OpenStreetMap Project for Data Wrangling with MongoDB

July 24, 2015

1 Katherine Tansey

1.1 Map Area: Area around Bridgewater NJ USA

2 Introduction

I choose this area of central New Jersey as it is where I grew up, so I am very famaliar with the area. I selected an area to meet the specified size (greater than 50MB), which encompassed more than just my hometown, and included other nearby towns.

3 Problems Encountered

3.1 Road Names

There are few highways in the area, and their names were not consistent, for example "US 206" and "Route 206 South", and "Rt-31" and "NJ 31 S" refer to the same road. These were standardized to have the same name throughout entries, and were changed "Route ##". There was also a street called "Road 3" which refered to "Route 3", and this was changed to align with the other highway names.

Some streets were contained multiple abbreviations, i.e. "W. High St.", and these were changed to have no abbreviations, i.e. "West High Street."

Some streets had older (no longer in use) names in the parenthesis (i.e. Mendham Road (Old Rt. 24) and W. Main Street (Old Rt. 24)), these were updated to only include the new names of the streets. There was also a road with extra information in parenthesis about direction (JOYCE KILMER AVENUE (S/N TWRS), this was also removed.

There was location that included very specific information for the street name, "80 Morristown Road Unit 17", and the ending information for the specific unit in the street name was removed so the street name was just the road name ("80 Morristown Road").

There were multiple street names which were all capitalized, i.e. "WINCHESTER ROAD", and these were changed to only have the first letter of each word was capitalized, i.e. "Winchester Road".

All street type abbreviations were standardized to be the same, i.e. "Ave", "Ave." and "AVE." were all changed to "Avenue".

See code titled "improving_street_names.py". This was also implemented in "preparing_for_database.py", the code inserted into this script looked liked this:

```
elif key == "addr:street":
 street = val
 for key in mapping_street:
      if street.endswith(key):
           street = street.replace(key, mapping_street[key])
           street = street.title()
           node['address']['street'] = street
      else:
                continue
 node['address']['street'] = street
```

Example of the output for the changes:
Mendham Road (Old Rt. 24) => Mendham Road
W. Main Street (Old Rt. 24) => West Main Street
W. High St. => West High Street
US 22 => Route 22
ROAD 3 => Route 3
ALLISON ROAD => Allison Road
NEW ENGLAND AVE. => New England Avenue
US 206 => Route 206
JOYCE KILMER AVENUE (S/N TWRS) => Joyce Kilmer Avenue
TITSWORTH PLACE => Titsworth Place
80 Morristown Road Unit 17 => 80 Morristown Road
HOES LANE WEST => Hoes Lane West

3.2 Zip codes

There are multiple zip codes in the area selected, and all of them were correct for the area choosen. Some of the zip codes included the extra 4 digits at the end, these were removed to leave just the 5 zip code designation.

See code titled "zip_codes.py". This was also implemented in "preparing_for_database.py", the code inserted into this script looked liked this:

```
elif key == "addr:postcode":
 if '-' in val:
      sep = '-'
      zip = val.split(sep, 1)[0]
      node['address']['postcode'] = zip
 else:
      node['address']['postcode'] = val
```

Example output of the change:
08854-3929 => 08854
08854-8019 => 08854
08854-8018 => 08854
08854-8030 => 08854
08854-8076 => 08854
08854-8012 => 08854

4 Prepare data for MongoDB

All changes were made in one document and the data reformatted from a XML to a JSON.

See code titled "preparing_for_database.py".

5 Data Overview

5.0.1 Import into MongoDB

Imported the JSON file into MongoDB using the mongoimport command as follows: mongoimport -d udacity -c maps –file /Users/katherinetansey/Dropbox/udacity/project3/bridgewater_nj.txt.json

This imported 406031 documents.

5.1 Size of the file

Intial XML File ('bridgewater_nj.txt') - 92 MB
Final JSON File ('bridgewater_nj.txt.json') - 116.3 MB

5.1.1 Number of documents

```
In [9]: from pymongo import MongoClient
        import pprint
        client  = MongoClient('localhost', 27017)
        db = client.udacity.maps

        db.find().count()
```

```
Out[9]: 432978
```

There are 432,978 documents in the database.

5.1.2 Number of nodes and ways

```
In [26]: db.find( { 'type' : 'node' } ).count()
```

```
Out[26]: 403764
```

```
In [27]: db.find( { 'type' : 'way' } ).count()
```

```
Out[27]: 29214
```

There are more documents of the type 'node' (403,764, ˜ 93.25%) than there are of 'ways' (29,214, 6.75%)

5.2 Contributing users

5.2.1 Total number of contributing users

```
In [23]: len(db.distinct("created.user"))
```

```
Out[23]: 346
```

There are 346 different contributing users to the data.

5.2.2 Top 5 contributing users

```
In [24]: results = db.aggregate( [ { '$match' : { 'created.user' : { '$exists' : 1 } } },
                        { '$group' : { '_id' : '$created.user' , 'count' : { '$sum' : 1 } } },
                        { '$sort' : { 'count' : -1 } },
                        { '$limit' : 5 } ] )
         pprint.pprint(list(results))
```

```
[{u'_id': u'NJDataUploads', u'count': 130595},
 {u'_id': u'woodpeck_fixbot', u'count': 115867},
 {u'_id': u'bhousel', u'count': 51600},
 {u'_id': u'choess', u'count': 22247},
 {u'_id': u'NE2', u'count': 8752}]
```

The top 5 contributors have in total added 329,061, about 76% of all the documents, with the two top contributing users contributing about 30.16% and 26.72% each respectively.

6 Additional Data Exploration using MongoDB

6.1 Top 10 amenitites for the area

```
In [25]: results = db.aggregate( [ { '$match' : { 'amenity' : { '$exists' : 1} } },
                        { '$group' : { '_id' : '$amenity' , 'count' : { '$sum' : 1 } } },
                        { '$sort' : { 'count' : -1 } },
                        { '$limit' : 10 } ] )
         pprint.pprint(list(results))
```

```
[{u'_id': u'parking', u'count': 664},
 {u'_id': u'school', u'count': 276},
 {u'_id': u'place_of_worship', u'count': 177},
 {u'_id': u'restaurant', u'count': 74},
 {u'_id': u'grave_yard', u'count': 49},
 {u'_id': u'bank', u'count': 38},
 {u'_id': u'fast_food', u'count': 34},
 {u'_id': u'fuel', u'count': 19},
 {u'_id': u'library', u'count': 17},
 {u'_id': u'pharmacy', u'count': 17}]
```

The most common amenity is parking, then schools. There are few restaurants (only 74) in comparison to religious buildings (places of worship is 276, almost 4 times as many).

6.2 Religion

There are a lot of Places of Worship in the area. We can query which religion is most prominent in the area.

```
In [26]: results = db.aggregate( [ { '$match' : { 'religion' : { '$exists' : 1 } } },
                        { '$group' : { '_id' : '$religion' , 'count' : { '$sum' : 1 } } },
                        { '$sort' : { 'count' : -1 } } ] )
         pprint.pprint(list(results))
```

```
[{u'_id': u'christian', u'count': 170},
 {u'_id': u'muslim', u'count': 2},
 {u'_id': u'unitarian_universalist', u'count': 2},
 {u'_id': u'hindu', u'count': 1},
 {u'_id': u'unitarian', u'count': 1},
 {u'_id': u'jewish', u'count': 1}]
```

The majority of places fo worship in the area are (by far) Christian. We can also break this down by specific denomination.

```
In [27]: results = db.aggregate( [ { '$match' : { 'denomination' : { '$exists' : 1 } } },
                        { '$group' : { '_id' : '$denomination' , 'count' : { '$sum' : 1 } } },
                        { '$sort' : { 'count' : -1 } } ] )
         pprint.pprint(list(results))
```

```
[{u'_id': u'catholic', u'count': 24},
 {u'_id': u'methodist', u'count': 15},
 {u'_id': u'baptist', u'count': 14},
 {u'_id': u'lutheran', u'count': 8},
 {u'_id': u'roman_catholic', u'count': 6},
 {u'_id': u'presbyterian', u'count': 5},
 {u'_id': u'jehovahs_witness', u'count': 2},
 {u'_id': u'mormon', u'count': 1},
```

```
 {u'_id': u'pentecostal', u'count': 1},
 {u'_id': u'Worldwide Missionary Movement (Movimiento Misionero Mundial)',
  u'count': 1},
 {u'_id': u'Jafri', u'count': 1},
 {u'_id': u'hare_krishna', u'count': 1}]
```

And we can see that the major denomination for Christian is Catholic followed by Methodist and Baptist.

6.3 Restaurants

Not all of the restaurants have a specific cuisine listed, but we can run a query to get the total number of restuarants per cuisine types for those with that information.

```
In [28]: results = db.aggregate( [ { '$match' : { 'cuisine' : { '$exists' : 1 } } },
                        { '$group' : { '_id' : '$cuisine' , 'count' : { '$sum' : 1 } } },
                        { '$sort' : { 'count' : -1 } } ] )
         pprint.pprint(list(results))
```

```
[{u'_id': u'american', u'count': 9},
 {u'_id': u'burger', u'count': 8},
 {u'_id': u'coffee_shop', u'count': 6},
 {u'_id': u'pizza', u'count': 5},
 {u'_id': u'italian', u'count': 4},
 {u'_id': u'ice_cream', u'count': 4},
 {u'_id': u'doughnut', u'count': 3},
 {u'_id': u'irish', u'count': 2},
 {u'_id': u'sandwich', u'count': 2},
 {u'_id': u'mexican', u'count': 1},
 {u'_id': u'donut', u'count': 1},
 {u'_id': u'indian', u'count': 1},
 {u'_id': u'portuguese BBQ', u'count': 1},
 {u'_id': u'Sandwiches, Salads', u'count': 1},
 {u'_id': u'asian', u'count': 1},
 {u'_id': u'seafood', u'count': 1},
 {u'_id': u'african', u'count': 1},
 {u'_id': u'chicken', u'count': 1},
 {u'_id': u'sushi', u'count': 1},
 {u'_id': u'chinese', u'count': 1},
 {u'_id': u'steak_house', u'count': 1},
 {u'_id': u'coffee', u'count': 1}]
```

The cuisine in the area is mostly american and burger places with a few other types. Examine information specific to Fast food places.

```
In [29]: results = db.aggregate( [ { '$match' : { 'amenity': 'fast_food', 'name': { '$exists' : 1} } },
                                  { '$group' : { '_id' : '$name' , 'count' : { '$sum' : 1 } } },
                                  { '$sort' : { 'count' : -1 } } ] )
         pprint.pprint(list(results))
```

```
[{u'_id': u'Burger King', u'count': 4},
 {u'_id': u"Wendy's", u'count': 3},
 {u'_id': u"McDonald's", u'count': 3},
 {u'_id': u'Subway', u'count': 2},
 {u'_id': u'Dunkin Donuts', u'count': 2},
 {u'_id': u'Taco Bell', u'count': 1},
 {u'_id': u'Creamery', u'count': 1},
```

```
{u'_id': u"Rita's Water Ice", u'count': 1},
{u'_id': u"Gianni's Pizzarama", u'count': 1},
{u'_id': u'Chipotle', u'count': 1},
{u'_id': u'McCormick & Schmick\u2019s', u'count': 1},
{u'_id': u'Qudoba Mexican Grill', u'count': 1},
{u'_id': u'Cold Stone Creamery', u'count': 1},
{u'_id': u'Baskin Robins', u'count': 1},
{u'_id': u"Don's", u'count': 1},
{u'_id': u'Blimpie', u'count': 1},
{u'_id': u'Cluck-U Chicken', u'count': 1},
{u'_id': u'25-Burgers', u'count': 1},
{u'_id': u'Cups Frozen Yogurt', u'count': 1},
{u'_id': u'Dunkin Dounuts', u'count': 1},
{u'_id': u'Muscle Maker Grill', u'count': 1},
{u'_id': u'Panera Bread', u'count': 1},
{u'_id': u'Donkin Donout', u'count': 1}]
```

The top fast food place is Burger King.

7 Other Ideas about the datasets

From my point of view, I think that the data was fairly clean, and just needed some minor edits in terms of standardizing street names and zip code designations. However, some of the street names needed to be standardized in how they were refered to (like highways), and this could be avoided in the future by establishing a standard system for inputting street names (i.e. no abbreviations, all highways are called Route ##, etc.). This could help prevent differences between individuals data contributors and streamline the information. This would potentially reduce the number of errors in street names, and also ensure that all roads are labelled the same.

There were two main contributing users who are responsible for over 60% of the data for the area. In total, there were 346 users that contributed data, and I feel this number is probably low for the size of the area/data, as there are still a lot of missing information for the area. Involving more people into the data curation would likely greatly increase the annotation of the area. One potential way to increase the number of contributing users might be to have contributing be part of a school project which would encourage people to take part. This would also help in the issues highlighted below, which refer to increasing the depth of the information included.

Parking was the most common amenity in the region. Even this amenity could have more valuable information, like if it is just a lot or a parking structure. Potentially there could also be information included for how many space there are for each parking amenity.

Inclusion of more information could also be extended to other amenities, like the schools or colleges in the area. These could be annnotated into types (elementary, middle, high, community college, 4-year university, etc.) to see what types of schools are the most prominent in the area, but this information was not in the XML. This information could be incorporated by using a new k which could be education level, with values coding for which level the school is.

There were not that many restaurants with most of them being burger or american places. Being familiar with the area, there are more restuarants than are in the database currently. Maybe integrating in information from other sources, like googlemaps, restaurant directories, or the yellow pages. Cross checking with other database will reveal missing information that then can be easily inserted into the openstreetmap project.

There was no information about retail stores (i.e. clothing store, electronic store, etc.) which should be included. These could be included in two steps, like for restuarants, where they are listed under amenity as a retail store and then another k tag to differentiate type of store, like clothing or electronic. This would greatly increase the annotation of the area and the wealth of the data.

8 Conclusions

Overall, I think the data for the area was interesting and fairly clean. There is room for streamlining and expanding the annotation in the area.