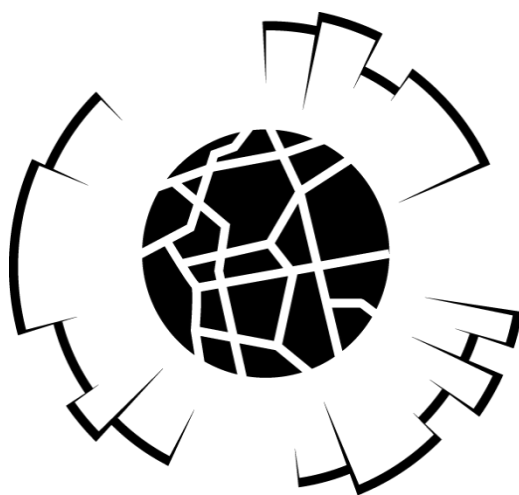




# OSpider 用户手册

v2.0.0



二〇一九年十二月



## 目录

1 版本与功能 .....	1
1.1 当前版本功能 .....	1
1.2 升级说明 .....	1
2 使用教程 .....	1
2.1 文件结构说明 .....	1
2.2 操作示范 .....	1
2.3 FQA .....	4
3 关于 .....	5
3.1 开发者 .....	5
3.2 Bug 报告与意见反馈 .....	5
附录 1 爬取原理与 OSpider 爬取偏差问题 .....	6
附录 2 WGS84 与 BD09 坐标问题 .....	7

## 1 版本与功能

### 1.1 当前版本功能

- 含界面免安装, 专注抓取城市及城市内部百度 POI, 并将坐标转化为 WGS84
- 附带插件: 获取行政区的外接矩形(菜单: 工具->获取政区坐标)

### 1.2 升级说明

- 版本号 1.0.1->2.0.0
- 增添了 UI 界面, 程序运行状态提示更加人性化
- 重构核心代码
- 自动记忆上一次输入的参数
- 新增用户群, 开辟交流新渠道

## 2 使用教程

### 2.1 文件结构说明







 OSpider_v2.0.0.exe	2019/12/29 13:53	应用程序	9,130 KB
 help.pdf	2018/7/31 14:38	PDF Document	1,167 KB
 icon.ico	2018/7/22 15:28	图片文件(.ico)	17 KB
 addin_regxy.html	2018/7/23 16:45	Chrome HTML D...	4 KB
 property.ini	2019/12/29 13:52	配置设置	1 KB
 results	2019/12/29 13:09	文件夹	

图 1 OSpider 相关文件结构图

- **OSpider.exe** 为 OSpider 的主程序, 双击后弹出 OSpider 应用主界面。如果杀毒软件报警, 请忽略, OSpider 很安全。
- **help.pdf** 为当前版本的用户手册, 帮助用户掌握 OSpider 操作技巧, 对软件使用过程中的常见问题给出解决方案, 并提供核心算法思路。
- **icon.ico** 为 OSpider 界面的图标, 不可删除, 删除将导致程序启动异常。
- **addin\_regxy.html** 为 OSpider 的辅助 web 工具, 用于查询指定行政区域的边界坐标信息。
- **property.ini** 为 OSpider 配置文件, 内含爬取参数设置。可当成.txt 打开, 根据需求调整参数 (2.0.0 及以上版本可以通过用户界面直接输入参数, 保留 property.ini 是为使软件向后兼容, 且使用灵活)。
- **results 文件夹** 是 POI 爬取结果的储存目录, 运行 OSpider.exe 后, 爬取结果以.txt 格式存储在该目录中。

### 2.2 操作示范

**需求:** 爬取武汉市内所有的烟酒出售点(liquor store), 并转化为 shp 文件, 为后续探究武汉市烟酒店空间格局打下数据基础。

**操作流程:** 确定参数->执行程序->业务使用

**STEP 1 (定义参数):**



工具->打开百度地图, 判断关键词是否合适, 并考量 POI 的量级 (400 以上以下两种抓法)。尝试‘烟酒店’、‘烟酒超市’、‘烟酒副食’、‘烟酒’作为关键词查询后, 认为‘烟酒’更为合适, 此时, 根据百度地图反馈武汉市有 4813 个相关 POI。

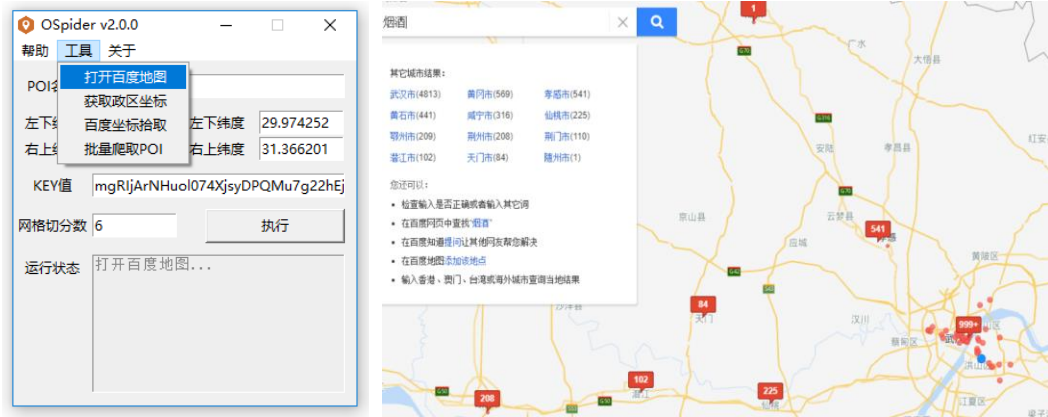


图 2 确认关键词及 POI 量级

工具->获取行政区坐标, 在查询框中输入“武汉市”, 查询得到武汉市行政区的边界信息。其中图 3 中红框标明部分是需要参数: “经纬度:左下角,右上角: 113.707695,29.972898;115.085775,31.367052”。(一般抓行政区 api 比较多, 如果想抓任意矩形, 可以用工具->百度坐标拾取来确定参数)

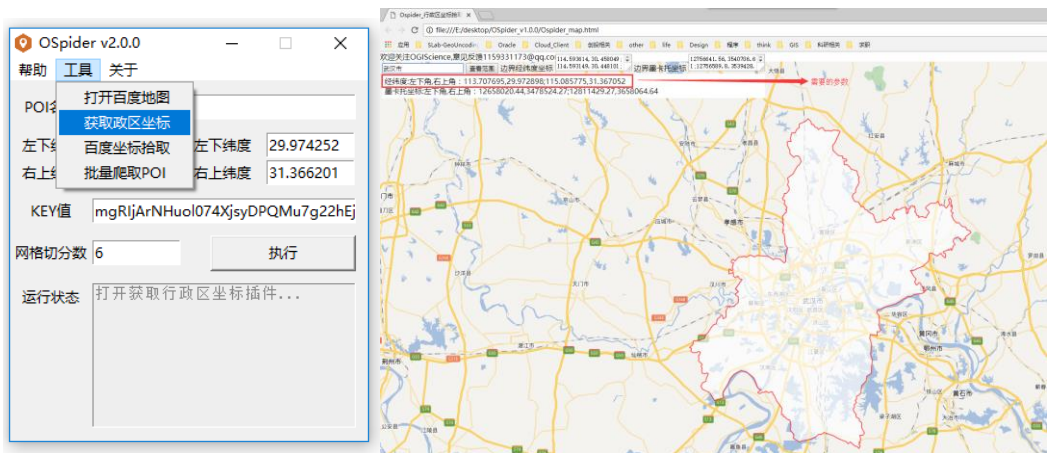


图 3 获得爬取区域的坐标

其他参数包括 KEY 值和网格切分数。KEY 值程序初始内置是小 O 的 KEY, 供大家测试以及应急使用。正常情况下, 希望大家使用自己的 KEY。KEY 可在百度地图开发者平台上申请。OSpider 在抓 POI 的时候先把区域拆分成  $n \times n$  的网格, 再对每一个小网格进行四分递归抓取。这里的网格切分数就上文中的“n”, 根据实验, 当 POI 量大于 400 时, n 取 6 或 9 的时候在城市尺度上能获得教好的爬取效果, 数据丢失极少, POI 量越大越明显。从性价比角度出发, 一般 n 取 6 即可。当 POI 量小于 400 时, n 取 1 直接抓就好, n 的增大不会显著改善数据质量 (有时甚至造成丢失), 反而会使得时间成本明显增多。

另外, 也可直接以文本形式打开 property.ini 设置参数, 其中 poiName-POI 名称、left-左下经度、down-左下纬度、right-右上经度、up-右上纬度、grid\_num-网格切分数、key-KEY 值。设置好 property.ini 后再打开 OSpider.exe, 参数会被自动读取。

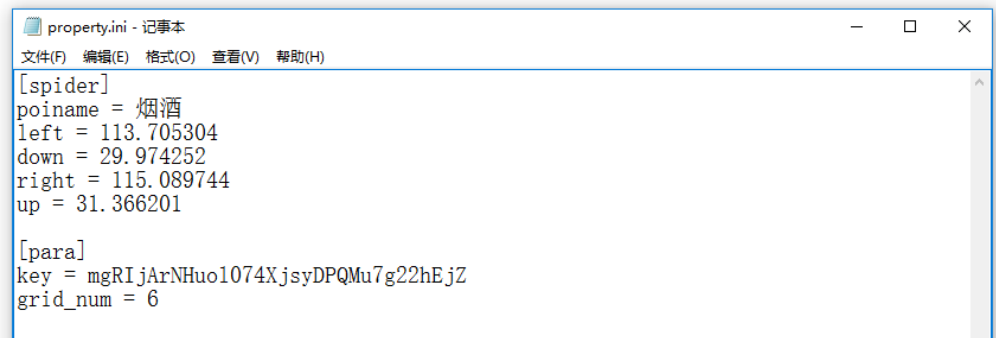


图 4 配置文件

## STEP 2 (执行程序):

完成参数设置后, 直接单击'执行'按钮, 运行程序。相关信息在运行状态栏输出, 目的是防止用户对于程序跑了好一会感到焦虑。程序运行时, 请勿频繁拖动程序。抓取完成后, 会统计耗时、KEY 用量和 POI 数量。这里我们总共抓取了 **4576 个 POI**, **数据完整度高达 95.08%**, 网上绝大多数软件或代码都无法达到这个精度。(抓几百、几十量级的 POI 的时候, 当前的 OSpider 抓取完整度会有迷之波动-但总体是可用的。接下来的版本, 将大幅提升算法在小量级 POI 抓取上的表现以及大量级上的抓取速度)。

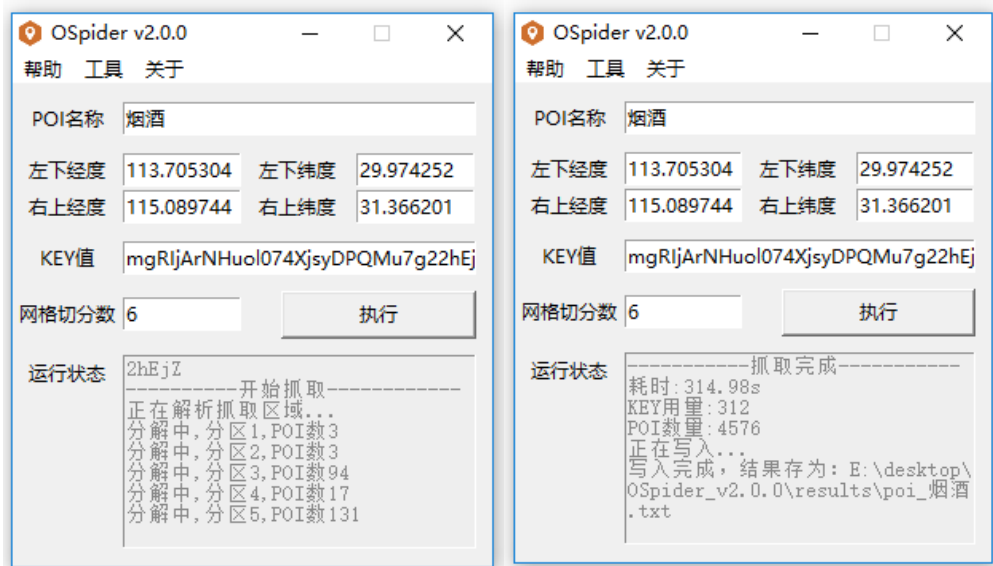


图 5 执行程序

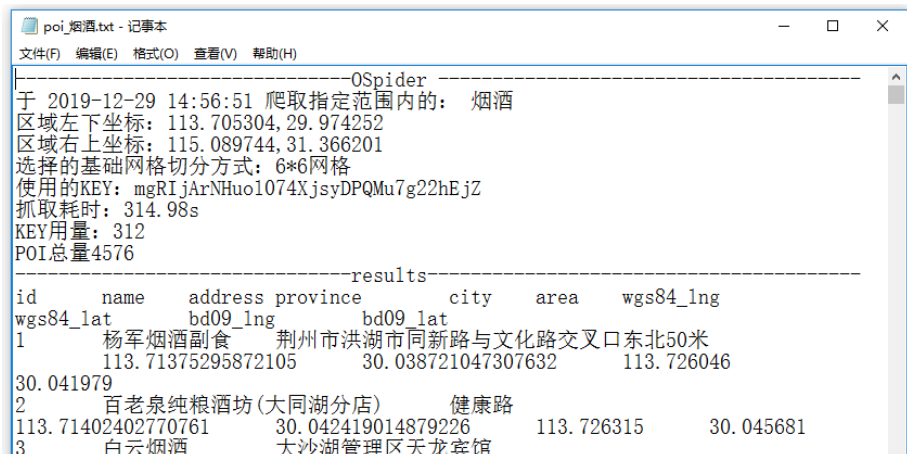


图 6 结果文件

目前每分钟抓取的 POI 在 300~2000 之间, POI 总体量较大的时候, 反而平均抓取速度较快。程序运行时, 稍安勿躁, 喝杯茶, 一会就出结果了。爬取结果以文本文件格式存储在 results 文件夹中, 文件包括爬取信息头和结果两个部分。数据与数据之间采用的是 \t 间隔, 这使得把数据复制到 Excel 表格中时, 能够自动分列。

### STEP 3 (业务使用) :

该部分根据实际需求来定。本范例中, 小 O 先把 txt 中的数据复制到.xlsx 文件中, 再加载到 Arcgis 中, 确认无误后导出为.shp 文件。

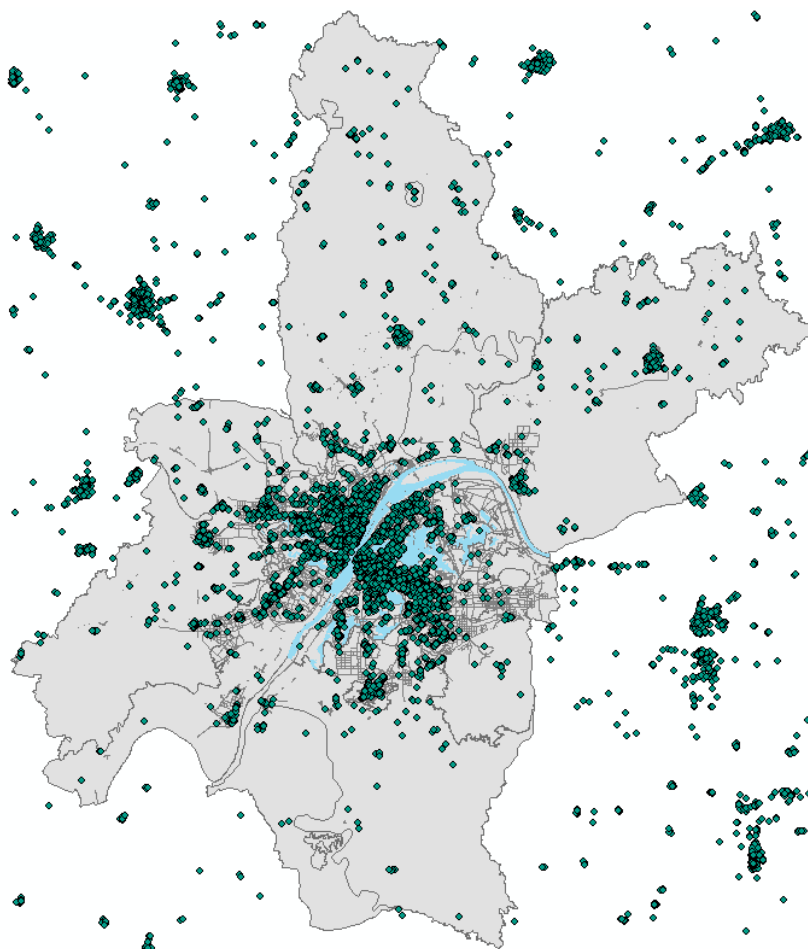


图 7 在 Arcgis 中加载爬取的 POI 数据

## 2.3 FQA

### ● 如何获取我想要爬取的区域的坐标 ?

如果您希望爬取某一行政区的 POI 数据, 请打开 OSpider\_map.html 查询行政区边界坐标 ;如果您希望其他区域的相关坐标, 请使用百度坐标拾取器。在 OSpider 2 中, 这些插件或在线工具已经嵌入在工具菜单中了。

### ● 为什么程序会闪退 ? 或停滞 ?

请检查 OSpider 可执行程序目录下是否有 results 目录, 若没有请手动添加后再次运行程序 ; 确认网络状况是否良好, 保持 ip 稳定 (使用 vpn 切换网络状态时会造成程序中断) ; 确认指定 key 对应的 Place api 是否达到最大使用次数, 可更换 key

值重新尝试。另外, 目前 OSpider 只能抓单个城市及城市之内的 POI, 抓取范围跨城市会报错。

如果问题无法解决, [请直接发送邮件到 1159331173@qq.com](mailto:1159331173@qq.com) 联系开发者解决。

- 为什么 `grid_name` 设置不同, 爬取的同区域同种 POI 数量不同? 为什么爬取的 POI 数量与百度地图 ([map.baidu.com](http://map.baidu.com)) 上显示的不一致?

参见“附录 1 爬取原理与 OSpider 爬取偏差问题”。

- 小 O 帅不帅?

帅!!!

## 3 关于

### 3.1 开发者

武大城市化研究室 小 O

“如果 OSpider 切实帮助了您, 欢迎向小 O 赞赏支持哦。”



### 3.2 Bug 报告与意见反馈

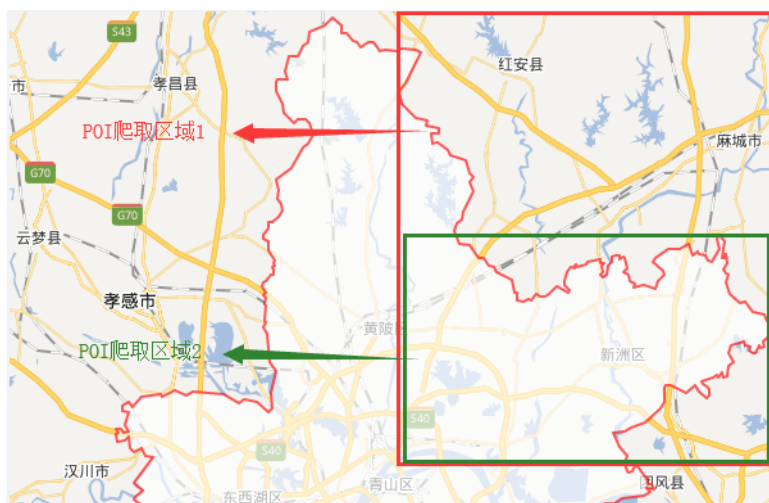
如果您在使用 OSpider 的过程中发现 Bug 或对 OSpider 的使用有什么意见或建议, 请向 1159331173@qq.com 发送邮件, 小 O 收到邮件后会尽快回复。另外, 非常欢迎加入 OSpider 用户群 (QQ) : 939504570。



## 附录 1 爬取原理与 OSpider 爬取偏差问题

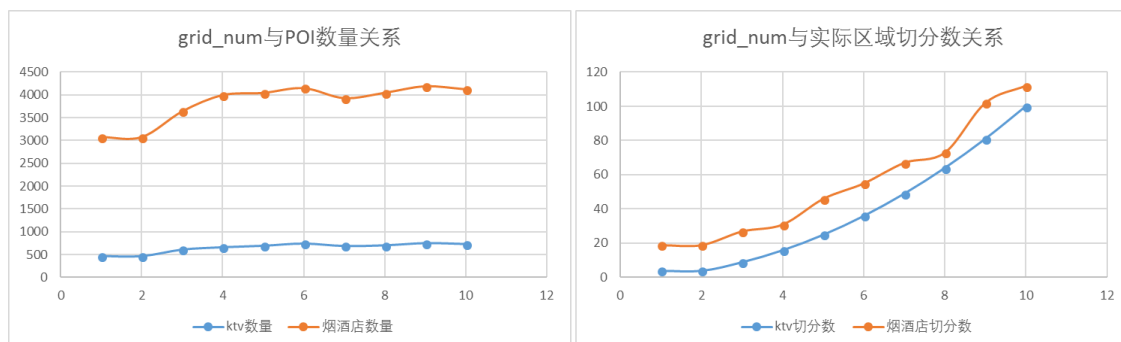
OSpider v1.0.0 使用百度的 Place API 进行爬取, 该 API 能够提供多种场景的 POI 检索功能, 具体包括城市检索、周边检索、矩形区域检索和地点详情检索。其中行政区划区域检索可以检索某一行政区内 (目前最细到城市级别) 的地点信息; 圆形区域检索允许开发者设置圆心和半径, 检索圆形区域内的地点信息; 矩形区域检索可通过设置检索区域左下角和右上角坐标, 检索坐标对应矩形内的地点信息。

使用 Place API 进行爬取的时候主要存在两个问题。第一个是, 不论是行政区还是区域, 结果列表最多只返回 400 个 POI ; 第二个问题是当检索区域存在多个行政区时, 结果只返回其中一个行政区的检索结果 (下图中, 区域 1 返回 POI\_ktv 127 个, 区域 2 返回 POI\_ktv 145 个)。



本文采用手动网格切分+自动四分递归的方法来进行 POI 爬取，完美突破第一个反爬虫机制，并部分绕过第二个反爬虫机制。程序运行时，读取配置文件中的 grid\_num，将爬取区域切分成 grid\_num\*grid\_num 的小区域，然后对每一个小区域进行四分递归爬取。具体而言为，判断该区域的 POI 数量是否在 400 及以上，若在 400 及以上就将该区域切分成四块，爬取其子区域的 POI；如果该区域的 POI 数量在 400 以下，便直接爬取 POI。

一方面由于百度地图 (map.baidu.com) 主页上的检索模式不仅仅是关键词检索, 另一方面百度地图 Place API 的第二种反爬虫机制, 当前 OSpider 爬取大范围 POI 的结果与百度地图查询显示的结果有一定出入, 且爬取结果随着 grid\_num 的不同而不同。虽然还存在一定的不足, 但是通过设置相对合理的 grid\_num, 爬取的城市尺度上的 POI 已经足够普通业务、科研和学习使用了。后续小 O 将持续性的改进算法, 力求



小 O 测试了 ktv 和烟酒店两种 POI 爬取与 grid\_num 的关系，结果如上图所示。建议爬取城市级 POI（以武汉为例）的时候，设置 grid num 为 6 或 9 较好（POI 量大于 400，小于



400 还是直接取 1 的好)。

## 附录 2 WGS84 与 BD09 坐标问题

wgs84 坐标是世界通用、科研常用的地理坐标系, google 采用的坐标系也是 wgs84, 国内从事科研时, 很多时候也用 GCGS2000 坐标系, 这个坐标系和 wgs84 基本一致, 除非特别高精度要求, 一般可以把 wgs84 的数据当 GCGS2000 的或把 GCGS2000 的当 wgs84 的来用。

国内测绘地理信息行业有保密条例, 要求对外使用的至少是经过国测局加密一次的 GCJ-02 坐标系 (火星坐标), 而不同互联网地图服务商又往往会在 GCJ-02 的基础上进行二次加密, 百度二次加密后的坐标系是 BD-09。

OSpider 内置了坐标转换程序, 输出结果中同时保留了 wgs84 和 bd09 坐标。下图中武汉市的底图是 GCGS 2000 地理坐标系加投影的, 红色的点是把 wgs84 坐标定义为 GCGS 2000 地理坐标系后加载入 Arcgis 的结果, 而绿色的点是把 bd-09 坐标定义为 GCGS2000 地理坐标系后加载入 Arcgis 的结果。可以看出, 绿色点偏差明显很多点掉入了江湖, 而红色点与底图相匹配。

