Project 3: Wrangle OpenStreetMaps Data

by Thomas Dräbing

In this project the map data of Hamburg (Germany) from the OpenStreetMaps database is wrangled. The aim is to create a MongoDB database containing the map data of Hamburg in a clean and corrected state. The used data can be found under the following url: https://s3.amazonaws.com/metro-extracts.mapzen.com/hamburg_germany.osm.bz2. The link to the corresponding area on openstreetmap.org is: https://www.openstreetmap.org/relation/2618040.

Problems encountered in the data

Within the scope of this project the address data will be given a closer look. There is a total of 2236 distinct keys in the data. To check them all for mistakes and inconsistencies would be out of scope.

Even in the small subset of data types a multitude of problems emerged:

- Keys containing the term 'fixme' exist, indicating that there are at least partly incorrect documents in the database.
- Not all the nodes, ways or relations are situated in Hamburg. This is probably due to the way the map subset was created by mapzen.com.
- The values for the states are sometimes abbreviated or in English instead of German.
- Most of the time the state is not given.
- There are postcodes, which do not exist.
- Postcodes may not fit the given city or the city is named inconsistently.
- The street names seem to be mostly clean. Only a few contain easy to find errors.
- There are some house numbers not existing.
- There is a high inconsistency in which format the house number is given.

'Fixme' –entries

Inspecting the values of fields named 'fixme' or with a similar term, it became clear that fixing the data would most of the time need one to be in Hamburg itself or extensive individual care. Thus for now the corresponding documents were moved to their own collection. Thus the data is still available in the database, but easier to be distinguished as unfinished data.

States

All states were renamed to their full-length German Names. (e.g. 'NS' or 'Lower Saxony' -> 'Niedersachsen'). Since the dataset is rather huge and the main interest was to build a database of Hamburg, all entries not situated directly in Hamburg were moved to another collection. This was comparably easy, since Hamburg is a city-state. After doing this, it became apparent that most of the time no value for the state was given, since barely any document was moved. Thus postcodes and cities were also used to select for entries not representing a place in Hamburg.

Post Codes

Although Hamburg is one big city it consists of several districts, thus we expect several post codes. But some post districts in the data are probably not part of Hamburg, as stated above. Thus the post codes were cross referenced with the google maps API to check in which state the respective post district can be found. In case that the state was not Hamburg, the document was moved to the collection containing data of other states.

The post code '22701' does not exist. Using the remaining address data, the correct post code (22765) was found using Google Maps and changed in the database.

Cities

The city-field contains several distinct values. Some are the different districts of Hamburg, but some are cities not actually belonging to Hamburg, thus being part of another state. The respective documents were again moved to another collection. City names containing problem chars were renamed. Additionally the post code and city were crosschecked using the Google Maps API, to investigate whether both values are assigned correctly. If this was not the case, the post code of Google maps was used instead.

Streets

The street data seems to be very clean. Four entries were manually cleaned since they contained latin numbers or information not belonging in this field. In Germany there is just one official abbreviation for street types and that is 'Str.' for 'Straße' ('street'). Searching the data, no abbreviated entry could be found. The analysis here does not investigate several other possible errors, like typos, variations in writing or not existing streets. Also on the street level it is rather difficult to check, whether it is situated in Hamburg, when neither state, post code nor city is given, since street names are often redundant.

House Numbers

The house numbers in the raw data set are represented in several different formats. Regular expressions were used to detect those and transform them into a more standardized format. Separation symbols were limited to commas and hyphens. Letters were capitalized, the use of spaces was standardized and some other changes were performed. For some little represented forms this was done manually (less than 40 out of more than 7000). Three house numbers were no house numbers at all and were set to *None*.

Data Overview

A few basic statistics were taken from the data used in this project:

Size of osm-file:	1.10 Gb
Size of JSON-file:	1.33 Gb
Size of audited database:	1.63 Gb
Number of documents:	5.822.728
Number of nodes:	4.985.068
Number of ways:	820.642
Number of relations:	687
Number of unique users:	4.344
Number of users posted once:	795

Additionally queries were performed on the data. For example the number of pizza places or the most common construction year was queried:

Number of pizza places: 83

Top 10 building years:

Construction Year	# count
None given	5822629
1840	13
1843	7
1832	6
1783	6
1802	5
1889	5
1886	5

Apparently users mostly mention construction buildings for old buildings.

Other ideas about the dataset

There are some more problems, which were so far not addressed. One of the most urgent would be, that a lot of fields seem to be redundant. A good example already audited would be the fixme – fields. Those should be aggregated. There are also a lot of keys containing colons, indicating the existence of subfields that should be represented accordingly in MongoDB syntax.

The database could be used to search for the closest food places using the longitude / latitude array, at the same time giving additional information like opening times.