OpenStreetMap Project

Data Wrangling with MongoDB

Juho Salminen

Map area: Lahti, Finland
https://www.openstreetmap.org/relation/345901#map=11/60.9724/25.6888
Latitude: 60.8590, 61.0855
Longitude: 25.3146, 26.0630

Postal code search

Problems encountered in the map

The downloaded map data was first audited using audit.py script. The overall quality of the data appears to be good.

- No missing or non-integer user ids
- All timestamps are in correct format
- All longitudes and latitudes are floats
- All ways have at least two node references
- Street names do not contain abbreviations
- House numbers contain only numbers

```
missing uids: 0
wrong ids: 0
wrong timestamps: 0
wrong longitude or latitude: 0
ways with less than 2 node references: 0
number of street abbreviations: 0
set()
number of unique tag names: 425
number of wrong postcodes: 27
['FI-15150', 'FI-15870 HOLLOLA', 'FI-15870  HOLLOLA', 'FI-15870
HOLLOLA', 'FI-15170 LAHTI', 'FI-15170 LAHTI', 'FI-15170 LAHTI', 'FI-
15170 LAHTI', 'FI-15870 HOLLOLA', 'FI-15870', 'FI-15870', 'FI-15170
LAHTI', 'FI-15870 HOLLOLA', 'FI-15870 HOLLOLA', 'FI-15150', 'FI-
15870 HOLLOLA', 'FI-15870 HOLLOLA', 'FI-15870 HOLLOLA', 'FI-15870
HOLLOLA', 'FI-15870 HOLLOLA', 'FI-15170 LAHTI', 'FI-15170 LAHTI',
'FI-15170 LAHTI', 'FI-15170', 'FI-15170 LAHTI', 'FI-15170',
'Villähde']
Cities:
{'Orimattila', 'HOLLOLA', 'Villähde', 'Lahti', 'Pennala', 'Hollola',
'Messilä', 'LAHTIS', 'Nastola'}
number of weird housenumbers: 0
[]
unexpected street names:
['Lahti', 'Lahti', 'Kokkokalliokantu', 'Pasaasi']
```

The audit still revealed a few issues in the data:

- Inconsistent postal codes.
- Inconsistencies in city names.
- A few unexpected street names.

Postal codes

The postal codes were mostly correct. They match the pattern of five numbers starting with 15, 16 or 17, as in 15140. Still 27 postal codes in the dataset contained excessive information, for example 'FI-15870 HOLLOLA'. The script used for reshaping the data for input into MongoDB (openstreetmap_munging.py) stripped the extra information from the postal codes so that only the five numbers were left. In one case the postal code contained the city name. This was corrected manually in MongoDB by adding the correct postal code for the address based on postal code search service.

```
> db.lahti.update({'address.postcode': 'Villähde'}, {'$set':
{'address.postcode': '15540'}})
```

Cities

Sometimes the names are on all capital letters, and both Finnish and Swedish name for Lahti is used. Using openstreetmap_munging.py, all city names were transformed to lowercase except for the initial character. Swedish city name was translated to Finnish.

Street names

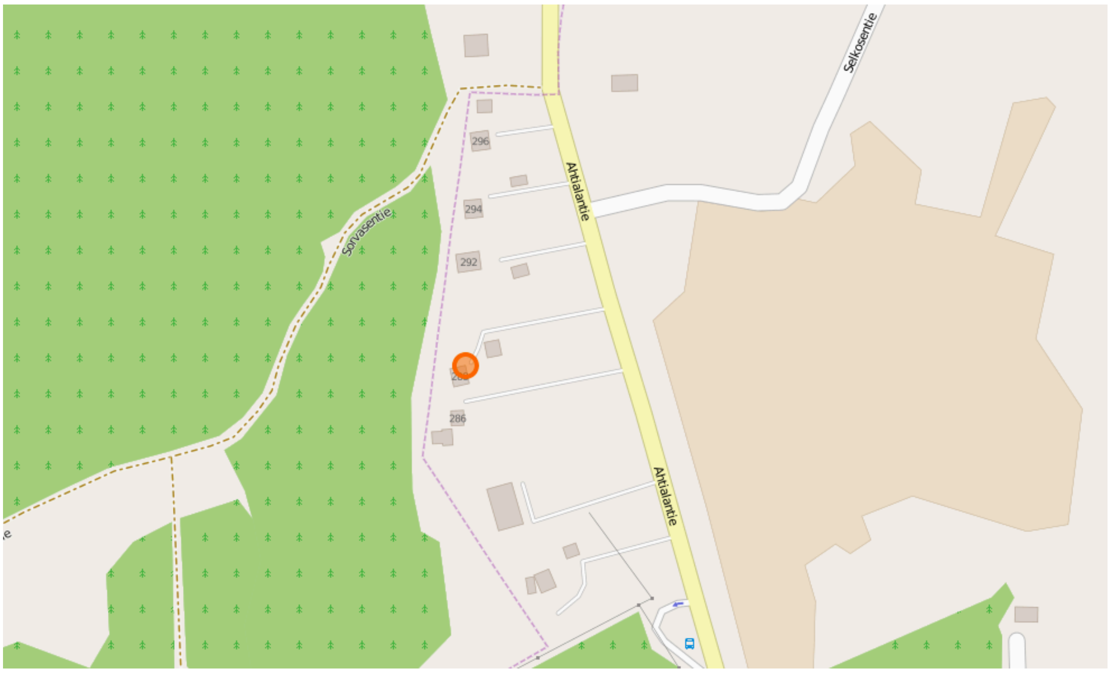
Four unexpected street names turned up during the audit. One of them was actually a correct street name and no further action was taken in that case. The rest of the mistakes were corrected manually in MongoDB.

```
> db.lahti.update({'address.street': 'Kokkokalliokantu'}, {'$set':
{'address.street': 'Kokkokallionkatu'}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```

Typo in the street name is now fixed.

```
> db.lahti.find({'address.street': 'Lahti'})
{ "_id" : ObjectId("558ac7c5ed60c86a715ccc0c"), "id" : "1285727844",
"created" : { "user" : "MakeLa", "changeset" : "8147983",
"timestamp" : "2011-05-15T06:41:58Z", "version" : "1", "uid" :
"454444" }, "type" : "node", "address" : { "housenumber" : "2",
"street" : "Lahti", "postcode" : "15320" }, "pos" : [ 61.0090216,
25.77853 ] }
{ "_id" : ObjectId("558ac7c8ed60c86a715e98c4"), "node_refs" : [
"588517829", "588517831", "588517832", "588517830", "588517829" ],
"id" : "46158927", "created" : { "user" : "MakeLa", "changeset" :
"8182231", "timestamp" : "2011-05-18T15:47:15Z", "version" : "2",
"uid" : "454444" }, "type" : "way", "address" : { "housenumber" :
"288", "street" : "Lahti", "postcode" : "15340" } }
```

The correct addresses for the two entries with city names in street name fields required more research. Eventually they could be corrected by checking the location of the reference nodes on OpenStreetmap and verifying the information on Google Maps. The information was updated manually on MongoDB.



Location of node 588517831. The location of the other mistaken address was identified similarly. Correct addresses were updated to MongoDB. Problems found in street names have now been fixed.

```
> db.lahti.update({'address.street': 'Lahti', 'address.postcode':
'15340'}, {'$set': {'address.street': 'Ahtialantie', 'address.city':
'Lahti'}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.lahti.update({'address.street': 'Lahti', 'address.postcode':
'15320'}, {'$set': {'address.street': 'Päivänsäteenkatu',
'address.city': 'Lahti'}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```

Verifying the data

The quality of the data after the fixes was confirmed with the following queries to the database.

All postcodes in the database match the correct format.

```
> db.lahti.find({'address.postcode': {'$exists': 1}}).count()
852
> db.lahti.find({'address.postcode': {'$regex': '(15|16|17)[0-
9]{3}'}}).count()
852
```

Problematic street names cannot be found anymore.

```
> db.lahti.find({'address.street': 'Lahti'}).count()
0
> db.lahti.find({'address.street': 'Kokkokalliokantu'}).count()
0
```

City names found in the database are consistent.

```
> db.lahti.distinct('address.city')
[
    "Lahti",
    "Messilä",
    "Orimattila",
    "Nastola",
    "Hollola",
    "Pennala",
    "Villähde"
]
```

Data overview

This section contains the basic statistics about the dataset.

lahti_finland.osm	96.5 MB
lahti_finland.osm.json	108.7 MB

Number of documents

```
> db.lahti.find().count()
507601
```

Number of nodes

```
> db.lahti.find({'type': 'node'}).count()
453135
```

Number of ways

```
> db.lahti.find({'type': 'way'}).count()
54466
```

Number of unique users

```
> db.lahti.distinct('created.uid').length
241
```

Top contributing users

The most active user:

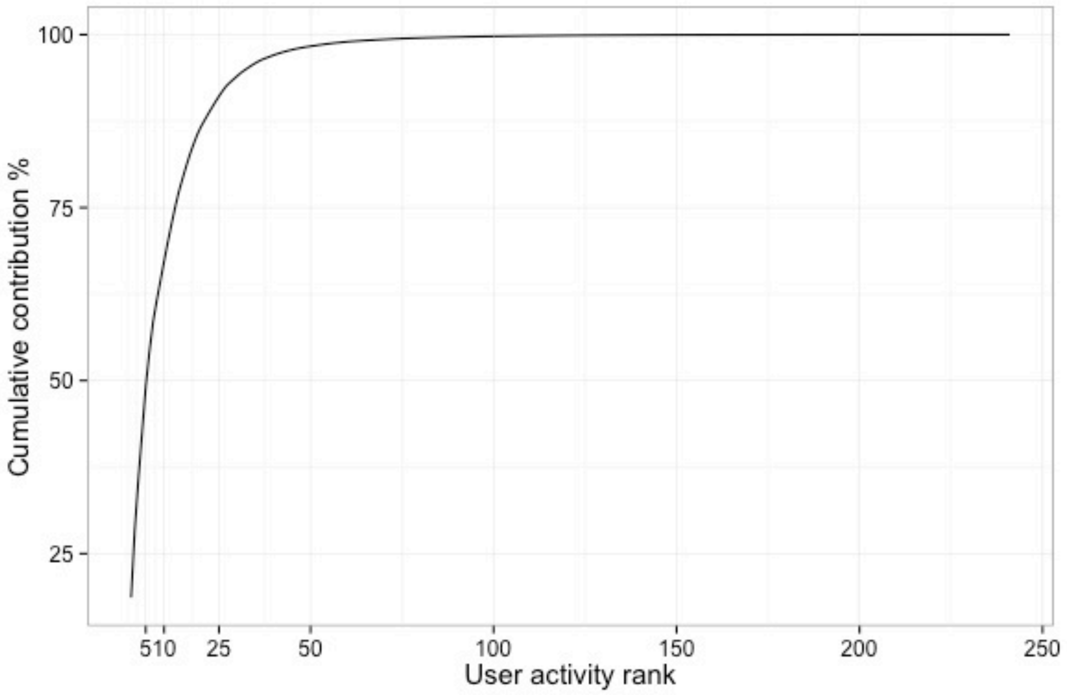
```
> db.lahti.aggregate([{'$group': {'_id': '$created.user', 'count':
{'$sum': 1}}},{'$sort': {'count': -1}}, {'$limit': 1}])
{ "_id" : "Daeron", "count" : 94630 }
```

Investigating the distribution of contribution among the users. First, user contribution counts were exported as a json document using the following script and shell command.

```
cursor = db.lahti.aggregate([{'$group': {'_id': '$created.user',
'count': {'$sum':1}}}, {'$sort': {'count': -1}}])
while(cursor.hasNext()){
      printjson(cursor.next());
}

mongo datascience mongo1.js > result.json
```

Resulting json file was loaded to R and then the following graph was created.



This figure shows that already the 5 most active users contributed about half of the content (49.1 %) and top 25 most active users take the cumulative contributions to over 90 % of the total.

Number of users appearing only once

```
> db.lahti.aggregate([{"$group":{"_id":"$created.user",
"count":{"$sum":1}}}, {"$group":{"_id":"$count",
"num_users":{"$sum":1}}}, {"$sort":{"_id":1}}, {"$limit":1}])
{ "_id" : 1, "num_users" : 44 }
```

Additional ideas

Contributions to nodes vs. ways

Are users contributing differently to nodes and ways? Do some users specialize?

```
> db.lahti.aggregate([{'$match': {'type': 'node'}},{'$group':
{'_id': '$created.user', 'count': {'$sum': 1}}},{'$sort': {'count':
-1}}, {'$limit': 10}])
{ "_id" : "Daeron", "count" : 85221 }
{ "_id" : "kallam", "count" : 42921 }
{ "_id" : "Nikerabbit", "count" : 33263 }
{ "_id" : "diibadaaba", "count" : 30741 }
{ "_id" : "ij_", "count" : 27724 }
{ "_id" : "jleh_nlsfi_import", "count" : 24467 }
{ "_id" : "Medar", "count" : 20580 }
{ "_id" : "teollisuus", "count" : 14135 }
{ "_id" : "geosapiens_nlsfi_import", "count" : 13640 }
{ "_id" : "JRA2", "count" : 12554 }

> db.lahti.aggregate([{'$match': {'type': 'way'}},{'$group': {'_id':
'$created.user', 'count': {'$sum': 1}}},{'$sort': {'count': -1}},
{'$limit': 10}])
{ "_id" : "Daeron", "count" : 9409 }
{ "_id" : "kallam", "count" : 7499 }
{ "_id" : "ij_", "count" : 5775 }
{ "_id" : "Nikerabbit", "count" : 3992 }
{ "_id" : "diibadaaba", "count" : 3010 }
{ "_id" : "JRA2", "count" : 2844 }
{ "_id" : "jleh", "count" : 2326 }
{ "_id" : "Medar", "count" : 1316 }
{ "_id" : "jleh_nlsfi_import", "count" : 1200 }
{ "_id" : "Geosapiens", "count" : 1105 }
```

Rankings of top contributing users to nodes and ways are similar. Most of the contributions are nodes: none of the most active users contributed more ways than nodes. Two users appear to have secondary user names ending with _nlsfi_import. These are likely to refer to National Land Survey of Finland, which recently made lots of map data publicly available. This data source could be used to improve the quality of the OpenStreetMaps in other parts of the Finland, or by adding more information to Lahti area. Available data includes elevation models, laser scanning data, topographic map rasters, road and place names and aerial photographs. With over 200 years of experience in land surveying and safeguarding land ownership, the data from NLS can be expected to be accurate and up to date. The main challenge in using this data is likely to be the compatibility of data formats. NLS offers the data in Esri shape and MapInfo MIF formats and topographic data in MAAGIS/XL and GML formats. These formats may not work directly with OpenStreetMaps, and as a result some serious data munging might be necessary. Ready-made scripts could still be available, as other people have already imported data from NLS to OpenStreetMaps.

Most common amenities

```
> db.lahti.aggregate([{'$match': {'amenity': {'$exists': 1}}},
{'$group': {'_id': '$amenity', 'count': {'$sum': 1}}}, {'$sort':
{'count': -1}}, {'$limit': 10}])

{ "_id" : "parking", "count" : 630 }
{ "_id" : "waste_basket", "count" : 61 }
{ "_id" : "restaurant", "count" : 55 }
{ "_id" : "school", "count" : 50 }
{ "_id" : "pub", "count" : 50 }
{ "_id" : "post_box", "count" : 41 }
{ "_id" : "fast_food", "count" : 33 }
{ "_id" : "fuel", "count" : 31 }
{ "_id" : "bench", "count" : 31 }
{ "_id" : "kindergarten", "count" : 30 }
```

Querying the database for some of the most common amenities listed above reveals that information available on them is inconsistent. For most amenities only the very basic information is listed, such as street address and name, while others have also opening times, phone numbers and websites. It should be possible to complement the dataset by adding information on amenities from some other source. Unfortunately information on companies is often behind a paywall, or at least not available through a public API that could be used programmatically. Additionally copyrights might apply, making the data incompatible with Creative Commons license used by OpenStreetMaps. Care should be taken to correctly match amenities and information extracted from elsewhere. Many amenities have same or similar names, and for instance confusing locations of different units of a retail chain are a risk.

Most common services

```
> db.lahti.aggregate([{'$match': {'service': {'$exists': 1}}},
{'$group': {'_id': '$service', 'count': {'$sum': 1}}}, {'$sort':
{'count': -1}}, {'$limit': 10}])

{ "_id" : "driveway", "count" : 1412 }
{ "_id" : "parking_aisle", "count" : 541 }
{ "_id" : "spur", "count" : 27 }
{ "_id" : "yard", "count" : 20 }
{ "_id" : "crossover", "count" : 9 }
{ "_id" : "alley", "count" : 3 }
{ "_id" : "emergency_access", "count" : 2 }
{ "_id" : "siding", "count" : 2 }
{ "_id" : "dealer;parts;repair", "count" : 1 }
{ "_id" : "drive-through", "count" : 1 }
```

Against my assumption services in OpenStreetMap (at least in this area) refer mostly to roads and parking instead of restaurants and dry cleaners.

Most common buildings

```
db.lahti.aggregate([{'$match': {'building': {'$exists': 1}}},
{'$group': {'_id': '$building', 'count': {'$sum': 1}}}, {'$sort':
{'count': -1}}, {'$limit': 10}])
{ "_id" : "yes", "count" : 26235 }
{ "_id" : "residential", "count" : 257 }
{ "_id" : "house", "count" : 69 }
{ "_id" : "industrial", "count" : 47 }
{ "_id" : "apartments", "count" : 45 }
{ "_id" : "garage", "count" : 20 }
{ "_id" : "canopy", "count" : 19 }
{ "_id" : "roof", "count" : 13 }
{ "_id" : "school", "count" : 9 }
{ "_id" : "terrace", "count" : 9 }
```

Apart from the first item on the list the results are expected. Perhaps “yes” is used as a placeholder when it is known that there is a building, but the exact type of the building is unknown. Adding information on building types could be a way to improve the quality of OpenStreetMap data. However, finding a suitable data source could be difficult.