Data Wrangle OpenStreetMaps Data

Udacity Data Analyst Nanodegree – Project 2

Zhanzhan He

June 2015

Contents

0. References

mackerski. (2007, February 28). Retrieved from OpenStreetMap: https://www.openstreetmap.org/user/mackerski

OSM Contributers. (2015). *Overpass Turbo*. Retrieved June 2015, from http://overpass-turbo.eu/

Postcodes in the United Kingdom. (2015, May). Retrieved June 1, 2015, from Wikipedia: http://en.wikipedia.org/wiki/Postcodes_in_the_United_Kingdom#Outward_code

Spinellis, D., & Louridas, P. (2008). The Collaborative Organization of Knowledge. *Communications of the ACM*, 68-73.

1. Dataset

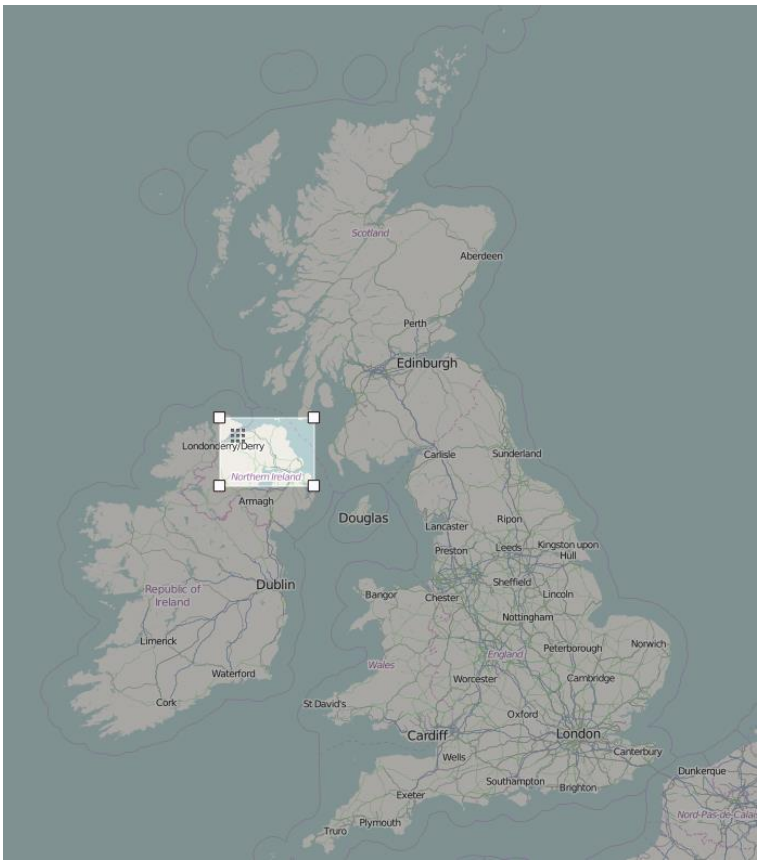

Figure 1 – Bounding box of the North Coast Dataset

Min Latitude	Min Longitude	Max Latitude	Max Longitude
54.5163	-7.3856	55.3198	-5.4794

The dataset examined includes the North Coast of Northern Ireland, where I grew up as well as the major cities of Belfast and Londonderry. The data was downloaded from the Overpass API (OSM Contributers, 2015) using a custom bounding box (Figure 1).

2. Problems Encountered

audit.py contains the code used to flag up the problems. Some of these are fixed automatically in *data.py*.

2.1 Street type

Street	Avenue	Boulevard	Drive	Court	Place
Square	Lane	Road	Trail	Parkway	Commons
Crescent	Dale	Glen	Green	Gardens	Walk
Way	Park	Parade	Manor	Meadow	Lodge
Heights	Grove	Close	Quay		

The street type audit was initially done with the set of street types highlighted in yellow treated as valid. However I found there were other street types (highlighted in green) that were also correct and added these to the valid set. Some of these are terms, e.g. 'Glen' are unusual outside of Ireland.

2.1.1 Inconsistent capitalization

There were instances where street types started with a lower case letter. These were replaced with the capitalized version, e.g. gardens -> Gardens

2.1.2 Over-abbreviation

Street names that were overly abbreviated were expanded, e.g. Rd-> Road

2.1.3 Single word street names

It isn't uncommon to have single word street names like 'Briarhill', 'Mainebank' and 'Meadowvale'. These have no clear pattern to audit automatically so I excluded them from the audit program.

2.1.4 Spelling errors

The audit program flagged up what appear to be spelling errors such as 'Garland Hil' and 'Marlborough Hheights'. These would be best checked against a standard dictionary and another dataset before fixing them as sometimes archaic words and spellings occur in place names.

2.2. Postcodes

UK postcodes are made up of an outward code and an inward code. These can each be any sequence of letters and numbers (Wikipedia, 2015). They are usually separated by a single space when written, e.g. SW11 1TN. A regex was added to audit.py to check postcodes conform to this format and postcodes were found with leading and trailing whitespace. Trimming of whitespace was then added for values in *data.py*.

3. Data Overview

data.py was used to process the osm file. The data was imported to mongodb using the command:

```
mongoimport --collection nc northcoast.osm.json
```

3.1 File sizes

northcoast.osm	190MB
northcoast.osm.json	220MB

3.2 Total Documents

```
> db.nc.count()
1024972
```

3.3 Number of Ways

```
> db.nc.count({'type':'way'})
79660
```

3.4 Number of Nodes

```
> db.nc.count({'type':'node'})
945297
```

3.5 Number of Unique Users

```
db.nc.distinct('created.user').length
669
```

3.6 Number of Cafes

```
> db.nc.count({amenity: 'cafe'})
92
```

3.7 Number of Restaurants

```
> db.nc.count({amenity: 'restaurant'})
134
```

3.8 User Contributions

3.8.1 Total User Contributions

```
> db.nc.count({'created.user': {'$exists': true}})
1024972
```

3.8.3 Top 5 User Contributions

```
> db.nc.aggregate([
      {'$group': {'_id': '$created.user', 'count': {'$sum': 1}}},
      {'$sort': {'count': -1}},
      {'$limit': 5}
])
{ "_id" : "Stephen_Co_Antrim", "count" : 390577 }
{ "_id" : "mackerski", "count" : 141274 }
{ "_id" : "Steve_NI", "count" : 84255 }
{ "_id" : "KDDA", "count" : 64082 }
{ "_id" : "Warofdreams", "count" : 43276 }
```

3.8.3 Percentage user contributions

```
> db.nc.aggregate([
      {'$group': {'_id': '$created.user', 'count': {'$sum': 1}}},
      {'$project': {
            'percentage': {
                  '$multiply': [
                        {'$divide': ['$count', 1024972]}, 100
                  ]
            }
      }},
      {'$sort': {'percentage': -1}},
      {'$limit': 5}
])
```

User	% Contribution
Stephen_Co_Antrim	38.1
mackerski	13.8
Steve_NI	8.2
KDDA	6.3
Warofdreams	4.2

The user contributions show that this area has benefited relatively little from automated editing. A quick look at the public profiles of the top 20 users shows primarily individual mapping enthusiasts who do GPS logging and manual editing, though the top contributions have involved importing data from other datasets (mackerski, 2007).

4. Additional Ideas

It might be possible to increase the amount and quality of OpenStreetMap data by making contributions easier for non-technical users. There are smartphone apps designed for OSM

contributors, but an option to contribute anonymous data to help improve OSM. Raw GPS traces as contributions, which are not put directly on the map but are used by contributors to add new routes and improve the accuracy of existing ones.

This trace data could be used to highlight areas where data is missing or of low quality, for example sections of the map with many contributed GPS traces but few corresponding nodes and ways. These could be marked as areas needing attention in a similar manner to Wikipedia's red links which helped drive Wikipedia's growth (Spinellis & Louridas, 2008).

5 Additional Data Exploration

5.1 Most frequent amenities

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

Amenity	Frequency
parking	1162
place_of_worship	427
school	253
pub	163
fast_food	156
fuel	149
restaurant	134
café	92
post_box	82
atm	70

5.2 Biggest Fast Food Chains

```
db.nc.aggregate([
      {'$match': {'amenity': {'$eq': 'fast_food'}, 'name': {'$exists': true}}},
      {'$group': {'_id': '$name', 'count': {'$sum': 1}}},
      {'$sort': {'count': -1}},
      {'$limit': 5}
])
```

Restaurant	Frequency
KFC	11
McDonald's	11
Burger King	6
Subway	5

This query revealed problems requiring further cleaning. There currently 14 fast food restaurants marked on the map which don't have a corresponding name. Also McDonalds is punctuated in 2 different ways, skewing the results.

5.3 Most frequent restaurants and cafes by cuisine

```
db.nc.aggregate([
      {'$match': {'$or': [
            {'amenity': {'$eq': 'restaurant'}, 'cuisine': {'$exists': true}},
```

```
        {'amenity': {'$eq': 'cafe'}, 'cuisine': {'$exists': true}},
    ]}},
    {'$group': {'_id': '$cuisine', 'count': {'$sum': 1}}},
    {'$sort': {'count': -1}},
    {'$limit': 5}
])
```

Cuisine	Frequency
coffee_shop	19
regional	8
sandwich	8
chinese	7
pizza	6
indian	5
american	4
international	3