Udacity DAND P3:Wrangling OpenStreetMap Data Project (SQL)

Author: Saugata Ghosh

Date : January 5, 2017

Map Area

New Delhi

This dataset contains information about New Delhi, the capital of India and the city where I live.

After downloading the map in xml format and running the 'count_tags' function ('OSMProject.ipynb') I found the following distribution of different types of tags. Since we are only interested in the **node, 'way', 'tag'** and **nd** tags the other tags were ignored for the purpose of analysis.

```
{'bounds': 1, 'member': 16133, 'meta': 1, 'nd': 297004, 'node': 245721, 'note': 1, 'osm': 1, 'relation': 596, 'tag': 58667, 'way': 45422}
```

Problems encountered in the map

Street Types

Tag key : addr:street

After downloading the map in xml format and running it against the auditing street types functions ('OSMProject.ipynb') we encountered the following problems:

- Streetnames include indigenous terms like 'Marg', 'Chowk', 'Bazaar', 'Gali', etc. I retained some of the terms in common usage and replaced the others (viz. 'Gali') with the English equivalent ('Lane').
- Street types are not always at the end of the string. e.g. 'Gali No 1'. In this case the function I used to clean up names (given below) as per mapping was used to replace instances of street types wherever they occur.
- Other inconsistencies in spelling ('Counnaught Circus' which should read 'Connaught Circus') , use of case ('lane' vs 'Lane'), abbrveviated street types ('Rd') and postal codes trailing at the end of the string, which were fixed with the mapping ('OSMProject.ipynb').

In [ ]:

```
# change string into titleCase except for UpperCase

def string_case(s):
    if s.isupper():
        return s
    else:
        return s.title()

# Update streetname by splitting the string to clean streetname wherever
# it appears in the string

def update_street_name(name, mapping):
    name = name.split(" ")
    for i in range(len(name)):
        if name[i] in mapping:
            name[i] = mapping[name[i]]
            name[i] = string_case(name[i])
        else:
            name[i] = string_case(name[i])
    name = " ".join(name)
    return name
```

Postal Codes

Tag key : addr:postcode

The following issues were detected and resolved:

- Invalid postcodes(not starting with 11) or those from adjoining state of Uttar Pradesh
- Valid postcodes had errors such as whitespace or an extra zero in the middle
- Vaid postcodes were cleaned ('OSMProject.ipynb) and tags with invalid or out-of--state

postcodes were passed over in the shape_element function ('OSMProject.ipynb').

City

Tag key : addr:city

- The value 'noida' referred to an area in the neighboring state of Uttar Pradesh and tags with that value were passed over in the shape_element function
- Spellings and case of 'Delhi' or 'New Delhi' were fixed and trailing information edited.

Data Overview

File Size of different files/databases used/created

'new_delhi.osm - 53.9 MB

'osmdb.db' - 37.2 MB

'nodes_project.csv' - 20.2 MB

'nodes_tags_project.csv' - 190 KB

'ways_project.csv' - 2.7 MB

'ways_tags_project.csv' - 1.7 MB

'ways_nodes_project.csv' - 7.3 MB

Basic Statistics about the tables(OSMProject.ipynb)

- Number of nodes - 245721
- Number of ways - 45422
- Number of nodes tags - 5036
- Number of ways tags - 51402
- Number of unique users - 449

Top amenities

(u'school', 123) (u'place_of_worship', 81) (u'restaurant', 58) (u'parking', 51) (u'atm', 41) (u'embassy', 40) (u'fast_food', 33) (u'hospital', 31) (u'fuel', 29) (u'bank', 25)

Popular religions

(u'hindu', 21) (u'muslim', 19) (u'sikh', 11) (u'christian', 7) (u'jain', 1) (u'zoroastrian', 1)

Information on historical sites

(u'archaeological_site', 30) (u'monument', 18) (u'tomb', 8) (u'memorial', 6) (u'ruins', 3) (u'city_gate', 2) (u'fort', 2) (u'mon', 1) (u'wayside_shrine', 1)

Additional Data Ideas

- The query in Openstreetmap for New Delhi can be improved to provide a more balanced representation. Right now the majority of postcodes in the tables are '11001' - '11003' and '110055'. Other areas of South, West and Central New Delhi can be better represented. However this would involve work on the original Openstreet Map data. Validation against results of the same query thrown up by Google Maps can be also be used too fine-tune the query.

- Delhi is a city rich in history and heritage. The values for the key 'historic' , such as 'archeological_site' and 'monument' can possibly be converted into keys of their own with more fine-grained values to represent the cultural heritage of the city in more detail. This can also be done for 'place_of_worship' under the 'amenities' key. Places of worship in Delhi are often important cultural and historical sites in their ownn right and merit more detailed depiction than being clubbed under amenities.

- There is some scope to rationalize the keys in the ways_tags and nodes_tags tables. There are times when keys such as 'atm' appear separately and as value under the 'amenities' key. Such inconsistencies need to be addressed.

- Out of 449 unique users about 30% have contributed only once. The map for New Delhi can be made more detailed and useful if more users contribute. Different ways of incentivizing users to contribute more to the data (through contests, etc.) can make the community develop the New Delhi map better.

- It may also be important to have some broad rules and conventions about inputting data in native languages. For a foreigner travelling to New Delhi some of the obscure street types such as 'Chowk, Marg, Bazaar' which really refer to 'Square', 'Road' and 'Market' may create confusion. A basic standardization of symbols without comprmosing on the unique local flavour of each area could be a medium-term objective of this collaborative project.

Conclusions

- The Openstreetmap dataset for New Delhi is reasonably clean and quite useful. Nonetheless there is scope for standardising street types, adding more details of historical monuments, rationalizing keys etc. The benefits of doing so are:

  ```
  * More clear, comprehensible and useful data especially for foreign travellers
  * Adoption of more internationally accepted standards in this great collaborative project
  ```

- The problems that can be anticipated in doing the above are:

  ```
  * More collaborative work on the actual datasets could be time-consuming unless contributors are suitably incentivized
  * Validation against external sources such as Google Maps could detract from the collaborative, open-source nature of the project
  ```

Sources used

Wikipedia, OpenStreetMap website, UDacity DAND disucssion forum, lessons for the project, stackoverflow.