OpenStreetMap Project

Data Wrangling with MongoDB

Liya Naumova

Map area: Saint Petersburg, Russia

1. Problems Encountered in the Map

After running test with audit.py on sample data from small area of the city I found several types of problems:

- tags with problematic characters (“hours of operation”)
- abbreviated street names (“пл.”)
- city names not in Russian (“St. Petersburg”)
- misplaced and incorrect postal codes ('172006', '8', '24/7')
- buildings with multiple addresses

Tags with problematic characters

I found several tags with characters that may cause problems like “ref:sobory.ru” and ‘hours of operation’’ . I decided to ignore all tags with such characters.

Abbreviated street names

Checking street names I faced with a problem that in Russian street “type” can stand before or after street “name’’ so I couldn’t just find last word and assume that it is street type. I decided to look for all words that start with lowercase letter because in Russian street types do not start with capital letters and found some abbreviated ones like “пл.” instead of “площадь”.

```
      #Looking for words starting with lowercase letter
      street_type_re = re.compile(ur'\b[а-я][а-яА-ЯЁё-]+\.?',
re.U,)
```

Then I made a mapping of problem street types and expected and made the replacement.

City names not in Russian

Some “city” tags contain different variations of city name, sometimes not in russian, so I decided replace all such tags with standard “Санкт-Петербург”.

Incorrect postal codes

All postal codes in Saint Petersburg start with ”19” and are 6 digits long (“190056”). I checked all postcodes against this requirements and found many that don’t follow this pattern:

- postcodes starting with “18” - these belong to Leningrad oblast because parts of it are included into Saint Petersburg metro area
- other valid postcodes - possibly misplaced values, i decided to delete them.
- not postcodes at all (“24/7”, “RU”) - also deleted

2. Data Overview

File size:

saint_petersburg.osm - 77,5 MB
saint_petersburg.osm.json - 88,9 MB

Number of documents:

```
db.elements.count()
412631
```

Number of nodes:

```
nodes = db.elements.find({"type": "node"}).count()
357412
```

Number of ways:

```
ways =  db.elements.find({"type": "way"}).count()
55146
```

Top 10 users

```
db.elements.aggregate([{"$group":{"_id":"$created.user",
                                  "count":{"$sum":1}}},
                      {"$sort":{"count":-1}},
                       {"$limit":10}])
{u'count': 76166, u'_id': u'Danidin9'}
{u'count': 34332, u'_id': u'GaM'}
{u'count': 22854, u'_id': u'paavolobja'}
{u'count': 20954, u'_id': u'Sergey Astakhov'}
{u'count': 15794, u'_id': u'xronos'}
{u'count': 15378, u'_id': u'russianin'}
{u'count': 13873, u'_id': u'serge56'}
{u'count': 12493, u'_id': u'fserges'}
{u'count': 9550, u'_id': u'Ilgn'}
{u'count': 8942, u'_id': u'EIjas'}
```

Street with most elements:

```
db.elements.aggregate([{"$match":{"address.street":{"$exists":1}}}
,
                     {"$group":{"_id": "$address.street",
                                "count":{"$sum":1} }},
                    {"$sort":{"count":-1}},{"$limit":1}])
     {u'count': 72, u'_id': u'Лесная улица'}
```

Top 5 types of amenities:

```
     db.elements.aggregate([{"$match":{"amenity":{"$exists":1}}},
                          {"$group":{"_id": "$amenity",
                                 "count":{"$sum":1} }},
                           {"$sort":{"count":-1}},
                           {"$limit":5}])
     {u'count': 518, u'_id': u'parking'}
     {u'count': 277, u'_id': u'waste_disposal'}
     {u'count': 224, u'_id': u'bench'}
     {u'count': 167, u'_id': u'cafe'}
     {u'count': 119, u'_id': u'kindergarten'}
```

Top 5 leisure amenity types:

```
     db.elements.aggregate([{"$match":{"leisure":{"$exists":1}}},
                          {"$group":{"_id": "$leisure",
                                 "count":{"$sum":1} }},
                           {"$sort":{"count":-1}},
                           {"$limit":5}])

     {u'count': 378, u'_id': u'playground'}
     {u'count': 216, u'_id': u'pitch'}
     {u'count': 186, u'_id': u'park'}
     {u'count': 74, u'_id': u'common'}
     {u'count': 27, u'_id': u'stadium'}
```

3. Additional ideas

Exploring the dataset I found some inconsistency in tags describing buildings with two addresses:

1. second address tagged with tags “addr:street2” and “addr:housenumber2”
2. second address tagged “addr2:street” and “ addr2:house number”.

I think it’s better for consistency to change the schema to allow one document to have an array of addresses to ease the search of such buildings.

Data for Saint Petersburg look like already cleaned but not complete, I think it is possible to fill information about many map objects using other geoinformational systems API like 2GIS or Yandex.maps which contain much more information but there can be difficulties concerning copyright.