"}).count()
```

```
178557
```

Number of total documents

```
collection.count()
```

```
1524761
```

Top 5 contribution users

```
top_user = collection.aggregate([
        {"$group": {"_id": "$created.user", "count": {"$sum": 1}}},
        {"$sort": {"count": -1}},
        {"$limit": 5}
    ])

list(top_user)
```

```
[{u'_id': u'nmixter', u'count': 287912},
 {u'_id': u'mk408', u'count': 151363},
 {u'_id': u'Bike Mapper', u'count': 81192},
 {u'_id': u'samely', u'count': 77905},
 {u'_id': u'dannykath', u'count': 72273}]
```

Top 5 Amenity

```
amenity =  collection.aggregate( [
        {"$group": {"_id": "$amenity", "count": {"$sum": 1}}},
        {"$sort": {"count": -1}},
        {"$limit": 5}
        ])
list(amenity)
```

```
[{u'_id': None, u'count': 1517593},
 {u'_id': u'parking', u'count': 1910},
 {u'_id': u'restaurant', u'count': 969},
 {u'_id': u'school', u'count': 534},
 {u'_id': u'fast_food', u'count': 494}]
```

Amenity in each postcode

```
amenity_by_postcode =  collection.aggregate( [
        {"$match": {"amenity": {"$exists":1},
                    "address.postcode": {"$exists": 1}}},
        {"$group": {"_id": { "postcode": "$address.postcode",
                             "amenity": "$amenity"},
                    "amenity_count": {"$sum": 1}}},
        {"$match": {"amenity_count": {"$gt": 2}}},

        {"$sort": {"amenity_count": -1 }},
        {"$group": {"_id": "$_id.postcode",
                    "amenity_by_postcode": {"$push": {
                                                "amenity": "$_id.amenity",
                                                "count": "$amenity_count"
                    }},
                    "total_count":  {"$sum":1}
                   }},

        {"$sort": {"total_count":-1}},
        {"$limit": 5}
        ])

list(amenity_by_postcode)
```

```
[{u'_id': u'95014',
  u'amenity_by_postcode': [{u'amenity': u'restaurant', u'count': 34},
   {u'amenity': u'cafe', u'count': 8},
   {u'amenity': u'fast_food', u'count': 6},
   {u'amenity': u'bank', u'count': 5},
   {u'amenity': u'place_of_worship', u'count': 4},
   {u'amenity': u'fuel', u'count': 3},
   {u'amenity': u'school', u'count': 3}],
  u'total_count': 7},
 {u'_id': u'94087',
  u'amenity_by_postcode': [{u'amenity': u'restaurant', u'count': 10},
   {u'amenity': u'fast_food', u'count': 4},
   {u'amenity': u'dentist', u'count': 3},
   {u'amenity': u'bank', u'count': 3},
   {u'amenity': u'cafe', u'count': 3}],
  u'total_count': 5},
 {u'_id': u'95051',
  u'amenity_by_postcode': [{u'amenity': u'restaurant', u'count': 12},
   {u'amenity': u'bank', u'count': 4},
   {u'amenity': u'cafe', u'count': 4},
   {u'amenity': u'school', u'count': 3},
   {u'amenity': u'fast_food', u'count': 3}],
  u'total count': 5},
```

Looks like 95014 area is the most common area with different kings of stores, banks, and schools. From Google search, 95014 is Cupertino area, where I usually go for lunch and dinner :)

Other ideas about the dataset

When I went through the database, there are a lot of information missing. For example, when I query the amenity, there are 1517593 none value

```
amenity =  collection.aggregate( [
        {"$group": {"_id": "$amenity", "count": {"$sum": 1}}},
        {"$sort": {"count": -1}},
        {"$limit": 5}
        ])
list(amenity)
```

```
[{u'_id': None, u'count': 1517593},
 {u'_id': u'parking', u'count': 1910},
 {u'_id': u'restaurant', u'count': 969},
 {u'_id': u'school', u'count': 534},
 {u'_id': u'fast_food', u'count': 494}]
```

In my opinion, it is probably because there are large amounts of data sources. Since each data source has different data structure, so sometimes, it may miss the "amenity" part. Also, we can see that most of the data sources are unknown!

```
[{u'_id': None, u'count': 1514035},
 {u'_id': u'bing', u'count': 4200},
 {u'_id': u'Yahoo', u'count': 1572},
 {u'_id': u'bing;survey', u'count': 1282},
 {u'_id': u'Bing', u'count': 911},
 {u'_id': u'photograph', u'count': 699},
 {u'_id': u'NHD', u'count': 516},
```

I recommend to use one stable data source like Google Map API to get data and remain the same data structure. In that way, we can easily audit, clean and maintain the dataset.

Conclusion

During this project, I audited, cleaned and explored San Jose OpenStreetMap data. I found two problems. One is the addr:street - street name is inconsistent such as "Rd" and "Road". I created a mapping for those street names and delete unexpected street names. The other problems is the postcode. I identified unexpected "CA" and "CU", and updated all 9 digits to 5 digits, which I think is a good for further analysis.

When I implemented this data improvement, I found:

Benefits:

1. We have cleaner data to do analysis – for example, use postcode to count something.
2. Now we know some tags have the problems, then we can do something during the data collection phase to prevent them.

Anticipated Problems:

1. Everyone has different approach to clean the data, it's hard to combine each person's dataset
2. We might lose some information when we clean the data

When I looking at tag at the very beginning of the project, I found that there are hundreds of tags there. So one of my suggestions is to reduce the amount of tags and created more general tag names. In that case, we can save time to audit and clean the dataset.

I also plan to audit and clean more tags in the near future to get more familiar with OpenStreetMap dataset.