Nanodegree Project 3 - Data Wrangle OpenStreetMaps Data

Mateusz Stankiewicz

Map Area: Poznań, Poland, Europe

OSM Link - Poznan

Problems Encountered with the map

After processing the xml file with parse_into_json.py, I have noticed two main issues with the data file:

1. Polish diacritics in street names
2. Street names contained house numbers

All other issues were solved by code used in Lesson 6.

Polish diacrtitics in street names

Polish names usually contain characters with diacritics (like ą,ś,ć,ź), which are well represented using UTF-8 encoding, but didn't translate well when uploaded to MongoDB. I have decided to remove the diacritics and leave only the character i.e. ą became a. The code responsible for change is available in *parse_into_json.py:change_diacritics()*.

Street names contain house numbers in name

Some of less common streets contained house numbers in their addr:street tags. Filtering them out was mainly manual labor, because I could not detect any pattern - some names contain only one word (i.e. Strzeszynska) or multiple (i.e. Osiedle Edwarda Raczynskiego) or combination of numbers and letters (i.e. 11 Listopada). Also, housenumbers are also very various ranging from simple numbers to numbers with dividers (6/8) or combination of numbers, letters and dividers (i.e. 1A/8 or 6/4B). Fortunately it was a very small subset containing about 40 elements to change.

Data Overview

Poznan dataset:

- poznan_poland.osm: 163MB
- poznan_poland.osm.json: 177MB

Initializing the DB connection:

```
client = pymongo.MongoClient("mongodb://localhost:27017")
db = client.maps
poznan = db.poznan
```

Number of documents:

```
poznan.find().count()
```

```
795240
```

Number of nodes:

```
poznan.find({"type": "node"}).count()
```

```
694626
```

Number of ways:

```
poznan.find({"type": "way"}).count()
```

```
100614
```

Number of distinct users:

```
len(poznan.distinct("created.user"))
```

```
708
```

Top 3 contributing users:

```
poznan.aggregate([
        {"$group": {"_id": "$created.user", "count": {"$sum": 1}}},
        {"$sort": {"count": -1}},
        {"$limit": 3}
    ])
```

```
[{u'count': 157109, u'_id': u'piottr'}, {u'count': 98892, u'_id': u'miko101'}, {u'count': 92
```

Average contributions per user:

```
avg = poznan.aggregate([
    {"$group": {"_id": "$created.user", "count": {"$sum": 1}}},
    {"$group": {"_id": "$created.user", "average": {"$avg": "$count"}}}
])

[{u'average': 1123.2203389830509, u'_id': None}]
```

Additional Ideas

Contributor statistics

This idea has been taken from Udacity Sample Project - I really liked it.

Top user contribution percentage (piottr) - 19.76%

Combined top 2 users' contribution (piottr and miko101) - 32.19%

Combined Top 10 users contribution - 77.44%

The user contributions are not as skewed as in Sample Project. We see here that there's a group of dedicated users, that have made most of the changes, but there's no monopoly of a single user. One username in top10 suggests that it is a bot, but it stays in the middle of the ranking.

Top 3 fuel station operators

```
res = poznan.aggregate([{"$match": {"amenity": "fuel"}},
                        {"$group": {"_id": "$operator", "count": {"$sum": 1}}},
                        {"$sort": {"count": -1}}, {"$limit": 3}])
[{u'count': 30, u'_id': None}, {u'count': 20, u'_id': u'PKN Orlen'}, {u'count': 15, u'_id':
```

Count of places accessible by wheelchairs

```
all_amenities = poznan.find({"amenity": {"$exists": "true"}}).count()

5139


wheelchair_accessible = poznan.find({"amenity": {"$exists": "true"}, "wheelchair": "yes"}).c

137
```

This value does not mean that there are so little places that are wheelchair accessible, but more probably the data is very incomplete.

Conclusion

After reviewing the OSM data for Poznań I can see that geospatial data is very detailed. However, amenities and other places are missing details, which could be beneficial for general use. I was pleased to see that there's thriving community, which improves the map constantly. The streetnames needed only very little cleaning, but that showed that using this method could further improve, already well prepared, dataset.