Contents

一 数据源

选取的是上海地区的 OSM 文件，大小 54M。链接如下：

https://s3.amazonaws.com/metro-extracts.mapzen.com/shanghai_china.osm.bz2

二 分析结果及及解决办法

1）分析结果

对原始的 OSM 数据进行分析后，发现如下一些问题

- 相同属性的取值不统一
    1. 当 tag 为“way“时，k=oneway 对应的 v 值不统一，根据官方的解释，应该为 yes 或者 no 但在数据集中有的值为-1。需要统一为 yes 或者 no。 官方解释链接如下：

       http://wiki.openstreetmap.org/wiki/Way

2. highway 和 Hwy 不一致，需要转换 Hwy 到 highway。

- 当 tag 为 node 或者 way 时，k=name 对应的 v 是中文名字时，有的 node 节点没有 k=name-en 属性。
- 当 tag 为 node 时，user 的值有些是中文。但绝大部分的 user 取值是英文字母.

2）解决办法

根据以上列出的问题，相应的解决办法如下

- 属性取值不统一的，在抽取数据时，把不规范或者不统一的值修改成规范或者统一值。
- 增加 k=name-en 属性，其 v 的取值采用百度开放的翻译 API 接口翻译成英文。
  代码片段如下：

```
def translate_func(text):
    httpClient = None
    myurl = '/api/trans/vip/translate'
    fromLang = 'zh'
    toLang = 'en'
    salt = random.randint(32768, 65536)
    sign = appid + text + str(salt) + secretKey
    m1 = md5.new()
    m1.update(sign)
    sign = m1.hexdigest()
    myurl = myurl + '?appid=' + appid + '&q=' + urllib.quote(
        text) + '&from=' + fromLang + '&to=' + toLang + '&salt=' + str(salt) + '&sign=' + sign

    try:
        httpClient = httplib.HTTPConnection('api.fanyi.baidu.com')
        httpClient.request('GET', myurl)
        response = httpClient.getresponse()
        result = eval(response.read())
        return result["trans_result"][0]["dst"]
    except Exception, e:
        print e
    finally:
        if httpClient:
            httpClient.close()
```

- 将 user 的取值为中文的，用汉语拼音方式改写。
  代码片段如下：

```
            if u'\u4e00' <= item.attrib["user"] <= u'\u9fff':
                node["user"] = lazy_pinyin(item.attrib["user"])
            else:
                node["user"] = item.attrib["user"]
```

三 数据抽取转换过程

- 完成以下内容
  1 只处理两种类型的顶级标记：“节点（node）”和“道路（way）”
  2 CREATED 数组中的属性应该添加到键“created”下
  3 经纬度属性应该添加到“pos”数组中，以用于地理空间索引编制。确保“pos”数组中的值是浮点型，不是字符串。
  4 数据存为 JSON 格式并导入 Mongo 数据库

  5 创建 Mongodb 数据库和集合

```
> use shaosm
switched to db shaosm
> show dbs
admin     0.000GB
examples  0.000GB
local     0.000GB
> db
shaosm
> db.createCollection(mapdata)
2017-09-22T21:20:28.988+0800 E QUERY    [thread1] ReferenceError: mapdata is not defined :
@(shell):1:1
> db.createCollection("mapdata")
{ "ok" : 1 }
```

四 数据统计结果

1 数据总数

```
> show dbs
admin     0.000GB
examples  0.000GB
local     0.000GB
shaosm    0.007GB
> use shaosm
switched to db shaosm
> db.mapdata.find().count()
83898
```

中文在 mongodb 里显示正常：

```
> db.mapdata.find({"name":"东方天郡东区5"}).co
{ "_id" : ObjectId("59c51112ec90280df4e27da1"), "name:en" : "Eastern day county east 5", "node_refs" : [ "4936510597", "4936510598", "4936510599", "4936510600", "4936510601", "4936
], "name" : "东方天郡东区5", "created" : { "changeset" : "49828883", "version" : "1", "uid" : "6144608", "timestamp" : "2017-06-26T07:10:36Z" }, "user" : "tangzhenchao", "type" :
9627" }
```

> db.mapdata.find().count()

83898

2 Node 节点的数量

> db.mapdata.find({"type":"node"}).count()

74650

3 Way 节点的数量

> db.mapdata.find({"type":"way"}).count()

9248

4 Suppermarket 的数量

```
> db.mapdata.find({"shop": "supermarket"}).count()
3
```

5 Highway 的种类以及个数

```
> db.mapdata.aggregate([{$group:{ _id: "$highway",result: { $sum: 1 }}}])
{ "_id" : "services", "result" : 1 }
{ "_id" : "tertiary_link", "result" : 9 }
{ "_id" : "platform", "result" : 1 }
{ "_id" : "pedestrian", "result" : 21 }
{ "_id" : "track", "result" : 62 }
{ "_id" : "living_street", "result" : 10 }
{ "_id" : "steps", "result" : 21 }
{ "_id" : "path", "result" : 36 }
{ "_id" : "trunk_link", "result" : 64 }
{ "_id" : "service", "result" : 584 }
{ "_id" : "raceway", "result" : 2 }
{ "_id" : "construction", "result" : 25 }
{ "_id" : "cycleway", "result" : 33 }
{ "_id" : "motorway_link", "result" : 195 }
{ "_id" : "footway", "result" : 206 }
{ "_id" : "motorway", "result" : 247 }
{ "_id" : "primary_link", "result" : 75 }
{ "_id" : "trunk", "result" : 60 }
{ "_id" : "secondary", "result" : 454 }
```

{ "_id" : "secondary_link", "result" : 25 }

五 分析数据的问题

通过第一次数据分析，发现如下一些问题：

1 当 type 是 way（道路）的时候，相关一些属性在有的节点没有记录，比如是否 ‘railway’ 还是 ‘highway’ 等其他属性，并没有记录。这样就造成后面数据统计的时候数据不准确。比如：

……

"user": "Steven Shen",

"type": "way",

"id": "240348083",

…….

2 当 type 是 way（道路）的时候，其 node_refs 的值里有些 node 其实并不存在。这些 node 并没有出现在 type 是 node（节点）。比如：

{

"node_refs": [

"3911958650", 这个值在 type 是 node（节点）时并不存在。

"3911958859"

],

"created": {

"changeset": "45942077",

"version": "2",

"uid": "445671",

"timestamp": "2017-02-09T09:37:50Z"

},

"user": "flierfy",

"railway": "rail",

"type": "way",

"id": "387966082"

}

六 数据处理建议

1 当 type 是 way（道路）时，有些节点属性缺失的情况，可能需要参考其他一些道路数据集，尽量将这些属性补全，如果没有，考虑将这些节点删除掉，因为这些节点对于后面的一些统计分析没有啥价值。

2 当 type 是 way（道路）时，对于 node_refs 的值里有些 node 其实并不存在的情况，同样需要参考其他一些数据集，或者放大采样范围，或许能将这一部分数据补齐。否则建议将这些不存在的 node 从 node_refs 里删除掉。

七 预期的益处和风险

益处：

补足缺失的属性后，当统计所有 type 是 way（道路）时，会得到更加准确的结果。比如：之前统计的“raceway”的数量为 2，（{ "_id" : "raceway", "result" : 2 }）。对 node 的真实性实施改进后，way 的 node_refs 值会更准确可靠，我个人觉得 node_refs 可以用于导航路线的生成，那么改进后生成的导航路线会更加准确和真实。

风险：

处理分析数据的难度会增大，增加更多的人力和时间成本，体现在如下几点：

1 需要选取有效的额外的数据集，这个需要花费大量的时间去搜索和比对，做前期数据的筛选校验。

2 如果数据集的存储格式不一致，数据分析处理过程不同，导致程序版本过多，后期迭代开发不方便。

3 整体数据分析过程复杂，很难做成统一的数据处理接口，