Project 4: Wrangle OpenStreetMap Data

Location: Mumbai India - https://mapzen.com/data/metro-extracts/your-extracts/693a2a74b296

Objective: Audit, clean the OSM dataset, convert from XML to JSON format and analyze insight within the data.

Refer file : process.py

1. Data Audit

function count_tags will parse through Mumbai dataset with ElementTree and count the number of unique elements to get an overview of the data and use pretty print to print the results.

For following functions: key_type & process_map. We check the "k" value for each tag and see if they can be valid keys in MongoDB, as well as see if there are any other potential problems. As we saw in the quiz earlier, we would like to change the data model and expand the "addr:street" type of keys to a dictionary like this: {"address": {"street": "Some value"}} So, we have to see if we have such tags, and if we have any tags with problematic characters.

For the function 'key_type', we have a count of each of three tag categories in a dictionary:

- "lower", for tags that contain only lowercase letters and are valid,
- "lower_colon", for otherwise valid tags with a colon in their names,
- "problemchars", for tags with problematic characters

2. Problems encountered

2.1 Street address abbreviation

One of the problem is the street name abbreviation inconsistency. Below we make the regex matching the last element in the string, where usually the street type is based.

  - audit_street_name function search the input string for the regex. If there is a match and it is not within the "expected" list, add the match as a key and add the string to the set.
- is_street_name function looks at the attribute k if k="addre:street"
- audit function will return the list that match previous two functions. After that, we would do a pretty print the output of the audit. With the list of all the abbreviated street types we can understand and fill-up our "mapping" dictionary.

function update_name takes the old name and update them with a better name

2.2 Zip codes

We can reuse part of the code above to clean the zipcodes.

Preparing for MongoDB by converting XML to JSON

To transform the data from XML to JSON, we should follow these rules:

- Process only 2 types of top level tags: "node" and "way"
- All attributes of "node" and "way" should be turned into regular key/value pairs, except: attributes in the CREATED array should be added under a key "created", attributes for latitude and longitude should be added to a "pos" array, for use in geospacial indexing. Make sure the values inside "pos" array are floats and not strings.
- If second level tag "k" value contains problematic characters, it should be ignored

- If second level tag "k" value starts with "addr:", it should be added to a dictionary "address"
- If second level tag "k" value does not start with "addr:", but contains ":", process it same as any other tag.
- If there is a second ":" that separates the type/direction of a street, ignore this tag

After all the cleaning is done, we use process_map function to convert the file from XML into JSON.

Refer File: mongo.py

3. Data Overview with MongoDB

File sizes
The original OSM file is 56.364893 MB
The JSON file is 65.88346 MB

Number of documents
303290

Number of unique users
778

Number of Nodes and Ways
Number of nodes: 262990
Number of ways: 40191

Name of top 5 contributors

```
 [{u'_id': u'PlaneMad', u'count': 31353},
 {u'_id': u'anthony1', u'count': 18584},
 {u'_id': u'Zulfiqarib', u'count': 16677},
 {u'_id': u'Nagarjunreddy', u'count': 15081},
 {u'_id': u'venkatkotha', u'count': 14686}]
```

4. Further data explaration with MongoDB

List of top 10 amenities in Mumbai

```
 [{u'_id': u'place_of_worship', u'count': 207},
 {u'_id': u'restaurant', u'count': 199},
 {u'_id': u'bank', u'count': 125},
 {u'_id': u'school', u'count': 117},
 {u'_id': u'hospital', u'count': 75},
 {u'_id': u'cafe', u'count': 57},
 {u'_id': u'college', u'count': 51},
 {u'_id': u'parking', u'count': 49},
 {u'_id': u'police', u'count': 43},
 {u'_id': u'fuel', u'count': 43}]
```

List of top 5 Foods in Mumbai

```
 [{u'Count': 115, u'Food': None},
 {u'Count': 29, u'Food': u'indian'},
 {u'Count': 7, u'Food': u'pizza'},
 {u'Count': 6, u'Food': u'vegetarian'},
 {u'Count': 5, u'Food': u'regional'}]
```

List of top 10 post code in Mumbai

```
 [{u'_id': u'400050', u'count': 635},
 {u'_id': u'400043', u'count': 81},
 {u'_id': u'400005', u'count': 79},
 {u'_id': u'400089', u'count': 54},
 {u'_id': u'400074', u'count': 42},
 {u'_id': u'400071', u'count': 38},
 {u'_id': u'400001', u'count': 30},
 {u'_id': u'400002', u'count': 28},
 {u'_id': u'400051', u'count': 22},
 {u'_id': u'400004', u'count': 21}]
```

Total users have unique post (post only one time)

```
 [{u'_id': u'yes', u'count': 26328},
 {u'_id': u'apartments', u'count': 387},
 {u'_id': u'residential', u'count': 250},
 {u'_id': u'commercial', u'count': 76},
 {u'_id': u'house', u'count': 69}]
```

Top 5 building type by count

```
   [{u'_id': u'yes', u'count': 26328},
    {u'_id': u'apartments', u'count': 387},
    {u'_id': u'residential', u'count': 250},
    {u'_id': u'commercial', u'count': 76},
 {u'_id': u'house', u'count': 69}]
```

4. Conclusion

Ideas to improve data quality of OSM: While auditing the data, we find that although there are minor human input errors, the dataset is fairly clean. For human errors we can have a srtuctured input form so everyone can input the same data format to reduce errors. Moreover, we can incentivize users in the contribution process, then we can create a recommendation engine to leverage this data, since OpenStreetMaps is an open source project, there're still a lot of areas left unexplored as people tend to focus on a certain key areas and left other part outdated. we can resolve this issue by cross-referencing/cross-validating missing data from other database like Google API. Since each node has a coordinate (lattitude & longtitude).

Potential cost of the implementation: There're few potential issues that may arise from the implementation of this solution. One of which is the amount of effort to engineer all this and the cost of creating, auditing & maintaining thus may require a dedicated team.