OpenStreetMap Data Analysis for San Diego, CA

Why San Diego?

I have traveled to many places but San Diego is one of my favorite cities. This project provides an unprecedented opportunity for me to dig deeper into the city. The following link was used to retrieve the data:

https://mapzen.com/data/metro-extracts/metro/san-diego_california/

Parsing the Data Using the ElementTree XML API

First, let's get an overall idea of the size of our dataset by counting the number of tags in the xml file. This is done using the script *tags.py*. Here's the output:

"defaultdict(<type 'int'>, {'node': 1041623, 'nd': 834206, 'bounds': 1, 'member': 13500, 'tag': 2556910, 'relation': 767, 'way': 94784, 'osm': 1})"

This is a pretty large dataset. We'll do similar analysis using sqlite queries later in the project.

We can also explore the data a bit more to see if there are tags with problematic characters. This is also done using the script *tags.py*. Here's the output:

"{'lower': 648901, 'lower_colon': 1871409, 'other': 36595, 'problemchars': 5}"

We have 5 tags with problematic characters.

Problems Encountered in the Map

1) Inconsistencies in the street name. The script *audit.py* lists all different formats that have been used to name the streets. There are several inconsistencies in the street names. It looks like we have some cleaning to do. The function "update_name()" in the script *open_street.py* maps the street names to the appropriate names. The clean data is then written into the csv files.

2) Inconsistencies in the phone numbers. A quick look at a small section of the data reveals that several formats have been used to list the phone numbers. The phone numbers have been updated to 1) keep the digits (hyphens, dots, and other characters were dropped), and 2) drop the country code if it exists. The clean data were used to write the csv files. The function "update_phone()" within the script *open_street.py* does this for us.

Importing CSV Files into SQL Databases

Next, we need to import the csv data into a sqlite database to execute some queries. The scripts *create_db.py* creates a sqlite database containing five different tables namely: “nodes_tags”, “ways_tags”, “nodes”, “ways”, and “ways_nodes”.

Data Overview and New Insights

To obtain an overall overview of the database, I first ran some basic queries. The script *db_query.py* contains commented queries.

1) The total number of nodes: 1041623

2) Total number of ways: 94784

This data is consistent with what we had before.

3) Total number of unique users: 1077. This is a significant number!

4) List of top 10 users: ('n76', 334892), ('Adam Geitgey', 158974), ('Sat', 125254), ('woodpeck_fixbot', 90764), ('TheDutchMan13', 26178), ('Zian Choy', 16856), ('Brian@Brea', 15758), ('TieFaith', 12902), ('stevea', 12506), ('evil saltine', 11942)

We see that a single user (the top user) has had more than 300,000 contributions. This user is likely a robot. However, there is no tag in the xml file that distinguishes human from robots. I think it would be very helpful to include a new tag into the xml files to distinguish human input from automated inputs. Such information would motivate many users to contribute to the OSM project. Perhaps some machine learning algorithms can automatically do this using the current set of data.

5) List of top 10 amenities: ('place_of_worship', 915), ('fast_food', 538), ('restaurant', 497), ('school', 299), ('bar', 275), ('cafe', 176), ('fuel', 107), ('bank', 83), ('drinking_water', 73), ('bench', 71)

6) I like “Chipotle Mexican Grill”, so I decided to calculate the total number of Chipotle restaurants in San Diego. There are 26 Chipotle in San Diego. That’s a good news!

- To see if this dataset is actively being updated, I counted the number of inputs that has been made since the beginning of the June, 2017. There has been ~30,000 inputs within the last month. This indicates that the dataset is being updated, that’s also a good news!