OpenStreetMap Data Project

Map Area

Miami, FL, United States

- https://www.openstreetmap.org/relation/1216769
- https://mapzen.com/data/metro-extracts/metro/miami_florida/

I have been interested in this city for a long time since I often visited there when I was in my graduate school University of Florida, so I'm very curious to see what database querying reveals, and I'd like an opportunity to contribute to its improvement on OpenStreetMap.org.

Problems Encountered in the Map

After initially downloading the map of the Miami area and running it against a small size of sample file, I noticed two main problems with the data, which I will discuss in the following order:

1. Over abbreviated street names ('Granada Ave', 'Bamboo Ln)
2. Inconsistent postal codes ('FL 33140', 'FL 33431-4403')

Abbreviated Street Names

To deal with correcting abbreviated street names, I will use regular expression to iterate over each last word in an address, and correct them to their respective mappings by the following function:

```
def update_name(name, mapping):
    m = street_type_re.search(name)
    if m:
        street_type = m.group()
        if street_type in mapping.keys():
            new_name = re.sub(street_type, mapping[street_type], name)
    return new_name
```

This updated all substrings in problematic address strings, for example:

“NW 66th Ave” becomes: “NW 66th Avenue”.

Post Codes

Postal code strings have a different sort of problem, so I need to strip all leading and trailing characters before and after the main 5digit zip code. This effectively dropped all leading state characters (as in “FL 33140”) and 4digit zip code

extensions following a hyphen (“FL 33431-4403”). I will use regular expression and iteration for each string in a postcode dictionary, as following:

```
post_code_re = re.compile(r'^\D*(\d{5}).*')

def update_postcode(postcode):

    m = post_code_re.search(postcode)

    if m:

        better_pc = m.group(1)
```

After standardizing inconsistent postal codes, I put these all post codes together to see if there are any other things to figure out with this aggregation:

```
select tags.value, count(*) as num\

from (select * from nodes_tags Union all select * from ways_tags) as tags\

where tags.key = 'postcode'\

group by tags.value\

order by num DESC;
```

Here are the top ten results, beginning with the highest count:

```
(u'33327', 6770),
(u'33326', 6472),
(u'33331', 3117),
(u'33135', 2150),
(u'33332', 1882),
(u'33125', 1796),
```

```
(u'33142', 1782),
(u'33133', 1369),
(u'33145', 1283),
(u'33126', 1210)
```

From the postcode ranking, we could found that this map is the Miami metropolitan area which is the metropolitan area including Miami and nearby communities. The metropolitan statistical area comprises Miami-Dade, Broward, and Palm Beach counties—Florida's three most populous, together forming South Florida—with principal cities including Miami, Fort Lauderdale, Pompano Beach, West Palm Beach, and Boca Raton.

And there are two postcodes ‘82941' and '91730', which are showed one time, respectively. ‘82941’ is the zip code for Pinedale, Wyoming and '91730' is for Rancho Cucamonga, CA. These type of error we could guess it was typo when people were entering into the map. Anyway, it is easy to be deleted from the map data document.

What is more, in order to prove what I found that this map is the Miami metropolitan area, I will sort cities in this map by count.

Sort cities by count, descending

Now we will sort the different cities in this map by this aggregation:

```
"Select tags.value, count(*) as num \

From (select * from nodes_tags Union all select * from ways_tags) as tags\

Where tags.key like '%city'\
```

Group by tags.value\

Order by num DESC;"

And, the results, edited for readability:

```
(u'Weston', 18244),
 (u'Miami', 14166),
 (u'Hialeah', 1382),
 (u'Miami Beach', 667),
 (u'Homestead', 471),
 (u'Coral Gables', 457),
 (u'Doral', 368),
 (u'North Miami', 365),
 (u'Miami Gardens', 341),
 (u'North Miami Beach', 315),
 (u'Fort Lauderdale', 267),
 (u'Opa Locka', 208),
 (u'Miami Springs', 158),
 (u'Cutler Bay', 145),
 (u'Palmetto Bay', 135),
 (u'West Palm Beach', 134),
 (u'Coral Springs', 116),
 (u'Sunny Isles Beach', 110),
 (u'Florida City', 107),
 (u'Wellington', 105)
 (u'Naranja', 89),
 (u'Boca Raton', 88),
 (u'Royal Palm Beach', 80),
 (u'Medley', 77),
 (u'Aventura', 73),
 (u'Miami Lakes', 63),
```

```
(u'South Miami', 61),
(u'West Miami', 58),
(u'Pelican Lake', 57),
(u'Pinecrest', 57)
```

I listed the cities which appearing over 50 times in the map. Then, this result confirms the thinking I mentioned above that this map is actually the Miami metropolitan area including Miami and nearby communities, such as ‘Fort Lauderdale’, ‘West Palm Beach’, and so on.

Data Overview and Additional Ideas

This section contains basic statistics about the dataset, the sql queries used to gather them, and some additional ideas about the data in context.

File sizes

```
Miami_florida.osm ......... 590.8 MB
Miami_florida.db .......... 342.5 MB
nodes.csv ............... 218.5 MB
nodes_tags.csv ........ 11.4 MB
ways.csv ............... 19.9 MB
ways_tags.csv ......... 56.3 MB
ways_nodes.cv ......... 69.9 MB
```

Number of Nodes

```
Select count(*) as num from nodes;
```

```
2521515
```

Number of Ways

```
Select count(*) as num from ways;
```

```
318815
```

Number of Unique Users

```
Select count(distinct (map.uid)) from (select uid from nodes\
UNION ALL select uid from ways) as map;
```

```
1629
```

Top 10 Contributing Users

```
Select map.user, count(*) as num from (select user from nodes\
UNION ALL select user from ways) as map\
group by map.user\
```

order by num DESC\

limit 10;

```
(u'MiamiBuildingsImport', 857814),
(u'grouper', 333762),
(u'woodpeck_fixbot', 228308),
(u'carciofo', 205148),
(u'Latze', 137144),
(u'freebeer', 117671),
(u'bot-mode', 55969),
(u'NE2', 51698),
(u'Seandebasti', 47406),

(u'westendguy', 46308)
```

Number of Users Appearing only Once

select count(*) from(select map.uid, count(*) as num from\

(select uid from nodes\

UNION ALL\

select uid from ways) as map\

group by map.uid\

having num =1) as users;

498

Additional Ideas

Contributor statistics

Here are some user percentage statistics:

- Top user contribution percentage (“`MiamiBuildingsImport`”) 30.20%
- Combined top 2 users' contribution (“`MiamiBuildingsImport`” and “`grouper`”) 41.95%
- Combined Top 10 users contribution 73.27%

The top two contributors are government organization “MiamiBuildingsImport” and some group called “grouper”, and there is also some automatic edit which includes “bot” in the username. From the top ten contributors statistics, I think the individual and handful edit still need to be stimulated by some incentive ways, such as prize, rewards, or grade system. These may make people more likely to edit the map. Also, if there are some competitions for bots, it is very possible to get more efficient editing bots for the map.

From these statistics, it inspires me for another idea which are some mobile games linking urban cities with data community probably to improve the map. Currently, urban-related data and geographic information are becoming mainstream in the Linked Data community due to the popularity of Location-based Services. For example, there is a game ‘UrbanMatch’, a mobile gaming application that joins data linkage and data quality assessment in an urban

environment. When people play this game on their cell phone, it is more possible for them to edit the openstreetmap data individually. For this new idea, there are some benefits and potential problems as following:

Benefits

1. More and more people will be motivated to join the map data community because of the popularity of mobile games with Location-based Services.
2. People are more and more used to “check-in” physical places with their mobile devices and to add small bits of data information related to their activities and actions in the physical world.

Anticipated Problems

1. This is still very new type of mobile games, so the further evaluation is needed to involve more people to consume, create and improve urban-related linked map.
2. Data linkage and content creation or assessment tasks need an active and productive engagement of users, but not all crowdsourcing campaigns are successful and effective. So making Games with a Purpose of Human Computation missions maybe a potential solution to this issue. But finding a good balance between the game rules and the purpose achievement is often a hard task.

Additional Data Exploration

Top 10 appearing amenities

```
select tags.value, count(*) as num from\

(select * from nodes_tags\

UNION ALL\

select * from ways_tags) as tags\

group by tags.value\

having tags.key = 'amenity'\

order by num DESC\

limit 10;
```

```
(u'parking', 2638),
(u'school', 2285),
(u'restaurant', 673),
(u'place_of_worship', 614),
(u'kindergarten', 529),
(u'fast_food', 493),
(u'fuel', 484),
(u'fire_station', 312),
(u'fountain', 261)
(u'bank', 240)
```

Biggest religion

```
select tags.value, count(*) as num from\
```

```
(select * from nodes_tags\

Union ALL\

select * from ways_tags) as tags\

where tags.key = 'religion'\

group by tags.value\

order by num DESC\

limit 3;
```

```
(u'christian', 564), (u'jewish', 17), (u'buddhist', 2)
```

Most popular cuisines

```
select tags.value, count(*) as num from\
(select * from nodes_tags\
Union ALL\
select * from ways_tags) as tags\
where tags.key = 'cuisine' or tags.key = 'restaurant'\
group by tags.value\
order by num DESC\
limit 10;
```

```
(u'burger', 188),
(u'sandwich', 64),
(u'pizza', 56),
(u'coffee_shop', 55),
(u'donut', 43),
```

```
(u'italian', 40),
(u'chicken', 34),
(u'american', 31),
(u'mexican', 27),
(u'chinese', 18)
```

Conclusion

In this Openstreetmap Data Project, firstly, iterating the big data file to clean and correct the data for two major problems: Over abbreviated street names and Inconsistent postal codes. Then we use sqlite to analyze the data file for different topics which we are interested in, such as contributors for this map, cities including in this map and so on. After analysis and review, we realize that the Miami map is not only for Miami city, but the whole Miami metropolitan area including Miami and nearby communities. Moreover, one thing I found interesting in this research is that there are some editing bots to edit this map. Comparing to handful edit and type by individual person, the bots maybe more efficient, and make more clean data in the original data file, which could save a lot of time for analysts to clean and correct data I think.

Reference

1. OpenStreetMap Data Case Study

   https://gist.github.com/carlward/54ec1c91b62a5f911c42#file-sample_project-md

2. Miami metropolitan area

   https://en.wikipedia.org/wiki/Miami_metropolitan_area

3. UrbanMatch – linking and improving Smart Cities Data

   http://events.linkeddata.org/ldow2012/papers/ldow2012-paper-10.pdf