Wrangle OpenStreetMap Data

W. Alexander Jenkins

Map Area: Savannah, Georgia, United States

https://www.openstreetmap.org/relation/119867
https://mapzen.com/data/metro-extracts/#savannah-georgia

Problems Encountered in the Map

After downloading the data for the Savannah area, I noticed a few challenges during my audit of the data. Some of these problems would cause issues when importing to MongoDB; others were inconsistencies among the data. The following problems were encountered, and a discussion of the cleaning process follows.

- Over-abbreviated street names and types, e.g. Sunbriar Ln
- House numbers as ranges or multiple numbers, e.g. 114-122
- Inconsistent phone number formats (“+1 (xxx) xxx-xxxx”,”+1-xxx-xxx-xxx”,”xxx-xxx-xxxx”)
- Duplicate features and feature names (“phone” vs. ”contact:phone”, “email” vs. “contact:email”)

Over-abbreviated Street Names

A search of the OSM XML data file revealed some inconsistencies among street names. Street types (street, boulevard, avenue, etc.) were abbreviated in some street names and were not in others. I updated all address strings to use the full street type where it appeared, such that “Ford Ave” becomes “Ford Avenue.”

House Numbers

After finding “Ford Avenue”, I noticed another issue with the house numbers. That entry’s house number field included the full street address. Upon searching for other instances of this problem, I found some house numbers appeared as ranges or lists of numbers. These instances represented buildings spanning across a block. All house numbers were transformed into tuples

of two numbers: the first, the minimum of the range and the second if it exists, the maximum of the range.

Phone Numbers

Some queries on the database in MongoDB revealed different formatting for phone numbers. Phone numbers may or may not have appeared with country prefixes, parentheses separating area codes, or dashes separating digits. All phone numbers have been formatted as “xxx xxx xxxx”; this will allow easier manipulation and/or conversion to the notation of the E.123 recommendations set forth by the International Telecommunications Union.

Duplicate Features

A problem found during the data audit is that some features are tagged by different names. For example, “contact:phone” would appear instead of “phone”, and “phone” is the annotation most commonly used for this feature. The data for the MongoDB insert is modified to use the common feature name “phone” and similarly “email.”

Also some documents showed the address being duplicated by data from a previous import of TIGER data (TIGER is the Topologically Integrated Geographic Encoding and Referencing System). The TIGER data separated street names and street types into individual fields, and abbreviations were used for street types. According to the OpenStreetMap project, there is a huge effort to fix the data from the TIGER imports. This data may not be reliable or outdated so it will not be included. There’s no need to bloat the database with questionable data.

There are two instances where the feature name is “type”; during the formatting of the dataset, this value would overwrite the type of document. The features will be ignored.

Find documents whose type is not “node” or “way”

```
> db.osm.find({"type":{'$nin':["node","way"]}})
```

Here are the two documents before the correction.

```
[{u'_id': ObjectId('56c8dceb0be4445fec07edfd'),
 u'address': {u'state': u'GA'},
 u'aeroway': u'aerodrome',
 u'closest_town': u'Savannah, Georgia',
 u'created': {u'changeset': u'36411636',
              u'timestamp': u'2016-01-06T19:43:02Z',
              u'uid': u'145231',
              u'user': u'woodpeck_repair',
              u'version': u'5'},
 u'ele': u'10',
 u'gnis': {u'county_name': u'Chatham',
           u'created': u'11/17/2008',
           u'feature_id': u'2512243',
           u'feature_type': u'Military'},
 u'iata': u'SVN',
 u'icao': u'KSVN',
 u'id': u'368997924',
```

```
 u'landuse': u'military',
 u'name': u'Hunter Army Airfield',
 u'operator': u'U.S. Army',
 u'pos': [32.0119068, -81.1426901],
 u'source': u'wikipedia',
 u'type': u'military',
 u'wikipedia': u'en:Hunter Army Airfield'},
{u'_id': ObjectId('56c8dcf00be4445fec092c05'),
 u'aeroway': u'navigationaid',
 u'created': {u'changeset': u'31492381',
              u'timestamp': u'2015-05-27T03:02:13Z',
              u'uid': u'1962916',
              u'user': u'hokieengr',
              u'version': u'2'},
 u'frequency': u'115.95 Mhz',
 u'id': u'1312079413',
 u'name': u'Savannah VOR',
 u'pos': [32.1462579, -81.1991353],
 u'type': u'vortac'}]
```

Data Overview

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

File sizes

```
savannah_georgia.osm ........... 70.9 MB
savannah_georgia.osm.json ...... 79.1 MB
```

Number of documents

```
> db.osm.find().count()
365568
```

Number of nodes

```
> db.osm.find({"type":"node"}).count()
333365
```

Number of ways

```
> db.osm.find({"type":"way"}).count()
32203
```

Number of unique users

```
> len(db.osm.distinct("created.user"))
299
```

Top 1 contributing user

```
> db.osm.aggregate([{"$group":{"_id":"$created.user",
"count":{"$sum":1}}},{"$sort":{"count":-1}},{"$limit":1}])
{u'_id': u'hokieengr', u'count': 116189}
```

Number of documents possibly created by bots

```
> db.osm.aggregate([{'$match':{"created.user":{'$regex':'bot'}}},
{'$group':{"_id":"$created.user","count":{'$sum':1}}},{'$sort':{"
count":-1}}]
[{u'count': 32990, u'_id': u'woodpeck_fixbot'}, {u'count': 3764,
u'_id': u'bot-mode'}, {u'count': 6, u'_id': u'xybot'}]
```

Total

```
> db.osm.aggregate([{'$match':{"created.user":{'$regex':'bot'}}},
{'$group':{"_id":"$created.user","count":{'$sum':1}}},{'$group':{
"_id":"Total Bot Docs","total":{'$sum':"$count"}}}]
[{u'total': 36760, u'_id': u'Total Bot Docs'}]
```

Number of different amenities

```
> len(db.osm.distinct("amenity"))
48
```

Ten most occurring amenities

```
> db.osm.aggregate([{"$match": {"amenity":{"$exists":True}}},
{"$group":{"_id":"$amenity","count":{"$sum":1}}},{"$sort":{"count
":-1}},{"$limit":10}])
[{u'count': 351, u'_id': u'place_of_worship'}, {u'count': 173,
u'_id': u'parking'}, {u'count': 155, u'_id': u'school'},
{u'count': 74, u'_id': u'restaurant'}, {u'count': 49, u'_id':
u'grave_yard'}, {u'count': 46, u'_id': u'bench'}, {u'count': 27,
u'_id': u'fast_food'}, {u'count': 18, u'_id': u'library'},
{u'count': 17, u'_id': u'hospital'}, {u'count': 16, u'_id':
u'fuel'}]
```

Additional Ideas

Additional Data Modeling

Further data modeling is needed in order the make this dataset more useful. Many of the relationships among features are lost due to the almost flat hierarchy. For example, “service” is

a common feature tag with many different values, and depending on the instance, its usage can mean different things. Here are just some of the usages of “service.”

```
> db.osm.aggregate([{"$match": {"service":{"$exists":True}}},
{"$group":{"_id":"$service","count":{"$sum":1}}},
{"$sort":{"count":-1}}])
[{u'count': 822, u'_id': u'parking_aisle'},
{u'count': 261, u'_id': u'spur'},
{u'count': 193, u'_id': u'driveway'},...
```

The value “parking_aisle” relates to the parking amenity. “Spur” refers to the railway service. And the value “driveway” is used to describe roads and footpaths. If one does not know what each value means or relates to, the feature’s relationship with other features can be lost. This feature’s importance to other features can be disvalued if the meanings and relationships of its values are unknown.

These related features could be nested inside dictionary items that themselves are data values of encompassing features. For example, after remodeling the following

```
{
u'_id': ObjectId('56c907a40be4445fec11b342'),
u'created':  {
             u'changeset': u'14014847',
             u'timestamp': u'2012-11-24T16:17:39Z',
             u'uid': u'57884',
             u'user': u'EdLoach',
             u'version': u'2'
             },
u'highway': u'service',
u'id': u'9079389',
u'node_refs': [u'67023613', u'67023614'],
u'service': u'parking_aisle',
u'type': u'way'
}
```

could become

```
{
u'_id': ObjectId('56c907a40be4445fec11b342'),
u'created': {
             u'changeset': u'14014847',
             u'timestamp': u'2012-11-24T16:17:39Z',
             u'uid': u'57884',
             u'user': u'EdLoach',
             u'version': u'2'
             },
u'highway': {u'service': u'parking_aisle'},
u'id': u'9079389',
u'node_refs': [u'67023613', u'67023614'],
u'type': u'way'
}
```

Modeling this data for ease of use and understanding can be difficult. Encompassing features inside other features without losing value would be the biggest challenge. Steps to differentiate features, their values, and their relations would need to be outlined.

Conclusion

The Savannah, Georgia dataset is not complete. Although this data has been cleaned for this exercise, it still may not be suitable for other usage. The validity of the data was questionable. The OpenStreetMap datasets are living documents and as such are subject to change. With enough forethought into the cleaning and modeling of this data, valid data could be input to OpenStreetMap to help in its efforts to fix the data.