Data wrangling with MongoDB

By Jaroslav Klen

Processed map

Chattanooga, Tennessee, US, dataset downloaded from mapzen

Files description

Chattanooga.osm (source dataset, size 117 MB)
Chattanooga.json (final dataset with indent = 2, size 195 MB)

Activities performed

1. Audit dataset
2. Restructure and clean dataset
3. Final dataset saved to JSON and MongoDB collection
4. MongoDB queries

Section 1: Audit and problems discovered

To get information about structure of the dataset and to identify problems, which are needed to be solved during cleaning process, I designed audit following:

General information about elements, which include:

Occurencies of each element

```
'bounds'        1
'member'        2834
'nd'            646096
'node'          579363
'osm'           1
'relation'      298
'tag'           276283
'way'           57067
```

Elements, which have children elements

```
'bounds'
        'without_children'    1
'member'
        'without_children'    2834
'nd'
        'without_children'    646096
'node'
        'with_children'       9553
        'without_children'    569810
'osm'
        'with_children'       1
'relation'
        'with_children'       296
        'without_children'    2
'tag'
        'without_children'    276283
'way'
        'with_children'       56989
        'without_children'    78
```

Elements, which reference other elements

```
'bounds'
        'without_ref'         1
'member'
        'without_ref'         2834
'nd'
        'without_ref'         646096
'node'
        'without_ref'         579363
'osm'
        'without_ref'         1
'relation'
        'without_ref'         298
'tag'
        'without_ref'         276283
'way'
        'with_ref'            56989
        'without_ref'         78
```

Exploration of 'tag' elements

This part of audit was focused on detailed examination of tag values content. As result, about each tag key we have got information how many values:

- Are only integer
- Are only float
- Are only text
- Are only alphanumeric
- Are in form of a sentence (includes more words separated with space)
- Contain street abbreviation or type in a sentence
- Contain possible phone or fax number
- Contain postcode
- Contain number as word at the beginning, in the middle or at the end of sentence
- Contain special characters

Neccesary was also to validate tag key, if it is suitable as key for MongoDB. Designing audit this way was a multi stage process, which together with examining distinct tag values or distinct words in sentences in problematic tags where it was possible, revealed most of common and rare inconsistencies, which in general include:

Wrong values

Some tags contained values, which should be paired with different tag. Example:

```
addr:housename = '406'
addr:housename = 'St. Elmo Ave'
name = '90'
```

Additional values

Some tags contained values, which included value as expected, but included also value, which should be placed in different tag. Example:

```
addr:street = '1812 Gunbarrel Road'
addr:housenumber = '103 Joyce Ave'
addr:street = '5728 Tennessee 58, Harrison, TN'
```

Incomplete values

Some tags contained values with incomplete information. Example:

```
addr:street  = '735 vine'
```

Other less serious problems include:

- Abbreviations

```
addr:street = 'Coffee Tree Ln'
addr:street = 'E 3rd St'
```

- Misspelings

```
addr:city = 'Chattannooga'
```

- Not titleized words

```
addr:city = 'red bank'
```

- Inconsistent formats

```
maxspeed = '20 mph', maxspeed = '40'
phone = '(423)899-4149', phone = '+1-423-396-9898'
```

- Unneccesary characters

```
addr:street = 'Market St #102'
```

Section 2: Fixing dataset

Before creating script which fixes mentioned problems, it was neccesary to design proper structure of document, which groups related information together[1], decide in which data type will be keys values stored, which tags do not contain important information[2] to leave them out and if it is neccesary to create new tags.[3]

The most challenging issue with this dataset, was to put the right address information into right tag, since addr:housename, addr:housenumber, addr:street, name, alt_name contained in a lot of cases part of address information, which should be paired with other address tags. Created logic for solving this problem was specific for this dataset and could be improved with some sort of street names validation from official or more relevant source, for example from cartographic.info website.

Section 3: MongoDB queries from final dataset

Top 5 most contributing users

```
[{'$group':{'_id':'$created.user',
                        'count':{'$sum':1}}},
          {'$sort':{'count':-1}},
          {'$limit':5}
]
```

```
[{u'_id': u'rjhale1971', u'count': 202349},
 {u'_id': u'T_9er', u'count': 52777},
 {u'_id': u'woodpeck_fixbot', u'count': 38683},
 {u'_id': u'ELadner', u'count': 27483},
 {u'_id': u'Thad C', u'count': 20755}]
```

How many users contributed

```
[{'$group':{'_id':'$created.user_id'}},
          {'$group':{'_id':'distinct_users',
                     'count':{'$sum':1}}}
]
```

```
[{u'_id': u'distinct_users', u'count': 622}]
```

Average elevation of grave yards

```
[{'$match':{'ele':{'$exists':1},
                     'amenity':'grave_yard'}},
          {'$group':{'_id':'grave yards average elevation',
                     'value':{'$avg':'$ele'}}}
]
```

```
[{u'_id': u'grave yards average elevation', u'value': 255.04954954954954}]
```

Top 3 streets, where is possible to drink or eat something

```
[{'$match':{'address.street':{'$exists':1},
                     '$or':[{'amenity':'pub'},
                            {'amenity':'bar'},
                            {'amenity':'restaurant'},
                            {'amenity':'cafe'},
                            {'amenity':'nightclub'},
                            {'amenity':'fast_food'}]}},
```

[1] For example tag keys, which begin with addr:, are TIGER, GNIS or NHD data, were put into separate dictionaries.

[2] I decided to process only node and way elements, since they contain for me most relevant information and created a list with tags to skip in processing, which are too rare or specific.

[3] Tag alt_name contained values, which represented street intersections closest to node. These values were put into new tag 'street_intersect' as a list of strings.

```
          {'$group':{'_id':'$address.street',
                     'count':{'$sum':1}}},
          {'$sort':{'count':-1}},
          {'$limit':3}
]
```

```
[{u'_id': u'Hamilton Place Boulevard', u'count': 12},
 {u'_id': u'Gunbarrel Road', u'count': 11},
 {u'_id': u'Brainerd Road', u'count': 10}]
```

Top 5 amenities

```
[{'$match':{'amenity':{'$exists':1}}},
          {'$group':{'_id':'$amenity',
                     'count':{'$sum':1}}},
          {'$sort':{'count':-1}},
          {'$limit':10}
]
```

```
[{u'_id': u'parking', u'count': 1220},
 {u'_id': u'place_of_worship', u'count': 533},
 {u'_id': u'school', u'count': 253},
 {u'_id': u'grave_yard', u'count': 223},
 {u'_id': u'restaurant', u'count': 186}]
```

Section 4: Summary and additional ideas

Creating audit and fixing procedures were created for this specific dataset and I was mainly focused on information related to address, where I found most errors and which I think was the most problematic part of whole project. Altough I think, that I discovered and fixed most of the problems, there are definitely space for improvements and more detailed examination. The way I solved mentioned problems were also influenced by size of this dataset, which was not very big, and I imagine, that with bigger datasets would be neccesary to use much more detailed approach in audit procedures and better way to organize fixing procedures, depending on the application of final data.

Except my experience when processing Chattanooga dataset I would like first to summarize interesting information from being novice contributor using iD editor and information from few research articles, OSM wiki and blogs about openstreetmaps, on which base I would like to propose gamification idea concept, which may improve contributors experience and openstreetmaps usage in general.

Users and data specifics

OSM is a collaborative and volunteered effort of people, which have special characteristics and motivation factors to contribute and to join volunteering communities. There are also differences in motivation within OSM community between serious and casual mappers, where the active ones are more oriented to community, learning, local knowledge and career, and casual, which are more oriented to general principles of free avalability of mappping data.[4] One way how to rise level of engagement and motivate to contribute with respect to mentioned motivation factors is gamification.

With user generated content of OSM comes also specifics of data which are users creating, when compared against authoritative datasets. According to few studies[5], which I encountered, quality of OSM data (thematic accuracy, positional accuracy, temporal accuracy, logical consistency and completness) is comparable to authoritative datasets in specific LU (land use) features and OSM could act as potential alternative data source for GIS applications using datasets created by professional mapping companies or national mapping agencies. Results of some researches

[4] http://abs.sagepub.com/content/57/5/548.abstract

[5] Interesting reading on this topic is book “OpenStreetMap in GIScience, Experiences, Research and Applications” from J. J. Arsanjani, Springer 2015

indicate, that populated areas, can reach higher positional accuracy and completness than authoritative datasets. These researches where conducted on specific areas and with different evaluation methods. Detailed information about quality of OSM data is still absenting, which is an obstacle to use the real potential of OSM and to wide spread usage. Specific examples of other obstacles include for example difference in coverage between urban and rural areas, sometimes unclear semantics, spatial and temporal difference in quality, more options to tag same feature, or one option to tag more features in some cases, and semantic heterogenity.

Gamification idea

Basic concept of the game "Tag the enemy" is following. OSM user will be added to a team according to his regional belongness. Each team has an assigned geographic area. Goal of each team is to tag enemy's geographic area and to survive as last in group of teams. The only team which lasts, wins. Certain types of tags, certain subareas, or tags made by experienced/novice users could be awarded for more or less points than others. Random factor could be included as well (more points for tagging in certain timeframe, areas locking/unlocking, ...) as upgrading system or other features, which will make it more entertaining. One way how to get approximate number of tags of each teams area, which should be tagged by the enemies, could be via created regression models with possible inputs from already tagged features from same area or surrounding areas and from other datasets, which provide socio-economic and demographic data.

With this game concept could be achieved better cooperation of users on regional basis, faster learning process of novice users, existing users will be motivated to bring new users to OSM community and motivation to tag "undertagged" areas.

Other ideas and suggestions:

- Basic QA, for example confirmation of created features from new user by experienced user,
- Scraping data from business registers,
- Cooperation with local tourism agencies,
- Finding area specific tags association rules and implement them in tag suggestion feature,
- Creating area specific audit procedures and implement them as "warning" feature