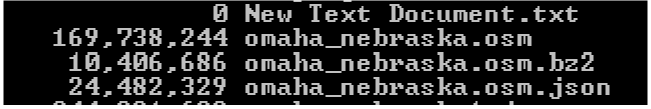
Data Wrangle OpenstreetMaps Data

Author : Vijayasarathi Balasubramanian

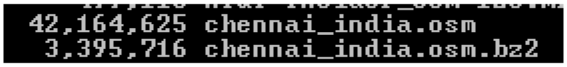
Problems encountered in your map

Initially I was thinking to select Atlanta or Chennai both of them were of big size, so my machine could not handle such a big data set. So I decided to choose some other city data set which of size less than 200 MB. I am planning to attend Nebraska next week for warren buffet conference, so thought it would be a better idea to choose. That way I will know about the city before I visit (I in fact have some idea about it as I had read about it before reserving travel and stay.



```
          0 New Text Document.txt
169,738,244 omaha_nebraska.osm
 10,406,686 omaha_nebraska.osm.bz2
 24,482,329 omaha_nebraska.osm.json
```

I tried to explore the Chennai,india data set as well, but the size was small, so I went with Omaha.



```
42,164,625 chennai_india.osm
 3,395,716 chennai_india.osm.bz2
```

I also explored the other parsing methodologies and settled with iterative parsing.

I also had tough time in resolving the street names to bring it to common format. As it had variety of names like blvd,st and numbered street. Took some time and explored to settle the final mapping list.

I was getting error like below, before I had the map for all of the different types of street names in Omaha.

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
<ipython-input-17-b92f6adb2f44> in <module>()
      1 if __name__ == '__main__':
----> 2     test()

<ipython-input-16-52592660f137> in test()
     22             #pprint.pprint(name)
     23             #pprint.pprint(mapping)
---> 24             better_name = update_name(name, mapping)
     25             print name, "=>", better_name
     26              #if name == "West Lexington St.":

<ipython-input-16-52592660f137> in update_name(name, mapping)
      3     better_name = name
      4     if m:
----> 5         better_street_type = mapping[m.group()]
      6         better_name = street_type_re.sub(better_street_type, name)
      7
```

```
KeyError: 'Rd'
```

Then I updated the mapping rules like below to take care of these gaps.

```
mapping = {  "St": "Street",
            "St.": "Street",
            "Ave": "Avenue",
            "A": "Avenue",
            "Rd.": "Road" ,
            "bing" : "Bing",
            'A': 'Avenue',
            'Blvd': 'Boulevard',
           'Dr': 'Drive'
            }
```

Overview of the Data

I took the Omaha, Nebraska dataset for analysis, to generate the .json file from the data set I used the file wrangling_preparing_db.py from iPython. I gave the omaha_nebraska.osm file as an input to the wrangling_preparing_db.py file.

The output file has been named as omaha_nebraska.osm.json

I had the mongo db installed in machine already, as I did the mongo db certification in the same machine. So I just have to connect db and load the data.

First, we start up MongoDB:

```
$ mongod --dbpath ~/data/db
```

Next, we import the data:

```
$ mongoimport --db map --collection map --file ./data/ omaha_nebraska.osm.json
```

Now start mongo:

```
$ mongo
> use test
switched to db test
```

Size of the file

The original OSM file is 161 MB. The JSON file generated from the OSM file is 233MB.

Let's look at the size of the actual collection:

```
> db.wrang.dataSize()
235210640
```

Data Wrangle OpenstreetMaps Data

Author : Vijayasarathi Balasubramanian

Number of unique users:

319 users have edited this map.

```
> db.wrang.distinct("created.user").length;
319
```

Number of nodes and ways:

The omaha map contains 694,980 nodes and 79232 ways:

```
> db.wrang.find({type:"node" }).length();
694980
> db.wrang.find({type:"way" }).length();
79232
```

number of chosen type of nodes

```
There are 34 cafes in omaha
```

```
> db.wrang.find({amenity:"cafe" }).length();

34
```

```
There are 645 shops in omaha
```

```
> db.wrang.find({shop:{$exists:true}}).count();

645
```

```
> db.wrang.aggregate([{
... '$group':{
... '_id':'$created.user',
... 'count':{
... '$sum':1
... }
... }
... }
... ,{ '$sort':{'count':-1}},{'$limit':1}])
{ "_id" : "Your Village Maps", "count" : 561099 }
```

Data Wrangle OpenstreetMaps Data

Author : Vijayasarathi Balasubramanian

- **Other ideas about the datasets**

Though open street map data covering most of the business locations, it would be good if the osm files can cover all of the business locations in the map. But it would be very tedious and time-consuming to add the appropriate information to all of them. I wonder whether it would even possible to programmatically get that data from the Google Maps API. But using information from Google Maps to add information to Open Street Map seems like data stealing and might violate the terms of agreement for Google Maps. May be if there is a way to extend the open street map to include the traffic in the roads along with the number vehicles passing through each of the major roads, that will help us to predict the traffic in the roads along with the highly used roads. So we can suggest government for the list of roads we need to maintain every year, every month, once in six month something like that, this will be a proactive approach instead of reacting for the road repairs.

Also, if it’s possible we can maintain the history of accidents in the roads, which will help us to predict the most dangerous roads or accident prone junctions. We can probably suggest the road department to reduce the speed in those junctions or divert the traffic to some other freeways. Or even we can add an alert to drivers when they pass through accidently prone zones and ask them to take the safe roads.

Though it will be good to have all (like accidents prone roads, highly used roads etch) of the information’s in the same file but the bigger the file analytics will become costly. Like we need more memory to store the data, need high performance machine to run any analytics on the dataset. Probably it’s better to have a separate data set focused on the roads and its usage another data set focuses on accidents history. However having separate copies creates duplicates, so it will become repetitive datasets. Based on the use case and project funding capacity we can decide the optimal approach.