Data Wrangling with MongoDB
P3 Final Project

Joshua Johnson

Map Area: Denver, CO, USA

https://www.openstreetmap.org/relation/253750
https://s3.amazonaws.com/metro-extracts.mapzen.com/denver-boulder_colorado.osm.bz2

1. Problems Encountered in Map Data

The Denver map OSM data was downloaded from the mapzen link above and a sample of the data was taken in order to quickly audit some of the fields. Two main parameters were audited on the sample OSM file - Street Names and Postcodes. Both of these parameters were cleaned programmatically.

Street Names

Not only were many of the street names abbreviated but many were misspelled and mis-capitalized including some well known local streets like “Baseline” spelled as “Baselin” and “Trail” spelled as “trail”. Other common abbreviations such as “Rd”, “St.”, and “Dr” were corrected to ensure clarity.

PostCodes

With nearly 100 postcodes in the map area, a python script was used to list out all of the postcodes found in the data field “addr: postcode”. Inconsistencies were found such as beginning with the state abbreviation, “CO 80439”, along with 4-digit extensions such as “80214-1807”. These outliers were then cleaned in order to maintain the 5-digit format. Both the street names and postcodes were cleaned in the data preparation python script which reformatted the data for upload to MongoDB by translating it into a JSON file.

One surprising find after importing the data into MongoDB was comparing the most common postcodes with the most common city names. The MongoDB queries are as follows:

```
> show dbs
local  0.000GB
test   0.498GB
> use test
switched to db test
> show collections
denver
names
> coll = db.denver

> coll.aggregate([{"$match": {"address.city": {"$exists":1}}}, {"$group": {"_id":
"$address.postcode", "count":{"$sum":1}}}, {"$sort":{"count":-1}},{"$limit":10}])
{ "_id" : "80026", "count" : 7246 }
{ "_id" : "80211", "count" : 6108 }
{ "_id" : "80205", "count" : 5110 }
{ "_id" : "80204", "count" : 5038 }
{ "_id" : "80203", "count" : 2880 }
{ "_id" : null, "count" : 2704 }
{ "_id" : "80218", "count" : 2158 }
{ "_id" : "80301", "count" : 1888 }
{ "_id" : "80202", "count" : 1818 }
{ "_id" : "80020", "count" : 1398 }

> coll.aggregate([{"$match": {"address.city": {"$exists":1}}}, {"$group": {"_id":
"$address.city", "count":{"$sum":1}}}, {"$sort":{"count":-1}},{"$limit":10}])
{ "_id" : "Denver", "count" : 25776 }
{ "_id" : "Lafayette", "count" : 7066 }
{ "_id" : "Boulder", "count" : 5174 }
{ "_id" : "Broomfield", "count" : 1884 }
{ "_id" : "Aurora", "count" : 1408 }
{ "_id" : "Louisville", "count" : 1270 }
{ "_id" : "Westminster", "count" : 1016 }
{ "_id" : "Commerce City", "count" : 518 }
{ "_id" : "Lakewood", "count" : 514 }
{ "_id" : "Idaho Springs", "count" : 448 }
```

After the queries were made, the US Zip code website was used to find the city for that code. Of note, the #1 postcode, 80026, is associated with Lafayette while this

town is #2 on the most frequent cities list. 80211 and 80205 codes are both associated with Denver, the #1 city in the OSM data, albeit two different neighborhoods - Highland and Whittier respectively. This makes sense when looking at the frequency of Denver - 25776 compared to the frequency of the top postcodes which are almost an order of magnitude smaller. Denver has multiple postcodes due to its size.

2. Data Overview

This section outlines basic statistics on the full Denver OSM dataset.

File sizes:
denver-boulder_colorado.osm …….…... 686MB
denver-boulder_colorado.osm.json ….. 774.9MB

 # Number of documents
> coll.find().count()
6862808

 # Number of nodes
> coll.find({"type":"node"}).count()
6189464

 # Number of ways
> coll.find({"type":"way"}).count()
672418

 # Number of distinct users
> coll.distinct("created.user").length
1649

 # Top 10 amenities
> coll.aggregate([{"$group":{ "_id":"$amenity",
"count":{"$sum":1}}},{"$sort":{"count":-1}},{"$limit":10}]).pretty()
{ "_id" : null, "count" : 6813586 }
{ "_id" : "parking", "count" : 23758 }
{ "_id" : "restaurant", "count" : 3604 }
{ "_id" : "school", "count" : 3012 }
{ "_id" : "fast_food", "count" : 1750 }

{ "_id" : "place_of_worship", "count" : 1744 }
{ "_id" : "bicycle_parking", "count" : 1624 }
{ "_id" : "fuel", "count" : 1260 }
{ "_id" : "bench", "count" : 1218 }
{ "_id" : "bank", "count" : 910 }

#Number of Mexican resaurants
> coll.find({"cuisine": "mexican"}).count()
484

#Top 1 Contributing User
> coll.aggregate([{"$group":{"_id":"$created.user", "count":{"$sum":1}}},
{"$sort":{"count":-1}}, {"$limit":1}])
{ "_id" : "Your Village Maps", "count" : 1299462 }

3. Additional Data Analysis Ideas

Additional data analysis that may be useful for businesses is to assess their presence in locations on a per capita basis. For instance, a business may want to know how many operating locations exist in a postcode and how many people live in that postcode. For areas that have a low ratio of operations per capita there may be potential for expansion. One way to accomplish this idea would be to add population data from the US census to the OSM data. This could be done at the postcode level in order to provide a higher level of resolution than just at a city or county level. The problem with this tactic would be that every amenity would have a population associated with it - which may be confusing to some and unnecessarily increase the size of the dataset. Another way to accomplish this would be to keep the OSM and population data as separate databases in MongoDB and to query them both in a pymongo script. For a given amenity, the OSM will contain the postcode and then the population could be easily queried from the population database.

Additional Data Exploration Using MongoDB Queries

#Top 10 amenities in most frequently listed postcode: 80026
> coll.aggregate([{"$match": {"address.postcode": "80026"}},{"$group":{ "_id":"$amenity",
"count":{"$sum":1}}},{"$sort":{"count":-1}},{"$limit":10}]).pretty()

{ "_id" : null, "count" : 7262 }
{ "_id" : "restaurant", "count" : 10 }
{ "_id" : "fuel", "count" : 6 }
{ "_id" : "parking", "count" : 4 }
{ "_id" : "car_wash", "count" : 4 }
{ "_id" : "hospital", "count" : 4 }
{ "_id" : "veterinary", "count" : 2 }
{ "_id" : "mortuary", "count" : 2 }
{ "_id" : "place_of_worship", "count" : 2 }
{ "_id" : "locksmith", "count" : 2 }

#Top 10 amenities in Denver
> coll.aggregate([{"$match": {"address.city": "Denver"}},{"$group":{ "_id":"$amenity", "count":{"$sum":1}}},{"$sort":{"count":-1}},{"$limit":10}]).pretty()
{ "_id" : null, "count" : 23432 }
{ "_id" : "restaurant", "count" : 842 }
{ "_id" : "cafe", "count" : 188 }
{ "_id" : "bar", "count" : 140 }
{ "_id" : "place_of_worship", "count" : 140 }
{ "_id" : "doctors", "count" : 108 }
{ "_id" : "fast_food", "count" : 96 }
{ "_id" : "clinic", "count" : 80 }
{ "_id" : "pub", "count" : 78 }
{ "_id" : "bank", "count" : 74 }

Conclusion

While several inconsistencies were noted when auditing the data in both the street names and postcodes, these problems were fixed programmatically. Otherwise the data seems to be fairly well cleaned other than a couple fields that have several “null” values. One of the more surprising findings of the data is that the most frequently used postcode does not belong to Denver proper but to one of the suburbs. Additionally, another somewhat surprising finding was that of the most frequently used postcode there were quite few cited restaurants. However, when expanded to include all of Denver proper, over 800 restaurants appeared. Such data may be useful to businesses who wish to evaluate their market penetration on a per capita basis. By using both the OSM and census population data a business may be able to spot opportunities for expansion.

4. Resources

- https://discussions.udacity.com/t/final-project-processes-taking-10-min/26540
- https://discussions.udacity.com/t/problem-launching-mongodb/40925
- https://discussions.udacity.com/t/importing-data-into-mondgodb/42524/5
- https://discussions.udacity.com/t/problem-launching-mongodb/40925
- http://stackoverflow.com/questions/14924495/mongodb-count-num-of-distinct-values-per-field-key