**携程网游客评论数据采集程序设计**

# 采集方案设计思路

## 1.1 目的与需求

根据课程作业要求，拟选择开发爬虫程序对“携程网”有关崂山景区的游客评论文本进行采集并进行分析。因此采集目标网站为携程网 (https://www.ctrip.com/) ，数据采集内容包括评论文本、评分、评论时间等。

## 1.2 目标网页分析

首先将访问崂山景区的携程页面，经过页面抓包分析得知评论内容来源于AJAX请求，该请求是一个post请求，包含了一般常见的请求头域。

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

图 1 请求方法和请求地址图

图形用户界面, 应用程序

描述已自动生成

图 2 请求标头图

请求参数部分包含了两个字符串请求参数和json载荷，如下图所示：

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

图 3 请求载荷图示

分析请求载荷内容可知，其中包含了当前页码、每页评论数和一些其他参数，具体而言，我们需要寻找pid、cid和x-id。

其中pid是景区id，只要传入了相对应的pid就可以获取相应景区的信息，因为本次主要采集崂山景区的评论文本，因此pid是固定的，进一步在网页中全局搜索pid，结果在相应页面js中定位到，如下图所示，由于信息被保存在js代码中且要求较少，考虑程序开发时使用正则表达式提取。

文本

中度可信度描述已自动生成

图 4 响应页面中的关键参数

其次，全局搜索\_fxpcqlniredt，在cookie中找到了该信息，因此为了本程序的稳定性，拟使用requests模块的session对象或httpx中的client对象来发送连续请求并存储cookie信息。

表格

描述已自动生成

图 5 cookie中的关键参数

全局搜索x-traceID，在js代码中找到了其生成位置，很显然局部变量e中缓存了\_fxpcqlniredt的值，而x-traceID由\_fxpcqlniredt、时间戳和一个随机数组成。

文本

描述已自动生成

图 6 x-tid的构建流程

至此，所有请求分析完毕，只要模拟该请求就可以获取所需要的响应。

## 1.3 采集思路设计

根据抓包分析的结果，拟定采集思路如下：

首先创建访问崂山景区的携程页面获取cookie及响应页面，提取需要的信息构建请求头；

其次构建并循环发送请求，遍历响应内容，解析和提取需要的信息，如评论文本、评分、评论时间等；

随后将提取的数据依次存入数据库，为避免数据重复，我们将评论内容字段设置为不重复，当相同的评论存入数据库时会被跳过；

最后，读取存储的结果并对其进行可视化分析。

# 程序运行流程图

爬虫程序将按照以下流程进行运行：发送HTTP请求获取页面内容；解析AJAX内容; 提取评论信息；进行数据去重与数据去重；将采集数据存储再MySQL数据库中；循环采集； 限制请求频率。

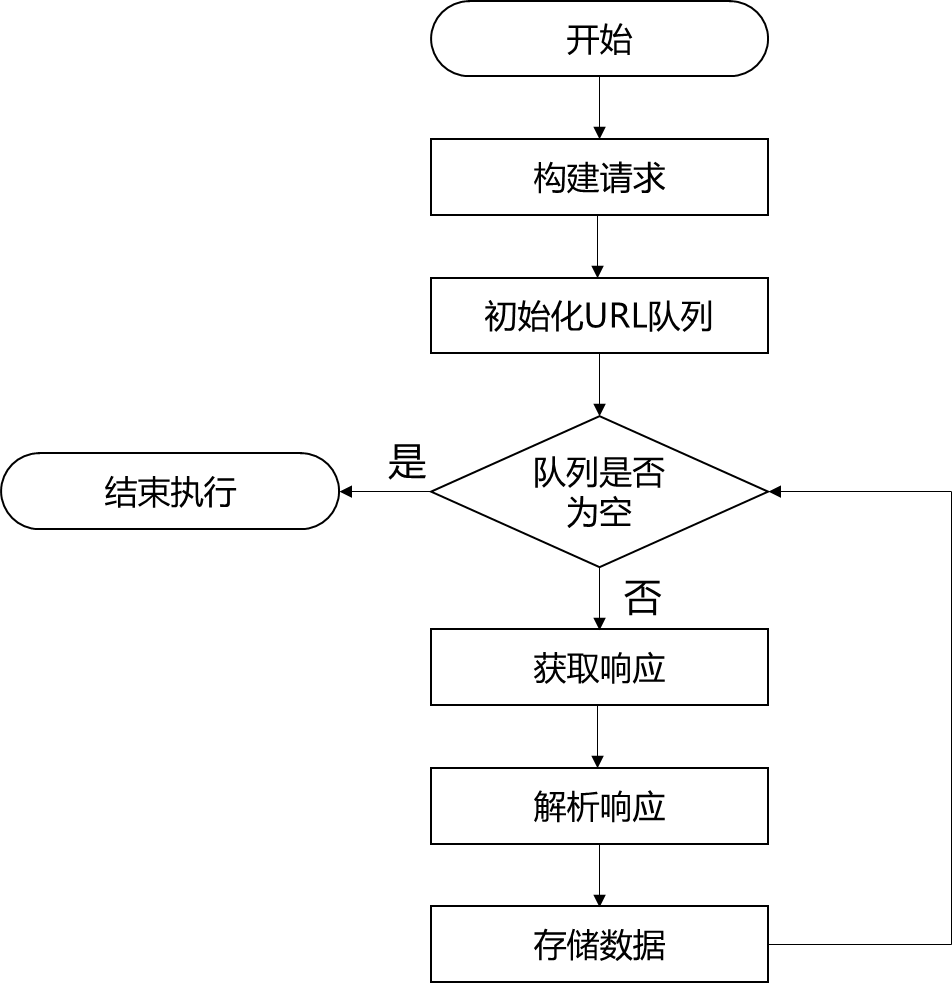


图 7 程序运行流程图

# 代码功能描述

本次爬虫开发采用面向对象的方式。

## 3.1初始化参数

首先设置实例的初始化参数，包括程序中对象通用的属性如下：

文本

描述已自动生成

图 8 初始化参数图

## 3.2 获取cookie和请求参数

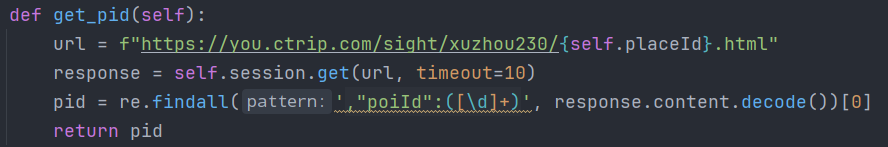


图 9 获取关键参数方法

定义方法访问崂山携程网主页获取cookie信息并使用正则表达式匹配获取关键参数pid。

## 3.3 发送请求

文本

描述已自动生成

图 10 构建参数与发送请求方法

根据前面的分析，在此处构造并发送请求，pid不再赘述，guid存储在session变量的cookie中，通过键GUID获取，由于本次需要构建post参数，因此直接将get参数置于URL中，如前面分析，x-tid是guid、时间戳和随机数拼接而成。

post参数中传入了 pid，guid和index，此处的index是变化的，代表着不同的页码，因此将其做出方法的参数传入，该方法最后返回未进行解析的服务器响应。

## 3.4 解析响应

文本

描述已自动生成

图 11 解析响应方法

解析响应的方法中传入请求来的响应对象，响应对象的json()方法可以直接将json字符串转化为python字典，通过对字典的操作获取评论列表，最后用列表推导式将所需数据提取出来。限于列表推导式的局限性，此处无法过滤掉空白评论，因此将对空白文本的筛选和去重操作写在数据库保存方法中。

考虑到评论文本复杂性质，此处对文本进行了初步清洗，主要包括去除特殊符号并将所有的字母转化为小写。

## 3.5 主要流程

屏幕的截图

描述已自动生成

图 12 程序入口与主流程

如上一节流程图所示，本程序首先收集请求需要的参数，其次使用range函数构建了URL序列并对其迭代，获取每一个URL所对应的数据并对其解析，最终保存在数据库中。

为了增加程序的健壮性，在循环处增加了随机等待，并建立异常捕获机制，一旦解析程序报错，说明爬取速度过快，因此等待10-15秒后再继续爬取。

背景图案

描述已自动生成

图 13 爬取结果示意图（部分）

合计12833条评论信息共计1284页，考虑随机等待时间下约每秒10条，约2分钟内采集完毕，本次作业要求300条数据以上，考虑到存在重复值异常值等原因，因此这里循环60次，拟采集数据600条。

# 网页数据存储方案

数据可以存储到本地文件（如CSV、JSON）或数据库（如MySQL、MongoDB），本次采集的数据选择存储在mysql数据库中。

## 4.1 数据库物理结构设计

结合网站响应数据的具体结构与作业要求，设计数据库物理结构如下：

表 1 评论信息字段表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 字段类型 | 字段长度 | 主键 | 重复 | 说明 |
| id | int | auto | 是 | 否 | 评论主键 |
| score | int | 2 | 否 | 否 | 游客评分 |
| content | varchar | auto | 否 | 是 | 评论内容 |
| datetime | datetime | 6 | 否 | 是 | 发表日期 |
| location | varchar | 20 | 否 | 是 | 发表位置 |

## 4.2 数据库存储代码

数据入库方法中，首先建立与数据库的链接，然后查询数据表如果没有将创建新表，并按照物理结构设计的结果设置数据库的物理结构，然后对所用评论数据进行遍历入库，最后关于数据库的连接。

文本

描述已自动生成

图 14 数据存储方法图

# 可视化方法

## 5.1 用户评分分布可视化图

读取爬取数据中的用户评分情况，计算评分分别为1-5分的用户占比情况，结果显示，崂山景区用户评分普遍较高，极少用户给出了3分的评分，大部分用户对崂山景区十分满意。

图表, 饼图

描述已自动生成

图 15 用户评分分布可视化图

## 5.2 用户评论词云图

读取用户评论文本数据，经过压缩去重，使用jieba分词并去除停留词后，使用python第三方库wordcloud生成词云图如下：

墙上有涂鸦

中度可信度描述已自动生成

图 16 用户评论词云图

## 5.3 用户地理分布可视化

读取用户地理位置数据，根据用户发表评论的地址信息，使用Python第三方包geopandas绘制地理可视化分布图如下：

地图

描述已自动生成

图 17 用户地理位置可视化图