OpenStreet Map Data Case Study

By Jiffin James

Map Area

https://www.openstreetmap.org/export#map=10/23.0217/72.5797

https://mapzen.com/data/metro-extracts/metro/ahmedabad_india/

I was at this place a few years ago and was really excited to see its available openstreet map. So I used this place for my openstreet map data case study project.

The Open Street Map data consists of Elements which are nothing but conceptual data model of the physical world.Elements consists following –

- Nodes (Defines points in space)
- Ways (Defines linear features)
- Relations (Used to explain how other elements work together)

All these can have tags associated to them which describe their meanings.The tags are made of keys and values.

Problems Encountered in the Map

- The map size was huge.
- Processing it caused my PC to crash a few times.
- There were around 1.5 million number of lines in the osm file.
- As expected there were short forms used to represent some of the names (Eg-"Av. For Avenue")

Overabbreviated Street Names

By opening the file using notepad,I observed that various short forms were used to represent names like Avenue,Road etc.To correct these problems I used regular expressions as taught in the Case Study.Some words did not have English meanings like "Marg" (**traditional/related to language formating**) which means road.So,I decided to keep it as it is because it is not an error

```
expected=["Rd","Av.","rd.","rd","Av"]
mapping = { "Rd": "Road",
            "Av.": "Avenue",
            "rd.": "Road",
            "rd": "Road",
            "Av": "Avenue",
            }
last_m = street_type_re.search(value)
                if last_m:
                    last_word= last_m.group()
                    if last_word in expected:
```

```
                              name1=re.sub(r'\b\S+\.?$', 
mapping[last_word], value, flags=re.IGNORECASE)
                              ntag['value']= name1
                         else:
                              ntag['value']= value
```

Here what I have done is that I created a list with all the errors and their respective mappings.So if our program encounters any of these in the last word of the name,It converts the text into the respective mappings.

If we observe closely,we find that there is a specific part of the word where mistakes are frequently made.The last words which mostly have following terms-“Avenue”,”Road” etc are mostly represented as Rd,Av etc.So to identify and clean these issues I used Regular Expressions.The symbol $ was used to access the last word from the given string.The expression used was as follows for finding abbreviations in Roads-

```
r'\b\S+\.?$'
```

We know that a data analyst spends around 70% of his/her time in data wrangling.Here,in this project I really experienced this process.Initially,I extracted the OSM(Open Street Map) file.Then,I used a word processor to understand the data.It was arranged in xml format.After understanding the structure of the data,I found various mistakes which was taken care of by cleaning it.It was then converted into dictionaries and lists.Then using various libraries,it was converted into CSV files.These CSV files were then converted into tables which were finally used in SQL.
Another problem which was regarding the size of the osm file,I used the script given Project Details of Wrangle OpenStreetMap Data to make a sample from the bigger osm file.

Data Overview

Size-

ahmedabad_india.osm=112 MB

nodes.csv=4.6 MB

nodes_tags.csv=29 KB

ways.csv=505 KB

ways_nodes.csv=1.6MB

ways_tags.csv=314 KB

Number of Nodes

sqlite>select count(*) from nodes;

```
56620
```

Number of ways

```
sqlite>select count(*) from ways;
```

```
8468
```

Number of unique users

```
sqlite>select count(distict(e.uid)) from (select uid from nodes union all select uid from ways) e;
```

```
239
```

Number of node tags

```
Sqlite>select count(*) from nodes_tags
```

Top ten contributing Users

```
sqlite>select use.user, count(*) as num from (select user from nodes union all select user from ways)
use group by use.user order by num desc limit 10;
```

```
uday01,17723
sramesh,13681
chaitanya110,12220
shashi2,4937
shravan91,2294
vkvora,2197
Rajsamand Local Guide,1443
Bhanu8,1258
Oberaffe,702
```

Additional Ideas

Number of One-Way in Ahmedabad

```
sqlite>select count(*) as total from ways_tags where key=”oneway”
```

```
179
```

Number of Water ways in Ahmedabad

```
sqlite>select count(*) as total from ways_tags where key=”waterway”
```

12

I have lived there and as far as I know there are far more waterways and one ways present there.It is difficult to find the exact number of these ways because google too doesn't have answers,but I asked my friends who live there and they too agreed that the data needs updates .This shows that the data is somewhat incomplete and requires updating.
After going through some of the timestamps details it can be said that it requires updates-

```
sqlite>select timestamp from nodes limit 10

2012-07-06T13:38:12Z
2015-02-07T06:38:10Z
2012-09-08T20:35:08Z
2008-12-29T18:44:32Z
2008-12-29T18:44:32Z
2008-12-29T18:44:33Z
2008-12-29T18:44:33Z
2008-12-29T18:44:34Z
2008-12-29T18:32:31Z
2008-12-29T18:32:31Z
```

Suggestions

1-The problem with these kind of data is that changes happen frequently which is not reflected immediately. Thus a bigger and active community of users is required for accurate data.

2-Another point is that there are very less locations of india available as metro extracts.This can be a problem for new users because no proper information is available on how to extract them manually.