OpenStreetMap Sample Project
Data Wrangling with MongoDB

Patrick Kennedy

Map Area: Austin, TX, United States

https://www.openstreetmap.org/relation/113314
http://metro.teczno.com/#austin

1. Problems Encountered in the Map

After downloading the Austin area OSM file and running it in a jupyter notebook file, I noticed three main problems with the data:

1. Over-abbreviated street names (“N. Lamar Blvd”, “W. HWY 290”)
2. Inconsistent postal codes (“TX 78745”, “TX 78759-3504”, “Texas”, “tx”)
3. Inconsistent phone numbers (“+1 (512) 714 3412”, “5124421283”) and instances of more than one number per id.

Over-abbreviated Street Names

For street names with an abbreviated cardinal direction and/or street type, I updated all substrings such that “N. Lamar Blvd” became “North Lamar Boulevard” and “W. HWY 290” became “West Highway 290”. Also, any street name with “US” in the name was changed to “U.S.”

Postal Codes

Postal codes showed several different standards, most prominently including the state abbreviation “TX” prior to the code. In some cases, only the word “Texas” appeared in the field. Codes that did not contain numbers were removed, one case of zip+4 was stripped of the trailing 4 digits and the state abbreviations preceding codes were removed.

Phone Numbers

Phone numbers showed a variety of different standards including parentheses around area codes, the presence of country codes with and without a ‘+’ character and dashes between digits. A **`X-XXX-XXX-XXXX`** standard was chosen and as such all special characters excluding the ‘-’ were stripped, a country code of ‘1’ was added if not present and dashes were added to link each piece of the phone number. If there was more than one phone number, each phone number was cleaned and multiple numbers were stored in a list excluding any text tag. Phone numbers with more than the standard 11 digits were removed.

2. Data Overview

This section contains basic statistics about the dataset and the MongoDB queries used to gather them.

File sizes

```
austin_texas.osm ......... 1.4 GB
austin_texas.osm.json .... 1.5 GB
```

Number of documents

```
> db.atx.find().count()
6977876
```

Number of nodes

```
> db.atx.find({"type":"node"}).count()
6319385
```

Number of ways

```
> db.atx.find({"type":"way"}).count()
658477
```

Number of unique users

```
> len(db.atx.distinct("created.user"))
1031
```

Top 1 contributing user

```
> db.atx.aggregate([{"$group":{"_id":"$created.user",
"count":{"$sum":1}}}, {"$sort":{"count":-1}}, {"$limit":1}])
[ { "_id" : "patisilva_atxbuildings", "count" : 2747363 } ]
```

Number of users appearing only once (having 1 post)

```
> db.atx.aggregate([{"$group":{"_id":"$created.user",
"count":{"$sum":1}}}, {"$group":{"_id":"$count",
"num_users":{"$sum":1}}}, {"$sort":{"_id":1}}, {"$limit":1}])
[ {"_id":1,"num_users":216} ]
# “_id” represents postcount
```

3. Additional Ideas

Contributor statistics

Contributions primarily come from users whose names include “_atxbuildings”. Eight of the top 10 users contain this postfix. These top 8 contributed `95.16%` of the data. In addition it looks as though one of the users, ccjjmartin_atxbuildings, has two profiles. One profile merely has an additional underscore in the name. There also seems to be a bot presence as user “woodpeck_fixbot” is the 6th highest contributing user with `226501` contributions. In contrast, the bottom 100 users contributed a total of `0.0019%` of the data.

Additional data exploration using MongoDB queries

Top 10 appearing amenities

```
> db.atx.aggregate([{"$match":{"amenity":{"$exists":1}}},
                    {"$group":{"_id":"$amenity",
                               "count":{"$sum":1}}},
                    {"$sort":{"count":-1}},
                    {"$limit":10}])

[ {"_id":"parking","count":1970},
  {"_id":"restaurant","count":710},
  {"_id":"waste_basket","count":591},
  {"_id":"school","count":554},
  {"_id":"fast_food","count":550},
  {"_id":"place_of_worship","count":491},
  {"_id":"fuel","count":391},
  {"_id":"bench","count":354},
  {"_id":"shelter","count":232},
  {"_id":"bank","count":173}]
```

Biggest religion

```
> db.atx.aggregate([{"$match":{"amenity":{"$exists":1},
"amenity":"place_of_worship"}},{"$group":{"_id":"$religion",
"count":{"$sum":1}}},{"$sort":{"count":-1}}, {"$limit":1}])
```

```
[ {"_id":"christian","count":446} ]
```

Most popular cuisines

```
> db.atx.aggregate([{"$match":{"amenity":{"$exists":1},
"amenity":"restaurant"}}, {"$group":{"_id":"$cuisine",
"count":{"$sum":1}}},{"$sort":{"count":-1}}, {"$limit":3}])

[{ "_id" : "None", "count" : 366},
{ "_id" : "mexican", "count" : 69},
{ "_id" : "american", "count" : 32}]
```

Conclusion

Upon inspection, it is clear the data is in need of some TLC. Many standards abound, especially with regard to phone numbers. Additionally, as evidenced by the most popular cuisines query, “None” seems to be the most popular in Austin. Having lived in Austin this may be more accurate than intended given the hipster-esque nature of the city. However, it is likely that many restaurants lack a cuisine value.

Here it may become useful to be able to take into account additional context from outside this particular dataset. For instance, checking restaurant names against wikipedia entries using address or pos as identifiers. By taking in additional context, a richer dataset can be populated. The trouble however is that scraping wikipedia adds yet another layer of complexity to the problem at hand. Under circumstances where it is necessary to gather this information, considerations would need to be paid to identify if the standard of cuisine type is consistent across both wikipedia entries and across the existing json file. Wikipedia itself may not be complete and another source may have to be identified. Yelp would be a good replacement however I wouldn’t know how amenable Yelp data is to scrape. And, again, consistency and reliability would need to be tested against this data source.

Manual coding could be explored and with it different types of complications arise. Data can be verified more easily, but research and entry takes an exponentially longer period of time. If the entry can be crowdsourced in the same manner Waze develops city road maps, the process can become less cumbersome. That takes building an entirely new platform of course and may not be worth the effort. So like in all things, the goal of the project must be weighed against the resources in order to identify the best path forward.