OpenStreetMap in Manhattan Data Wrangling with MongoDB

---Hang Zhu

Fort Lee
Cliffside Park
Fairview
New York

Map Area: Manhattan, New York City, US
Min latitude: 40.6978,
Min longitude: -74.0224,
Max latitude: 40.8704,
Max longitude: -73.9160

1. Problems encountered in the map

There isn`t available data for Manhattan area and the data for New York City is huge. Therefore, selecting data manually would be a better way. However, there are still some problems about it.

- This area includes some part of New Jersey and Brooklyn or Long Island which should be removed
- There are errors about the street names ('East 80th Street', 'East 80th Street, 21G')
- City names are not uniform and include upper and lower case ('New York city', 'new york')

This area includes some part of New Jersey and Brooklyn or Long Island

From the area chosen from Overpass API, some data in New Jersey and Brooklyn or Long Island is included. I used two methods to remove them. (keep_manhattan.py)

1. Extract only data with postal code which belongs to Manhattan
2. Use latitude and longitude as criteria to filter data

Errors about the street names

Remove the information after ',' in each street string ('East 80th Street, 21G' → 'East 80th Street') (osm_to_json.py)

City names are not uniform

Remove all data with other city names ('Long Island City') and transform New York/ New York City into New York. (keep_manhattan.py)

2. Procedure of data processing:

1) Obtain data (node(40.6978, -74.0224, 40.8704, -73.9160);<;);out meta; ('manhattan.osm')
2) Run osm_to_json.py ('manhattan.json')
3) Run keep_manhattan.py ('manhattan_only.json')
4) Import data in MangoDB

3. Overview of the Data

This section contains basic statistics about data and MangoDB queries

Size of files

manhattan.osm …. 207M
manhattan_only.json …. 92M

Number of documents

```
db.manha.find().count()
389967
```

Number of node

```
db.man.find({"type":"node"}).count()
289752
```

Number of way

```
db.man.find({"type":"way"}).count()
100211
```

Number of unique users

```
len(db.man.distinct('created.uid'))
1268
```

Number of unique amenity

```
len(db.man.distinct('amenity'))
75
```

Distinct city name

```
db.man.distinct("address.city")
['NEW YORK']
```

Top 3 frequent users

```
    pipeline = [
        {'$group': {'_id': '$created.user',
                'count': {'$sum': 1}}},
        {'$sort': {'count': -1}},
        {'$limit': 3}
        ]
{u'_id': u'Rub21_nycbuildings', u'count': 248365},
{u'_id': u'robgeb', u'count': 20540},
{u'_id': u'Korzun', u'count': 14072}
```

Top three most frequent amenities

```
    pipeline = [
        {'$group': {'_id': '$amenity',
                    'count': {'$sum': 1}}},
        {'$sort': {'count': -1}},
        {'$limit': 4}
    ]
{u'_id': u'parking', u'count': 558},
{u'_id': u'bicycle_parking', u'count': 548},
{u'_id': u'place_of_worship', u'count': 387}
```

Summary:

There are 1268 users in total. The most frequent user is ‘Rub21_nycbuildings’, which takes a great percentage among all users (64%). For 75 unique amenities, the numbers of ‘parking’, ‘bicycle_parking’ and 'place_of_worship' are the most.

4. Other ideas about data

Manhattan is a great place for people who like theater and food. Therefore, in this part, I explored mainly about restaurant and theaters.

Amenity type Imputation Suggestion

Query: db.man.find({'amenity': {'$exists': 1}}).count() → 3764
Amenity type is a very important feature which can be used to assess how a city or area is developing and let people to know the feature of the city. However, in the data I extracted, there are only 3764 records with amenity type information out of 389963 which means a lot of missing data about it. Therefore, imputation within same node could be a good way make up these data.
Benefit: Because there is pretty enough relevant information like the name of this position and name can represent amenity very well, data imputation can be achieved pretty well.
Limits: To impute this feature, machine learning process may be involved. More specifically, maybe some techniques from natural language process should be included. Thus, the whole process maybe a little complicated.

Additional data exploration using MongoDB queries

Top 3 frequent restaurants

```
    pipeline = [
        {'$match': {'amenity': {'$eq': 'restaurant'}}},
          {'$project': {'restaurant': 'name'}
        {'$group': {'_id': '$name',
                    'count': {'$sum': 1}}},
         {'$sort': {'count': -1}},
         {'$limit': 5}
    ]
{u'_id': u'Bareburger', u'count': 3},
{u'_id': u'Blockheads', u'count': 3},
{u'_id': u"Patsy's Pizzeria", u'count': 2},
```

The street with most frequent restaurants

```
   pipeline = [
            {'$match': {'address.street': {'$exists': 1}, 'amenity': 'restaurant'}},
            {'$group': {
                 '_id': '$address.street',
                 'count': {'$sum': 1}
            }},
            {'$sort': {'count': -1}},
            {'$limit': 1}
      ]
{u'_id': u'Amsterdam Avenue', u'count': 48},
```

The street with most theaters

```
      pipeline = [
            {'$match': {'address.street': {'$exists': 1}, 'amenity': 'theatre'}},
            {'$group': {
                 '_id': '$address.street',
                 'theater': {'$addToSet': '$name'},
                 'count': {'$sum': 1}
            }},
            {'$sort': {'count': -1}},
            {'$limit': 1}
      ]
      {u'_id': u'West 45th Street',
        u'count': 7,
        u'theater': [u'Lyceum Theatre',
                     u'Music Box Theatre',
                     u'John Golden Theatre',
                     u'Al Hirschfeld Theatre',
                     u'Bernard B. Jacobs Theatre',
                     u'Gerald Schoenfeld Theatre',
                     u'Booth Theatre']}
```

Top three most frequent place_of_worship

```
pipeline = [
            {'$match': {'amenity': {'$eq': 'place_of_worship'}}},
            {'$group': {'_id': '$name',
                    'count': {'$sum': 1}}},
            {'$sort': {'count': -1}},
                {'$limit': 5}
                ]
{u'_id': u"Kingdom Hall of Jehovah's Witnesses", u'count': 3},
{u'_id': u'A.M.E. Metropolitan Church', u'count': 2},
{u'_id': u'First Christian Church Of The Valley', u'count': 2}
```

Conclusion

After cleaning and exploration of data, it gives pretty much useful insights such as the street with most restaurants and theaters. However, the information is still incomplete(like amenity type) and will not absolutely represent the real situation. In addition to that, I didn`t include many other information because there is a lot of missing or hard understood information for them.