OpenStreetMap Data Case Study

Map Area

Newcastle, United Kingdom

- http://overpass-api.de/api/map?bbox=-2.0009,54.9035,-1.3033,55.1349 (http://overpass-api.de/api/map?bbox=-2.0009,54.9035,-1.3033,55.1349)

This city is where my University is located at. I missed the city so much and would like this chance to revisit Newcastle again but in OpenStreetMap.

I have downloaded the original Newcastle osm file which is 150mb and created a sample data to meet the 10mb file size requirement. After creating the sample data, several scripts will be ran to explore, clean and analyse the data.

Problems Encountered in the Map

I have downloaded the original Newcastle osm file and created a sample data to meet the 10mb file size requirement. After creating the sample data, several scripts will be ran to explore, clean and analyse the data.

- Shortened street names (“Chapel House Dr”)

Content

- Map Parser
- Tags Assigned
- Users
- Audit
- Data Conversion
- Data Overview (SQL)
- Data Exploration

The python codes are removed from this printed pdf to meet the requirement of maximum 6 pages.

Map Parser

Map Parser script is just to see what tags there are and also how many of each tag.

```
{'member': 5984,
 'nd': 46931,
 'node': 38368,
 'osm': 1,
 'relation': 29,
 'tag': 22832,
 'way': 6409}
```

Tags Assigned

2_tags.py checks what kind of tags there are and uses regex to check whether there are any problems with the tags.

Output:

```
highway
created_by
created_by
created_by
highway
highway
amenity
name
...
direction
highway
name
created_by
created_by
highway
name
ref
created_by


{'lower': 18438, 'lower_colon': 1949, 'other': 2445, 'problemchars': 0}
```

Users

3_users.py will find out how many unique users that have contributed to this map.

Output:

```
set(['-Matt-',
     'AAEmmerson',
     'AJGUY94',
     'AMB',
     'AcousticNewt',
     'Adam Boardman',
     'AdamantUK',
     'Al_K',
     'Aled',
     'Alex McKee',
     ...
     'Alexander92',
     'Alpin100',
     'Amaroussi',
     'AndiBing',
     'Anti-Distinctlyminty',
     'ArnHH',
     'BCNorwich',
     'B_i_B',
     'Ben',
     'Benfll'])
```

Audit

By auditting the data, we can solve the issue of shortened street names. It will be better if the street names are all in standardised format. This will also write a new XML file named "audit.osm". This XML file will be the auditted file that has all the street names standardised. Problem: e.g. Shortened street names (“Chapel House Dr”)

Output:

```
{'4': set(['Cross Villa Place No 4']),
 'Aenue': set(['Fellgate Aenue']),
 'Approach': set(['Western Approach']),
 'Ashgill': set(['Ashgill']),
 'B1303': set(['Station Road          B1303']),
 'Bank': set(['Blaydon Bank',
              'Bottle Bank',
              'Chowdene Bank',
              'Dog Bank',
              'Long Bank',
              'Rectory Bank',
 ...
 'Cycleway': set(['John Reid Road Cycleway']),
 'Dam': set(['Mill Dam']),
 'Dr': set(['Chapel House Dr']),
 'Earlsway': set(['Earlsway']),
 'East': set(['Front Street East',
 'Harton House Road East',
 'Market Street East',
 'Victoria Road East'])
 ...}

Old George Yard => Yard
Meresyde => Meresyde
Mariners Wharf => Wharf
Chapel House Dr => Drive
St. Bede Wynd => Wynd
Chowdene Bank => Bank
Bottle Bank => Bank
Stepney Bank => Bank
Whiteside Bank => Bank
```

Data Conversion

5_data.py will allow the XML data format to be converted into 5 CSV format files. The CSV files will then be able to import into SQL database.

Data Overview (SQL)

Files:

- osm.db (96mb)

    ```
    Tables
    ```

    - nodes
    - nodes_tags
    - ways
    - ways_nodes
    - ways_tags

- nodes.csv (53mb)
- nodes_tags.csv (4.5mb)
- ways.csv (6.5mb)
- ways_nodes.csv (19.6mb)
- ways_tags.csv (9.6mb)

Data Exploration (SQL)

Number of nodes

```
SELECT COUNT(*) FROM nodes;
```

```
652248
```

Number of ways

```
SELECT COUNT(*) FROM ways;
```

```
108948
```

Number of unique users

```
SELECT COUNT(DISTINCT(e.uid))
FROM (SELECT uid FROM nodes UNION ALL SELECT uid FROM ways) e;
```

```
750
```

Top 10 contributing users

```
SELECT e.user, COUNT(*) as num
FROM (SELECT user FROM nodes UNION ALL SELECT user FROM ways) e
GROUP BY e.user
ORDER BY num DESC
LIMIT 10;
```

```
"James Derrick",319876
SkaBook,58585
GrahamS,46104
bigalxyz123,30371
LeedsTracker,27340
INeilC,24359
Rydium,18202
CreakyBike,12290
UniEagle,10768
"Chris Parker",10730
```

Number of users post less than 10 times

```
SELECT COUNT(*)
FROM
    (SELECT e.user, COUNT(*) as num
     FROM (SELECT user FROM nodes UNION ALL SELECT user FROM ways) e
     GROUP BY e.user
     HAVING num<10)  u;
```

```
323
```

Top 10 appearing amenities

```
SELECT value, COUNT(*) as num
FROM nodes_tags
WHERE key='amenity'
GROUP BY value
ORDER BY num DESC
LIMIT 10;
```

```
waste_basket,234
bench,174
telephone,110
post_box,83
parking,59
atm,37
post_office,36
toilets,25
bicycle_parking,23
place_of_worship,19
```

Top 4 Religion

```
SELECT value FROM nodes_tags
WHERE nodes_tags.key='religion'
GROUP BY nodes_tags.value
LIMIT 4;
```

1. christian
2. jewish
3. multifaith
4. muslim

Types of Food

```
SELECT nodes_tags.value, COUNT(*) as num
FROM nodes_tags
WHERE nodes_tags.key='cuisine'
GROUP BY nodes_tags.value
ORDER BY num DESC;
```

```
indian,7
sandwich,5
chinese,4
fish_and_chips,3
regional,2
```

Conclusion

From the analysis of the dataset, there are a wide range of values from the tags in the XML files of Newcastle. By sampling the size of the dataset, the values or analysis coming from this analysis will not be significant due to the wide range of values. One example will be the Types of Food analysis where the general search of the key "cuisines" only returned a small number of results for the top cuisine.

The data retrieved from OpenStreetMap can be done better by having more precise shapes of the countries or cities.