

※

高德接口抓取数据：利用高德官方开放接口，抓取全国的poi数据

源码github地址：

<https://github.com/liujiao111/poi>

高德接口：接口文档

<https://developer.amap.com/api/webservice/guide/api/search>

高德平台：创建应用、申请key值、查看配额

<https://console.amap.com/dev/flow/manage>

源码给出了高德和百度的数据抓取示例代码

下面的修改仅针对高德poi的抓取做了修改。

修改后代码位置：

<http://192.168.16.234:81/svn/GIS/Al/grab-gaode-poi>

后期修改：

针对高德的抓取修改了抓取的逻辑

原来的代码只支持对城市，省，区县等分别抓取，修改成按照当前级别以及下属级别分别抓取，最终截止等级为区县。

原来的代码只支持按照一层poi类别抓取，修改成按照当前类别以及下属级别分别抓取，最终截止等级为第三级别。

原来的代码一次请求，最多返回900条数据（按照每页25个，多页抓取），修改成如果数量超过900条，则按照下属的poi类别抓取，从而得到多于900的结果

原来的多边形抓取，也是只请求一次，最多返回900条数据（按照每页25个，多页抓取），修改成按照四叉树向下切分子网格，按照子网格继续循环获取，得到最多数据的poi

运行：

1. 准备环境：conda create -n grabpoi python=3.7

2. 安装依赖包：

```
conda install pandas
```

```
pip install xpinyin
```

```
conda install requests
```

3. 运行抓取poi:

```
cd D:\Work-yun\poi-master\poi-master\gaode\poi-province
```

```
python app.py
```

4. 运行分析和统计poi:

```
cd D:\Work-yun\poi-master\poi-master\gaode\poi-province
```

```
python analysis.py
```

5. 抓取和分析的参数设置，全部在key.py中设置。

逻辑和参数:

1. 按照行政区划抓取结合poi类别抓取: 调用关键词搜索接口

打开app.py

需要设置的参数，在key.py中设置:

amap_web_key: 高德官网申请的接口key值，多个，按照顺序使用，每个id每天限额3万个请求

grabtype: 此处配置为2, 1: 按照行政区划的边界，调用多边形搜索接口，划分四叉树逐级搜索 2: 按照行政区划逐级，调用关键词搜索接口，搜索上面的类别

keyword: 支持一级，二级，三级 poi类别，代码参考poiclass.py 和文档“高德地图POI分类编码表.xlsx”

adminname: 目标行政区划的名称或者编码，如果是省和市，可以直接用名称，区县强烈要求用编码比较好（避免有重名的可能），名称和编码参考“高德地图城市编码表.xlsx”

adminlevel: 指明目标行政区划的级别: 1为省，2为市，3为区、县

抓取过程的逻辑:

查询目标行政区划的等级，一直查询到区县级别，然后在区县下，逐个poi类别调用关键词搜索接口抓取数据，如一次接口的返回数量多于800条，则继续采用的区县的别介更细化的抓取。

弊端:

会有少量的重复数据，因为poi是有多类别的情况。

2. 按照行政区划的边界抓取结合poi类别抓取：调用多边形接口，推荐此种类型的抓取

grabtype : 此处配置为1
其余参数与上面同。

抓取过程的逻辑：

查询目标行政区划的边界，分子网格四个，调用多边形抓取数据，如网格范围返回的数量多于800个，则采用四叉树继续细分子网格持续抓取。

海淀区抓取结果和网购的样本数据对比：

右侧为抓取的数据

[支持多条件过滤, 例如: 北京 上海]

名称

选项

☒ (全选) (9999+)

☒ 1 (1239)

☒ 2 (192)

☒ 3 (269)

☒ 4 (57)

☒ 5 (12305)

☒ 6 (17093)

☒ 7 (16587)

☒ 8 (2913)

☒ 9 (3138)

☒ 10 (2116)

☒ 11 (1771)

☒ 12 (7863)

☒ 13 (6712)

☒ 14 (11556)

☒ 15 (12533)

☒ 16 (2185)

☒ 17 (12922)

☒ 18 (22)

☒ 19 (23209)

☒ 20 (1962)

☒ 22 (7)

☒ 97 (15351)

☒ 99 (13465)


buy

165468


名称

修改日期


类型

 poi-hai-dian-qu-01-1204.csv


2021/3/25 17:37 XLS 工作表

 poi-hai-dian-qu-02-224.csv


2021/3/25 17:37 XLS 工作表

 poi-hai-dian-qu-03-294.csv


2021/3/25 17:37 XLS 工作表

 poi-hai-dian-qu-04-60.csv


2021/3/25 17:37 XLS 工作表

 poi-hai-dian-qu-05-11721.csv


2021/3/25 17:45 XLS 工作表

 poi-hai-dian-qu-06-15967.csv


2021/3/25 18:21 XLS 工作表

 poi-hai-dian-qu-07-15655.csv


2021/3/25 18:25 XLS 工作表

 poi-hai-dian-qu-08-2814.csv


2021/3/25 18:26 XLS 工作表

 poi-hai-dian-qu-09-3060.csv


2021/3/25 18:26 XLS 工作表

 poi-hai-dian-qu-10-1902.csv


2 XLS 工作表

 poi-hai-dian-qu-11-1879.csv


2 XLS 工作表

 poi-hai-dian-qu-12-7675.csv


2021/3/25 18:30 XLS 工作表

 poi-hai-dian-qu-13-5940.csv


2021/3/25 18:32 XLS 工作表

 poi-hai-dian-qu-14-10739.csv


2021/3/25 18:34 XLS 工作表

 poi-hai-dian-qu-15-12274.csv


2021/3/25 18:38 XLS 工作表

 poi-hai-dian-qu-16-2108.csv


2021/3/25 18:39 XLS 工作表

 poi-hai-dian-qu-17-12394.csv


2021/3/26 9:08 XLS 工作表

 poi-hai-dian-qu-18-23.csv


2021/3/26 9:08 XLS 工作表

 poi-hai-dian-qu-19-22825.csv


2021/3/26 9:14 XLS 工作表

 poi-hai-dian-qu-20-1780.csv


2021/3/26 9:15 XLS 工作表

 poi-hai-dian-qu-22-7.csv

2021/3/26 9:30 XLS 工作表

 poi-hai-dian-qu-97-16029.csv

2021/3/26 9:34 XLS 工作表

 poi-hai-dian-qu-99-13207.csv

2021/3/26 9:37 XLS 工作表

147,507

3. 数据分析

采用了pandas来进行数据的统计，效率非常高。

<https://www.py pandas.cn/>

以下是pandas使用中一些总结：

<https://note.youdao.com/s/48MK35mz>

