Data Wrangling with MongoDB - Calgary Area

August 14, 2017

1 Data Wrangling with MongoDB -- Calgary Area

1.0.1 Map Area: Calgary, AB, Canada

I lived in Calgary for a few years ago. So I am interested in looking into the data quality of city calgary and check what data query revealed in openmap. I would like an opportunity to contribute to its improvement on OpenStreetMap.org.

1.1 Section 1. Problems in the Map

After downloading a small sample of the Calgary area data from OpenStreetMap, we use the data.py file to run a test and review the quality of the data on map information. I notice that most information on Calgary OpenStreetMap are well maintained. But there are still very few problem on the information of map:

- The format of postal code is not uniform.
- The format of address unit is not unform.
- The Inconsistent postal Code.
- Typo of postal code.

1.1.1 Postal Code

The postal code in calgary is not in uniform format. In some area postal codes are 'T2L1H4' while other area are in the form 'T3A 2H4'. In offical postal code, there is a space between the first 3 letters and the next three letters. The offical format postal code benefits MongoDB aggregation calls on postal codes.

There are errors in postal code in osm file such as 'T2T 0A7;T2T 0A7'. This may be caused from typo caused by contributors. We will implement the data clean process to correct the error of the postal code into the correct one.

There are also inconsistant format of postal code in osm fiel such as 'AB T2S 2N1'. The data cleaning process deletes the province letters in front.

1.1.2 Address Unit

The address unit in calgary is not in uniform format. In some house unit, the addresses are in the form of '106', while others are in the form of '#104'. We will use the offical house unit number format, that is in the form of number '106'. The offical format unit benefits MongoDB aggregation calls on house unit.

1.2 Section 2. Data Overview

This section we invest the basic statistics about the dataset and use Mongodb queries to gather them.

1.2.1 Size of files

Calgary.json ..... 86.8 MB
Calgary.osm ..... 55.2 MB

1.2.2 Number of documents

> db.calgarymap.find().count()
> 271435

1.2.3 Number of nodes and ways

> db.calgarymap.find({"type":"node"}).count()
> 233027
> db.calgarymap.find({"type":"way"}).count()
> 38408

1.2.4 Number of unique users

> db.calgarymap.distinct("created.user").**len**()
> 606

1.2.5 Top 10 contibuting user

```
> db.calgarymap.aggregate([{"$group":{"_id":"$created.user", "count":{"$sum":1}}},
{"$sort":{"count":-1}}, {"$limit":10}])
```

> {u'count': 48600, u'_id': u'abDoug'}
{u'count': 33729, u'_id': u'Zippanova'}
{u'count': 27837, u'_id': u'JamesBadger'}
...

1.2.6 The number of coffee store

> db.calgarymap.find({'cuisine' : 'coffee_shop'}).count()
> 48

1.3 Section 3. Addtional Ideas

The contribution of users seems skewed. Here are some user percentage statistics:

- Top user contribution percentage ("abDoug") - 17.90%
- Combined top 2 users' contribution ("abDoug" and "Zippanova") - 30.33%
- Combined Top 10 users contribution - 70.76%

1.3.1 Additional data exploration using mongoDB queries

Top 10 amenity: `> db.calgarymap.aggregate([{"$match":{"amenity":{"$exists":1}}}, {"$group":{"_id":"$amenity", "count":{"$sum":1}}}, {"$sort":{"count":-1}}, {"$limit":10}])`
> {u'count': 1051, u'_id': u'parking'}
{u'count': 356, u'_id': u'restaurant'}
{u'count': 274, u'_id': u'fast_food'}
...

Top 10 cuisine: `> db.calgarymap.aggregate([{"$match":{"cuisine":{"$exists":1}}}, {"$group":{"_id":"$cuisine", "count":{"$sum":1}}}, {"$sort":{"count":-1}},{"$limit":10}])`
> {u'count': 48, u'_id': u'coffee_shop'}
{u'count': 41, u'_id': u'burger'}
{u'count': 34, u'_id': u'pizza'}
...

1.4 Conclusion

After this review of the data in calgary area on openstreetmap, there are obviously a few errors and incompletion on map. It interets for me to notice there are a fare mount of contributors on openstreetmap on calgary area.

Idea(s) concerning improving the data quality of OSM:

- There are a few problems encountered in raw calgary.osm file, for instance, 'The format of postal code is not uniform' and 'The Inconsistent postal Code'. In order to prevent these issues in the future, we can apply the data cleaning process, for example, using data.py file, to correct the raw data when contributor input their data into map.

  Benefits: Improving the quality of data.

- The contribution of the information on maps are limited to a few group of contributors. We notice that top 10 users contribute around 70% of map informations. We should encourage more contributors, especially locals to provide us more robust map data and contribute on the error correction process. A possible way to stimulate the new contributors is to award them virtual metal in websites for recognizing their contributions. We are also interested in simplying the map data providing and cleaning process so more non technique contributors could join.

  Benefits: Improving the users' participating.

- We could also us Cross-referencing/Cross-validating incorrect or missing data from other databases like Google API.

Benefits: Improving the quality of data.

With a more robust data processor and working contributor together wiht more robust data processor, I think it would be possible to get a great mount of cleaned data to openstreetmap.

In [ ]: