Contents

[一 数据源 1](#_Toc494098207)

[二 分析结果及及解决办法 1](#_Toc494098208)

[1）分析结果 1](#_Toc494098209)

[2）解决办法 2](#_Toc494098210)

[三 数据抽取转换过程 2](#_Toc494098211)

[四 数据统计结果 3](#_Toc494098212)

[1 数据总数 3](#_Toc494098213)

[2 Node节点的数量 3](#_Toc494098214)

[3 Way节点的数量 3](#_Toc494098215)

[4 Suppermarket的数量 4](#_Toc494098216)

[5 Highway的种类以及个数 4](#_Toc494098217)

[五 分析数据的问题 5](#_Toc494098218)

[六 数据处理建议 6](#_Toc494098219)

[七 预期的益处和风险 6](#_Toc494098220)

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>

1. 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

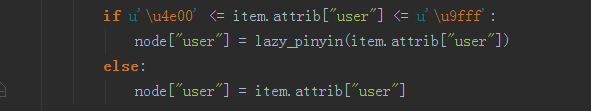
* 属性取值不统一的，在抽取数据时，把不规范或者不统一的值修改成规范或者统一值。
* 增加k=name-en属性，其v的取值采用百度开放的翻译API接口翻译成英文。

代码片段如下：



* 将user的取值为中文的，用汉语拼音方式改写。

代码片段如下：

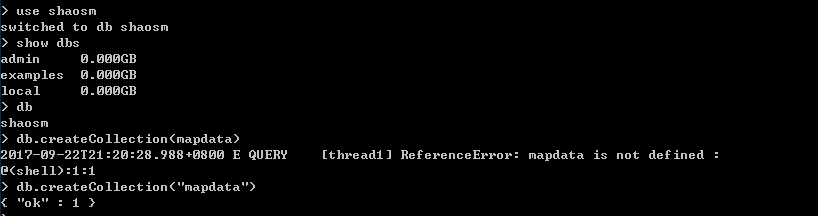


三 数据抽取转换过程

* 完成以下内容

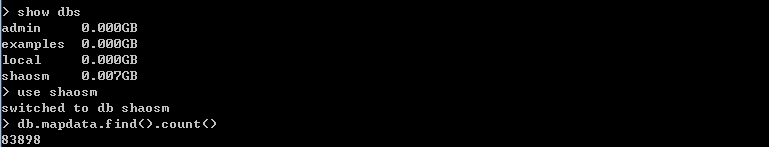
1. 只处理两种类型的顶级标记：“节点（node）”和“道路（way）”
2. CREATED 数组中的属性应该添加到键“created”下
3. 经纬度属性应该添加到“pos”数组中，以用于地理空间索引编制。确保“pos”数组中的值是浮点型，不是字符串。
4. 数据存为JSON格式并导入Mongo 数据库

5 创建Mongodb数据库和集合

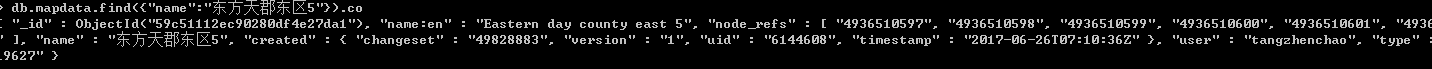


四 数据统计结果

1 数据总数



中文在mongodb里显示正常：



> 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，