OpenStreetMap Project - Data Wrangling with MongoDB

Introduction

Downloading data for the entire San Francisco Bay area was a pretty large undertaking. The data file itself was 850MB uncompressed. 4.4 million records.

I tried to be creative and, in addition to standardizing street names, I added phone number and postal code standardization logic to the file conversion process.

I also decided to try to handle colon fields by creating hierarchical dictionaries similar to how addresses were handled in the Ch 6 exercises. I thought being flexible and supporting all colon-separated fields would be easy… but due to the sheer amount of data, I had many problems clashing with non-colon fields with similar names. (See problems below)

In the end, through some renaming logic in my scripts, I was able to load every record into the MongoDB database and run queries against it.

So: How good is the San Francisco Bay Area OpenStreetMap dataset?

Problems Encountered in the Map

Before loading the data into MongoDB, I had to convert the XML data into a JSON file. While writing the conversion scripts, I noticed some peculiarities with the OpenStreetMap data:

Contact Information is stored inconsistently:

OpenStreetMap supports a **contact:** key for storing contacts:

http://wiki.openstreetmap.org/wiki/Key:contact

Unfortunately many contributors ignore this key. Information is stored in different formats:

- phone vs. contact:phone
- email vs. contact:email
- fax vs. contact:fax

I prefer grouping this information together so I went ahead and applied the contact: key to all phone numbers, fax numbers, and email addresses

---

Phone number formats are not consistent

Phone numbers in the United States consist of a three digit area code and seven digit number. Unfortunately even though all US phone numbers are the same number of digits, there are various ways to write them.

From the OpenStreetMap data:

+1 510 5491085
+1 415 682 7789
650-685-5821
650.343.2049
(650) 347-1023

The XML cleansing script uses a regular expression to extract a 10-digit phone number from these formats. The phone numbers are then rendered in the following format: (area code) <prefix>-<suffix>.

e.g. (123) 456-7890

Records with phone numbers which could not be processed are registered in an error file (with source IDs) for later reference. I noticed several reasons for these types of errors:

1) Letters in the phone number: '415-397-BROS'
2) Corrupt phone numbers '+49'
3) Missing area code: '885-2222'
4) Incomplete phone numbers: '415 242 960'
5) URL in the phone number field: 'http://www.pastapastaco.com/'
6) Unrelated information in phone number field: 'B Street & Vine', 'yes', 'fire'

These errors occurred quite infrequently. Of the 2,665 OpenStreetMap records with phone numbers, only only 16 phone numbers were unable to be standardized. These were loaded into MongoDB unchanged.

Street type standardization is difficult due to many factors

Unlike phone numbers and postal codes, street names are varied and unpredictable. Most street names end in familiar street types (Avenue, Lane, Road, Court, etc.). When attempting to standardize these street names (and replace abbreviations with the full street type), I noticed many problems with the OpenStreetMap data for the San Francisco Bay Area. These problems fell into the following categories:

1) Incomplete street names

‘24th’ instead of ‘24th Street’
‘9th’ instead of ‘9th Street’
‘Brannan’ instead of ‘Brannan Street’
‘Van Ness’ instead of ‘Van Ness Avenue’

2) Misspelled street types:

‘Garvin **Avenie**’ instead of ‘Garvin Avenue’
‘Columbus **Abenue**’ instead of ‘Columbus Avenue’

3) Non-standard street names:

‘El Camino Real’ is the Spanish name of a famous, historical road in the SF Bay Area. Many business are on this address.
‘Broadway’ includes the street type in its name.
‘Alameda de las Plugas’, another Spanish name
Translated means ‘Avenue of the Fleas’
(I actually learned this during this exercise!)
This road also has an alias: “The Alameda”, which is also non-standard and frequently used in the OSM data.

4) Unrelated data:
‘Multi Use Building’
Weird unicode characters including longitude and latitude in the street field

After filtering out the non-standard street names, there were still 849 standardization ‘errors’ in the errors file.

grep Address san-francisco_california_errors.txt | wc

Of these, 384 errors were likely due to misspelling and aliases of “Alameda De Las Plugas”.

grep Address san-francisco_california_errors.txt | grep Alameda | wc

We could add logic to our conversion script to process these records and increase the consistency of our data.

Clashes between colon-separated keys and standard keys.

Some records contain tags where one colon-separated tag's prefix is also used independently. The following XML has 4 similarly named tags:

```
    <node changeset="18720543" id="1241239103" lat="37.7666991" lon="-122.4303369"
timestamp="2013-11-04T21:46:50Z" uid="119881" user="Clorox" version="8">
        <tag k="name" v="Rolf Klotz watch &amp;amp; clock maker" />
        <tag k="building" v="commercial" />
        <tag k="addr:street" v="Market Street" />
        <tag k="building:levels" v="2" />
        <tag k="addr:housenumber" v="2166" />
        <tag k="building:material" v="wood;stucco;brick" />
        <tag k="building:levels:underground" v="1" />
    </node>
```

Similar to the exercises and their treatment of address fields, my migration scripts try to organize colon-separated key names into parent:child JSON objects. Unfortunately, when the above XML record is parsed, if a key (building) exists with the same name as the prefix (**building**:levels), a name collision when I try to create a dictionary variable with the same key as an existing field.

'building' and 'levels' cannot be both strings and dictionaries. They have to be one or the other.

The JSON file creation script gets around this by adding an underscore to the colon prefix fields when creating the respective dictionaries.

Overview of the Data

The data for the San Francisco Bay Area is pretty huge.

The OSM XML file from MapZen is 154.7MB zipped and almost 850MB unzipped.
The JSON file generated from my conversion script was 1.2GB.

There are 4,412,192 records in the target Mongo DB. These represent nodes and ways in the OSM file:

```
> db.osm.stats()
{
        "ns" : "users.osm",
        "count" : 4412192,
        "size" : 1042451198,
        "avgObjSize" : 236,
        "storageSize" : 288849920,
        "capped" : false,
```

There are 3.97 million nodes. Most of these represent geographic locations (points) in the bay area with no detailed descriptions.

Here is a query for all node records:

> db.osm.find({type : 'node'}).count()
3971007

These records include a ‘name’ that can be used to identify the location.
> db.osm.find({type : 'node', name : {$exists : true}}).count()
13960

13,960 records out of 3,971,007 records means that only 0.35% of the nodes contain descriptive names which we can query on.

If we limit the nodes to amenities (stores), we get an even smaller set:

> db.osm.find({type : 'node', amenity : {$exists : true}}).count()
10567

There are 441,071 ways. These represent roads, bridges, highways, etc.

> db.osm.find({type : 'way'}).count()
441071

Buildings and commercial enterprises are registered with the ‘amenity’ tag. This tag is supposed to be used for nodes only, but I see many of them in <way> elements as well. Perhaps this is to associate them with geographic locations that are bigger than a single node.

> db.osm.find({type : 'way', amenity : {$exists : true}}).count()
6022

I tried to identify the location breakdown of amenities, but unfortunately most records lack city-level address details.

> db.osm.aggregate([{$match : {amenity : {$exists : true} }}, {$group : {_id : '$address.city', count : {$sum : 1}}}, {$sort : {count : -1}}, {$limit : 10}])
{ "_id" : null, "count" : 14052 }
{ "_id" : "San Francisco", "count" : 1209 }
{ "_id" : "Berkeley", "count" : 359 }
{ "_id" : "Redwood City", "count" : 211 }
{ "_id" : "Oakland", "count" : 196 }
{ "_id" : "Union City", "count" : 75 }
{ "_id" : "Burlingame", "count" : 49 }
{ "_id" : "Alameda", "count" : 46 }
{ "_id" : "Pacifica", "count" : 38 }
{ "_id" : "Richmond", "count" : 33 }

It’s late and I am hungry! Which fast food restaurants are popular in the Bay Area?

```
> db.osm.aggregate([{$match : {amenity : 'fast_food' }}, {$group : {_id : '$name', count : {$sum : 1}}}, {$sort : {count : -1}}, {$limit : 15} ])
{ "_id" : "McDonald's", "count" : 57 }
{ "_id" : "Subway", "count" : 47 }
{ "_id" : "Burger King", "count" : 32 }
{ "_id" : "Taco Bell", "count" : 26 }
{ "_id" : "Jamba Juice", "count" : 17 }
{ "_id" : "Jack in the Box", "count" : 15 }
{ "_id" : "Wendy's", "count" : 14 }
{ "_id" : null, "count" : 9 }
{ "_id" : "KFC", "count" : 8 }
{ "_id" : "Togo's", "count" : 8 }
{ "_id" : "Noah's Bagels", "count" : 6 }
{ "_id" : "Panda Express", "count" : 6 }
{ "_id" : "Chipotle", "count" : 6 }
{ "_id" : "In-N-Out Burger", "count" : 6 }
{ "_id" : "Carl's Jr.", "count" : 5 }
```

Are there only 6 In-N-Out burgers in the bay area? That does not sound right. Maybe some are not classified as ‘fast food’?

Removing the amenity filter and usi a regular expression to search by name, I see some more (10 total) records:

```
> db.osm.find({'name' : /.*In.*N.*Out/}, {_id : 0, name : 1, amenity : 1})
{ "amenity" : "fast_food", "name" : "In-N-Out Burger" }
{ "amenity" : "fast_food", "name" : "In-N-Out Burger" }
{ "amenity" : "restaurant", "name" : "In N Out" }
{ "amenity" : "fast_food", "name" : "In-N-Out Burger" }
{ "amenity" : "fast_food", "name" : "In-N-Out Burger" }
{ "amenity" : "fast_food", "name" : "In-N-Out Burger" }
{ "amenity" : "fast_food", "name" : "In N Out Burger" }
{ "amenity" : "fast_food", "name" : "In-N-Out Burger" }
{ "amenity" : "fast_food", "name" : "In-N-Out" }
{ "amenity" : "restaurant", "name" : "In-N-Out Burger" }
```

The amenity tag is being inconsistently applied (‘restaurant’? Really?). The chain name is not consistent (In-N-Out Burger, In N Out, In-N-Out) either.

What kinds of cuisine are available in the bay area?

```
> db.osm.aggregate([{$match : {amenity : 'restaurant' }}, {$group : {_id : '$cuisine', count : {$sum : 1}}}, {$sort : {count : -1}}, {$limit : 10} ])
{ "_id" : null, "count" : 854 }
{ "_id" : "mexican", "count" : 199 }
{ "_id" : "pizza", "count" : 152 }
```

{ "_id" : "chinese", "count" : 126 }
{ "_id" : "italian", "count" : 112 }
{ "_id" : "japanese", "count" : 109 }
{ "_id" : "thai", "count" : 96 }
{ "_id" : "american", "count" : 92 }
{ "_id" : "vietnamese", "count" : 65 }
{ "_id" : "burger", "count" : 62 }

It seems almost 1/3 of the restaurants (854 / 2480) do not have the 'cuisine' field set. This makes it hard to make a confident statement, but from the data we have, it seems that Mexican restaurants are the most popular. There are also many Asian restaurants. Chinese, Japanese, Thai, and Vietnamese restaurants are all in the top 10.

Sightseeing - Golden Gate Bridge

I also checked the Golden Gate Bridge to see how it is represented in Open Street Map. The Golden Gate Bridge is interesting because it is part of a highway (US101), but there is a walking path on one side and a bicycle path on the other.

> db.osm.find({'name' : "Golden Gate Bridge"}, {_id : 0, name : 1, type : 1, foot : 1, bicycle : 1, highway : 1})
{ "bicycle" : "no", "name" : "Golden Gate Bridge", "type" : "way", "highway" : "motorway" }
{ "bicycle" : "no", "name" : "Golden Gate Bridge", "type" : "way", "highway" : "motorway" }
{ "name" : "Golden Gate Bridge", "foot" : "designated", "bicycle" : "designated", "type" : "way", ***"highway" : "footway"*** *}*
{ "name" : "Golden Gate Bridge", "foot" : "no", "bicycle" : "yes", "type" : "way", ***"highway" : "cycleway"*** *}*
{ "bicycle" : "no", "highway" : "motorway", "foot" : "no", "name" : "Golden Gate Bridge", "type" : "way" }

The walking path, bicycle path, and road are all registered as separate 'way' records. The 'highway' tag can be used to differentiate these records, although I also see multiple roads (highway = 'motorway') which likely represent different motor vehicle-only sections of the bridge.

Other Ideas about DataSets

The OpenStreetMap data is a really exciting project that provides a public data source (and forums) for mapping locations and creating metadata to represent businesses and homes and addresses.

Perhaps due to its reliance on voluntary contributions from the public, there is a lot of room for improvement. Data is inconsistently entered, some address / phone number fields are incorrect, and generally the data is sparse.

I can think of a couple of ways to improve this data:

1) Ask Google, Apple, Yelp, Redfin, and other providers of location-based services if they can share all or even some of their data via one-way feeds. Working with an open source project would be great for public relations for a company. There might be benefit gained by having outside eyes checking data for errors and raising concerns where needed.
2) Work with education providers (high school and university computer science programs, online education providers… :)) to incorporate OpenStreetMap into curriculums.
3) Organize Meetup groups in cities to raise awareness of this service. Contributions from the public (including non-engineers through the website) are welcome and will improve the data. Organizing meetings in restaurants or cafes would be a way to keep these sessions low cost as well as providing opportunities for people to network with each other.
4) The tags are also inconsistent (see the “phone” vs. “contact:phone” issue). Even worse, while checking out neighborhoods in the city, I had problems locating famous San Francisco neighborhoods.

> db.osm.find({place : 'neighborhood', "gnis_.County" : 'San Francisco'}, {_id : 0, name : 1})
>

Hmmm, this is strange, I could have sworn that this tag exists….

From http://wiki.openstreetmap.org/wiki/Neighbourhood

Current state of mapping

There is now a tag which has been approved via the wiki proposal process:

- place=neighbourhood

Other previous and related tags include:

- place=suburb or place=hamlet - these were not intended for use with neighbourhoods
- boundary=neighborhood - used 154 times as of 2011/04/14, though the vast majority were last edited by one user, NE2, in Orlando, FL, USA
- place=subdivision - used 638 times as of 2011/04/14, though the vast majority were last edited by one user in the Philippines, though also used in Cincinnati, OH, USA

Wait… 2 different spellings? I can understand (and complain ;)) about the British spelling being used in one instance, but the fact that the American spelling is also used elsewhere (and on the same wiki page) is not ideal. I think the OpenStreetMap project should standardize on one spelling.

> db.osm.find({place : 'neighbourhood', "gnis_.County" : 'San Francisco'}, {_id : 0, name : 1})
{ "name" : "Seacliff" }
{ "name" : "Richmond District" }
{ "name" : "Marina District" }
{ "name" : "Crocker-Amazon" }
{ "name" : "Haight-Ashbury" }
{ "name" : "Presidio Terrace" }
{ "name" : "Bayview District" }
{ "name" : "Western Addition" }
{ "name" : "Potrero Hill" }
{ "name" : "Ingleside" }
{ "name" : "Japantown" }
{ "name" : "Saint Francis Wood" }
{ "name" : "Mission District" }
{ "name" : "North Beach" }
{ "name" : "Forest Hill" }
{ "name" : "Jordan Park" }
{ "name" : "Sunset District" }
{ "name" : "Pacific Heights" }

References

Phone Number Regular Expression
http://www.diveintopython.net/regular_expressions/phone_numbers.html

San Francisco Bay Area OpenStreetMap data provided by MapZen:
https://mapzen.com/data/metro-extracts

OpenStreetMap Documentation
http://wiki.openstreetmap.org/wiki/Key:contact