OpenStreetMap Project
Data Wrangling with MongoDB

Joohun Lee

Map Area: San Jose, CA, United State
http://www.openstreetmap.org/relation/112143

1. Problems Encountered in the Map

- Over-abbreviated street names (e.g. 353 W Main Ave -> 353 W Main Avenue)
- "address" is already used as one of the keys which is redundant to the key as a dictionary that includes "street" and "postcode" and "housenumber" etc.
- Include "incorrect" postal codes (San Jose zip codes begin with "951", but it includes neighbor county's zip codes)
- Postal code information is included in the full "address" key.

Over-abbreviated Street names

As already practiced in the example of "audit.py", I've noticed that the street names are overly abbreviated. I queried abbreviated strings of all streets and found out followings could be replaced with full street name as below table.

```
mapping = { "St": "Street",
            "St.": "Street",
            "Ave": "Avenue",
            "ave": "Avenue",
            "Rd.": "Road",
            "Rd" : "Road",
            "Dr" : "Drive",
            "Ct" : "Court",
            "Ln" : "Lane",
            "Sq" : "Square",
            "Blvd" : "Boulevard"
            }
```

"address" is already used as one of the keys

As a part of the requirement for the last task for "data.py", any tags begin with "addr" should be wrapped by the "address" dictionary. However, Some of the data, "address" is already used as a key that contains value for "entire address". Hence there is a conflict with this key "address" and new generated key "address". So I have to change the existing "address" to "full address" to generate new key as "address".

e.g)

```
<node id="3061406142" lat="37.3707738" lon="-122.0320404" version="1"
timestamp="2014-09-05T19:02:43Z" changeset="25254734" uid="816433"
user="f2003104">
      <tag k="name" v="Sunnyvale Optometric Center"/>
      <tag k="address" v="510 S Murphy Avenue, Sunnyvale, CA 94086"/>
</node>
```

Incorrect postal codes

I found that query result for San Jose also includes postal codes from neighbors' cities. San Jose zip codes should begin with "951" but some of postal codes turned out to belong to "Mountain view", "Sunnyvale", "Campbell".

For accurate analysis, I had to screen all postal codes which doesn't begin with "951" .

Postal code information is included in "address" key.

There are cases where no "postal code" information was found but after converting to json format, postal code was found from "full address". To weed out all data not belong to San Jose area, I had to parse full address to capture postal code and eliminate this data if not belong to San Jose Zip code.

e.g.)

{"name": "Sunnyvale Optometric Center", "created": {"changeset": "25254734", "user": "f2003104", "version": "1", "uid": "816433", "timestamp": "2014-09-05T19:02:43Z"}, "pos": [37.3707738, -122.0320404], **"full address": "510 S Murphy Avenue, Sunnyvale, CA 94086**", "type": "node", "id": "3061406142"}

2. Data Overview

File Sizes

San-jose_california.osm - 222MB

San-jose_california.osm.json - 248MB

 # Number of documents

```
> db.sanjose_california.find().count()
1121378
```

 # Number of nodes

```
> db.sanjose_california.find({"type":"node"}).count()
1004266
```

 # Number of ways

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

117077

 # Number of unique users

```
> db.sanjose_california.distinct("created.user").length
997
```

3. Additional Ideas

Additional data exploration using MongoDB queries

#Top 10 amenities

```
> db.sanjose_california.aggregate([{"$match":{"amenity":{"$exists":1}}},
{"$group":{"_id":"$amenity", "count":{"$sum":1}}}, {"$sort":{"count":-1}},
{"$limit":10}])
{ "_id" : "parking", "count" : 1597 }
{ "_id" : "restaurant", "count" : 724 }
{ "_id" : "school", "count" : 524 }
{ "_id" : "fast_food", "count" : 393 }
{ "_id" : "place_of_worship", "count" : 329 }
{ "_id" : "fuel", "count" : 202 }
{ "_id" : "cafe", "count" : 194 }
{ "_id" : "bicycle_parking", "count" : 166 }
{ "_id" : "bench", "count" : 163 }
{ "_id" : "toilets", "count" : 160 }
```

 # Top 5 restaurants

```
> db.sanjose_california.aggregate([{"$match":{"amenity":{"$exists":1},
...           "amenity":"restaurant"}},
...  {"$group":{"_id":"$cuisine",
...           "count":{"$sum":1}}},
...  {"$sort":{"count":-1}},
...  {"$limit":5}])
{ "_id" : null, "count" : 217 }  : 51.78%
{ "_id" : "mexican", "count" : 69 }
{ "_id" : "vietnamese", "count" : 50 }
{ "_id" : "chinese", "count" : 48 }
{ "_id" : "pizza", "count" : 45 }
```

#Top 5 fast food restaurant

```
> db.sanjose_california.aggregate([{"$match":{"amenity":{"$exists":1},
"amenity":"fast_food"}}, {"$group":{"_id":"$cuisine",
"count":{"$sum":1}}},   {"$sort":{"count":-1}},   {"$limit":5}])
{ "_id" : null, "count" : 98 } : 33.79%
{ "_id" : "burger", "count" : 70 }
```

{ "_id" : "sandwich", "count" : 63 }
{ "_id" : "mexican", "count" : 36 }
{ "_id" : "pizza", "count" : 23 }

From query result of top 10 amenities, I found the categories that could potentially be inseparable in terms of types of cuisine, for example, "restaurant" vs. "fast_food". Both categories could include the same types of cuisines that could change the result of analysis.
For instance, If I want to find out top five cuisine, querying with "restaurants" returns "Mexican(69)" -> "Vietnamese(50)" ->"Chinese(48)" -> "Pizza(45)" from top . But "Mexican" and "Pizza" are also included in "fast_food" amenity. Hence in order to get the accurate top 5 cuisine, I had to combine restaurant with fast_food.
By this approach, the top 5 result get changed in the order of "Mexican(105)",->"burger(70)" -> pizza(68)" -> "sandwitch(63)" -> "Vietnamese(50)".
However, this analysis is conducted with "Null" data 51.78% from "restaurant" and 33.79% from "fast_food" . For more accurate analysis, it'd be necessary to fill the correct type of cuisine and decrease the number of NULL" data. One of my idea is to capture key word from the name of restaurant and guess the types of cuisine base on the key word. For example, if "Pho" captured from the name, we can assume it is Vietnamese restaurant as below.
e.g)
{"amenity": "restaurant", "name": "Pho Hoa Beef Noodle", "created": {"changeset": "34942469", "user": "Eureka gold", "version": "3", "uid": "532783", "timestamp": "2015-10-29T07:30:23Z"}, "brand": "Pho Hoa", "pos": [37.306556, -122.032767], "type": "node", "id": "2099014851"}

 # Top 5 religions

```
> db.sanjose_california.aggregate([{"$match":{"amenity":{"$exists":1},
... "amenity":"place_of_worship"}},
... {"$group":{"_id":"$religion", "count":{"$sum":1}}},
...
... {"$sort":{"count":-1}}, {"$limit":5}])
{ "_id" : "christian", "count" : 288 }
{ "_id" : null, "count" : 20 }
{ "_id" : "jewish", "count" : 4 }
{ "_id" : "muslim", "count" : 3 }
{ "_id" : "buddhist", "count" : 3 }
```

From the query of top 5 religion, Christian is most popular religion in San Jose area with no doubt as the gap between Christian and the rest are too big.

#Top 10 building types

```
> db.sanjose_california.aggregate([{"$match":{"building":{"$exists":1}}},
{"$group":{"_id":"$building", "count":{"$sum":1}}}, {"$sort":{"count":-1}},
{"$limit":10}])
{ "_id" : "yes", "count" : 45524 }
{ "_id" : "house", "count" : 5079 }
{ "_id" : "residential", "count" : 3896 }
{ "_id" : "apartments", "count" : 367 }
{ "_id" : "roof", "count" : 317 }
{ "_id" : "school", "count" : 254 }
{ "_id" : "commercial", "count" : 166 }
{ "_id" : "office", "count" : 157 }
{ "_id" : "retail", "count" : 105 }
{ "_id" : "garage", "count" : 98 }
```

: 81.34%

San Jose is densely populated area as most of IT companies are located in bay area. I was trying to find out the quantity of the residential structures(such as house/apartment/townhouse/ etc) and hoped to come out with the result about ratio for population vs. residential structure available. however, 81.34% of data filled with the building type as "yes". Not enough information was available on how it ended up filling with this data. Reducing this "garbage " data is the challenge for future in order to conduct accurate analysis.

e.g.)

{"building": "yes", "created": {"changeset": "22116825", "user": "YongYang", "version": "2", "uid": "1408707", "timestamp": "2014-05-04T01:22:10Z"}, "pos": [37.3199555, -121.991699], "type": "node", "id": "2834928599"}

Contributors of major garbage data and suggestion

As conducted in the sample project, querying top 10 contributors result in the data shows 71% of data were contributed by top 10 creators.

```
> db.sanjose_california.aggregate([{"$group":{"_id":"$created.user", "count":
{"$sum":1}}},{"$sort":{"count":-1}}, {"$limit":10}])
{ "_id" : "nmixter", "count" : 274705 }
{ "_id" : "mk408", "count" : 167103 }
{ "_id" : "Bike Mapper", "count" : 81463 }
{ "_id" : "n76_cupertino_import", "count" : 68114 }
{ "_id" : "n76", "count" : 59762 }
{ "_id" : "erjiang_imports", "count" : 36402 }
{ "_id" : "matthieun", "count" : 36293 }
{ "_id" : "Minh Nguyen", "count" : 33399 }
{ "_id" : "doug_sfba", "count" : 22279 }
{ "_id" : "woodpeck_fixbot", "count" : 20119 }
```

And again I queried top 10 contributors who provided garbage data for building types as “yes” and queried again the top 10 contributors who provided “NULL” cuisine types in restaurant/fast_food as below.

* Creators in blue belong to top 10 contributors of map data

```
>
db.sanjose_california.aggregate([{"$match":{"building":{"$exists":1},"building":"
yes"}},
... {"$group":{"_id":"$created.user", "count":{"$sum":1}}},
... {"$sort":{"count":-1}}, {"$limit":10}])
{ "_id" : "nmixter", "count" : 27535 }
{ "_id" : "Bike Mapper", "count" : 5008 }
{ "_id" : "n76", "count" : 2431 }
{ "_id" : "n76_cupertino_import", "count" : 2034 }
{ "_id" : "Minh Nguyen", "count" : 1718 }
{ "_id" : "erjiang_imports", "count" : 846 }
{ "_id" : "oldtopos", "count" : 446 }
{ "_id" : "matthieun", "count" : 385 }
{ "_id" : "stevea", "count" : 368 }
{ "_id" : "thibautRe", "count" : 344 }
```

Data indicated that majority of creator who filled building type “yes” turned out they belong to top 10 contributors as above.

```
> db.sanjose_california.aggregate([{"$match":{"amenity":{"$exists":1},
"amenity":"restaurant"}},
... {"$group":{"_id":"$created.user","count":{"$sum":1}}},
... {"$sort":{"count":-1}}, {"$limit":10}])
{ "_id" : "Minh Nguyen", "count" : 237 }
{ "_id" : "n76", "count" : 64 }
{ "_id" : "nmixter", "count" : 52 }
{ "_id" : "njaard", "count" : 37 }
{ "_id" : "Bike Mapper", "count" : 35 }
{ "_id" : "oldtopos", "count" : 33 }
{ "_id" : "Jim Gibson", "count" : 12 }
{ "_id" : "stevea", "count" : 10 }
{ "_id" : "Jonathan ZHAO", "count" : 9 }
{ "_id" : "fredmart", "count" : 9 }

> db.sanjose_california.aggregate([{"$match":{"amenity":{"$exists":1},
"amenity":"fast_food"}}, {"$group":{"_id":"$created.user","count":{"$sum":1}}},
{"$sort":{"count":-1}}, {"$limit":10}])
{ "_id" : "Minh Nguyen", "count" : 142 }
{ "_id" : "nmixter", "count" : 30 }
{ "_id" : "Bike Mapper", "count" : 23 }
{ "_id" : "andrewpmk", "count" : 18 }
```

{ "_id" : "njaard", "count" : 18 }
{ "_id" : "n76", "count" : 18 }
{ "_id" : "SomeoneElse_Revert", "count" : 14 }
{ "_id" : "mk408", "count" : 9 }
{ "_id" : "doug_sfba", "count" : 7 }
{ "_id" : "Jim Gibson", "count" : 6 }
Data above indicated that majority of creators who provided restaurant/fast_food data belong to top 10 contributors again.

Therefore, according to the analysis, the majority of "NULL or garbage" data were created mainly by the top 10 contributors of the San Jose map. In the other word, the more these creators contributed to the map data, the more he/she may create more prone to garbage data. It could be various reasons why data were created in this way. Potential reason I can assume is it could be either these top contributors lack in information for attributes of places that they filled in map or they couldn't fully understand the format how the data should be filled in such way the data will be easily queried and analyzed as useful ones by analysts. My suggestion is in order to prevent further garbage data, release standard formats that should be included by creators as a minimum requirement especially to potential top contributors. And even if the data is already existed or created by other users, allowing other user to review/revise the data should be necessary to reduce the garbage data.

4. Conclusion

There are still too many "Null" data or "Garbage" data by execution of data.py/audit.py. Like said, we can improve this data by assuming the data type with key word through algorithms or increase the participation of users who are familiar to each address and have them to correct the garbage data which will be the next challenge.