Data Analyst Nanodegree

Project 3
Wrangle OpenStreetMap Data

Section 1: Problems encountered in the map

After downloading a map of Detroit, Michigan using Map Zen, several problems were immediately noticeable. The problems which I programmatically reduced and/or solved were non-standard street type names and inconsistent zip code formats.

Non-standard Street Type Names

Auditing the street type names revealed many values which were not listed by the USPS as official street type names. To generate a list of official capitalized street type names, I used a series of regular expressions to scrape the HTML used to generate the main table at the following url: http://pe.usps.gov/text/pub28/28apc_002.htm

Although all of the street type names in the dataset could not be converted to the official street type names, many of them could. I was able to programmatically replace street type names such as:

- Ave
- Ave.
- Blvd
- Blvd.
- Ct
- DR
- Pkwy
- etc...

with the official street type names listed the referenced USPS web page.

Inconsistent Zip Code Formats

All of the zip codes provided in the dataset were not in a consistent format. Some examples of provided zip codes are:

- 48060
- 48068-9998
- N7T 7B4
- ON N8N 0B8

Since some of the zip codes were in Canada while others were in the United States, there was not a single format which was suitable for all of the zip codes. However, I chose a signal format to use for each country and programmatically formatted each zip code to conform to the relevant format if it already did not. After this formatting step, all the US zipcodes were in the form "48068" while all the Canadian zip codes were in the form "N8N 0B8"

Section 2: Overview Of the Data

Size of Uncompressed XML File:
kejas-MacBook-Air:p3 krowe$ ls -lah detroit_michigan.osm
-rw-r-----@ 1 krowe staff **694M** Nov 29 15:11 detroit_michigan.osm

Size of Uncompressed JSON File:
kejas-MacBook-Air:p3 krowe$ ls -lah detroit_michigan.osm.json
-rw-r--r-- 1 krowe staff **793M** Dec 2 08:38 detroit_michigan.osm.json

Number of unique users:
››› len(db.detroit.distinct('created.user'))
1434

Number of nodes and ways:
››› db.detroit.find({"type":"node"}).count()
3378176
››› db.detroit.find({"type":"way"}).count()
310164

Most common amenities in Detroit
››› [doc for doc in db.detroit.aggregate([
... {"$group":{"_id":"$amenity","count":{"$sum":1}}},
... {"$sort":{"count":-1}}
...])]

[{u'count': 3671592, u'_id': None},
{u'count': 5990, u'_id': u'parking'},
{u'count': 2910, u'_id': u'school'},
{u'count': 1739, u'_id': u'place_of_worship'},
{u'count': 922, u'_id': u'restaurant'}...

Section 3: Other ideas about the Dataset

The dataset could benefit from further cleaning. Although some of the street types were successfully cleaned, there are many instances where the street type is simply not provided. To determine the street type, perhaps a database of local road names could be used. If the other fields in the address such as street name, house number, and zipcode are cross referenced against this other database, then it might be possible to determine the street type when it has not been provided in the OpenStreetMap dataset.

Additionally there are quite a few keys in the dataset which are unclear and hard to interpret. Some examples of such keys are the following:

- tiger:name_direction_suffix_2
- ref:bag
- point_kilo
- cutting

I saw that the OpenStreetMap wiki provided some explanation of some keys as well as discussion of the grammar / syntax which should be used when providing the values for those keys. However, it seems that a relatively small number of keys are described in the wiki. Allowing users to define arbitrary keys and provide values for them do give the OpenStreetMap documents a lot of flexibility. However, in cases such as those above, that flexibility comes at the cost of uniformity and clarity.