OpenStreetMap Sample Project: Data Wrangling with MongoDB

Map Area: Barcelona, Spain

https://mapzen.com/data/metro-extracts

Problems encountered in the map

I focused my work with this dataset into three different problems I spotted during the audit:

1. A plethora of phone number formats
2. Inconsistent language (Catalan / Spanish) and street abbreviations
3. Inconsistent use of tags for the same data (`phone` = * and `contact:phone`=*, etc.)
4. Unspecific `url` tags.

Phone Number Formatting

According to the OSM wiki, the correct format for phone and fax numbers is one of those two:

> `phone=+<country code> <area code> <local number>`
> `phone=+<country code>-<area code>-<local number>`

However, a rapid exploration of the phone and fax numbers in the dataset reported that this was not the case. I implemented the function `formatPhoneNumber` to consistently format numbers and faxes prior to inserting them to the MongoDB. Some examples of this process can be found below.

Phone Type	Before Formatting	After Formatting
Regional Phone	`+34 935729621`	`[‘+34 93 5729621’]`
	`(+34) 934 73 41 28`	`[‘+34 93 4734128’]`
	`0933880630`	`[‘+34 93 3880630’]`
	`93-383-58-03`	`[‘+34 93 3835803’]`
Mobile Phone	`667 253 318`	`['+34 667 253318']`
Emergency Phone	`112`	`['112']`
Multiple numbers	`677780096, 935142424, 932212202`	`['+34 677 780096', '+34 93 5142424', '+34 93 2212202']`
	`933770606 -.934744208`	`['+34 93 3770606', '+34 93 4744208']`

Street Name Abbreviations & Language

This dataset is especially interesting regarding street names, because in Catalunya there are two official languages: Spanish and Catalan. In consequence, depending on the preferred language of the submitter, some street names are submitted in Spanish, or Catalan. Although harmonizing and translating all the data into a single language is a gigantic task, I implement the function `fixStreet` to start by changing the abbreviations (both Spanish and Catalan) used to its Catalan full name, as it's what can be read in Barcelona street plates. As such, 'C/ Sant Jaume' would be converted to 'Carrer Sant Jaume', and 'Rbla. Catalunya' would be 'Rambla Catalunya' and 'Plaza Francesc Macià' would be formatted to 'Plaça Francesc Macià'.

Reshaping subcategory tags

In OSM, the tags `'phone'`, `'fax'`, `'email'`, `'website'`, `'facebook'`, `'twitter'`, `'linkedin'`, `'google_plus'`, `'instagram'`, `'diaspora'`, `'xing'`, `'webcam'` and `'vhf'` can be entered directly or under the 'contact' category (such as `'contact:phone'` or `'contact:website'`). Although both approaches are valid, I decided to group all the tags under the contact category to sanitize and harmonize the files and ease the search and comparison of those tags in the MongoDB database.

Then, I generalized the function to become `reshapeCategory`, and also created subdocuments for the tags under the `'created'` (as seen in the lesson examples) and `'address'` categories.

Reshaping url tags

Exploring the dataset, I noticed that a lot of nodes had a `'url'` tag, while it being quite unspecific. In consequence, I implemented the function `reshapeUrl` to address with two main objectives

1. All urls now start with http:// or https://
2. The `url` is categorized into more specific website subtypes if possible (e.g. `'contact:facebook'` or `'contact:twitter'`.

Data Overview

In this section I provide some statistics about the data I used:

File sizes:

```
barcelona_spain.osm : 188.5 MB
barcelona_spain.osm.json : 213.1 MB
```

Number of documents, nodes and ways

```
> show dbs
local  0.078GB
osm    0.453GB
> use osm
switched to db osm
> show collections
barcelona
system.indexes
> db.barcelona.count()
960395
> db.barcelona.find({"type":"node"}).count()
850657
> db.barcelona.find({"type":"way"}).count()
109441
```

Note: although I only processed nodes and ways to include in the database, some users used the key `type` inside the node tag. I could solve that by replacing it with the correct type at the end of the tag processing, but I would leave it like these to manually curate and edit those entries.

Number of unique users

```
> db.barcelona.distinct("created.user").length
1679
```

Tourism amenities and its type in the Barcelona Metropolitan Area

```
> db.barcelona.aggregate([{"$match":{"tourism":{"$exists":1}}},
{"$group":{"_id":"$tourism","count":{"$sum":1}}},  {"$sort":{"count":-
1}},  {"$limit":25}])
{ "_id" : "hotel", "count" : 288 }
{ "_id" : "information", "count" : 162 }
{ "_id" : "attraction", "count" : 156 }
{ "_id" : "artwork", "count" : 92 }
{ "_id" : "viewpoint", "count" : 72 }
{ "_id" : "museum", "count" : 64 }
[...]
```

Tourism amenities and its type in Barcelona

```
> db.barcelona.aggregate([{"$match":{"address.city" : "Barcelona",
"tourism":{"$exists":1}}},
{"$group":{"_id":"$tourism","count":{"$sum":1}}}, {"$sort":{"count":-
1}}])
{ "_id" : "hotel", "count" : 64 }
{ "_id" : "museum", "count" : 15 }
{ "_id" : "attraction", "count" : 13 }
{ "_id" : "hostel", "count" : 5 }
{ "_id" : "guest_house", "count" : 3 }
{ "_id" : "artwork", "count" : 2 }
```

Where are the rest of tourism nodes?

```
> db.barcelona.aggregate([{"$match":{"tourism":{"$exists":1}}},
{"$group":{"_id":"$address.city", "count":{"$sum":1}}},
{"$sort":{"count":-1}}])
{ "_id" : null, "count" : 821 }
{ "_id" : "Barcelona", "count" : 102 }
[...]
{ "_id" : "barcelona", "count" : 1 }
[...]
```

Not surprisingly, the great majority of the tourism amenities (as the majority of the nodes) in this dataset do not have completed values for the address tags, such as city and postcodes, and a few of them are wrongly spelled or differently formatted, e.g.:

```
>
db.barcelona.distinct('addres
s.city')
[
[...]
"el Prat del Llobregat
Barcelona",
"Barelcona",
"L'Hospitalet de Llobregat,
Barcelona",
"sant feliu de llobregat",
"barcelona",
"cerdanyola del Vallès",
"Mira-sol. Sant Cugat del
Vallès",
"Bacelona",
"Barcelana",
"Santa coloma de Cervelló",
"Sant Climent de Llobregat",
"Gava",
"BARCELONA",
"08005",
[...]
]
```

Upon inspecting the information for the ‘`address:city`’, it seems that there is complete address information for more nodes in ‘Santa Coloma de Cervelló’ than in Barcelona, which is striking taking into account the importance difference between this two cities:

```
> db.barcelona.aggregate([{"$match":{"type":"node","address.city":
{"$exists":1}}}, {"$group":{"_id":"$address.city","count":{"$sum":1}}},
{"$sort":{"count":-1}},  {"$limit":25}])
{ "_id" : "Santa Coloma de Cervelló", "count" : 1750 }
{ "_id" : "Barcelona", "count" : 979 }
[...]
```

We can see how this fact is mainly due to a unique user very active in his (probably) hometown:

```
db.barcelona.aggregate([{"$match":{"type":"node","address.city": "Santa
Coloma de Cervelló"}},
...   {"$group":{"_id":"$created.user","count":{"$sum":1}}},
...   {"$sort":{"count":-1}},
...   {"$limit":25}])
{ "_id" : "Luis Peña", "count" : 1646 }
{ "_id" : "Rodrigo Rega", "count" : 94 }
{ "_id" : "geodreieck4711", "count" : 8 }
{ "_id" : "Robowolfer", "count" : 1 }
{ "_id" : "HolgerJeromin", "count" : 1 }
```

Other ideas about the dataset

There is quite a bit work we could do to improve with this dataset. For example, as mentioned before, knowing that we tend to miss information for the `city` tag but have the coordinates, we could extract this information from the points near the one of interest using the MongoDB geospatial indexes. This would provide solid information for big cities, but probably would have to determine the maxDistance experimentally not to misclassify nodes in smaller towns.

Although a little more difficult (especially in this dataset due to the language differences), we could also update the postcodes for specific addresses using publicly available information, which is formatted as follows:

```
Postcode: Street Name (Min/Max Odd numbers) (Min/Max Even Numbers)
08027:CAN BERDURA,Passatge
08028:CAN BRUIXA, DE (Impares del 1 al 19)  (Pares del 2 al 22)
08014:CAN BRUIXA, DE (Impares del 21 al final)  (Pares del 24 al final)
```

Due to the high error potential using this methodology, implementing initiatives from the city councils or on-site gamification would probably be a safer bet to improve the OSM dataset in this region. While it is true that most of the errors in the dataset are user-driven, without the user submission we would not have any dataset to start with, and a large enough user base would curate the data itself.