OpenStreetMap Project
Data Wrangling with MongoDB

Map Area: San Jose, CA, United States
https://mapzen.com/data/metro-extracts/

1. Problems Encountered in the Map

After initially downloading a small sample size of the San Jose area and running it against a provisional data.py file, I noticed three main problems with the data, which I will discuss in the following order:

- Over-abbreviated street names(st, Ln, Hwy etc.)
- City name is not unified (San Jose, San José, san Jose, sj, etc)
- State name is not unified(CA, Ca, California)
- Not unified phone number

Over-abbreviated street names

Once match the over-abbreviated at the end the street field, we replace the word apply the following mapping relation:

```
street_mapping = { "St": "Street",
            "St.": "Street",
            "Ave":"Avenue",
            "Rd.":"Road",
            "Winchester" : "Winchester Street",
            'Ln':'Lane',
            'Rd':"Road",
            'ave':'Avenue',
            'Hwy':'Highway',
            'Dr':"Drive",
            'Cir':'Circle',
            'street':'Street'
    }
```

City and state name is not unified

I just unified the city, state apply the following relation:

```
city_mapping = {
u'San José':'San Jose',
'San jose':'San Jose',
'san Jose':'San Jose',
'san jose':'San Jose',
'SUnnyvale':'Sunnyvale',
'Sunnyvale, CA':'Sunnyvale',
'cupertino':'Cupertino',
'Saj Jose':'San Jose',
```

```
'santa clara':'Santa Clara',
'Los Gato':'Los Gatos',
'Los Gatos, CA': 'Los Gatos'
}

state_mapping = {
'Ca':'CA',
'California':'CA',
'ca':'CA'
}
```

Phone format

I use the following python code to get all number and unified the phone.

```
def phone_cleaning(raw_phone):
    phone_string = ''
    phone_type_re.findall(raw_phone)
    for num in phone_type_re.findall(raw_phone):
        phone_string = phone_string + str(num)
        if len(phone_string) == 10:
            phone_string = str('+1')+phone_string
        elif len(phone_string) == 11:
            phone_string = str('+') + phone_string
    return phone_string
```

2. Data Overview

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

File sizes:

```
-rw-r--r--@  1 zhaoslin  110237970   259M Apr 21 20:35 san-jose_california.osm
-rw-r--r--   1 zhaoslin  110237970   304M Apr 26 21:33 sjs.json
```

Number of documents

```
> db.sjs.find().count()
1401066
```

Number of Nodes

```
 > db.sjs.find({'node_type':'node'}).count()
1241702
```

Number of ways

```
> db.sjs.find({'node_type':'way'}).count()
158173
```

Number of unique user

```
> db.sjs.distinct("created.user").length
```

1095

Top 1 contributing user

```
>db.sjs.aggregate([{$group:{_id:"$created.user",user_contribute_count:{$sum:1}}
},{$sort:{user_contribute_count:-1}},{$limit:1}]).pretty()
{ "_id" : "nmixter", "user_contribute_count" : 281626 }
```

3. Additional Ideas

Top 10 appearing amenities

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

how many street in San Jose.

```
>db.sjs.aggregate({$group:{_id:"$address.city",street_count:{$sum:1
}}}).pretty()

{ "_id" : "San Jose", "street_count" : 723 }
{ "_id" : "Sunnyvale", "street_count" : 3387 }
{ "_id" : "santa Clara", "street_count" : 2 }
....
```

Most popular cuisines

```
> db.sjs.aggregate([{$match:{amenity:{$exists:1},
amenity:"restaurant"}},{$group:{_id:"$cuisine",count:{$sum:1}}},{$sort:{count:-
1}},{$limit:10}])
{ "_id" : "mexican", "count" : 76 }
{ "_id" : "chinese", "count" : 62 }
{ "_id" : "pizza", "count" : 52 }
{ "_id" : "vietnamese", "count" : 52 }
{ "_id" : "japanese", "count" : 40 }
{ "_id" : "american", "count" : 34 }
{ "_id" : "indian", "count" : 33 }
{ "_id" : "italian", "count" : 32 }
{ "_id" : "thai", "count" : 24 }
```

The longest way is 9197 ft which is runway

```
> db.sjs.aggregate([{$match:{'node_type':'way','length':{$exists:1}}},{$sort:{"length":-
1}}],{$limit:1})
{ "_id" : ObjectId("571f6eafe3f064018eef7ec6"), "node_refs" : [ "26403564",
"26409038", "26403545" ], "created" : { "changeset" : "18256668", "version" : "5",
"user" : "Charles_Smothers", "timestamp" : "2013-10-08T22:48:04Z", "uid" : "160148" },
"length" : "9197 ft", "width" : "200 ft", "node_type" : "way", "aeroway" : "runway", "ref" :
"14L/32R", "id" : "4338473" }
```

4. Other idea about the datasets

Improving and analyzing the data

I found lot of the street name without any city name. We can use the GPS position to mapping the city field. The are several ways to mapping the city. For the first one, we could add several cube block of position for certain city. If the half the the address point fill into the cube block. We can modify the city field in address. There is also second way to fix that is use the API which provide by google map could give us the city of the location.

Discussion about the benefits as well as some anticipated problems

There are more dimensions of the data, the are more information we can inspect. Which mean that we can get more information from that data. But on the other side, we don't want the use mess the data. There are several ways to solve this problem.
For the UI side, we recommend use select the data instead of input the data. Such as we can allow use to select the city from the dropdown menu.
For the data generating, we could use the information generated from device as much as possible. Such as GPS location, device ID etc.

Conclusion

After this review of the data it's obvious that the San Jose area is incomplete, lots of street without have city name and the value is null. The people in San Jose is very like Mexican and Chinese food. May be there are a lot of Mexican and Chinese in this city.