OpenStreetMap Project
Data Wrangling with MongoDB
Peterson C. de Oliveira

Map Area: Curitiba, PR, Brazil

http://metro.teczno.com/#curitiba-brazil

1. Problems Encountered in the Map

A sample of the Curitiba area was downloaded and a python script was used to audit, clean and save the data into a JSON file. After this process the data was imported in Mongodb using the follow command:

mongoimport --db openstreet -c curitiba --type json --file /path/curitiba_brazil.osm.json --jsonArray

While auditing the file with a python script some problems were noticed, as the excessive use of street names abbreviation, language translation problems and wrongly recorded data.

Language translation problems

At the audit stage some language issues were detected. This will be a problem when you work with data around the globe, mainly because there is not a common language so the exceptions and regex matches will be diverse becoming more difficult to have a general rule to deal with every data.

For example if you search for way types, as Street, Avenue and etc you will find almost 0 matches because the word position to indicate the type is on the beginning in Portuguese language instead of at the end as it is in English.

So, the street type regex should be changed from re.compile(r'\b\S+\.?**$**', re.IGNORECASE) to re.compile(r'\b\S+\.?', re.IGNORECASE).

Another point to change is the mapping structure, that should be

```
mapping = { "R.": "Rua",
            "Aveninda": "Avenida",
            "Av.": "Avenida",
            "Av ": "Avenida ",
            "RUA": "Rua",
            "Rod.": "Rodovia",
            "Praça": "Praca"
            }
```

instead of

```
mapping = { "St": "Street",
            "St.": "Street",
            "Ave": "Avenue",
            "Rd.": "Road"
            }
```

At the cleaning stage I updated almost every problem encountered at the .osm file, the only exception was the wrongly recorded data which would be impossible to update without some specialized help.

Over-abbreviated Street Names

The use of over abbreviation was common mainly for Street, Avenue, Road, and Park. the result obtained was as follow,

```
{'Alfred': set(['Alfred Charvet']),
 'Av': set(['Av Comendador Franco',
            'Av Nossa Senhora de Lourdes',
            'Av das Torres']),
 'Av.': set(['Av. Comendador Franco', 'Av. Sete de Setembro']),
 'Aveninda': set(['Aveninda Marechal Floriano Peixoto']),
 'Av\x7fenida': set(['Av\x7fenida Portugal']),
 'BR': set(['BR 116 km 93']),
 'BR-116': set(['BR-116']),
 'BR-277': set(['BR-277']),
 'BR116': set(['BR116']),
 u'Bar\xe3o': set([u'Bar\xe3o do Serro Azul']),
 'Br': set(['Br 376']),
```

'Carlito': set(['Carlito Dissenha']),
u'Centro': set([u'Centro Polit\xe9cnico da UFPR, Caixa Postal 19100']),
'Comendador': set(['Comendador Franco']),
'Filipinas': set(['Filipinas']),
'Francisco': set(['Francisco Caetano Coradim', 'Francisco Dranka']),
'Galeria': set(['Galeria Lustosa']),
u'Hospital': set([u'Hospital \xd4nix Hospital \xd4nix 2321 Av. Vicente Machado']),
'Jacarezinho': set(['Jacarezinho']),
'Linha': set(['Linha Verde']),
'Manoel': set(['Manoel Ribas']),
u'Pra\xe7a': set([u'Pra\xe7a 19 de Dezembro',
u'Pra\xe7a Divina Pastora',
u'Pra\xe7a Doutor Vicente Machado',
u'Pra\xe7a General Os\xf3rio',
u'Pra\xe7a Generoso Marques',
u'Pra\xe7a Gibran Khalil',
u'Pra\xe7a Guido Viaro',
u'Pra\xe7a Marechal Alberto Ferreira de Abreu',
u'Pra\xe7a Nossa Senhora da Salete',
u'Pra\xe7a Nossa Senhora de Salete',
u'Pra\xe7a Os\xf3rio',
u'Pra\xe7a Rui Barboca',
u'Pra\xe7a Rui Barbosa',
u'Pra\xe7a Tiradentes',
u'Pra\xe7a Zacarias',
u'Pra\xe7a do Redentor']),
u'R.': set([u'R. At\xedlio B\xf3rio',
'R. Escola de Oficiais Especialistas',
'R. Mateus Leme',
'R. Sargento Erwin',
'R. Sargento Lafayette',
'R. Sargento Milano',
'R. Sgt. Roberto Maciel']),
'RUA': set(['RUA VICENTE DE CARVALHO \x7f ']),
'Rod.': set(['Rod. BR-376, Km 23,5 (sentido Joinville/ Curitiba']),
'Rui': set(['Rui Barbosa']),
'R\x7fua': set(['R\x7fua Itacolomi', 'R\x7fua Sinke Ferreira']),
'rua': set(['rua cruz machado'])}

After the audit stage we applied a cleaning process to translate all the abnormalities discovered at the audit phase into a cleaned data ready to be recorded in a JSON file. For example the Av Comendador Franco became Avenida Comendador Franco and so on.

Addresses with errors

At the audit stage some data were different from the expected, as Alfred Charvet, Linha Verde, Aveninda and Hospital Ônix Hospital Ônix 2321 Av. Vicente Machado. Most of them seems to be some typos errors and other seems to be the common sense name from places which, if you are not a local you will have difficulties in classifying the point as a road, street, park or avenue.

2. Data Overview

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

File sizes

curitiba-brazil.osm ..... 79,6 MB
curitiba-brazil.osm.json .... 123,6 MB

 # Number of documents

```
> db.curitiba.find().count()
411035
```

 # Number of nodes

```
> db.curitiba.find({"type":"node"}).count()
366064
```

 # Number of ways

```
> db.curitiba.find({"type":"way"}).count()
44961
```

 # Number of unique users

```
> db.curitiba.distinct('created.user').length
432
```

 # Top 10 contributing users

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

 # Top 10 amenities

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

```
{ "_id" : "parking", "count" : 384 }
{ "_id" : "restaurant", "count" : 271 }
{ "_id" : "fuel", "count" : 237 }
{ "_id" : "bank", "count" : 185 }
{ "_id" : "school", "count" : 156 }
{ "_id" : "pharmacy", "count" : 144 }
{ "_id" : "place_of_worship", "count" : 132 }
{ "_id" : "fast_food", "count" : 111 }
{ "_id" : "pub", "count" : 69 }
{ "_id" : "university", "count" : 58 }
```

Some additional data exploration using MongoDB queries

 # Top 10 cuisines

```
>  db.curitiba.aggregate([{$match:{amenity:{$exists:1},  amenity:"restaurant"}},  {$group:{_id:"$cuisine",
count:{$sum:1}}},{$sort:{count:-1}}, {$limit:5}])
```

```
{ "_id" : null, "count" : 99 }
{ "_id" : "regional", "count" : 45 }
```

```
{ "_id" : "pizza", "count" : 33 }
{ "_id" : "steak_house", "count" : 23 }
{ "_id" : "italian", "count" : 20 }
{ "_id" : "japanese", "count" : 11 }
{ "_id" : "sandwich", "count" : 4 }
{ "_id" : "german", "count" : 3 }
{ "_id" : "asian", "count" : 3 }
{ "_id" : "barbecue", "count" : 3 }
```

 # Banks ranking

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

```
{ "_id" : null, "count" : 160 }
{ "_id" : "Banco do Brasil", "count" : 6 }
{ "_id" : "Itaú", "count" : 5 }
{ "_id" : "Bradesco", "count" : 4 }
{ "_id" : "Santander", "count" : 3 }
{ "_id" : "HSBC", "count" : 2 }
{ "_id" : "Caixa Economica", "count" : 2 }
{ "_id" : "Banco Central do Brasil", "count" : 1 }
{ "_id" : "Itau", "count" : 1 }
{ "_id" : "Caixa Economica Federal", "count" : 1 }
```

3. Additional Ideas

The use of Machine Learning techniques to improve the quality of the data available and do some predictions to complete the missing information would be very interesting. This can be very useful to solve the problem of incomplete information. One example can be obtained using the following query.

```
db.curitiba.aggregate([
        {$match:{amenity:{$exists:1}, amenity:"bank"}},
        {$group:{_id:"$operator", count:{$sum:1}}},
```

```
        {$sort:{count:-1}},
        {$limit:10}
])
```

From that we can see that,

```
{ "_id" : null, "count" : 160 }
{ "_id" : "Banco do Brasil", "count" : 6 }
{ "_id" : "Itaú", "count" : 5 } .....
```

we have more null information than correct information at the 'operator' field. Analysing a little bit more the data we can see that the operator data can be obtained from the 'name' attribute. This can be achieved with the use of some supervised algorithm, to predict the value of the operator attribute based on some train data available. The downside of this approach is the required extra processing power to do all those analysis automatically, another problem with this type of solution is that you train dataset should be reliable, thus gathering a reliable training set would be a challenge. Because the training set needs to be representative of the real-world use of the function. Thus, a set of input objects is gathered and corresponding outputs are also gathered, either from human experts or from measurements. [2]

After the training process, we need to evaluate the accuracy of the learned function. After parameter adjustment and learning, the performance of the resulting function should be measured on a test set that is separate from the training set.

Another suggestion would be to use a geographic database or module, to give more power to the analysis, including spatial queries to display useful information to the users. Some possible queries would be for example, obtain the banks with more parking lots near, or the best cafe near a position (lat,long).
This would converge to a GIS solution. With GIS technology, people can compare the locations of different things in order to discover how they relate to each other. For example, using GIS, the same map could include sites that produce pollution, such as gas stations, and sites that are sensitive to pollution, such as wetlands. Such a map would help people determine which wetlands are most at risk.[1]

From [3] and [4] we can see that some disadvantages of implementing a GIS system are:

- Without adequate data, GIS is not very useful. It requires an enormous amount of data inputs to be practical for some tasks.
- The earth is round and geographic error is increased as you get into a larger scale.
- GIS layers, may lead to costly mistakes when property agents interpret a GIS map, or engineer's design around GIS utility lines.
- The fourth area relates to technology – specifically computer hardware, GIS software and training.

- The fifth area concerns methods – assuming the previous data and technological problems have been resolved – how can GIS be used to improve our understanding of the problem? Spatial statistical analysis is a newly developing field and has no agreed upon or standard methodologies.

Conclusion

After this review of the data it's obvious that the Curitiba area is incomplete, many information are missing. I think that with the use our data.py scrpt and the help of some machine learning algorithm we can obn betteresult in termsf qualitof the da.whether by scripting a map editing bot or otherwise. With a rough GPS data processor in place and working together with a more robust data processor similar to data.py think it would be possible to input a great amount of cleaned data to OpenStreetMap.org.