Data Wrangling with MongoDB

Maksym Osmanov
Udacity Nanodegree program
max.osmanov@gmail.com
27.04.2015

Outline

Problems encountered in the map

Working with a small sample part of map of Kyiv I encounter some problems which will be discussed below. The size of a test file was 2.96 MB and area was chosen from a central part of city, to cover as many objects as it is possible. Later, analyzing the whole area I didn't notice any other problem different from following

- Different formats were used to save phone numbers. Some examples are
    - `<tag k="phone" v="+380 (44) 223-98-61"/>`
    - `<tag k="phone" v="0038-044-234-52-61"/>`
    - `<tag k="phone" v="(044) 495 03 97"/>`
    - `<tag k="phone" v="+38 044 234-15-74"/>`
    - `<tag k="phone" v="+38 0672409109"/>`

  These different formats can be converted to a single one with a help of a regular expressions in Python. To solve this problem I use the material of [1] where algorithm, pieces of code and helpful advises are given.
- Kyiv is a central city of Ukraine where official language is Ukrainian. The worst for our task is that it is based on the Cyrillic script. Depending how the data are encoded, parsing of XML can be done not very correctly. For example, I have following line in a converted json file

  ```
  {"visible": "true", "historic": "memorial", "name":
  "\u042f\u0440\u043e\u0441\u043b\u0430\u0432\u0443
  \u041c\u0443\u0434\u0440\u043e\u043c\u0443", "created": {"changeset": " …
  ```

where name is absolutely unreadable! However, it is not a big problem for the project, because anyway, I don’t need to report here texts in Ukrainian.
To contribute to OpenStreetMap project one has to solve this problem with encoding. Python allows to do this

```
text.decode(‘cp1252’).encode(‘utf8’)
```

- Third problem in a map is also related to the language. In addition to an element “name” also similar elements are introduced in OpenMap “name:en”, “name:de” and so on. However, often none of these additional elements (most importantly “name:en”) is filled. It’s not even necessary to translate them from Ukrainian to English. One can use so-called Romanization of Ukrainian conversion [2] to give foreigners at least a chance to be able to read this name. There is a pytils package for Python which allows to do so [3]. However, more exciting thing is to use methods of machine learning and artificial intelligence to make a real translation of the name. For example, Google’s methods can be used [4]. We leave this problem for future investigations.
- The last we want to note that the map evolves very quickly because of the active participation and also because of rapid development of a city itself. For example, recently more than 25 streets were renamed, because of the Soviet Past [5]. Of course, all this changes affect a map as well. However, using Python one can change all names if a few minutes.

Data Analysis

Data overwiev

In this section we write basic statistics about the dataset. XML file taken from OpenStreetMap has a size 222 MB and converted json file 246 MB.

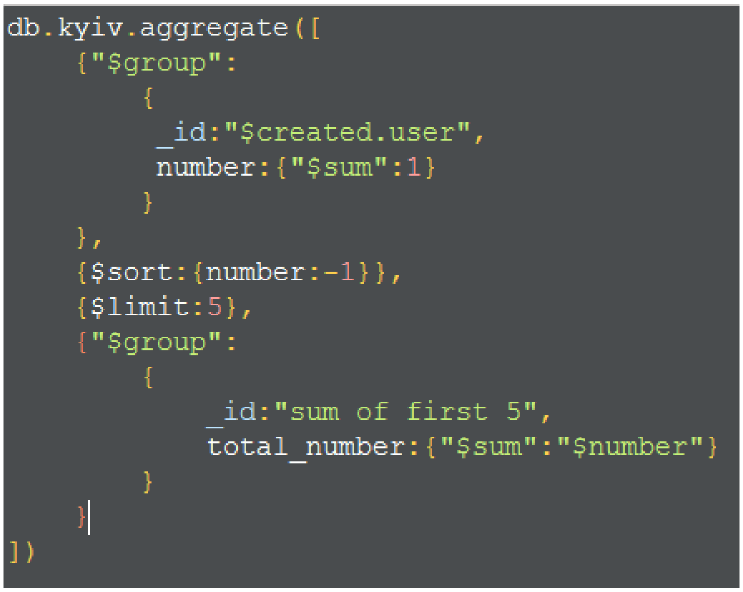
Using mongoimport we import the json file in MongoDB and analyze it with MongoDB queries.

Dataset contains 1153104 documents (db.kyiv.find().count())
Dataset contains 1029968 nodes (db.kyiv.find({"type":"node"}).count())
Dataset contains 122988 ways (db.kyiv.find({"type":"way"}).count())

In total 1257 distinct users participated in creating the map (db.kyiv.distinct("created.user").length)

Analysis of contributions

To analyze contributions of first five users we use following query



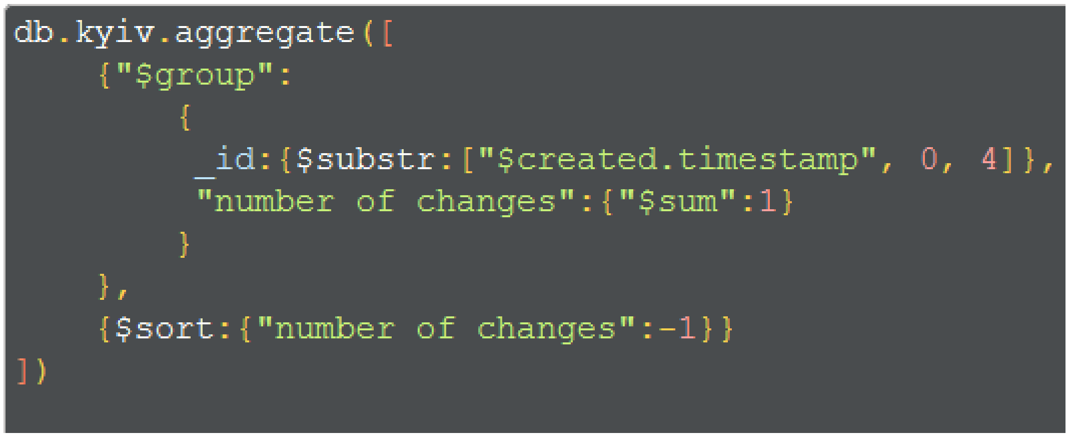
```
db.kyiv.aggregate([
    {"$group":
        {
         _id:"$created.user",
         number:{"$sum":1}
        }
    },
    {$sort:{number:-1}},
    {$limit:5},
    {"$group":
        {
            _id:"sum of first 5",
            total_number:{"$sum":"$number"}
        }
    }
])
```

In total, first 5 users with the largest contributions edited 405383 documents or about 35% of all changes. First 10 users about 50% and 50 users modified 80 % out of all documents. It is interesting to mention that 202 users edited only a single element.

Analysis of evolution

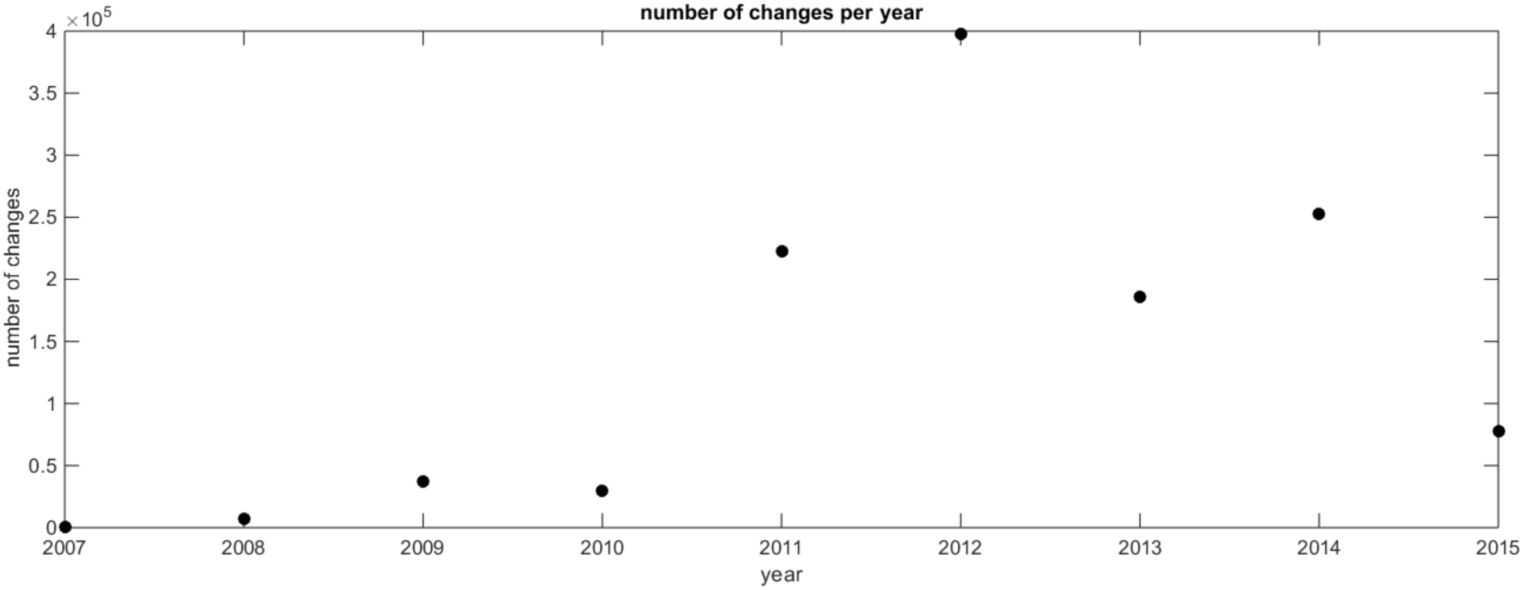
The next question we want to discuss is in which year the largest number of changes was made. It is an interesting question, because on can try to speculate that growth should be either linear, or it should decrease monotonically (excluding small fluctuations). We will show that evolution is not monotonical and try to explain this.

We use following query to show a list of changes made every year. (In principle, there is $year expression in MongoDB which returns the year portion of date, but timestamp is BSON string and this expression gives a mistake).



```
db.kyiv.aggregate([
    {"$group":
        {
         _id:{$substr:["$created.timestamp", 0, 4]},
         "number of changes":{"$sum":1}
        }
    },
    {$sort:{"number of changes":-1}}
])
```

We plot the distribution of changes according to year



One see that maximum of changes was done in 2012 (2015 is not finished yet ad we plot only a fraction made until the 1 of March). It should be noted that in 2012 Ukraine hosted UEFA Euro 2012 which definitely plays certain role for fact that 2012 has maximum of changes.

Some interesting facts

In addition to previous analysis we want to show some tables with interesting facts about Kyiv

Top 6 most appearing amenities.

Amenity	Parking	Bank	Pharmacy	Restaurant	Café	School
#	1439	765	558	496	495	479

Top 7 most popular cuisines.

Cousine	Regional	Pizza	Burger	Italian	Japanese	Coffee	Intnl
#	73	65	31	31	29	18	13

5 street with the largest number of objects

Street	Shevchenko	Franka	Khmelnitskogo	khreschatyk	**Lenina**
# of objects	230	165	165	136	116

Lenina street contains 116 objects and few weeks ago it was renamed.

7 possibilities to park a car

Parking	Surface	Underground	Garage_boxes	Multy-storey	Garage
#	466	37	28	16	8

Some important destinations for tourists

Name	Hotel	Museum	Viewpoint	Attraction	info	Artwork	hostel	Motel	Zoo
#	114	60	51	31	27	24	17	12	9

How many floors different building have

# of floors	5	1	9	2	16	3	4	10	12	6
# of objects	1344	1324	1251	866	441	330	273	162	114	90

The biggest number of levels is 31 (according to the map) and there is the only such building.

Additional ideas

I have two ideas how to improve the dataset. The first one is related to the issue discussed at the beginning of the report, where the Cyrillic fonts make impossible for foreign citizens get any understanding about the object on the map. The second idea is about the description of many objects which is still lacking.

Cyrillic fonts

. In addition to an element "name" also similar elements are introduced in OpenMap "name:en", "name:de" and so on. However, often none of these additional elements (most importantly "name:en") is filled. My idea is to use the methods of artificial intelligence which based on the information from the internet to find the most suitable translation. The most suitable tool, in my opinion, is the Bayesian methods nicely described in the Udacity Course
"Artificial intelligence for robotics"
https://www.udacity.com/course/artificial-intelligence-for-robotics--cs373

Description of objects

Finding the best possible translation one can go even further. It is possible to use the same methods and improve the description of every possible object on the map. It will require much more efforts, but the result will improve the quality of the map a lot.

In conclusion, I want to say, that I improve my understanding of data wrangling. Using the Python I cleaned the initial dataset and use MongoDB to analyze the obtained one. Finally, I give some additional ideas how to improve the dataset and resulting OpenMap data and listed some interested facts about the city Kyiv.

References

1. Mark Pilgrim. Dive Into Python. Case Study:Parsing Phone Numbers. http://www.diveintopython.net/regular_expressions/phone_numbers.html
2. Romanization of Ukrainian. http://en.wikipedia.org/wiki/Romanization_of_Ukrainian
3. Simple tools for processing strings in Russian. https://pypi.python.org/pypi/pytils
4. Can Google break the computer language barrier? http://www.theguardian.com/technology/2010/dec/19/google-translate-computers-languages

5. Ukraine Ditches Soviet Street Names: Ukrainian seek to distance nation from totalitarian past. http://uatoday.tv/society/some-of-ukraine-s-streets-named-after-lenin-are-to-be-renamed-390462.html
6. UEFA Euro 2012. http://en.wikipedia.org/wiki/UEFA_Euro_2012