**OpenStreetMap Project with MongoDB**

Thomas Woodside

Map Area: Mid San Francisco Peninsula (minlat: 37.3909, minlon: -122.315, maxlat : 37.5606, maxlon: -121.996)

**Problems Encountered in the Map**:

**General Human Misunderstanding:**

There were a few locations where humans clearly misunderstood the way the correct formatting of the data. For example, the key for “shop” was often tied to a value of “yes”. The value was supposed to include the kind of shop instead. Another example could be found in the street names, which occasionally included more information about the address than just the street, such as the house number.

**Locally Unique Street Name Endings**:

When I was cleaning, my expected street names list only included the few in the Udacity example. However, there were many perfectly legitimate street name endings that I would not have expected. These included trail, commons, circle, crescent, plaza, terrance, walk, and way. I also found South, North, East, and West at the end of several street names. This is also perfectly correct. Additionally, as the Spanish originally colonized California, many of the street names are also Spanish. I added “Real” to my expected street name endings to make up for that Spanish street ending. Additionally, I added “Pulgas”, which allowed “Alameda de Las Pulgas” through. Pulgas actually means flees, (Alameda being the real street “ending”) but I included it anyway.

**Abbreviated and Missing Street Name Endings**:

Many of the street names that I cleaned had abbreviated or uncapitalized street endings such as 'St' or 'avenue'. This was easily fixed with a simple Python dictionary. There were also a good number of street names that didn't have a valid street ending (e.g. 'Palm' instead of 'Palm Drive'). These proved very difficult to fix, especially if there were multiple streets with the same base name, only separated by the street ending. I left these as is without cleaning.

**Unexpected Types Of Data:**

Coming into this project I expected all data to be contained in either a node or a way. However, I found 14 'business' headers, 7 'water' headers, 6 'site' headers, 4 'boundaries', and many more (including one 'multipolygon'). Overall this didn't impact my analysis much, as the total number of these was only around 0.004% of the whole.

**Overview of the Data:**

**Size:**

xml original file: 188.2MB

json: 209.5MB

**Number of Unique Users:**

unique\_users = db.OSM.aggregate([

unique\_users = db.OSM.aggregate([

{'$group': {'\_id': '$created.user'}},

{'$group': {'\_id': 'null', 'count': {'$sum': 1}}}

])

{'count': 838, '\_id': 'null'}

838 unique users

**Number of Nodes and Ways:**

nodesways = db.OSM.aggregate([

{'$match': {'type': {'$exists': **True**}}},

{'$group': {'\_id': '$type', 'count': {'$sum': 1}}},

{'$sort': {'count': -1}}

])

{'\_id': 'node', 'count': 828679}

{'\_id': 'way', 'count': 95130}

{'\_id': 'business', 'count': 14}

{'\_id': 'water', 'count': 7}

{'\_id': 'site', 'count': 6}

{'\_id': 'boundary', 'count': 4}

{'\_id': 'Public', 'count': 2}

{'\_id': 'multipolygon', 'count': 1}

{'\_id': 'Private', 'count': 1}

{'\_id': 'bocce', 'count': 1}

{'\_id': 'multi-storey', 'count': 1}

**Most Commonly Occurring “Leisure” nodes:**

leisure = db.OSM.aggregate([

{'$match': {'leisure': {'$exists': True}}},

{'$group': {'\_id': '$leisure', 'count': {'$sum': 1}}},

{'$sort': {'count': -1}},

{'$limit': 3}

])

{'count': 399, '\_id': 'pitch'}

{'count': 280, '\_id': 'park'}

{'count': 163, '\_id': 'swimming\_pool'}

**Additional Ideas:**

Clearly, judging from the inconsistencies that I found, many of the fields are confusing as to what is supposed to go in them. An example of this is “shop”, which was often interpreted as having a “yes” or “no” value. It would be better if it were called “type\_shop” to show the real purpose. This more descriptive naming scheme also applies widely to other fields.

Additionally, formatting and intent of the data should be standardized across the board, such as a standard abbreviation practice. On the beginners’ guide on the OpenStreetMap wiki, nothing is said about the specific fields that could be used or standardization of abbreviations. Another page on the Wiki, “Good Practices,” states that abbreviations should not be used, but again no mention is made of specific fields.

It seems there is no standard set of fields that can be used in OpenStreetMap. Putting a standard set of fields on the Wiki with accepted values that should go in these fields would go a long way to improve the data.

There are, however, some problems with this approach. First, standard fields would have to be agreed upon, and as the uses for fields seem to vary widely this could be contentious. Second, there might be too many fields for any person to read all of them. Third, a massive effort would have to be undertaken to bring all of the existing data up to par. Fourth, even if standards were adopted, they might not be adhered to, as we saw with the no abbreviations practice.

Overall, the main issue with OpenStreetMap is a lack of standardization.