OpenStreetMap data wrangling with MongoDb Documentation

Release

Ignacio Toledo

March 10, 2015

Contents

1 Introduction

The goal of this project is to use data wrangling (munging) techniques over data obtained from the OpenStreetMap. The data correspond to the area encompassing Santiago de Chile. The data wrangling consisted in :

- Obtaining the data from an API in OSM format.
- Parse the data using an python xml library.
- Audit the data set and determine which problems we will solve
- Once some functions to fix the chosen problems are created, a second parsing is done from the OSM file, but this time fixing some of the issues.

- Transform the parsed data into a valid JSON file to be imported by MongoDB
- Import the JSON file into MongoDB

2 Problems encountered in our map area

2.1 Problems related to the parsing of the data

First we described some basic problems that need to be solved in order to convert the XML file in a JSON file that can be imported by mongodb.

- Problematic characters: the fields names of the JSON document should include only alphabet character. Also, according to the OSM wiki they should be avoided in tags. We removed all *tags* that were matched by the regular expression:

  ```
  /[=\+/&<>;\'"`\?%#$@\,\. \t\r\n]/
  ```

- Tags with two sets of colons. Colons are present in the OSM's *tags* to allow the use of namespaces. It is not recommend to overuse this feature, so we removed any tags with more than one namespace (i.e., more than one colon)

Both of these problems are solved using the appropriate algorithms that are coded in the python files for this project.

2.2 Problems related to the data audition

Here a list of the problems we found within the data is presented. The will be classified by the kind of problem defined in the Data Wrangling class.

2.3 Validity

- As in the Lesson 6 exercise, the street's name in the data set do not follow a standard format when dealing with the 2 main kinds of streets used in Santiago's street names: *Avenida* and *Pasaje*. In many cases they are abbreviated, and not in a uniform way. **This problem was solved in the final parsing of the OSM file**
- Many phone numbers do not follow a valid representation for Santiago de Chile. The phone numbers should be in the format: *+(Country_Code) (Area_Code) (Number)*.
  - The *Country_Code* should be 56, and in many cases is not present, nor the plus symbol at the beginning.
  - The *Area_Code* should be always 2 for residential lines, or 9X (with X a number between 5 and 9) for cell phones.
  - The *Number* should consist in an 8 digit number always starting with a 2 for residential lines; if a cell phone, the number should have 7 digits.

  Only 129 phone numbers (of 993) are in the correct formatting.

  This problem is resolved by doing a query with pymongo, and then using the 'update' procedures. The functions needed are at the end of the *osm_to_json2.py* file.
- E-mail address with no valid domain or plainly invalid (forbidden characters or missing @). Examples:
- Fields with multiple values without using semicolons to separate the different items
- Highways/streets with no 'name' tag.
- Address with multiple values for the street key.
- Ways without any tags

- Cross-field issues between *address.street* and street *name*: there are lots of nodes with the field *adress.street* where the street is not present in the list of streets that we can retrieve from the highway ‘way‘s.

2.4 Completeness

- There are missing schools, traffic lights, hospitals, etc. A gold standard database to fix completeness is not available.

2.5 Consistency

- There are some nodes, very close in the same position, but with different tags. For example, there is the case of a two nodes within a *way* defining a school area. However one of the has the tag *amenity=school* and the other *amenity=fuel*: do we have a school or a gas station here? Is the *area* a school?

3 Overview of the data

3.1 File size:

santiago.xlm 216 MB (OSM file) santiago.xlm.json 237 MB (JSON file)

3.2 Number of documents

```
> db.santiago.find().count()
1015506
```

3.3 Number of nodes

```
> db.santiago.find({type: 'node'}).count()
814484
```

3.4 Number of ways

```
> db.santiago.find({type: 'way'}).count()
201022
```

3.5 Number of unique users and top 10 users

```
> db.santiago.distinct("created.user").length
1003

> db.santiago.aggregate(
      [{$group: {'_id': '$created.user',
                 'count': {$sum:1}}},
       {$sort: {'count': -1}},
       {$limit: 10}])
 { "_id" : "Zambelli Limitada", "count" : 244613 }
 { "_id" : "Fede Borgnia", "count" : 204199 }
 { "_id" : "felipeedwards", "count" : 107799 }
 { "_id" : "chesergio", "count" : 61124 }
 { "_id" : "dintrans_g", "count" : 56281 }
 { "_id" : "madek", "count" : 34397 }
 { "_id" : "ALE!", "count" : 30174 }
 { "_id" : "toniello", "count" : 27401 }
 { "_id" : "OttoPilot", "count" : 16241 }
 { "_id" : "Chilestreet", "count" : 15715 }
```

4 Additional Ideas about the data set

There are several tools available to do a quality assurance over an OSM file (JOSM/Validator) or online (Osmose, for selected areas). However they only deal with validity, consistency and uniformity, while accuracy and completeness are left out.

This is understandable, since the accuracy and completeness depend on having *gold standard* databases that we can use to perform these steps in the measure of the data quality. A great project would be to help the local city government to update and maintain the gold standard databases.

Currently, in Chile there is web page associated to the National Government that aims to provide this information to the users that require it. The information can't only be queried through the web interface, but an API has also been implemented. However many problems remain:

- Data is outdated and no longer maintained: besides some small number of databases, like the database with all the public transportation information for Santiago and the database with all the schools in the Santiago area, most of the geolocation data sets are outdated.
- Several geolocation data sets are missing: a database of the streets of Santiago with valid names, a database related to hospitals and clinics, a database with police stations, etc. They might actually exist, but they are not publicly available in this web page nor through the API.

Another good project would consist in resume the work at the WikiProject Chile OpenStreetMap page. There is some work already done there, which aims mainly to use a consistent set of tags and values to give consistency and uniformity to chilean map features, but is still clearly incomplete and kind of dead, since no modifications have been done in the last year.

4.1 Additional data exploration using MongoDB queries

Number of schools in Santiago

```
> db.santiago.find({'amenity':'school'}).count()
2550
```

Number of police stations in Santiago

```
> db.santiago.find({'amenity':'police'}).count()
72
```

Number of highways (streets) without name

```
> db.santiago.find({highway:{$exists:1}, type:'way', name:''}).count()
47
```

The 10 districts with the higher number of bus stops

```
> db.santiago.aggregate(
    {$match: {type: 'node', highway:'bus_stop'}},
    {$group: {'_id': '$is_in:city', 'count': {$sum:1}}},
    {$sort: {'count':-1}},
    {$limit:10})

 { "_id" : "Puente Alto", "count" : 800 }
 { "_id" : "Maipú", "count" : 746 }
 { "_id" : "La Florida", "count" : 668 }
 { "_id" : "Santiago", "count" : 620 }
 { "_id" : "San Bernardo", "count" : 476 }
 { "_id" : "Pudahuel", "count" : 416 }
 { "_id" : "Las Condes", "count" : 408 }
 { "_id" : "Ñuñoa", "count" : 362 }
 { "_id" : "La Pintana", "count" : 353 }
 { "_id" : "Peñalolén", "count" : 352 }
```