openstreemap

June 28, 2016

1 OpenStreetMap Data Wrangling with MongoDB

Kai Wang
Map Area: Beijing, China
https://s3.amazonaws.com/metro-extracts.mapzen.com/beijing_china.osm.bz2

1.1 1. Problems Encountered in the Map

After downloading the xml file and running test.py to see the general idea of the data, four main problems were noticed.

- Over-abbreviated street names ("Dongzhimen Outer **St**").
- There is wrong postcode, perhaps phone number(010-62332281).
- A variety of different forms of fax and phone ('+86 10 5960 2233', '+86-010-69079600', '86-10-64577779', '(+86)10/65125126').
- Multiple key have the same meaning (疏散人数（万）, 应急避难场所疏散人数万人, 疏散人数, 应急避难场所疏散人口万人).

Over-abbreviated street For those over-abbreviated address, we updated using function *correct_street_type* in audit.py.

```
In [1]: def correct_street_type(street_name):
            changed = 0
            street_name = street_name.strip()
            words = street_name.split()
            newwords = ''
            # change street abbreviation to full name
            for w in words:
                if w in mapping:
                    newwords += mapping[w] + ' '
                else:
                    newwords += w + ' '
            if street_name != newwords[:-1]:
                #print street_name + '=>' + newwords[:-1]
                changed = 1

            return newwords[:-1], changed
```

wrong postcode This is phone-format string in postcode tag: "010-62332281". Maybe it's a wrong input by mistake. So we can easily ignore this wrong input by detecting "-" and skip this tag value.

Fix fax and phone formate There are quite a lot of different forms of fax and phone((‘+86 10 5960 2233’, ‘+86-010-69079600’, ‘86-10-64577779’, ‘(+86)10/65125126’),‘+86 10 8438 8088’) and some tag value contain two number separated by “;” or “/”. The ‘/’ is very tricky because there are also number like:“(+86)10/65133366”, which should be treated as one number. But “00861065323114/008613901017417” should be split into two numbers. Function *correct_number* were write to deal with the fax and phone tags.

Multiple key have the same meaning “应急避难场所疏散人数万人” and “应急避难场所疏散人口万人” are two k tags that have the same meaning. They all mean “the number of people that a emergency shelter can has (10 thousands unit)”. “疏散人数（万）” and “疏散人数” also have similar meaning: “number of people can be evacuated”. But the former is 10 thousands unit. So the former is the later divied by 10000. This can be due to inputting the same item by different people. We will combine the similar keys in data.py.

1.2 2. Data Overview

After downloaded and uncompressed the data file, we can see that it’s not quite a large dataset, but big enough to force us to consider using cElementTree and iterate through the data instead of reading all into memory.

```
$ du -sh beijing_china.osm
152M    beijing_china.osm
```

Basic statistics about the dataset were fetched by MongoDB queries and listed below

```
In [2]: import  pymongo
        client = pymongo.MongoClient('192.168.32.200', 27017)
        db = client['test']
        db.authenticate('test','test')

        # Number of documents
        doccount = db.openstreet.find().count()
        print "Number of documents: %d" % doccount
        # Number of nodes
        nodecount = db.openstreet.find({"type":"node"}).count()
        print "Number of nodes: %d" % nodecount
        # Number of ways
        waycount = db.openstreet.find({"type":"way"}).count()
        print "Number of ways: %d" % waycount
        # Number of unique users
        uniqusers = len(db.openstreet.distinct("created.user"))
        print "Number of unique users: %d" % uniqusers
        #Top 10 contributing user
        R = db.openstreet.aggregate([{"$group":{"_id":"$created.user", "count":{"$sum":1}}
        print "Top 10 contributing user:"
        for r in R:
            print '\t' + r['_id'] + ': ' +  str(r['count'])
        # Number of users appearing only once (having 1 post)
        R = db.openstreet.aggregate([{"$group":{"_id":"$created.user", "count":{"$sum":1}}
        for r in R:
            print "Number of users appearing only once: %d" % r["num_users"]
```

```
Number of documents: 817151
Number of nodes: 711770
Number of ways: 105369
Number of unique users: 1374
Top 10 contributing user:
        Chen Jia: 194010
        R438: 151265
        ij_: 52067
        hanchao: 47770
        katpatuka: 24074
        m17design: 21999
        Esperanza36: 19123
        nuklearerWintersturm: 17233
        RationalTangle: 14493
        u_kubota: 9411
Number of users appearing only once: 275
```

1.3 3. Additional Ideas

Mix-use of Chinese and English In some tag, Chinese and English words are mixed together. Like the stree name listed below. It's better to create streetName_en and streetName_ch to store English street name and Chinese street name separately.

```
In [3]: sR = db.openstreet.distinct("address.street")
        for s in sR[:10]:
            print s
```

```
学院路
团结湖北口
Wangfuijing Street
新街口外大街
西二旗大街
林萃路
酒仙桥北路 甲 10 号院电子城 IT 产业园 107 楼 6 层
荷清路
中关村大街
北四环中路
```

Additional data exploration using MongoDB queries

```
In [4]: # Top 10 appearing amenities
        amenitiesR = db.openstreet.aggregate([{"$match":{"amenity":{"$exists":1}}}, {"$grou
        print "Top 10 appearing amenities:"
        for r in amenitiesR:
            print r["_id"] + ': ' + str(r['count'])
```

```
Top 10 appearing amenities:
restaurant: 1016
parking: 613
bank: 362
school: 350
toilets: 332
fuel: 276
fast_food: 256
cafe: 182
hospital: 156
telephone: 150
```

```
In [5]: # Biggest religion
        religionR = db.openstreet.aggregate([{"$match":{"amenity":{"$exists":1}, "amenity"
                           {"$group":{"_id":"$religion", "count":{"$sum":1}}},
                           {"$sort":{"count":-1}}, {"$limit":1}])
        print "Biggest religion:"
        for r in religionR:
            print r["_id"] + ': ' + str(r['count'])
```

```
Biggest religion:
buddhist: 40
```

```
In [6]: # Most popular cuisines
        cuisinesR = db.openstreet.aggregate([{"$match":{"amenity":{"$exists":1},"cuisine":
                           {"$group":{"_id":"$cuisine", "count":{"$sum":1}}},
                           {"$sort":{"count":-1}}, {"$limit":10}])
        print "Most popular cuisines:"
        for r in cuisinesR:
            print str(r["_id"]) + ': ' + str(r['count'])
```

```
Most popular cuisines:
chinese: 115
japanese: 11
pizza: 10
regional: 10
italian: 8
international: 5
american: 4
asian: 3
german: 3
thai: 3
```

There are 755 restaurants don't have a "cuisine" tag. But it's pretty mush the case that most Chinese love Chinese Food.

1.4 4. Conclusion

We've import the OpenStreetMap data of Beijing into MongoDB. Although it can be called "clean data", we should keep in mind that there's still possible that some mistake in the data. Maybe after further use of it, we will be forced to come back and add more clean functions. But for now, it looks fine.