Project Wrangle OpenStreetMap Data

Map Area

Charlotte, NC, United States

- https://mapzen.com/data/metro-extracts/metro/wuhan_china/

Wuhan is the capital of Hubei province,China, and is the most populous city in Central China. It lies in the eastern Jianghan Plain at the intersection of the middle reaches of the Yangtze and Han rivers. Arising out of the conglomeration of three cities, Wuchang, Hankou, and Hanyang, Wuhan is known as "九省通衢" (the Nine Provinces' (China's) Leading Thoroughfare)"; it is a major transportation hub, with dozens of railways, roads and expressways passing through the city and connecting to other major cities. Because of its key role in domestic transportation, Wuhan was sometimes referred to as "the Chicago of China" by foreign sources.

To be honest, I choose it, just because it's of the right size.

Problems Encountered in the Map

After initially downloading a small sample size of the Charlotte area and running it against a provisional data.py file, I noticed five main problems with the data, which I will discuss in the following order:

- To organize data, all road related tags should be put under road node. Road type nodes have special tags, like:

```
      <tag k="lanes" v="3"/>
      <tag k="oneway" v="yes"/>
      <tag k="highway" v="primary"/>
```

- Inconsistent names, not all tags value are written in chinese *(“Teaching* building*”)*
- Some tags as "bench", "shelter", "atm" always appear with the same value ‘yes’. It seems to me, they can be noted in a list, as facilities provided.

Road related

This collects subnodes under road node, such as:

```
<way id="77752303" version="5" timestamp="2016-12-02T23:03:41Z"
changeset="44126232" uid="2500" user="jamesks">
                <nd ref="914416109"/>
                <nd ref="914418477"/>
                <tag k="ref" v="S112"/>
                <tag k="name" v="平江大道"/>
                <tag k="lanes" v="2"/>
                <tag k="layer" v="1"/>
                <tag k="bridge" v="yes"/>
                <tag k="oneway" v="yes"/>
                <tag k="source" v="Bing"/>
                <tag k="highway" v="primary"/>
</way>
```

becomes:

```
"road": {"bridge": "yes", "lanes": "2", "highway": "primary", "oneway":
"yes"}
```

Not chinese names

Sometimes, the tag ‘name’ isn’t written in Chinese. I use a regex to recognize tags that doesn't contain chinese charactor, which falls
between \u4e00 and \u9fa5 Then replace it with value of tag ‘name:zh’.

```
def audit_chinese(s):
    '''
    This function audit if parameter s is pure Chinese character, returns
True or False.
    :param s: string
    :return: string
    '''
    s = unicode(s)
    u_chinese = re.compile(u"[\u4e00-\u9fa5]+")
    rlt = u_chinese.search(s, 0)
    if not rlt:
        return False
    return True
```

Facilities

Some tags as "bench", "shelter", "atm" always appear with the same value ‘yes’. It seems to me, they can be noted in a list, as facilities provided.

```
if tag.attrib['k'] in FACILITY:
    srv.append(tag.attrib['k'])
    if len(srv) > 0:
            node['facility'] = srv
```

For example, xml element like:

```
<tag k="bus" v="yes"/>
<tag k="bench" v="yes"/>
<tag k="shelter" v="yes"/>
```
becomes:

```
"facility": ["bus", "bench", "shelter"]
```

Data Overview and Additional Ideas

This section contains basic statistics about the dataset, the MongoDB queries used to gather them, and some additional ideas about the data in context.

File sizes

```
wuhan_china.osm ......... 63.6 MB
wuhan_china.osm.json .... 51.7 MB
```

Number of nodes

```
db.getCollection('map').find({"type":"node"}).count()
```
321244

Number of ways

```
db.getCollection('map').find({"type":"way"}).count()
```
34791

Top 5 contributing users

```
db.getCollection('map').aggregate([
{'$group': {'_id': '$created.user', 'count':{$sum:1}}},
{'$sort': {'count': -1}},
{'$limit': 5},
])
/* 1 */
{
    "_id" : "GeoSUN",
    "count" : 111540.0
}

/* 2 */
{
    "_id" : "Soub",
    "count" : 47736.0
}
```

```
/* 3 */
{
    "_id" : "jamesks",
    "count" : 24378.0
}

/* 4 */
{
    "_id" : "Gao xioix",
    "count" : 17894.0
}

/* 5 */
{
    "_id" : "katpatuka",
    "count" : 17225.0
}
```

Number of users appearing only once (having 1 post)

```
db.getCollection('map').aggregate([
{'$group': {'_id': '$created.user', 'count':{$sum:1}}},
{'$match': {"count":{$lte:1}}},
])
/* 1 */
{
    "_id" : "poornibadrinath",
    "count" : 1.0
}

/* 2 */
{
    "_id" : "ouleyang",
    "count" : 1.0
}

/* 3 */
{
    "_id" : "xdpitest",
    "count" : 1.0
}

/* 4 */
{
    "_id" : "freenavi",
    "count" : 1.0
}

/* 5 */
```

```
{
    "_id" : "LESLIELZX",
    "count" : 1.0
}

/* 6 */
{
    "_id" : "PeterFritz",
    "count" : 1.0
}

/* 7 */
{
    "_id" : "ulrich_",
    "count" : 1.0
}

/* 8 */
{
    "_id" : "大鸟哥",
    "count" : 1.0
}

/* 9 */
{
    "_id" : "dading007",
    "count" : 1.0
}

/* 10 */
{
    "_id" : "homer__simpsons",
    "count" : 1.0
}

/* 11 */
{
    "_id" : "small ball",
    "count" : 1.0
}

/* 12 */
{
    "_id" : "TPOB",
    "count" : 1.0
}

/* 13 */
{
    "_id" : "raojing",
    "count" : 1.0
}
```

```
/* 14 */
{
    "_id" : "Junqiao Zhao (John)",
    "count" : 1.0
}

/* 15 */
{
    "_id" : "邹华_可视化",
    "count" : 1.0
}

/* 16 */
{
    "_id" : "YUCHUN TSAO",
    "count" : 1.0
}

/* 17 */
{
    "_id" : "wantmore",
    "count" : 1.0
}

/* 18 */
{
    "_id" : "徐超伟 1",
    "count" : 1.0
}

/* 19 */
{
    "_id" : "xbd1519722757",
    "count" : 1.0
}

/* 20 */
{
    "_id" : "大白兔",
    "count" : 1.0
}

/* 21 */
{
    "_id" : "weir",
    "count" : 1.0
}

/* 22 */
{
```

```
    "_id" : "asyncJun",
    "count" : 1.0
}

/* 23 */
{
    "_id" : "Aleks-Berlin",
    "count" : 1.0
}

/* 24 */
{
    "_id" : "foxyflash",
    "count" : 1.0
}

/* 25 */
{
    "_id" : "Zaw Lin Tun",
    "count" : 1.0
}

/* 26 */
{
    "_id" : "金逸陶",
    "count" : 1.0
}

/* 27 */
{
    "_id" : "Damodar Dhakal",
    "count" : 1.0
}

/* 28 */
{
    "_id" : "KirillRBT",
    "count" : 1.0
}

/* 29 */
{
    "_id" : "gpp 超爱吃芒果",
    "count" : 1.0
}

/* 30 */
{
    "_id" : "Qartallo",
    "count" : 1.0
}
```

```
/* 31 */
{
    "_id" : "Yegor Kirpichev",
    "count" : 1.0
}

/* 32 */
{
    "_id" : "Kong Cobain",
    "count" : 1.0
}

/* 33 */
{
    "_id" : "spasstard",
    "count" : 1.0
}

/* 34 */
{
    "_id" : "dgitto",
    "count" : 1.0
}

/* 35 */
{
    "_id" : "陈小花",
    "count" : 1.0
}

/* 36 */
{
    "_id" : "Maeve Ward1",
    "count" : 1.0
}

/* 37 */
{
    "_id" : "easy_life",
    "count" : 1.0
}

/* 38 */
{
    "_id" : "Wrightbus",
    "count" : 1.0
}

/* 39 */
{
    "_id" : "maimaihu",
```

```
    "count" : 1.0
}

/* 40 */
{
    "_id" : "wheelmap_visitor",
    "count" : 1.0
}

/* 41 */
{
    "_id" : "JeremyList",
    "count" : 1.0
}

/* 42 */
{
    "_id" : "漫游 GIS",
    "count" : 1.0
}

/* 43 */
{
    "_id" : "Virgil Guo",
    "count" : 1.0
}

/* 44 */
{
    "_id" : "七彩田",
    "count" : 1.0
}

/* 45 */
{
    "_id" : "漱石枕流",
    "count" : 1.0
}

/* 46 */
{
    "_id" : "noliver",
    "count" : 1.0
}

/* 47 */
{
    "_id" : "chinhsuen",
    "count" : 1.0
}
```

```
/* 48 */
{
    "_id" : "Trung Lupin",
    "count" : 1.0
}

/* 49 */
{
    "_id" : "eeqinfeng",
    "count" : 1.0
}

/* 50 */
{
    "_id" : "AndiG88",
    "count" : 1.0
}
```

Additional Ideas

Dataset reduction suggestion

As there are a lot of duplicated nodes with same name tag, provided by different users. It could be more efficient, if we come up with a neutral template, which combines those tags all together. This approach can massively decrease duplication in this giant dataset. The downside is that we might miss out those unique tags, that only some users noted.

Additional Data Exploration

Top 5 duplicated name

```
db.getCollection('map').aggregate([
{'$group': {'_id': '$name', 'count':{$sum:1}}},
{'$sort': {'count': -1}},
{'$match':{'_id':{'$nin':['',null]}}},
{'$limit': 5},
])
/* 1 */
{
    "_id" : "京珠高速",
    "count" : 167.0
}

/* 2 */
{
```

```
    "_id" : "三环线",
    "count" : 126.0
}

/* 3 */
{
    "_id" : "二环线",
    "count" : 87.0
}

/* 4 */
{
    "_id" : "汉蔡高速",
    "count" : 73.0
}

/* 5 */
{
    "_id" : "武鄂高速公路",
    "count" : 67.0
}
```

All tourism places

```
db.getCollection('map').aggregate([
{'$group': {'_id': '$tourism', 'count':{$sum:1}}},
{'$sort': {'count': -1}},
{'$match':{'_id':{'$nin':['',null]}}},
])
/* 1 */
{
    "_id" : "hotel",
    "count" : 94.0
}

/* 2 */
{
    "_id" : "attraction",
    "count" : 23.0
}

/* 3 */
{
    "_id" : "museum",
    "count" : 22.0
}

/* 4 */
{
    "_id" : "hostel",
```

```
    "count" : 4.0
}

/* 5 */
{
    "_id" : "artwork",
    "count" : 4.0
}

/* 6 */
{
    "_id" : "viewpoint",
    "count" : 3.0
}

/* 7 */
{
    "_id" : "zoo",
    "count" : 3.0
}

/* 8 */
{
    "_id" : "motel",
    "count" : 2.0
}

/* 9 */
{
    "_id" : "chalet",
    "count" : 2.0
}

/* 10 */
{
    "_id" : "information",
    "count" : 2.0
}

/* 11 */
{
    "_id" : "theme_park",
    "count" : 1.0
}

/* 12 */
{
    "_id" : "caravan_site",
    "count" : 1.0
}

/* 13 */
```

```
{
    "_id" : "alpine_hut",
    "count" : 1.0
}
```