OpenStreetMap Data Case Study

By CHEN XINRUI

Map Area

Chicago, USA

- Original https://www.openstreetmap.org/relation/122604
- Sample https://pan.baidu.com/s/1kVFgBptO_QDXR3tDbUMltA

Chicago is my favorite city. I'd like an opportunity to contribute to its improvement on OpenStreetMap.org.

Problems Encountered in the Map

After initially downloading the whole document of Chicago area, I use python to extract a smaller sample of it for exploration. I noticed some main problems with the data, which I will discuss in the following order:

- There exists many problematic elements in the nodes, which are completely meaningless.
- There are many abbreviations for street types, which seem chaotic. For example, 'Road' can be written in the form of 'Rd', 'rd' and so forth.

Data Cleaning

Then, I write code to change all these chaotic forms into normal ones. The functions are as followings.

```
mapping = { "St.": "Street","Rd": "Road", "rd": "Road","RD":"Road",
            "Trl":"Trail","Rd.":"Road", "Ave": "Avenue",
            "Blvd": "Boulevard","Ct":"Court", "Ln":"Lane"}

def update_name(name, mapping):
    shortname = mapping.keys()
    for word in shortname:
        if word in name:
            name = name.replace(word,mapping[word])
    return name
```

In this way, all tags like 'E main st' will be like 'E main street'.

Data Overview

After cleaning the data, I import the data to the SQLite and explore it.

Number of nodes

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

	COUNT(*)
1	434363

Find new nodes in Jan.2017

```
1    SELECT id
2    FROM nodes
3    WHERE version =1 and
4    timestamp >='2017-01-01' and
5    timestamp <='2017-01-31'
```

	id
1	4583794021
2	4583794041
3	4583794061
4	4583823513
5	4583823522

5066 行数据在 369ms 内返回

The position of nodes

```
1    SELECT ways_nodes.node_id,
2    ways_nodes.position,
3    nodes.lat,
4    nodes.lon
5    from ways_nodes join nodes
6    on ways_nodes.node_id=nodes.id;
```

	node_id	position	lat	lon
1	734055802	8	41.79802	-87.8318394
2	4270282384	6	42.0017585	-87.9161179
3	26749765	4	41.7955778	-87.9946907
4	1316160487	5	41.9574637	-87.6640053

Top 10 appearing amenities

```
1    SELECT key,value, COUNT(*) as num
2    FROM nodes_tags
3    WHERE key='amenity'
4    GROUP BY value
5    ORDER BY num DESC
6    LIMIT 5;
```

	key	value	num
1	amenity	place_of_worship	145
2	amenity	school	102
3	amenity	restaurant	73
4	amenity	fast_food	46
5	amenity	bench	26

Number of ways

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

	COUNT(*)
1	61244

Top 5 values for ways_tags

```
SELECT key,value,count(*) as num
FROM ways_tags group by key order by count(*) desc limit 5
```

	key	value	num
1	building	yes	43688
2	building_id	890067	36348
3	street	Skokie Boulevard	24638
4	housenumber	9353	24592
5	street:name	Hermitage	24142

Top 5 contributing users for ways

```
select ways.user,ways.uid,count(*) as num
from ways
group by ways.uid order by count(*) desc
limit 5;
```

	user	uid	num
1	chicago-buildings	674454	39191
2	Umbugbene	567034	3684
3	bot-mode	451693	1700
4	Oak_Park_IL	5387019	632
5	boeleman81	1557342	602

Top 5 streets containing most nodes

```
select nodes_tags.key,nodes_tags.value,count(*) as num
from nodes_tags
where key='street'
group by nodes_tags.value order by count(*) desc
limit 5;
```

	key	value	num
1	street	Central Street	27
2	street	Sherman Avenue	27
3	street	Dodge Avenue	26
4	street	Ashland Avenue	23
5	street	Forest Avenue	20

Top 5 postal codes

```
SELECT tags.value, COUNT(*) as count
FROM (SELECT * FROM nodes_tags
UNION ALL
SELECT * FROM ways_tags) tags
WHERE tags.key='postcode'
GROUP BY tags.value
ORDER BY count DESC
limit 5;
```

	value	count
1	60201	471
2	60202	392
3	60305	88
4	60564	71
5	60136	62

Top 5 highway usage

```
SELECT tags.key,tags.value, COUNT(*) as count
FROM (SELECT * FROM nodes_tags UNION ALL
SELECT * FROM ways_tags) tags
WHERE tags.key LIKE '%highway'
GROUP BY tags.value
ORDER BY count DESC
limit 5;
```

	key	value	count
1	highway	residential	5254
2	highway	service	3656
3	highway	footway	1036
4	highway	turning_circle	696
5	highway	primary	679

Top 5 railways values

```
1    select nodes_tags.key,nodes_tags.value,count(*) as num
2    from nodes_tags
3    where key='railway'
4    group by nodes_tags.value order by count(*) desc
5    limit 5;
```

	key	value	num
1	railway	level_crossing	215
2	railway	switch	125
3	railway	station	22
4	railway	crossing	18
5	railway	buffer_stop	7

Additional ideas

We can see that tags in the openstreetmap have many problems. There are many expressions for one thing. For example, street can be written as st, ST, and so forth. Users might add tags in forms they like, which makes the tags chaotic. So I think something can be done.

Suggestion: I suggest the map automatically gives certain list of formal options when users add tags.

Benefit:

- The forms will be more unified and standardized.

Potential problems:

If users can only use the given options as names:

- Problems appear when all stored options cannot match the situation.
- Users' passion might get hurt for not having his own choice.

If users can use other names other than the given options

- Users might ignore the given options.