Openstreet Map

Map Area

HartfordCountry, CT, United States

https://www.openstreetmap.org/relation/1839541#map=10/41.7933/-72.7185
I have been living in this county for more than four years, so I’m interested in looking into the data of this area, and I’d like an opportunity to contribute to its improvement on OpenStreetMap.org.

1. Problems Encountered in the Map

After initially extracting a small sample size of the Hartford Country area and running it against a provisional data.py file, I noticed three main problems with the data, which I will discuss in the following order:

- Inconsistent street names ("Road" vs "Rd", "Highway" vs "Hwy", etc.)
- Inconsistent city names ("New Haven, CT", "Hartford, CT:New Haven, CT", "Litchfield, CT:New Haven, CT")
- Inconsistent phone numbers (“+1 203 5740096”, “+1 860 223 2885”, “860-667-4000”, & "8602161255")

Problem 1 Inconsistent street names

All substrings in problematic address strings were updated, with the full names of street types replacing the abbrevations， by using this following update_street function.

In [4]:

```
def update_street(name, mapping):

    sorted_keys = sorted(mapping.keys(), key=len, reverse=True)

    for abbrv in sorted_keys:
        if(abbrv in name):
            return name.replace(abbrv, mapping[abbrv])

    return name
```

Here are some examples after update_street function was used:

Ln => Lane
Expy => Expressway
Rd => Road
Pl => Place
Hwy => Highway
Dr => Drive
Trl => Trail
Tpke => Turnpike
St => Street
Cir => Circle
Blvd => Boulevard

Ave => Avenue
Ter => Terrace
Ct => Court

Problem 2 Inconsistent city names

All substrings in problematic city strings were updated to make them consistent to one another, by using this following update_city function.

In [10]:

```
def update_city(name, mappingcity):

    sorted_keys = sorted(mappingcity.keys(), key=len, reverse=True)

    for abbrv in sorted_keys:
        if(abbrv in name):
            return name.replace(abbrv, mappingcity[abbrv])

    return name
```

Here are some examples after update_city function was used:

New London, CT:Tolland, CT => Tolland, CT
Hartford, CT; Tolland, CT:Tolland, CT => Tolland, CT
Hartford, CT:Middlesex, CT => Middlesex, CT
Hartford, CT:New Haven, CT => New Haven, CT
Middlesex, CT:New Haven, CT => New Haven, CT
Hartford, CT; Tolland, CT => Tolland, CT
Middlesex, CT:New London, CT => New London, CT

Litchfield, CT:New Haven, CT => New Haven, CT
Hartford, CT:Litchfield, CT => Litchfield, CT
; Hartford, CT => Hartford, CT
Hartford, CT:New London, CT => New London, CT

Problem 3 Inconsistent phone numbers

All substrings in problematic phone numbers were updated to make them consistent to one another by using this following update_phone function.

In [9]:
```
def update_phone(name, mappingphone):

    sorted_keys = sorted(mappingphone.keys(), key=len, reverse=True)

    for abbrv in sorted_keys:
        if(abbrv in name):
            #print(abbrv)
            return name.replace(abbrv, mappingphone[abbrv])

    return name
```

Here are some examples after update_phone function was used:

+1 203 5740096 => 203-574-0096
8602161255 => 860-216-1255
+1 860 223 2885 => 860-223-2885
+860 793 7815 => 860-793-7815
+1 860 666 2009 => 860-666-2009

+1 860 666 2000 => 860-666-2000
+1 860 922 5329 => 860-922-5329

2. Data Overview

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

File Size

Hartford.db .... 139 MB
HartfordCountry.osm .... 244 MB
nodes.csv .... 96.5 MB
nodes_tags.csv .... 8.30 MB
ways.csv .... 7.16 MB
ways_nodes.csv .... 30.7 MB
ways_tags.csv .... 14.9 MB

Number of nodes

In [ ]:

```
sqlite> SELECT COUNT(*) FROM nodes;
```

1155917

Number of ways

In [ ]:

```
sqlite> SELECT COUNT(*) FROM ways;
```

115980

Number of unique users

In [ ]:

```
sqlite> SELECT COUNT(DISTINCT uid) FROM (SELECT uid FROM nodes UNION ALL SELECT uid FROM ways);
```

716

Number of cafes in nodes

In [ ]:

```
sqlite> SELECT COUNT(DISTINCT id) FROM nodes_tags WHERE value="cafe";
```

45

Ways with most nodes top 10

In [ ]:

```
sqlite> SELECT ways_nodes.id, ways_tags.value, COUNT(*) as num FROM ways_nodes JOIN ways_tags on (ways_nodes.id=ways_tags.id)
WHERE ways_tags.key = "name" GROUP BY ways_nodes.id ORDER BY num DESC LIMIT 10;
```

```
348352747|Metacomet Trail (blue blazes)|1358
347769348|Metacomet Trail|1200
193301986|white blazes|1149
71465383|Barkhamsted Reservoir|879
352700201|Metacomet Trail (blue blazes)|806
418306914|(blue-white-dot blazes)|777
361350649|(blue blazes)|752
362076670|mountain bike trail|732
396331569|(blue blazes)|729
364020745|mountain bike trail|696
```

How many users have made contribution

In [ ]:

```
sqlite> SELECT COUNT(DISTINCT user) as num FROM (SELECT user from nodes UNION ALL SELECT user from ways);
```

```
716
```

Top 10 users

In [ ]:

```
sqlite> SELECT user, COUNT(*) as num FROM (SELECT user from nodes UNION ALL SELECT user from ways)
GROUP BY user ORDER BY num DESC LIMIT 10;
```

```
jremillard-massgis|424054
```

```
David Reik|124752
maxerickson|101175
Leo22|64540
KindredCoda|48410
Tomash Pilshckik|40390
FourOhFour|38943
MassGIS Import|38847
abar|36871
J_Hutch|24740
```

3. Additional Ideas

Which type of shops is the most popular in Hartford area?

In [ ]:

```
sqlite> SELECT value, count(*) as num FROM notes_tags WHERE key="shop" GROUP BY value ORDER BY num DESC LIMIT 1;
```

```
supermarket|52
```

What is the proportion of supermarket in all shops in Hartford are?

In [ ]:

```
sqlite> SELECT count(*), 52.00/count(*) as proportion FROM nodes_tags WHERE key="shop";
```

```
384|0.1354
```

Therefore 13.54% of the 384 shops in Hartford area are supermarkets.

Contribution of TIGER (Topologically Integrated Geographic Encoding and Referencing system) to way data

In [ ]:

```
sqlite> SELECT count(DISTINCT id) FROM ways_tags WHERE type="tiger";
```

```
26424
```

In [ ]:

```
sqlite> SELECT 26424.0/(SELECT count(DISTINCT id) FROM ways_tags);
```

```
0.2286
```

Therefore tiger contributed 22.86% of the way information.

4. Discussion and Conclusion

After reviewing the data, I found more investigations are needed to clarify some details of Hartford area, for example some way nodes have parenthesis on their names without any explanation. It looks like a fair amount of information of ways was collected through TIGER. Hundreds of users have helped complete OpenStreetMap, and the person who made the most contribution provided more than 400 thousand pieces of information. In general, while the data is not 100% clean, I believe it was sufficiently cleaned for the purposes of this project, and I definitely learned some new information about Hartford area through the map.

I think the maps can still be improved in some respects. The fact that many individual users are making contribution to the OpenStreetMap makes it grow very quickly and very promising. However, one possible problem is that as a map that can be edited by anyone, it is relatively more vulnerable. It will be more convenient to map users if, in addition to the map information, they could also get the information about how credible of the map in this specitic area, using something like users' rating. If someone find a serious mistake on the map, the ideal situation is that they correct it immediately. However, if they are not able to do that right away, at least they could remind other people it is not accurate through rating. In this way, both quantity and quality of the information on the map can be more guaranteed.

In [ ]: