Project 2 Udacity Nanodegree OpenSteetMap Data Wrangling with MongoDB

Michal Nadolny

Map Area: Mobile, Alabama
https://mapzen.com/metro-extracts/ (find Mobile, Alabama)
https://www.openstreetmap.org/relation/254046

Project Overview

Openstreetmap.org allows anyone on the internet to crowdsource information on geographic locations throughout the work. The downfall is the majority of this information is human generated which leads to human errors that need to be audited. Luckily there people out there that find a way to systematically find these errors and fix them so that the data is more accurate. Initially wanted to audit New York/New Jersey but the exported file from Mapzen brought the file to be over 2GB. Instead I opted for Mobile, Alabama because the data was small enough for my computer to process really fast, and it reminded me of the of how warm it is down south as I look out my window and there is still snow in New Jersey this time of year. This pdf is in this directory along with the python notebook and code. Here is a link back to the document

Data Overview

Here I gather some basic details on the data through xml.etree.ElementTree

File Size:

Number of ways, nodes, and other counts:

{'bounds': 1,
'member': 1461,
'nd': 287739,
'node': 239873,
'osm': 1,
'relation': 117,
'tag': 256925,
'way': 32687}

Count of non distinct contributors to this data:

156

Street name problems found (bolded)

{'Ave': set(['Holcombe Ave', 'S Mobile Ave']),
'Avenue': set(['North Washington Avenue', 'Spring Hill Avenue']),
'Blvd': set(['Airport Blvd']),
'Boulevard': set(['Airport Boulevard', 'Eastern Shore Boulevard']),

```
 'Court': set(['Green Court', 'Southern Way Court']),
 'Dr': set(['Grishilde Dr', 'Yacht Club Dr']),
 'Drive': set(['Bass Pro Drive',
               'Dunlap Drive',
               'Gaillard Drive',
               'Golf Way Drive',
               'Museum Drive']),
 'Highway': set(['North Craft Highway']),
 'Laurel': set(['Laurel']),
 'Rd': set(['Old Shell Rd']),
 'Road': set(['Addsco Road',
              'Cody Road',
              'Howells Ferry Road',
              'North Beach Road',
              'Old Shell Road']),
 'South': set(['Schillinger Road South']),
 'Street': set(['Dauphin Street',
                'Government Street',
                'Saint Francis Street',
                'South Broad Street',
                'South Claiborne Street']),
 'Trail': set(['Old Spanish Trail']),
 'Way': set(['Cypress Way']),
 'West': set(['Hwy. 90 West'])}
```

Changes needed:

mapping = { "Ave": "Avenue", "Blvd": "Boulevard", "Dr": "Drive", "Rd": "Road"}

At this point I converted XML to JSON

File named: mobile.alabama.osm.json

Import JSON into MongoDB

To start Mongodb:

```
>cd C:\Program Files\MongoDB 2.6 Standard\bin
>mongod
```

To run Mongo Console: (in new window)

```
>cd C:\Program Files\MongoDB 2.6 Standard\bin
>mongo
```

Import (in new window)

```
>cd C:\Program Files\MongoDB 2.6 Standard\bin>
>mongoimport --db openstreetmap --collection mobile --file mobile_alabama.osm.json
```

```
connected to: 127.0.0.1
2015-03-01T16:10:14.028-0400        Progress: 11039689/65264541    16%
2015-03-01T16:10:14.031-0400             55000   18333/second
2015-03-01T16:10:17.001-0400        Progress: 27152823/65264541    41%
2015-03-01T16:10:17.001-0400             135300  22550/second
2015-03-01T16:10:20.002-0400        Progress: 41885847/65264541    64%
2015-03-01T16:10:20.002-0400             207400  23044/second
2015-03-01T16:10:22.906-0400 check 9 272560
2015-03-01T16:10:22.906-0400 imported 272560 objects
```

More Analysis on this data with MongoDB

After importing the clean JSON file into a MongoDB database here are more advanced queries with were run:

```
> use openstreetmap
switched to db openstreetmap

> db.mobile.dataSize()
86849792
```

The data in MongoDB is about 86mb.

Count of distinct contributors:

```
> db.mobile.distinct("created.user").length
153
```

Count of ways:

```
> db.mobile.find({"type":"way"}).count()
32685
```

Count of bars:

```
> db.mobile.find({"amenity":"bar"}).count()
2
```

Count of restaurants:

```
> db.mobile.find({"amenity":"restaurant"}).count()
17
```

Count of pubs:

```
> db.mobile.find({"amenity":"pub"}).count()
2
```

Count of nightclubs:

```
> db.mobile.find({"amenity":"nightclub"}).count()
0
```

Count of ice_cream:

```
> db.mobile.find({"amenity":"ice_cream"}).count()
0
```

Count of bbq:

```
> db.mobile.find({"amenity":"bbq"}).count()
0
```

Problems Encounter

I used a small data set, sharding techniques could have been set up to process a larger datasets which might have revealed some more human errors in larger data sets. There were a couple street names that could have further been cleaned up 'Highway': set(['North Craft Highway']), 'Laurel': set(['Laurel']), 'West': set(['Hwy. 90 West'])}, but in the US there are some weird names like this that make sense to leave them like they are.

References:

http://wiki.openstreetmap.org/wiki/OSM_XML
http://effbot.org/zone/element-iterparse.htm
https://docs.python.org/2/library/re.html
http://docs.mongodb.org/manual/reference/operator/aggregation/group/
http://wiki.openstreetmap.org/wiki/Key:amenity
https://docs.google.com/document/d/1F0Vs14oNEs2idFJR3C_OPxwS6L0HPliOii-QpbmrMo4/pub