OpenStreetMap Sample Project
Data Wrangling with MongoDB
Kosuke Murakami

1. Problems Encountered in the Map

In the process of analyzing map data of San Francisco, I noticed four problems.

- Over-abbreviated street names (St, Blvd, Cres)
- Inconsistent postal codes (CA 94111, 515)
- Post codes indicate outside San Francisco
- NULL values

Over-abbreviated Street Names

Many street name abbreviations are inconsistent in downloaded data, so I updated all data so that all the street name could be consistent such that “Oxford St.” becomes “Oxford Street”.

Postal Codes

Postal codes are inconsistent as well. Some include the character of “CA” in the postal codes or some only have three digits numbers even though basically postal codes are supposed to have five numbers, for example “CA 94111” or “515”. Also, I updated all the data with postal code into legitimate 5 digits’ codes

```
db.map_2.aggregate([{"$match":{"address.postcode":{"$exists":1}}},
{"$group":{"_id":"$address.postcode",
"count":{"$sum":1}}},{"$sort":{"count":-1}},{"$limit":5}])
```

This code enables me to see top five postal codes with highest count. The problem is that two out of five postal codes are outside San Francisco, which are Oakland’s postal codes. Therefore, I sorted city by count in a below code.

db.map_2.aggregate([{"$match":{"address.city":{"$exists":1}}},
{"$group":{"_id":"$address.city",
"count":{"$sum":1}}},{"$sort":{"count":-1}},{"$limit":5}])

A city with highest count is not San Francisco, it is Redwood city. Redwood city has about twice number of San Francisco's count.

Here is result.

{ "_id" : "Redwood City", "count" : 23520 }
{ "_id" : "San Francisco", "count" : 13915 }
{ "_id" : "Berkeley", "count" : 5503 }
{ "_id" : "Piedmont", "count" : 3812 }
{ "_id" : "Palo Alto", "count" : 1643 }

Even though original data is named san_francisco_california.osm, its includes data around San Francisco.

- NULL values
  In the process of parsing data, I sometimes face the problem of NULL value, so I needed to check the existence of its value before parsing it.

  #Python code
  >>res = street_type_re.search(tag.attrib['v'])
  >>if res:

2. Data Overview

san_francisco_california.osm>>>>>>>>>947.1MB
san_francisco_california.osm.json>>>1000.06MB

#Number of documents
>db.map.find().count()

5062041

#Number of nodes
>db.map.find({"type":"node"}).count()
4561127

#Number of ways
> db.map.find({"type":"way"}).count()
500914

#Number of unique users
> db.map.distinct("created.user").length
2160

#Top 1 contributing user
>db.map.aggregate([{"$group":{"_id":"$created.user",
"count":{"$sum":1}}}, {"$sort":{"count":-1}}, {"$limit":1}])

{ "_id" : "ediyes", "count" : 945415 }

Number of users appearing only once (having 1 post)
db.map.aggregate([{"$group":{"_id":"$created.user",
"count":{"$sum":1}}},{"$group":{"_id":"$count",
"num_users":{"$sum":1}}}, {"$sort":{"_id":1}}, {"$limit":1}])

{ "_id" : 1, "num_users" : 483 }
"_id" represents postcount

3. Additional Ideas

Contributor statistics

#Top 5 contributors

db.map_2.aggregate([{"$group":{"_id":"$created.user",
"count":{"$sum":1}}}, {"$sort":{"count":-1}}, {"$limit":5}])

{ "_id" : "ediyes", "count" : 945415 }
{ "_id" : "Luis36995", "count" : 719876 }
{ "_id" : "Rub21", "count" : 411277 }
{ "_id" : "RichRico", "count" : 222962 }
{ "_id" : "calfarome", "count" : 183859 }

- Top user contribution percentage ("ediyes") – 18.67%
- Combined top 2 users' contribution ("ediyes" and "Luis36995") – 32.89%
- Combined Top 10 users' contribution – 62.65%
- Only once appearance – 0.009%

This open street map data is made by multiple contributors.

4. Additional data exploration using MongoDB queries

Top 10 appearing amenities

> db.map.aggregate([{"$match":{"amenity":{"$exists":1}}},
{"$group":{"_id":"$amenity",

"count":{"$sum":1}}}, {"$sort":{"count":1}}, {"$limit":10}])

{ "_id" : "parking", "count" : 3902 }

{ "_id" : "restaurant", "count" : 2566 }

{ "_id" : "school", "count" : 1305 }…

#This data reflects how cars are important in San Francisco

 # Top 10 services

```
>db.map.aggregate([{"$match":{"service":{"$exists":1}}},
{"$group":{"_id":"$service", "count":{"$sum":1}}}, {"$sort":{"count":-
1}}, {"$limit":10}])
```

```
{ "_id" : "parking_aisle", "count" : 7401 }

{ "_id" : "driveway", "count" : 1897 }

{ "_id" : "alley", "count" : 659 }…
```

#As I expected, most services are related to cars

#Generally, there are many data related cars because San Francisco is famous for huge traffic. Therefore, I thought it is more convenient if there are more elaborated data about parking. From my experiences to stay in San Francisco for a few days, finding a place to park a car is really annoying since most good parking spots are occupied. If I were a contributor, I would add new information about parking price or capacity for parking cars.

However, especially parking price could be tentative information. If I would add that information, I need to update information so that I would not provide wrong information.

4. Conclusion

I was surprised that tons of map data were made by many contributors. In the process of parsing data, there were tons of missing and poor data, but through this course I successfully managed that kind of data.