Data Wrangle OpenStreetMaps Data

Alexander Baranov

Map Area: Nizhniy Novgorod, Russia

https://s3.amazonaws.com/metro-extracts.mapzen.com/nizhniy-novgorod_russia.osm.bz2

1. Problems Encountered in the Map

In general, I did not meet issues that I would expect to meet. There is no tricky abbreviation as it happens in other areas, no issues with postal code etc.

Anyway, there are two general issues:

- Main issue I should fix was different order in street name. In Russian, street names 'улица Петрова' and 'Петрова улица' are the same street while for program they are definitely different. Therefore, I needed to use one of options and all names in another should be translated into it. I did it in update_name(name) function of convertToJson module.
- Another issue is 'empty' nodes. Ones that do not contain any information but longitude and latitude, like:

  ```
  {
   "created": {
    "user": "Kato Kontenta",
    "timestamp": "2012-12-30T18:59:54Z",
    "version": "9",
    "changeset": "14468387",
    "uid": "624774"
   },
   "pos": [
    56.2819501,
    43.8966164
   ],
   "type": "node",
   "id": "76481136"
  }
  ```

  I am not sure why they were created. Some of them have many revisions. This kind of issues can not be fixed on my side without losing information. I have no way to add missed information. In the same time, I cannot skip those nodes since others could reference them.

The rest of data is fine.

2. Data Overview

File sizes

Filename	File size, MB
nizhniy-novgorod_russia.osm	184
nizhniy-novgorod_russia.osm.json	261
nizhniy-novgorod_russia_sample.osm	18,5
nizhniy-novgorod_russia_sample.osm.json	26,1

Number of documents

```
> db['nizhniy-novgorod_russia'].find().count()
```

952266

Number of nodes

> db['nizhniy-novgorod_russia'].find({"type":"node"}).count()

807533

Number of ways

> db['nizhniy-novgorod_russia'].find({"type":"way"}).count()

144482

Number of unique users

> db['nizhniy-novgorod_russia'].distinct("created.user").length

352

Top 1 contributing user

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

{ "_id" : "Kato Kontenta", "count" : 347569 }

Number of users appearing only once (having 1 post)

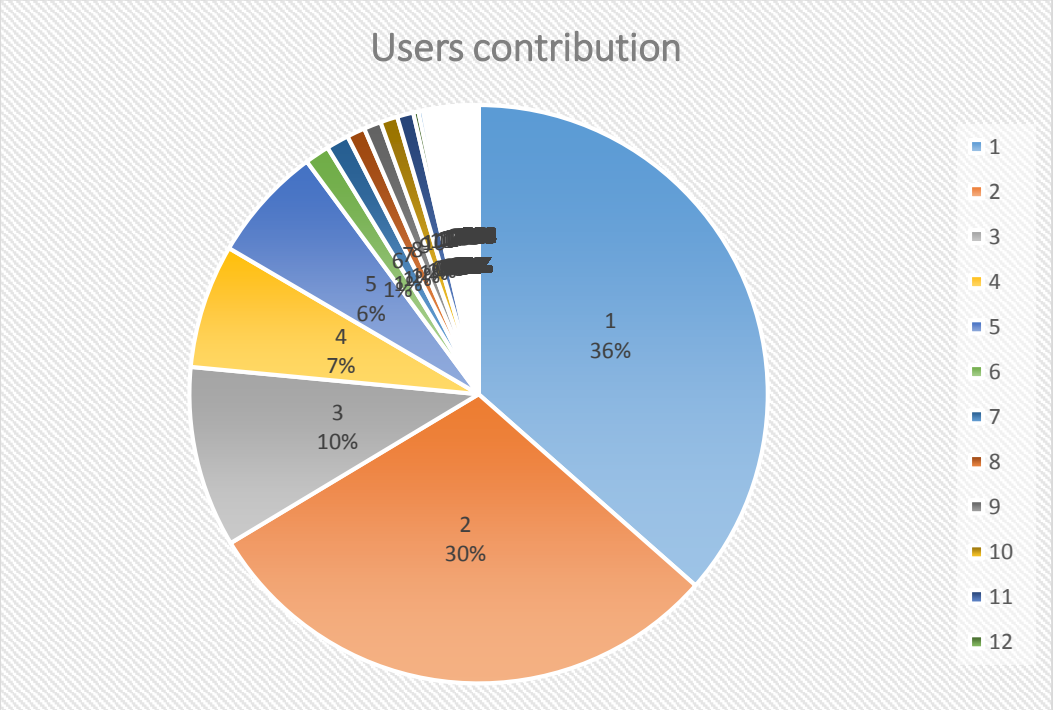
> db['nizhniy-novgorod_russia'].aggregate([{"$group":{"_id":"$created.user", "count":{"$sum":1}}}, {"$group":{"_id":"$count", "num_users":{"$sum":1}}}, {"$sort" : {"_id" : 1}}, {"$limit":1}])

{ "_id" : 1, "num_users" : 65 }

3. Additional Ideas

Users contribution

There are two main users accounted for 2/3 of contributions. 5 top users accounted for 89 % of contributions. The rest are minor contributors entered from one to few records.



Top 10 amenities

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

```
{ "_id" : "parking", "count" : 473 }
{ "_id" : "bench", "count" : 328 }
{ "_id" : "waste_basket", "count" : 276 }
{ "_id" : "kindergarten", "count" : 263 }
{ "_id" : "cafe", "count" : 206 }
{ "_id" : "waste_disposal", "count" : 172 }
{ "_id" : "school", "count" : 169 }
{ "_id" : "bank", "count" : 154 }
{ "_id" : "pharmacy", "count" : 150 }
{ "_id" : "fuel", "count" : 140 }
```

Chart of cuisines

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

```
{ "_id" : null, "count" : 64 }
{ "_id" : "japanese", "count" : 5 }
{ "_id" : "english", "count" : 1 }
{ "_id" : "italian", "count" : 1 }
{ "_id" : "european;japanese", "count" : 1 }
{ "_id" : "pizza", "count" : 1 }
{ "_id" : "russian", "count" : 1 }
{ "_id" : "ukrainian", "count" : 1 }
{ "_id" : "uzbek", "count" : 1 }
```

Conclusion

I have controversial feeling about the data. On one hand, entered data are cleaned enough. I did not spend much effort to fix them and to start using them. On the other hand, the data contains too much information that I would count as garbage like information about each bench and each wastebasket. It looks like local people entered information for local people which local people know already why I would like to see information from local people to tourists. However, this information is missed mostly. For instance most of restaurants (64 / 76) do not have cuisine field entered. There is still much work to enter needed information.

To improve data wrangling for further processing in addition to what I have done I would do following:

- Filter our noise and meaningless nodes. Probably there should be different levels of filtration: for local peoples, for tourists, for drivers etc. So different filtration rule would be applied. Meaningless nodes should be defined as ones that do not care information and are not referenced by others.
- Add default values for missing values of important fields. For instance, if restaurants cuisine is missed we count that it is Russian or European (not sure which one is more popular in Nizhniy-Novgorod.