OpenStreetMap 数据清洗项目

评审老师您好，笔者是零基础学编程和Python，一个多月磕磕碰碰到达项目一的，十分期待您的评审和建议，感谢您的时间。

地图区域

因为在上海工作比较熟悉，所以选择了中国上海市（主要地区），地图和数据集链接如下：

- http://www.openstreetmap.org/relation/913067#map=8/31.272/122.093
- https://mapzen.com/data/metro-extracts/your-extracts/bcabf6bcade4

数据来自OpenStreetMap，因为发现大都市数据包含有许多其他城市的数据，所以只在上海范围内选择了一部分区域的数据，导出文件大小符合项目要求（非压缩状态大于 50 MB）。

地图中的问题

由于只清洗了tags里的街道名，以下均为街道名的问题（其他问题思路大致一致）：

- 非汉字街道名，如：

> Huaihai Middle Road
> #3999 XiuPu Road
> لندينگ هوم 1

- 汉字、数字、英文和其他符号混合的街道名，如：

> 仙霞路333号10楼
> 河南中路531-541弄
> 红松路 81 弄
> 黄渡 • 绿苑路
> |注：“黄渡”是镇名，这里是必要的，因为上海有两个不同的区内都有“绿苑路”
> 世纪大道 Century Avenue
> Zhao Jia Bang Road (肇嘉浜路)

- 纯汉字，但不是纯街道名，有“市”、“区”等上一级命名，如：

> 上海市徐汇区淮海西路55号

- 纯汉字，但包含多个同级的命名（比如含有两个“路”字），如：

> 淮海中路,
> 近茂名路
> |注：这是这个数据集的特殊情况，原始代码为 v=”淮海中路， 近茂名路”
> |注：其他数据集里还有比如“淮海中路淮海中路” 这种情况，这个回答省略了没有处理这种情况（看起来这个数据集没有这个问题，但没有仔细探索）

- 纯汉字，但非街道名，而是商店、学校等名称，如：

> 新华村

以下为更新街道名的代码：

```
import re
import sys
stdout = sys.stdout
reload(sys)
sys.setdefaultencoding('utf-8')
sys.stdout = stdout

pattern1 = ur'^[a-z\.\\(\)\-\,\?]+$'
pattern2 = ur'[\u4E00-\u9FA5]+'
pattern3 = ur'^[\u4E00-\u9FA5]+$'

st_type_re1 = re.compile(pattern1, re.IGNORECASE) #1-不含中文和数字
st_type_re2 = re.compile(pattern2, re.IGNORECASE) #2-含有至少一个中文
st_type_re3 = re.compile(pattern3)  #3-只有中文

expected = [u"段", u"路", u"道", u"线", u"街"]

def mapping(t,new_name):
    if new_name.find(u"区") != -1:
        #如果含有 '区'
        t.attrib['v'] = new_name[new_name.find(u"区")+1: ]
    elif re.search(re.compile(ur'\('),new_name):
        #如果含有'('
        t.attrib['v'] = new_name[new_name.find(u"(")+1:]
    elif re.search(re.compile(ur'\.|\·|(\s\•\s)'),new_name):
        #如果含有 '.', '·', ' • '
        s = re.search(re.compile(ur'\.|\·|(\s\•\s)'),new_name).group()
        new_name = new_name.split(s)
        t.attrib['v'] = new_name[0]+"."+new_name[-1] #统一为 .
    elif re.search(re.compile(ur'\s'),new_name):
        #如果含有空格
        new_name = new_name.split(u" ")
        t.attrib['v'] = new_name[0]+"."+new_name[-1]
    else:
        #如果是其他情况，删除该tag
        t.attrib['v'] = None
    return t


def update_name(tag):
    st_name = tag.attrib['v']
    if not st_name:
        tag.attrib['v'] = None
    elif re.search(st_type_re1,st_name):#1
        tag.attrib['v'] = None
    elif st_type_re2.search(st_name):#2
        if st_name.find(u"路") == -1 and st_name.find(u"道") == -1 and \
            st_name.find(u"街") == -1 and st_name.find(u"线") == -1 and \
            st_name.find(u"段") == -1 :
            tag.attrib['v'] = None
        else:
            if st_name[-1] not in expected:
                for s in expected:
                    if st_name.find(s) != -1:
                        i = st_name.rfind(s)
                        new_name = st_name[:i+1]
                        #例：Zhao Jia Bing Lu（肇嘉浜路） => Zhao Jia Bing Lu（肇嘉浜路
                        if st_type_re3.search(new_name):
                            #3
                            if new_name.find(u"区") == -1:
                                tag.attrib['v'] = new_name
                            else:
                                mapping(tag,new_name)
                        else:
                        #新字符内的中文取出，比如：Zhao Jia Bing Lu（肇嘉浜路 => 肇嘉浜路
                            nm = st_type_re2.search(new_name).group()
                            tag.attrib['v'] = nm
            elif st_name[-1] in expected:
                if st_type_re3.search(st_name):
                   #3
                    if st_name.find(u"区") == -1:
                        pass
                    else:
```

```
                    mapping(tag, st_name)
                else:
                #不只有中文的
                    if st_name.find(u"近茂名路") != -1:
                        #该数据集的特殊情况
                        tag.attrib['v'] = u"淮海中路"
                    else:
                        mapping(tag, st_name)

    else:
        tag.attrib['v'] = None


'''检测上述代码是否生效'''
def is_st_name(tag):
    return (tag.attrib['k'] == "addr:street")

if is_st_name(t):
    print "Before: ", t.attrib['v']
    update_name(t)
    print "After: ", t.attrib['v']

    # if tag.attrib['v']: #如果更新后的街道名为真，则写入
```

代码结果：

```
Before: Huaihai Middle Road
After: None
Before: #3999 XiuPu Road
After: None
Before: 1 لندينگ هوم
After: None
Before: 仙霞路333号10楼
After: 仙霞路
Before: 河南中路531-541弄
After: 河南中路
Before: 红松路 81 弄
After: 红松路
Before: 黄渡 • 绿苑路
After: 黄渡.绿苑路
Before: 世纪大道 Century Avenue
After: 世纪大道
Before: Zhao Jia Bang Road (肇嘉浜路)
After: 肇嘉浜路
Before: 上海市徐汇区淮海西路55号
After: 淮海西路
Before: 淮海中路，近茂名路
After: 淮海中路
Before: 新华村
After: None
```

SQL查询任务

- 文件大小
- 唯一用户的数量
- 节点和途径的数量
- 便利店的数量
- “上海马戏城 ” 的经纬度

1. 文件大小

```
shanghai_major.osm ....... 137.8 MB
P_Shanghai.db ............ 73.6 MB
```

```
nodes.csv ................ 51.2 MB
nodes_tags.csv ........... 2.1 MB
ways_tags.csv ............ 5.3 MB
ways_nodes.csv ........... 8.3 MB
```

2. 唯一用户的数量

```
SELECT COUNT(DISTINCT(n.uid))
FROM
(SELECT uid FROM nodes UNION ALL SELECT uid FROM ways) as n;
```

1430

以上是清洗后的唯一用户数据，原始数据可能更多

3. 节点和途径的数量

```
SELECT COUNT (id)  FROM nodes;
SELECT COUNT (id)  FROM ways;
```

节点数：802518
途径数：89695

4. 便利店的数量

```
SELECT key, COUNT(*)
FROM nodes_tags
WHERE key = 'shop' AND value = 'convenience';
```

shop 290

5. “上海马戏城”的经纬度

```
SELECT nodes.id, nodes_tags.[value], nodes.lat, nodes.lon, nodes.[user]
FROM nodes
JOIN  nodes_tags
ON nodes.id = nodes_tags.id
WHERE nodes_tags.[value] = '上海马戏城'
GROUP BY nodes.id;
```

id	value	lat	lon	user
470079347	上海马戏城	31.2811762	121.4475394	Edward
651013995	上海马戏城	31.2800115	121.4469268	adong33
3800386164	上海马戏城	31.281178	121.4475769	aighes
4559664017	上海马戏城	31.2811737	121.4475097	aighes

由于住在马戏城附近，所以好奇查询。
可以看到有三位用户贡献过上海马戏城的经纬度，其中用户aighes贡献过两次。
几次位置差异并不大。
GPS识别到某手机的经纬度匹配到这个经纬度范围内，就可以通过地图数据集实时返回位置信息，手机导航的逻辑大概是这样吗？

关于数据集的其他想法

不要轻易相信数据，尽量克制手动修改原始数据（尽管有些小问题很难匹配到）。
发现问题时，要及时查询相关领域知识，辨别出真正的问题数据模式，不可盲目归到已有的清洗规则中，以免误伤了原本可能有效的数据。

比如“黄渡•绿苑路”，搜索绿苑路，结果发现上海市嘉定区和上海市闵行区都有绿苑路，而黄渡恰好是嘉定区的一个镇；如果不知道这个格式出现的原因，可能会错误地把“黄渡”也清洗掉。

预期问题：
我这里统一改成“黄渡.绿苑路”的格式也是自己定义的，对于他人来说这个格式可能仍然是脏数据，所以需要让使用的人知道这类清洗规则或格式规范。

另外一种解决方案，是可以直接把“黄渡•”去掉，但同时添加一个新的addr字段，比如zh:“镇”，然后把“黄渡”加进去。

另外有一些中文行政区划街道名库，也可以补充到OpenStreetMap里。
中华人民共和国行政区划数据【省、市、区县、乡镇街道】中国省市区镇三级四级联动地址数据

项目参考文档链接

1. 正则表达式测试：
   http://tools.jb51.net/regex/javascript#replace
2. python 2 编码问题，如何处理中文字符
   http://ask.csdn.net/questions/346937
   https://blog.ernest.me/post/python-setdefaultencoding-unicode-bytes
3. 类和实例：
   https://www.liaoxuefeng.com/wiki/001374738125095c955c1e6d8bb493182103fac9270762a000/0013868200407737 6d2d7f8cc8a4e2c9982f92788588322000