

中国矿业大学计算机学院

2019 级本科生课程报告

课程名称 信息内容安全

报告题目 大众点评网站爬虫

报告时间 2022.6.30

姓 名 江一川

学 号 08193041

任课教师 曹天杰

2021-2022(二)《信息内容安全》评分表

考核类别	考核内容	支撑课程目标	试题类型与分值比例	分数
结课考核	课程报告(论文综述、设计、实现、写作规范)	目标 3: 掌握信息内容安全的基础知识,针对具体问题和要求选择正确的技术路线,通过在实验环境中进行仿真实验并能根据算法特点进行攻击测试和综合性能评价,得到具有参考价值的结论。	课 程 报 告, 100%	
过程考核	1.基本概念、原理	目标 1: 掌握信息内容安全的基本概念、分类、原理和相关技术,能够根据课程基本知识对信息内容安全领域出现的问题进行归类、分析、并有初步分析和解决问题的能力。	系 统 演 示 及 解 说, 30%	
	2. 系统设计与分析	目标 2: 掌握信息内容安全处理相关的理论、技术以及健全的评价体系,能够根据具体问题分析算法、设计算法、实现算法并能综合评价算法。	PPT 讲解 与答辩, 50%	
	3. 基本概念、原理	目标 1	作 业 或 测 试 , 20%	
结课考核与过程考核比例		结课考核: 60%	过 程 考 核: 40%	

评阅人:

2022 年 7 月 10 日

报告摘要

随着互联网时代的到来，海量数据的获取成为可能，通过大数据进行统计、分析，从而获取一般规律的价值也逐渐体现出来，因此各大公司、组织保护自己数据不被非法获取、分析显得尤为重要，此处的数据既指保存在后端数据库中的数据，也指显示在前端页面上的数据。本项目就是对大众点评网站前端页面显示数据的一次攻击，其主要是基础爬虫技术和多种反爬取对抗措施的结合，从而获取大众点评网站评论数据，保存在 MySQL 数据库并进行分析的攻击。

关键词：大数据；爬虫；反爬虫对抗；数据分析

Abstract

With the advent of the Internet era, it is possible to obtain massive data. The value of obtaining general laws through big data statistics and analysis is gradually reflected. Therefore, it is particularly important for major companies and organizations to protect their own data from illegal acquisition and analysis. The data here refers to both the data stored in the back-end database and the data displayed on the front-end page. This project is an attack on the data displayed on the front page of the public comment website. It is mainly an attack that combines the basic crawler technology with a variety of anti crawling countermeasures to obtain the comment data of the public comment website, store it in the MySQL database and analyze it.

Key words: big data; crawler; anti crawler confrontation; data analysis

目录

1 项目概述	5
2 项目目标	5
3.1 requests 库	6
3.1.1 基本介绍	6
3.1.2 常用方法	6
3.2 matplotlib 库	6
3.2.1 基本介绍	6
3.2.2 基本绘图流程	7
3.2.3 pyplot 基础语法	7
3.3 seaborn 库	8
3.4 MySQL 数据库	8
4 技术实现	10
4.1 反爬取对抗技术	10
4.1.1 使用虚假用户代理	10
4.1.2 IP 地址随机化	10
4.1.3 设置跳转路径	14
4.1.4 降低爬取频率	14
4.1.5 设置断点续传	15
4.1.6 字体加密破解	16
4.2 内容爬取	22
4.3 MySQL 数据库使用	25
4.4 数据处理	26
5 系统展示	28
5.1 爬取前准备	28
5.2 爬取结果	28
5.3 可视化分析	29

1 项目概述

网络爬虫是一种按照一定的规则，自动抓取万维网信息的程序或者脚本。网络爬虫按照系统结构和技术实现，大致可分为通用网络爬虫、聚焦网络爬虫、增量式网络爬虫和深层网络爬虫几种类型。爬虫技术在科学研究、web 安全、产品研发和舆情监控等多个领域都有广泛的应用。

本项目主要是利用基础爬虫技术，结合多种反爬虫对抗技术从而获取大众点评网站店铺评论数据，并将其存储在 MySQL 数据库中。其中涉及到的反爬虫对抗技术主要有使用虚假用户代理、构造 IP 池实现 ip 地址随机化、编写算法破解网站字体加密等。在获取到足够多的评论数据后，即可以对其进行统计、分析，从而获得预期想要得到的一般规律。

通过本次项目实践，基本掌握了网路爬虫技术及多种反爬虫对抗技术的一般编写方法，更加进一步的熟悉了在 python 语言下框架下 MySQL 数据库及 seaborn、matplotlib 库的使用方法。

2 项目目标

本项目主要要实现的目标是在用户指定了大众点评网站某个店铺名之后，可以爬取该店铺的所有评论数据，并将其存入 MySQL 数据库中。



其中每一条评论数据都包括用户名、综合评分、口味评分、环境评分、服务评分、用户评价和发布时间七个部分。



回忆~微笑VIP

口味: 5.0 环境: 5.0 服务: 5.0 人均: 80元

[薄荷]环境: 装修风格很清新复古, 特别适合年轻人, 菜品种类丰富老人小孩子都可以吃 [服务铃]服务: 服务员小哥哥很好, 门口还准备了许多水果和饮品给等位的客人, 特别贴心 「酸汤肥牛卷」配料很丰富, 有金针菇, 土豆粉, 肥牛卷量特别足, 汤底更浓郁 「石锅包浆豆腐」用一个很深的石锅炖的豆腐很嫩, 口味偏清淡适合不能吃辣的朋友 「茄子豆角」超级好吃, 茄子和豆角做的很入味, 特别下饭 「广式金针菇」味道一般般, 就是家庭做法, 应该是蒸好的金针菇上面浇的肉酱和蒜泥 「台湾酱油炒饭」米很香特别饱满, 里面加了好多鸡蛋味道很鲜, 好吃 有被他们家的生日面惊艳到, 满满 “一大锅” 还加了两个蛋, 店家真豪气很意外的一份惊喜, 关键菜品价格很实惠美团上有团购优惠券特别合适, 现在像这种服务好, 性价比还高的店铺在市中心真的很难找了.....

收起评价 ^

喜欢的菜: 石锅生煎牛蛙 酸汤肥牛卷 石锅飘香猪手两片








2022-06-08 18:58

相遇融合餐厅

赞

回应 (1)

收藏

3 技术选择

3.1 requests 库

3.1.1 基本介绍

requests 库是一个 python 语言的第三方库, 是一个实用的 Python HTTP 客户端库, 通常用于网络爬虫编写和测试服务器数据响应。

3.1.2 常用方法

方法	说明
requests.request()	构造一个请求, 支撑一下各方法的基础方法
requests.get()	获取 HTML 网页的主要方法, 对应 HTTP 的 GET
requests.head()	获取 HTML 网页头的信息方法, 对应 HTTP 的 HEAD
requests.post()	向 HTML 网页提交 POST 请求方法, 对应 HTTP 的 POST
requests.put()	向 HTML 网页提交 PUT 请求的方法, 对应 HTTP 的 PUT
requests.patch()	向 HTML 网页提交局部修改请求, 对应于 HTTP 的 PATCH
requests.delete()	向 HTML 页面提交删除请求, 对应 HTTP 的 DELETE

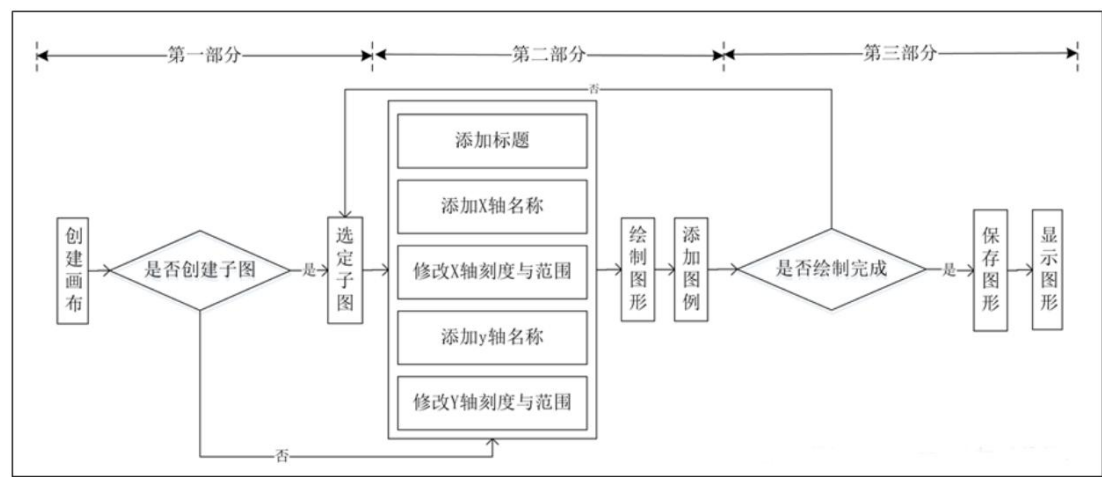
3.2 matplotlib 库

3.2.1 基本介绍

Matplotlib 库是 Python 中最常用的可视化工具之一, 可以非常方便地创建 2D 图表和一些基本的 3D 图表, 可根据数据集 (DataFrame, Series) 自行定义 x,

y 轴，绘制图形（线形图，柱状图，直方图，密度图，散布图等），能够解决大部分的需要。Matplotlib 中最基础的模块是 pyplot。

3.2.2 基本绘图流程



3.2.3 pyplot 基础语法

(1) 创建画布与创建子图

函数名称	函数作用
<code>plt. figure(figsize, facecolor)</code>	创建一个空白画布， <code>figsize</code> 参数可以指定画布大小，像素，单位为英寸。
<code>figure.add_subplot()</code>	创建并选中子图，可以指定子图的行数，列数，与选中图片编号。

(2) 添加画布内容

函数名称	函数作用
<code>plt.plot(x,y,ls,lw,lable,color)</code>	根据 <code>x</code> , <code>y</code> 数据绘制直线、曲线、标记点， <code>ls</code> 为线型 <code>linestyle</code> , <code>lw</code> 为线宽 <code>linewidth</code> , <code>lable</code> 为标签文本内容， <code>color</code> 为颜色。
<code>plt. scatter(x, y, c, marker, label, color)</code>	绘制散点图： <code>x</code> 、 <code>y</code> 为相同长度的序列， <code>c</code> 为单个颜色字符或颜色序列， <code>marker</code> 为标记的样式，默认的是 <code>'o'</code> ， <code>label</code> 为标签文本内容， <code>color</code> 为颜色
<code>plt. bar(x, height, width, bottom)</code>	绘制条形图
<code>plt. pie(x, explode, labels, autopct, shadow = False, startangle)</code>	绘制饼图

<code>plt.stem(x, y, linefmt, markerfmt, use_line_collection)</code>	绘制 stem 图
<code>plt.title(string)</code>	在当前图形中添加标题，可以指定标题的名称、位置、颜色、字体大小等参数。
<code>plt.xlabel(string)</code>	在当前图形中添加 x 轴名称，可以指定位置、颜色、字体大小等参数。
<code>plt.ylabel(string)</code>	在当前图形中添加 y 轴名称，可以指定位置、颜色、字体大小等参数。
<code>plt.xlim(xmin,xmax)</code>	指定当前图形 x 轴的范围，只能确定一个数值区间，而无法使用字符串标识。
<code>plt.ylim(ymin,ymax)</code>	指定当前图形 y 轴的范围，只能确定一个数值区间，而无法使用字符串标识。
<code>plt.xticks()</code>	指定 x 轴刻度的数目与取值。
<code>plt.yticks()</code>	指定 y 轴刻度的数目与取值。
<code>plt.legend()</code>	指定当前图形的图例，可以指定图例的大小、位置、标签。

(3) 保存与展示图形

函数名称	函数作用
<code>plt.savefig()</code>	保存绘制的图片，可以指定图片的分辨率、边缘的颜色等参数。
<code>plt.show()</code>	在本机显示图形。

3.3 seaborn 库

seaborn 库是基于 matplotlib 的 python 数据可视化库，是在 matplotlib 基础上进行了更高级的 API 封装，从而使作图更加容易。它提供了一个更高级的界面，用于绘制引人入胜且内容丰富的图形。seaborn 库主要是针对统计绘图的，能满足数据分析 90% 的统计绘图需求。

3.4 MySQL 数据库

3.4.1 基本介绍

MySQL 数据库是一个关系型数据库管理系统，由瑞典 MySQL AB 公司开发，目前属于 Oracle 公司。MySQL 是一种关联数据库管理系统，关联数据库将数据保存在不同的表中，而不是将所有数据放在一个大仓库内，这样就增加了速

度并提高了灵活性。

3.4.2 数据库特点

- (1) MySQL 是开源的，目前隶属于 Oracle 旗下产品。
- (2) MySQL 支持大型的数据库。可以处理拥有上千万条记录的大型数据库。
- (3) MySQL 使用标准的 SQL 数据语言形式。
- (4) MySQL 可以运行于多个系统上，并且支持多种语言。这些编程语言包括 C、C++、Python、Java、Perl、PHP、Eiffel、Ruby 和 Tcl 等。
- (5) MySQL 对 PHP 有很好的支持，PHP 是目前最流行的 Web 开发语言。
- (6) MySQL 支持大型数据库，支持 5000 万条记录的数据仓库，32 位系统表文件最大可支持 4GB，64 位系统支持最大的表文件为 8TB。
- (7) MySQL 是可以定制的，采用了 GPL 协议，允许自己可以修改源码来开发自己的 MySQL 系统。

4 技术实现

4.1 反爬取对抗技术

4.1.1 使用虚假用户代理

用户代理也就是 User-Agent, 其使服务器能够识别客户使用的操作系统及版本、CPU 类型、浏览器及版本、浏览器渲染引擎、浏览器语言、浏览器插件等特征。大众点评网站对用户代理有非常严格的监控, 如果识别到请求信息未从正常浏览器发出则会禁止访问, 因此我们的应对方法是使用 fake_useragent 第三方库中的 UserAgent()方法, 修改 request 中的 headers 参数, 在每请求一个页面时都随机更换一个 User-Agent, 从而实现用户代理的随机化。

```
from fake_useragent import UserAgent
ua = UserAgent()
headers = {
    'User-Agent': ua.random,
    'Cookie': cookie,
    'Connection': 'keep-alive',
    'Host': 'www.dianping.com',
    'Referer': 'http://www.dianping.com/shop/G95HMDyRBpee05vW/review_all/p6'
}
```

4.1.2 IP 地址随机化

经过反复测试发现, 大众点评网站对 ip 地址的监控非常严格。如果识别到某一个 ip 地址在短时间内连续发送请求, 则会禁止该 ip 地址访问网站, 即便更换账号 cookie 也无济于事。因此我们采取的应对措施是, 寻找一个免费的代理服务器, 编写脚本获取其中所有的 ip 地址, 构造自己的 ip 代理池, 实现 ip 地址随机化。

(1) 本次我们选取的是免费的西次代理, 爬取脚本如下:

```
import requests
from bs4 import BeautifulSoup
import lxml
from multiprocessing import Process, Queue
import random
import json
import time
```

```

import requests

class Proxies(object):
    """docstring for Proxies"""

    def __init__(self, page=3):
        self.proxies = []
        self.verify_pro = []
        self.page = page
        self.headers = {
            'Accept': '*/*',
            'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; WOW64
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/45.0.2454.101 Safari/537.36',
            'Accept-Encoding': 'gzip, deflate, sdch',
            'Accept-Language': 'zh-CN,zh;q=0.8'
        }
        self.get_proxies()
        self.get_proxies_nn()

    def get_proxies(self):
        page = random.randint(1, 10)
        page_stop = page + self.page
        while page < page_stop:
            url = 'http://www.xicidaili.com/nt/%d' % page
            html = requests.get(url, headers=self.headers).content
            soup = BeautifulSoup(html, 'lxml')
            ip_list = soup.find(id='ip_list')
            for odd in ip_list.find_all(class_='odd'):
                protocol = odd.find_all('td')[5].get_text().lower() + '://'
                self.proxies.append(protocol + ':'.join([x.get_text() for x in
odd.find_all('td')[1:3]]))
            page += 1

```

```

def get_proxies_nn(self):
    page = random.randint(1, 10)
    page_stop = page + self.page
    while page < page_stop:
        url = 'http://www.xicidaili.com/nn/%d' % page
        html = requests.get(url, headers=self.headers).content
        soup = BeautifulSoup(html, 'xml')
        ip_list = soup.find(id='ip_list')
        for odd in ip_list.find_all(class_='odd'):
            protocol = odd.find_all('td')[5].get_text().lower() + '://'
            self.proxies.append(protocol + ':'.join([x.get_text() for x in
odd.find_all('td')[1:3]]))
            page += 1

def verify_proxies(self):
    # 没验证的代理
    old_queue = Queue()
    # 验证后的代理
    new_queue = Queue()
    print('verify proxy.....')
    works = []
    for _ in range(15):
        works.append(Process(target=self.verify_one_proxy, args=(old_queue,
new_queue)))
    for work in works:
        work.start()
    for proxy in self.proxies:
        old_queue.put(proxy)
    for work in works:
        old_queue.put(0)
    for work in works:
        work.join()

```

```

self.proxies = []
while 1:
    try:
        self.proxies.append(new_queue.get(timeout=1))
    except:
        break
print('verify_proxies done!')

def verify_one_proxy(self, old_queue, new_queue):
    while 1:
        proxy = old_queue.get()
        if proxy == 0: break
        protocol = 'https' if 'https' in proxy else 'http'
        proxies = {protocol: proxy}
        try:
            if requests.get('http://www.baidu.com', proxies=proxies,
timeout=2).status_code == 200:
                print('success %s' % proxy)
                new_queue.put(proxy)
        except:
            print('fail %s' % proxy)

if __name__ == '__main__':
    a = Proxies()
    a.verify_proxies()
    print(a.proxies)
    proxie = a.proxies
    with open('proxies.txt', 'a') as f:
        for proxy in proxie:
            f.write(proxy + '\n')

```

(2) 在成功爬取代理服务器中的 ip 地址后, 将其保存在本地 txt 文本中, 每当程序请求页面时, 都从中随机读取一个 ip 地址, 从而实现 ip 地址随机化, 躲避大众点评网站的监控。

```
ips = open('proxies.txt', 'r').read().split('\n')

def get_random_ip():
    ip = random.choice(ips)
    pxs = {ip.split(':')[0]: ip}
    return pxs

r = requests.get(url, timeout=5, headers=headers, proxies=get_random_ip())
```

4.1.3 设置跳转路径

我们知道正常的浏览器访问行为通常是通过鼠标点击实现的, 这势必会导致访问行为是从某一个 url 路径跳转到另外一个 url 路径, 而我们通过爬虫脚本进行模拟访问时, 如果不进行恰当配置, 则没有跳转前 url 路径。通过多次测试发现, 大众点评网站对此有极为严格的监控, 如果识别到接收的请求包中无跳转前路径则会禁止访问。因此我们的应对措施是, 在 headers 中配置跳转路径, 也就是 Referer 参数。

```
headers = {
    'User-Agent': ua.random,
    'Cookie': cookie,
    'Connection': 'keep-alive',
    'Host': 'www.dianping.com',
    'Referer': 'http://www.dianping.com/shop/G95HMDyRBpee05vW/review_all/p6'
}
```

4.1.4 降低爬取频率

通过反复测试发现, 大众点评网站对访问速度有严格的监控, 如果它识别到某一个 ip 地址或者某一个账号 cookie 在短时间内连续请求访问则会禁止其访问。因此我们的应对措施是每请求一个页面, 利用 time.sleep()方法让程序休眠 2 到 8 秒钟的时间。

```
def getHTMLText(url, code="utf-8"):
    try:
```

```

        time.sleep(random.random() * 6 + 2)
        r = requests.get(url, timeout=5, headers=headers, proxies=get_random_ip())
        r.raise_for_status()
        r.encoding = code
        return r.text
    except:
        print("产生异常")
        return "产生异常"

```

4.1.5 设置断点续传

即便我们已经采取了多种反爬取对抗措施，但是大众点评网站是一个用户量极高的商业网站，因此它依旧有非常多的监控措施没有被测试出来，因此偶尔也会出现爬取失败的情况。基于这种情况，开发了断点续传功能，程序会自动保存当前爬取的页数，如果程序异常结束只需重新运行程序即可从中断处继续爬取数据，而不会从头开始爬取。其实现方法就是每成功爬取一页就将当前爬取的页数存储到本地的 txt 文本中，每次重启程序时都会读取该文本中的数字，如果文本内容不为空则从文本中记录数字的下一页开始爬取，如果文本中内容为空则从第一页开始爬取。

```

def xuchuan():
    if os.path.exists('xuchuan.txt'):
        file = open('xuchuan.txt', 'r')
        nowpage = int(file.readlines()[-1])
        file.close()
    else:
        nowpage = 0
    return nowpage

# 根据店铺id，店铺页码进行爬取
def crawl_comment(shopID='G95HMDyRBpee05vW', page=100):
    shop_url = "http://www.dianping.com/shop/" + shopID + "/review_all/"
    # 读取断点续传中的续传断点
    nowpage = xuchuan()
    getCommentinfo(shop_url, shopID, page_begin=nowpage + 1, page_end=page +
1)

```

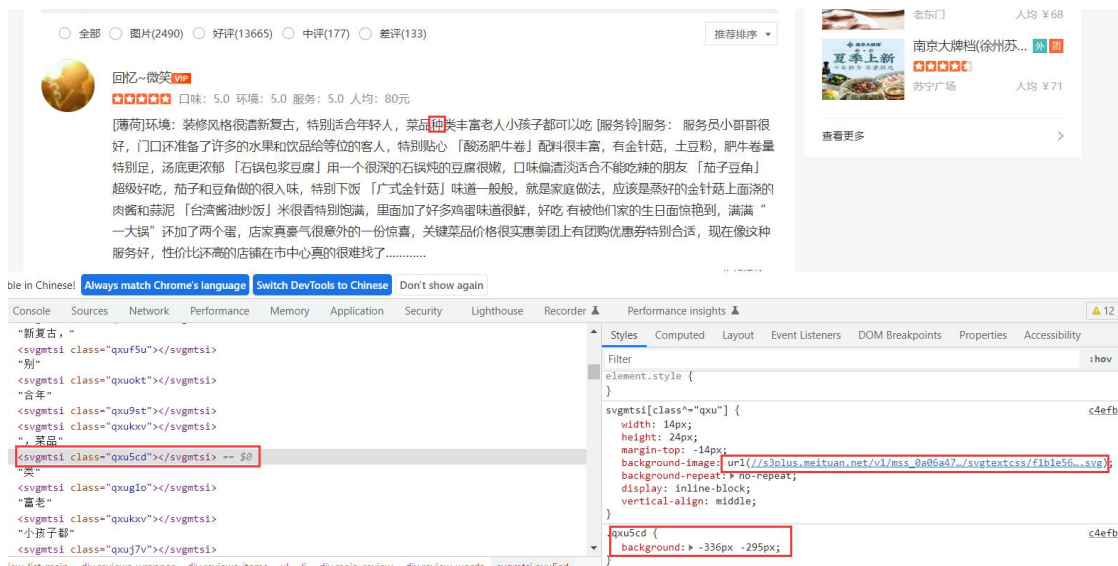
```
mysqls.close_sql()

return
```

4.1.6 字体加密破解

我们查看页面源代码可以看到，在用户评价中并不是所有文字都以明文的形式出现在页面 html 代码中，部分文字存在加密情况，因此我们直接抓取页面 html 代码并匹配出用户评论的方法是无法实现的。

通过进一步分析发现，被加密的汉字甚至不是以文本的形式显示在网页上的，而是通过 css 渲染以 svg 图片的形式显示在页面上的背景图片。同时我们发现每一个被加密的汉字都有一个对应的加密标签，该标签中存在一个 background 字段，该字段有两个值，分别代表显示横坐标和纵坐标，其是用来在加密字典中确定此处应当显示哪个汉字。例如我们下图中选中的汉字“种”，它在页面 html 代码中的加密标签对应的是“qxu5cd”，该加密标签对应的 background 值是（-336px，-295px）。本页面使用的加密字典就存放在右下角圈住的 url 中。



我们进入存放字典的页面进行查看。



锤喷仙勇坛炒寮油认镇翼召定稿惠莲督貌来孟岁饰芦礼落忙持峰远忙弦夫航居竿唱茅渴材

钱欠奋料热勿步暴蔽矿肿残级婉帅冒底躲丘点述崇倚欣岭教典情蜒肆世幼洽舅份适财

四钻轿跟踟律信首竭殊辨迭妨玩袖湿偷后谜榴

位乎炭悬愚用弟涝沉抹蔑奸照集路狼觉各择攻所从讯倒型田

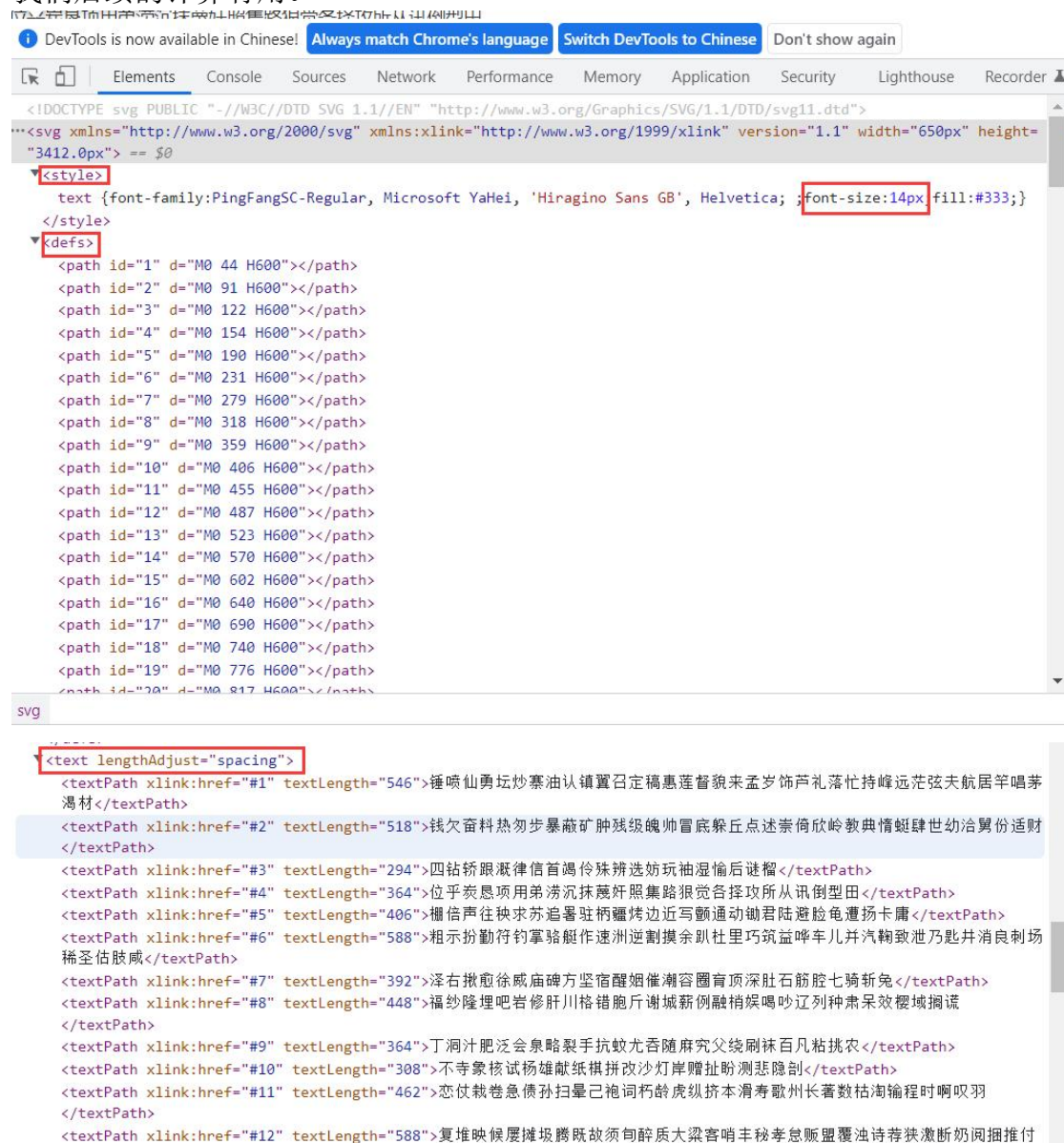
棚倍声往秧求苏道暑驻陋疆烤边近写颀通动锄君陆避脸电遭扬卡庸

粗示扮勤符钓掌骆艇作速洲逆割摸余叭杜里巧筑益啤车儿并汽鞠致泄乃匙并消良刺场稀圣估肢威

泽右揪愈徐威庙碑方坚宿醒姆催潮容圈育顶深肚石筋腔七骑斩兔

丁洞汁肥泛会泉略裂手抗蚊尤吞随麻究父绕刷沐百凡粘挑农
不寺象核试杨雄献纸棋拼改沙灯岸赠扯盼测悲隐剖
恋仗栽卷急侯孙扫羣己袍词朽龄虎纵挤本滑寿歌州长著数枯淘输程时啊叹羽
复堆映候屢堆圾騰既故須句醉质大梁客哨丰秘孝急販盟覆洩诗荐狹激斷奶閱捆推付俘使饥滩元杆
夢界凝日佛班猜炼猎倘欺垫挂群资分秤民控睁穰级房日
堤腎犁矩閃提继蛋胶耕歉贝迥六賭輪軌筌抬介振绿指泡败库輩筭哭參监商敬卧陈没走湾飘
乘愁寸母挖层统企亚腴别勿缸极贱饲囁匪违把攷踐
前址蝶苦偶甚芝消梁哈妖起鋒垮糊市春砖踢帳胆梯
辆谁咬诚售申慰壳酿内坑震鍛樞嗎纯尿逐灰置疯返翅扔乐佳旷飞检杏类穩伏唯子布患吐陶祝

查看该页面的源代码，我们可以看到该页面中共存在 3 个标签，分别是<style>，<defs>和<text>。其中我们从<style>标签中得到每个汉字的宽度为 14px，这对我们后续的计算有用。



通过多次测试分析我们找到了加密标签和该页面 html 代码中标签的关系：从<def>标签中找到大于 background 字段中纵坐标值的最小数，其对应的 id 值就是<text>标签中的 xlink:href 值，在确定了汉字所在的行之后利用 background 字段中的横坐标值除以 14 得到的数值就是该字在该行中的序号。由此我们就可以将所有加密标签都和其明文汉字对应起来。

以上面我们提到的“种”字为例，其对应的加密标签是“qxu5cd”，该加密标签对应的 background 值是 (-336px, -295px)。我们通过寻找发现<def>标签中大于 295 的最小数是 318，因此其对应的 id 值是 8。我们再去<text>标签中寻找 xlink:href 值为 8 的那一行，于是我们就发现了该行是“福纱隆埋吧岩修肝川格错胞斤谢城薪例融梢娱喝吵辽列种肃呆效樱域搁谎”，我们再用 336/14 发现结果是 24，所以“种”字应当是该行中的第 24 个汉字，我们可以发现结果是正确的。

▼<defs>

```
<path id="1" d="M0 44 H600"></path>
<path id="2" d="M0 91 H600"></path>
<path id="3" d="M0 122 H600"></path>
<path id="4" d="M0 154 H600"></path>
<path id="5" d="M0 190 H600"></path>
<path id="6" d="M0 231 H600"></path>
<path id="7" d="M0 279 H600"></path>
<path id="8" d="M0 318 H600"></path>
<path id="9" d="M0 359 H600"></path>
<path id="10" d="M0 406 H600"></path>
<path id="11" d="M0 455 H600"></path>
<path id="12" d="M0 487 H600"></path>
<path id="13" d="M0 523 H600"></path>
<path id="14" d="M0 570 H600"></path>
<path id="15" d="M0 602 H600"></path>
<path id="16" d="M0 640 H600"></path>
<path id="17" d="M0 690 H600"></path>
<path id="18" d="M0 740 H600"></path>
<path id="19" d="M0 776 H600"></path>
<path id="20" d="M0 817 H600"></path>
<path id="21" d="M0 867 H600"></path>
<path id="22" d="M0 907 H600"></path>
<path id="23" d="M0 955 H600"></path>
<path id="24" d="M0 1000 H600"></path>
<path id="25" d="M0 1049 H600"></path>
```

```
.qxu5cd {
  background: -336px -295px;
}
```

▼<text lengthAdjust="spacing">

```
<textPath xlink:href="#1" textLength="546">稚喷仙勇坛炒寨油认镇翼召定稿惠莲督貌来孟岁饰芦礼落忙持峰远茫弦夫航居竿唱茅鸿材</textPath>
<textPath xlink:href="#2" textLength="518">钱欠窗料热匆步暴森矿肺球级魄帅冒底察丘点迷崇倚欣岭教典情颀肆世幼诒翼份适财</textPath>
<textPath xlink:href="#3" textLength="294">四钻轿跟凝律信首竭伶殊辨选妨玩袖湿偷后谜幅</textPath>
<textPath xlink:href="#4" textLength="364">位乎炭息项用弟涝沉抹露肝照集路狼觉各择攻所从讯倒型田</textPath>
<textPath xlink:href="#5" textLength="406">槽倍声往袂求苏迨暑驻柄疆烤边近写颞通动键君陆邀脸龟遭扬卡庸</textPath>
<textPath xlink:href="#6" textLength="588">相示扮勤符钩掌验艇作速洲逆制摸余肌杜里巧筑益啤车儿并汽鞠致泄乃匙并消良刹场稀至佑肢咸</textPath>
<textPath xlink:href="#7" textLength="392">泽右撒愈徐威庙碑方坚宿醒媚催潮容圆育顶深肚石筋腔七骑斩免</textPath>
<textPath xlink:href="#8" textLength="448">福纱隆埋吧岩修肝川格错胞斤谢城薪例融梢娱喝吵辽列种肃呆效樱域搁谎</textPath> == $0
```


通过上述的分析我们已经知道了如何获取加密标签和明文汉字之间的关系，但是由于大众点评网站使用的加密字典是随机生成的，每个页面的加密字典都不相同，因此在每请求一个页面时就要重新获取一次明文和密文之间的对照表。

以下是获取的基本过程：

(1) 首先通过测试发现加密标签和 background 字段的值都存放在某一个 url 页面中，而该 url 路径存放在页面 html 代码中，因此我们分析该 url 的基本特征，利用正则表达式即可将其匹配出来，并将读取出的内容存放在 css 文件中。

```
<head>
<!-- 新页头 -->
<title>“相遇融合餐厅(徐州苏宁店)”的全部点评 - 徐州美食 - 大众点评网</title>
<!--网页标题左侧显示-->
<link rel="icon" type="image/x-icon" href="//www.dpfile.com/app/pc-common/dp_favicon.a4af753914321c8e82e402e2b4be01d7.ico">
<!--收藏夹显示图标-->
<link rel="shortcut icon" type="image/x-icon" href="//www.dpfile.com/app/pc-common/dp_favicon.a4af753914321c8e82e402e2b4be01d7.ico">
<meta name="Keywords" content="相遇融合餐厅(徐州苏宁店)评价, 相遇融合餐厅(徐州苏宁店)好不好, 相遇融合餐厅(徐州苏宁店)怎么样"/>
<meta name="Description" content="此商户已有13975条评价, 想知道相遇融合餐厅(徐州苏宁店)口碑好不好? 服务怎么样? 徐州大众点评为您提供更丰富更真实的探店消费评价和
<meta name="location" content="province=江苏;city=徐州"/>
<meta http-equiv="mobile-agent" content="format=xhtml; url=http://m.dianping.com/shop/152BoQ312BMNHLDc/review_all">
<!--1. 首先引入页头模块css, 保证页头模块css在前, 首先渲染 -->
<link rel="stylesheet" type="text/css" href="//www.dpfile.com/app/pc-common/index_min.1b782a80b8aba41a0307fdd6b470542a.css">
<link rel="stylesheet" type="text/css" href="//s3plus.meituan.net/v1/mss_0a06a471f9514fc79c981b5466f56b91/svgtextcss/c4efb5b77e8202dc334136ce6d328f30.css">
<!--2. 引入页头模块 js -->
<script type="text/javascript" src="//www.dpfile.com/app/pc-common/index_min.f2491848f6ed02c16c39faad2febfa93.js"></script>
<!--3. 注入页头需要的参数 -->
<script type="text/javascript">
    window._DP_HeaderData = {
        'cityId': 92,
        'cityChName': '徐州',
        'cityEnName': 'xuzhou',
        'userId': '1421648354',
        'userName': '浮生',
        'pageType': 'channel',
        'channelId': '10',
        'shopId': '152BoQ312BMNHLDc',
        'shopName': '相遇融合餐厅'
    }
</script>
```

利用以下代码实现：

```
css_url = re.findall('<link rel="stylesheet" type="text/css"
href="(//s3plus.meituan.*?)>', response)

css_url = 'http:' + css_url[0]

css_response = requests.get(css_url)

with open('css 样式.css', mode='w', encoding='utf-8') as f:
    f.write(css_response.text)
```

获取结果如下：

```
q.xuk2s{background:-126.0px -2468.0px;}.qxupg6{background:-546.0px -1499.0px;}.tecscs{background:-8.0px -14.0px;}.
p.uv4ih{background:-476.0px -180.0px;}.p.uvdmx{background:-140.0px -87.0px;}.qxuia2{background:-0.0px -1716.0px;}.
q.xum1a{background:-266.0px -753.0px;}.qxuqt7{background:-42.0px -547.0px;}.qxulk3{background:-420.0px -295.0px;}.
q.xu2md{background:-308.0px -753.0px;}.qxucq3{background:-322.0px -1264.0px;}.qxucrz{background:-70.0px -336.0px;}.
q.xuk90{background:-168.0px -579.0px;}.qxuedw{background:-168.0px -2183.0px;}.qxu82k{background:-308.0px -2108.0px;}.
q.xuo9e{background:-434.0px -2225.0px;}.p.uvcc1{background:-238.0px -14.0px;}.qxui2h{background:-322.0px -131.0px;}.
q.xu68d{background:-322.0px -2387.0px;}.qxumib{background:-266.0px -383.0px;}.p.uvh8m{background:-70.0px -87.0px;}.
q.xura3{background:-210.0px -2108.0px;}.qxuc0h{background:-126.0px -1982.0px;}.p.uvdot{background:-70.0px -149.0px;}.
p.uvyl3{background:-196.0px -87.0px;}.qxuc0r{background:-168.0px -1264.0px;}.qxuj6e{background:-210.0px -884.0px;}.
q.xuuku{background:-28.0px -3121.0px;}.qxujr6{background:-112.0px -2635.0px;}.qxurx3{background:-84.0px -3260.0px;}.
}cc[class^="tec"]{width: 14px;height: 16px;margin-top: -7px;background-image: url(//s3plus.meituan
net/v1/mss_0a06a471f9514fc79c981b5466f56b91/svgtextcss/3efed938ddb0ef05492166065cd6ace3.svg);background-repeat:
no-repeat;display: inline-block;vertical-align: middle;margin-left: -6px;}.qxubs5{background:-28.0px -2602.0px;}.
q.xumez{background:-350.0px -2301.0px;}.qxuryc{background:-378.0px -1424.0px;}.p.uvc7r{background:-14.0px -121.0px;}.
p.uvm32{background:-392.0px -149.0px;}.qxuko8{background:-126.0px -2225.0px;}.qxupzu{background:-28.0px -1759.0px;}.
q.xuw2x{background:-350.0px -2029.0px;}.qxu541{background:-238.0px -2553.0px;}.qxupqp{background:-490.0px -464.0px;}.
p.uve7v{background:-42.0px -149.0px;}.qxub6e{background:-126.0px -1104.0px;}.qxux3{background:-98.0px -2790.0px;}.
q.xuphe{background:-420.0px -432.0px;}.qxuo5g{background:-140.0px -256.0px;}.qxuprt{background:-238.0px -167.0px;}.
q.xut08{background:-182.0px -2635.0px;}.qxuq6y{background:-70.0px -1982.0px;}.qxu15g{background:-126.0px -500.0px;}.

```

(2) 通过第一步获取的内容可知, 该 css 文件中不仅存放了加密标签和 background 字段之间的对应关系, 还确定了该页面使用的是哪一张密码表, 因此我们再次使用正则匹配, 将存放字典的 url 链接匹配出来, 并将从该页面读取出的内容存放在 svg 文件中。

```
svg_url = re.findall(r'svgmts\[\class\^="qxu"\].*?background-image: url\(((.*?)\);',  
css_response.text)  
svg_url = 'http:' + svg_url[0]  
svg_response = requests.get(svg_url)  
with open('svg 映射表.svg', mode='w', encoding='utf-8') as f:  
    f.write(svg_response.text)
```

获取结果如下:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>  
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN" "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd"  
<svg xmlns="http://www.w3.org/2000/svg" version="1.1" xmlns:xlink="http://www.w3.org/1999/xlink"  
<style>text {font-family:PingFangSC-Regular, Microsoft YaHei, 'Hiragino Sans GB', Helvetica; ;fon  
<defs><path id="1" d="M0 44 H600"/><path id="2" d="M0 91 H600"/><path id="3" d="M0 122 H600"/><pa  
<text lengthAdjust="spacing">  
<textPath xlink:href="#1" textLength="546">锤喷仙勇坛炒寨油认镇翼召定稿惠莲督貌来孟岁饰芦礼落忙持峰远茫弦夫航居  
<textPath xlink:href="#2" textLength="518">钱欠奋料热匆步暴蔽矿肿残级魄帅冒底躲丘点述崇倚欣岭教典情蜓肆世幼治舅  
<textPath xlink:href="#3" textLength="294">四钻轿跟潞律信首谒伶殊辨选妨玩袖湿偷后谜榴</textPath>  
<textPath xlink:href="#4" textLength="364">位乎炭息昂用弟涝沉抹蔑奸照集路狠觉各择攻所从倒型田</textPath>  
<textPath xlink:href="#5" textLength="406">棚声往袂求苏追暑驻柄疆烤边近写颤通动锄君陆避险龟遭扬卡庸</textPath>  
<textPath xlink:href="#6" textLength="588">粗示扮勒符钧掌骆艇作速洲逆割摸余舂杜里巧筑益啐丰儿并汽鞠致泄乃匙井洋  
<textPath xlink:href="#7" textLength="392">泽右揪愈徐威庙碑方坚宿醒烟催潮容圈育顶深肚石筋腔七骑斩免</textPath>  
<textPath xlink:href="#8" textLength="448">福纱隆埋吧岩修肝川格错胞斤谢城薪例融梢娱喝吵辽列种肃呆效樱域搁谎</t  
<textPath xlink:href="#9" textLength="364">丁润汁肥乏会泉略裂手抗蚊尤吞随麻究父绕刷袜百凡粘挑农</textPath>  
<textPath xlink:href="#10" textLength="308">不寺象核试杨雄献纸棋拼改沙灯岸赠扯盼测悲隐剖</textPath>  
<textPath xlink:href="#11" textLength="462">恋仗裁卷急债孙扫晕已袍词朽龄虎纵挤本滑寿歌州长著数枯洵输程时啊叹羽  
<textPath xlink:href="#12" textLength="588">复堆映候屡摊极腾既故须旬醉质大梁客哨丰秘孝怠贖盟覆渔诗荐狭漱断奶阅  
<textPath xlink:href="#13" textLength="336">萝界凝旧佛班清炼猎倘欺垫挂群资分秤民控睁穗缀房目</textPath>  
<textPath xlink:href="#14" textLength="546">堤肾犁矩閃提继蛋胶耕歉贝逗六赌轮轨签抬介振绿指泡败痒辈箐哭参监阁敬  
<textPath xlink:href="#15" textLength="308">乘葱寸母挖层统企亚腓别勿缸极贱饲嚷匪违把狡践</textPath>  
<textPath xlink:href="#16" textLength="308">前址蝶苦偶甚芝捐梁哈妖起锋垮糊市春砖踢帐胆梯</textPath>  
<textPath xlink:href="#17" textLength="560">辆谁咬诚售串慰壳醃内坑震锻橘鸣纯尿逐灰置疯返翅扔乐佳旷飞检杏类稳伏  
<textPath xlink:href="#18" textLength="392">匀港狂贤搅花希角勺释乳蒸愿哥伍曾驰秋李濯正泥竟宋利席遗骂</textPath>  
<textPath xlink:href="#19" textLength="546">习阔责刊掘蛮芳背娃棕烂雀夹拆恩庆另蓝俊妻炊韵悄荣广住允杂历晃树盈双  
<textPath xlink:href="#20" textLength="308">劈浮滨南青坐形纳珍兆较痰榆逼向忆装朴散碌拉机</textPath>  
<textPath xlink:href="#21" textLength="434">困减紧镜录脏卸奴电扰悉抽货剩连婚捡保敢忘弹构固士窑丛誓湖糖甩他</t  
<textPath xlink:href="#22" textLength="588">练遭惨昂罪菲绒刑匹冀权谈环迎康你捧坊煎秆庭忍刹牵掌旗先宣恒秃秀优协
```

(3) 在通过上述两步获取到破解密文所需的数据之后, 即可通过之前分析的映射关系, 编写脚本, 实现明文和密文的对应。

```
svg_html = svg_response.text  
lengths = re.findall(r'<path\sid="(\d+)"\s\d="M0\s(\d+)\sH600"/>', str(svg_html))  
sel = parsel.Selector(svg_html)  
# 加载映射规则表
```



```

texts = sel.css('textPath')
lines = []
for text in texts:
    lines.append([int(re.findall(r'textlength="(.*?)"', str(text.get()))[0]),
text.css("textPath::text").get()])
# 获取所有类名及位置
css_text = css_response.text
class_map = re.findall('\.(qxu\w+)\{background:-(\d+)\.0px -(\d+)\.0px;\}', css_text)
for i in range(len(class_map)):
    class_map[i] = class_map[i][0], int(class_map[i][1]), int(class_map[i][2])
character = []
# 获取类名与汉字的对应关系
for map in class_map:
    for length in lengths:
        if map[2]>int(length[1]):
            pass
        else:
            line=lines[int(length[0])-1][1][int(map[1]/14)]
            character.append([map[0],line])
            break
return character

```

部分结果如下：

```

[['gjjg12', '梢'], ['gjcgt', '谨'], ['gjdx1', '酷'], ['gjme6', '骂'], ['gj5d4', '贤'], ['gjhur', '阔'],
['gjkvq', '梢'], ['gj56l', '僚'], ['gjcof', '孙'], ['gjica', '觉'], ['gj9dr', '盒'], ['gjjfq', '作'],
['gjume', '魂'], ['gjneb', '脖'], ['gji7e', '别'], ['gjiwi', '歪'], ['gjk0', '关'], ['gjbhf', '总'],
['gjukm', '响'], ['gju6o', '营'], ['gjphh', '煎'], ['gjbqg', '辨'], ['gjz01', '布'], ['gjfno', '扩'],
['gjf8t', '修'], ['gj6v0', '坡'], ['gj67u', '捉'], ['gjae7', '丈'], ['gj2nn', '浅'], ['gjvqj', '哀'],
['gjyet', '线'], ['gjfb0', '洽'], ['gjw1', '垒'], ['gjnm9', '赚'], ['gjhio', '铃'], ['gjeuj', '镜'],
['gjm5g', '皂'], ['gj4sh', '连'], ['gjvuk', '驾'], ['gjlkp', '交'], ['gjnoz', '浇'], ['gje64', '时'],
['gjzrm', '宰'], ['gjfb7', '傍'], ['gjols', '冈'], ['gj0cs', '序'], ['gj1zq', '短'], ['gjw23', '餐'],
['gj5ju', '再'], ['gjfbr', '醉'], ['gj1sb', '郊'], ['gjatn', '劈'], ['gjnnw', '唐'], ['gjr8u', '斤'],
['gj6z5', '四'], ['gjbhu', '艘'], ['gj1sy', '季'], ['gj9p5', '煤'], ['gjxbn', '湾'], ['gjzl8', '区'],
['gjtm5', '沃'], ['gjbzv', '德'], ['gjatp', '刚'], ['gjdw4', '欧'], ['gj0fj', '滩'], ['gj18t', '杏'],
['gjswd', '穿'], ['gjvox', '效'], ['gjgkf', '析'], ['gjlle', '林'], ['gjm6a', '坑'], ['gjvzp', '辱'],
['gjvud', '荣'], ['gja2l', '算'], ['gj7ll', '键'], ['gja7n', '哲'], ['gjg35', '步'], ['gjnrj', '桃'],
['gjakp', '刺'], ['gjep7', '熔'], ['gjntb', '蝴'], ['gj7wm', '神'], ['gj9f9', '室'], ['gjek7', '骄'],
['gj6hy', '缝'], ['gj4cp', '皇'], ['gj5ul', '段'], ['gjbpc', '力'], ['gjs5s', '文'], ['gjcr1', '以'],
['gjfwy', '推'], ['gjiah', '头'], ['gj01g', '执'], ['gj9sl', '年'], ['gju4r', '凉'], ['gjb3v', '膀'],
['gjimo', '骡'], ['gjgou', '屋'], ['gjbj', '红'], ['gjlt', '宫'], ['gj67p', '她'], ['gj6t0', '手'],
['gj7di', '答'], ['gj11q', '故'], ['gjhl8', '朽'], ['gjzrv', '慢'], ['gj93n', '暖'], ['gjmr8', '鹅'],
['gj1ny', '尺'], ['gj1se', '才'], ['gjju0', '悠'], ['gjwh2', '宝'], ['gjxse', '婚'], ['gjxxh', '倘'],

```

4.2 内容爬取

在完成以上对于反爬取对抗措施的代码编写，基础爬虫部分的实现还是比较简单的：首先要求用户输入要爬取的店铺的 ID，以及要爬取评论的页数，程序即可根据用户输入自动构造评论页的 url 链接，从第一页评论开始爬取，直到爬取到用户要求的页数停止。每一页评论的爬取过程是相同的，首先直接抓取该页面的 html 代码，接着将 bs4 库提供的解析方法和正则匹配相结合，将我们需要的用户名，综合评分，发布时间等元素匹配出来，并存储在 MySQL 数据库中，其中在匹配评论的时候因为涉及明密文的转换，需要编写一个双层 for 循环，在识别到一个密文的时候就遍历一次密码表。

当然如果出现爬取异常的情况，程序中也设置了许多 try-except 语句块来处理异常。

```
def getCommentinfo(shop_url, shpoID, page_begin, page_end):
    for i in range(page_begin, page_end):
        try:
            url = shop_url + 'p' + str(i)
            html = getHTMLText(url)
            character = exspider_2(html)
            infoList = parsePage(html, shpoID, character)
            print('成功爬取第{}页数据,有评论{}条'.format(i, len(infoList)))
            for info in infoList:
                mysqls.save_data(info)
                # 断点续传中的断点
            if (html != "产生异常") and (len(infoList) != 0):
                with open('xuchuan.txt', 'a') as file:
                    duandian = str(i) + '\n'
                    file.write(duandian)
            else:
                print('休息 60s...')
                time.sleep(60)
        except:
            print('跳过本次')
            continue
    return
```

```

def getHTMLText(url, code="utf-8"):
    try:
        time.sleep(random.random() * 6 + 2)
        r = requests.get(url, timeout=5, headers=headers, proxies=get_random_ip())
        r.raise_for_status()
        r.encoding = code
        return r.text
    except:
        print("产生异常")
        return "产生异常"

def parsePage(html, shpoID, character):
    infoList = [] # 用于存储提取后的信息, 列表的每一项都是一个字典
    soup = BeautifulSoup(html, "html.parser")
    for item in soup('div', 'main-review'):
        common =
re.findall(r'<div\class="review-words\sHide">([\s\S]+?)<div\class="less-words">',
str(item))
        if len(common) == 0:
            common = re.findall(r'<div\class="review-words">([\s\S]+?)</div>',
str(item))[0]
        else:
            common =
re.findall(r'<div\class="review-words\sHide">([\s\S]+?)<div\class="less-words">',
str(item))[0]
            common = re.findall(r'(\S+)', common)
            common = ".join(i for i in common)
            common = common.replace('<svgmtsiclass=', '')
            common = common.replace('></svgmts>', '')
            cus_common = ""
            i = 0
            while i < len(common):

```

```

        if common[i] == 'q' and common[i + 1] == 'x' and common[i + 2] ==
'u':

            for j in character:
                if common[i:i + 6] == j[0]:
                    cus_common += j[1]
                    i += 5
                    break

            else:
                cus_common += common[i]
                i += 1
cus_id = item.find('a', 'name').text.strip()
comment_time = item.find('span', 'time').text.strip()
try:
    comment_star = item.find('span',
re.compile('sml-rank-stars')).get('class')[1]
except:
    comment_star = 'NAN'
scores = item.find('span', 'score').text
kouwei = re.findall(r'口味: (.+)\s', scores)[0]
huanjing = re.findall(r'环境: (.+)\s', scores)[0]
fuwu = re.findall(r'服务: (.+)\s', scores)[0]

infoList.append({'cus_id': cus_id,
                  'comment_time': comment_time,
                  'comment_star': comment_star,
                  'cus_comment': remove_emoji(cus_common),
                  'kouwei': kouwei,
                  'huanjing': huanjing,
                  'fuwu': fuwu,
                  'shopID': shpoID})

return infoList

```


4.3 MySQL 数据库使用

在主函数中需要频繁的进行 MySQL 数据库的数据添加操作，因此单独在一个.py 文件中编写了关于对 MySQL 数据库进行操作的函数，其中包括数据库连接，新建表格以及向表格中插入数据的方法。这些方法都是利用 pymysql 库编写完成的，语法非常简洁。

```
import pymysql

#连接 MYSQL 数据库
db=pymysql.connect(host='localhost',user='spider',password='010316',charset='utf8',database='spider')
cursor = db.cursor()

#在数据库建表
def creat_table():
    cursor.execute("DROP TABLE IF EXISTS DZDP")
    sql = "CREATE TABLE DZDP(
        cus_id varchar(100),
        comment_time varchar(55),
        comment_star varchar(55),
        cus_comment text(5000),
        kouwei varchar(55),
        huanjing varchar(55),
        fuwu varchar(55),
        shopID varchar(55)
    );"
    cursor.execute(sql)
    return

#存储爬取到的数据
def save_data(data_dict):
    sql = "INSERT INTO
DZDP(cus_id,comment_time,comment_star,cus_comment,kouwei,huanjing,fuwu,shopID) VALUES(%s,%s,%s,%s,%s,%s,%s,%s)"
```

```

value_tup = (data_dict['cus_id']
             ,data_dict['comment_time']
             ,data_dict['comment_star']
             ,data_dict['cus_comment']
             ,data_dict['kouwei']
             ,data_dict['huanjing']
             ,data_dict['fuwu']
             ,data_dict['shopID']
             )

try:
    cursor.execute(sql,value_tup)
    db.commit()

except:
    print('数据库写入失败')

return

#关闭数据库
def close_sql():
    db.close()

creat_table()

```

4.4 数据处理

在用户所需评论数据都成功抓取并存储在 MySQL 数据库后，即可进行简单的数据分析。

(1) 首先利用 pymysql 库提供的方法连接数据库，并从表中读取用户的综合评分数据保存在变量 data 中，接着对评分数据进行处理让其转换为纯数字，最后利用 seaborn 库提供的方法绘制关于综合评分的柱形图，可以直观的观察某一店铺的综合评分分布情况。

```

import pandas as pd
import pymysql
import seaborn as sns

db =

```

```

pymysql.connect(host='localhost',user='spider',password='010316',charset='utf8',database='spider') #服务器: localhost, 用户名: root, 密码: (空), 数据库: TESTDB
sql = "select * from dzdp;"
data = pd.read_sql(sql,db)
db.close()

data.loc[data['comment_star'] == 'sml-str1','comment_star'] = 'sml-str10'
data['stars'] = data['comment_star'].str.findall(r'\d+').str.get(0)
data['stars'] = data['stars'].astype(float)/10
sns.countplot(data=data,x='stars')

```

(2) 同样在连接数据库后，从表中读取发布时间数据并从中提取出年、月、日和小时的数据段，基于此可以分析某一店铺的评论时间特性。此处分析的是一天中各小时评论数和时间的关系，并利用 matplotlib 库提供的方法绘制折线图。

```

from matplotlib import pyplot as plt

data.comment_time =
pd.to_datetime(data.comment_time.str.findall(r'\d{4}-\d{2}-\d{2} .+').str.get(0))
data['year'] = data.comment_time.dt.year
data['month'] = data.comment_time.dt.month
data['weekday'] = data.comment_time.dt.weekday
data['hour'] = data.comment_time.dt.hour

fig1, ax1=plt.subplots(figsize=(14,4))
df=data.groupby(['hour', 'weekday']).count()['cus_id'].unstack()
df.plot(ax=ax1, style='-.')
plt.show()

```

(3) 同样在连接数据库后，读取表中评价数据并计算出每条评价的长度，即可结合之前读出的综合评分数据分析这两者之间的关系，并绘制出关系图。

```

data['comment_len'] = data['cus_comment'].str.len()
fig2, ax2=plt.subplots()
sns.boxplot(x='stars',y='comment_len',data=data, ax=ax2)
ax2.set_ylim(0,600)

```

5 系统展示

5.1 爬取前准备

(1) 首先输入要爬取店铺的店铺 ID（店铺 ID 可以在点击进入该店铺后在 url 中获取），接着输入要爬取的评论页数。

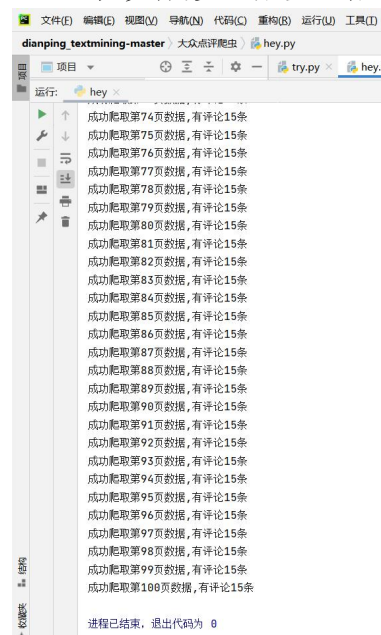
```
# 根据店铺id, 店铺页码进行爬取
def crawl_comment(shopID='G95HMDyRBpee05vW', page=100):
    shop_url = "http://www.dianping.com/shop/" + shopID + "/review_all/"
    # 读取断点续传中的续传断点
    nowpage = xuchuan()
    getCommentinfo(shop_url, shopID, page_begin=nowpage + 1, page_end=page + 1)
    mysqls.close_sql()
    return
```

(2) 接着配置账户 cookie 进行爬取。

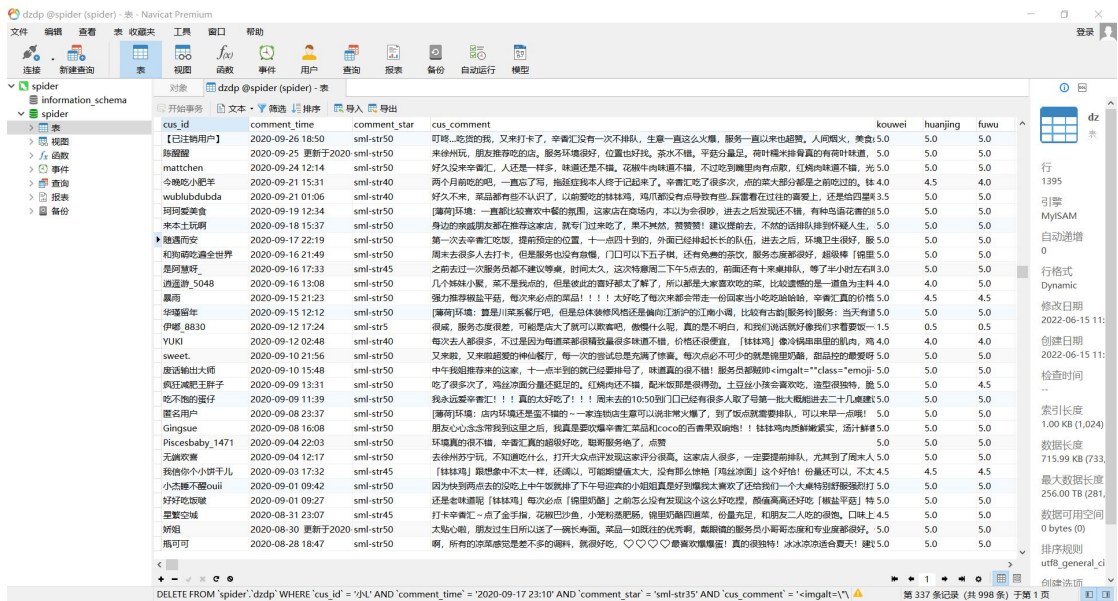
```
# 设置cookies
cookie="__mta=247289147.1654615733654.1654615733654.1654615733654.1; _lxsdk_..."
headers = {
    'User-Agent': ua.random,
    'Cookie': cookie,
    'Connection': 'keep-alive',
    'Host': 'www.dianping.com',
    'Referer': 'http://www.dianping.com/shop/G95HMDyRBpee05vW/review_all/p6'
}
```

5.2 爬取结果

(1) 在完成以上配置之后程序即可开始爬取。



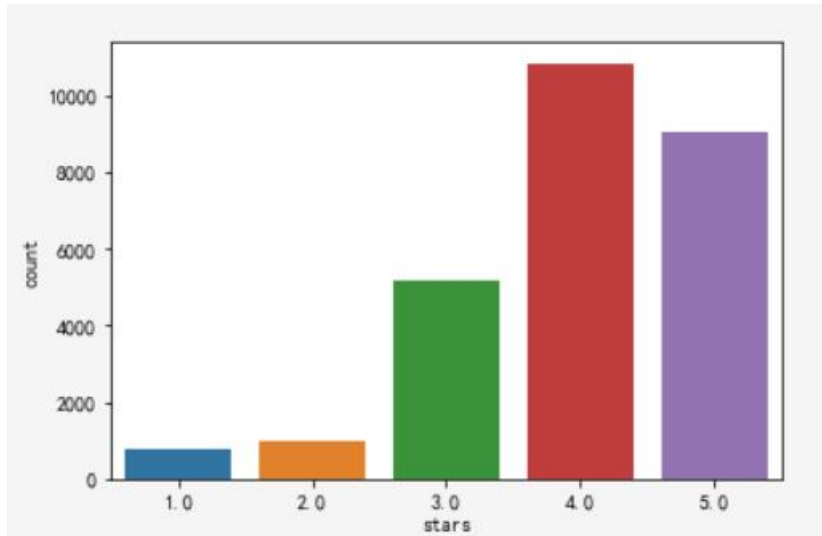
(2) 爬取的内容都保存在 MySQL 数据库中。



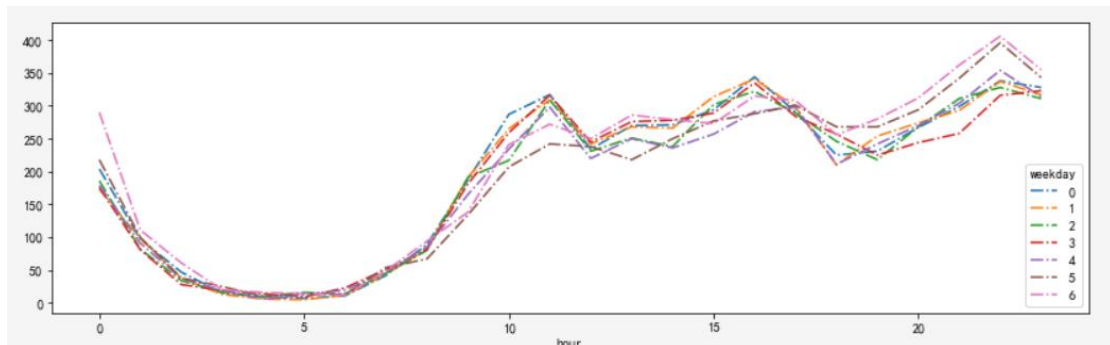
cus_id	comment_time	comment_star	cus_comment
【已注册用户】	2020-09-26 18:50	sml-str50	叮咚...吃货的我，又来打卡了，辛香汇没有一次不排队，生意一直这么火爆，服务一直以来也超赞，人间烟火，美食！
陈醒醒	2020-09-25 更新于2020	sml-str50	来徐州玩，朋友推荐吃的店，服务环境很好，位置也好找，茶水不错，平菇分量足，荷叶糯米排骨真的荷叶味道，
matchen	2020-09-24 12:14	sml-str50	好久没来辛香汇，人还是一样多，味道还是不错，花雕牛肉味道不错，不过吃到碗里肉有点散，红烧肉味道不错，光
今晚吃小肥羊	2020-09-21 15:31	sml-str40	两个月前吃的吧，一直忘了写，拖延症本人终于记起来了，辛香汇吃了很多次，点的大部分都是之前吃过的，辣
wububububuda	2020-09-21 01:06	sml-str40	好久不来，菜品都有点不认识了，以前爱吃的钵钵鸡，鸡爪都没有导致有点，蒜香骨在过往的喜爱上，还垫给四星
珂珂爱美食	2020-09-19 12:34	sml-str50	[薄荷]环境：一直都比较喜欢中餐的氛围，这家店在商场内，本以为是会很多，进去之后发现还不错，有种包场
来本土玩啊	2020-09-18 15:37	sml-str50	身边的亲戚朋友都在推荐这家店，就专门过来吃了，果然不赖，赞赞！建议提前去，不然的话排队排到怀疑人生，
隐逸而安	2020-09-17 22:19	sml-str50	第一次去辛香汇吃饭，提前预定的位置，十一四十到的，外面已经排起长长的队伍，进去之后，环境卫生很好，服
和雨晴吃遍全世界	2020-09-16 21:49	sml-str50	周末去很多人去打卡，但是服务也不建议等，门口可以下五子棋，还有免费的茶饮，服务态度都很好，超级棒！
是阿慧呀	2020-09-16 17:33	sml-str45	之前去过一次服务员都不建议等，时间太久，这次特意周二下午5点去的，前面还有十来桌排队，等了半小时左右
逍遥游_5048	2020-09-16 13:08	sml-str50	几个钵钵鸡，菜不是我的点，但是彼此的好都太了解了，所以都是大家喜欢吃的菜，比较遗憾的是没有鱼为主料
晨雨	2020-09-15 21:23	sml-str50	强力推荐椒盐平菇，每次来必点的菜品！！太好吃了每次来都会带一份回家当小吃吃油锅焗，辛香汇真的价格
华耀留年	2020-09-15 12:12	sml-str50	[薄荷]环境：真是川菜系啊，但是总体装修风格还是偏向江浙沪的江南小调，比较有古典(服务)服务：当天有
伊娜_8830	2020-09-12 17:24	sml-str5	环境，服务态度很差，可能是店大了就可以欺负吧，不懂什么呀，真的不明白，和我们说话就好像我们求着要
YUKI	2020-09-12 02:48	sml-str40	每次去人多很多，不过是因为每道菜都做得数量多味道不错，价格也便宜，[钵钵鸡]香浓入味且很入味，鸡
sweet	2020-09-10 21:56	sml-str50	又来啦，又来啦超超的神仙餐厅，每一次的尝试总是充满了惊喜，每次点必不可少的就是钵钵鸡，甜滋滋的
陈淑瑜比大师	2020-09-10 15:48	sml-str50	中午我超超超的这家，十一点半到的就已经要排队了，味道真的不错！服务员都配得
疯狂减肥王胖子	2020-09-09 13:31	sml-str50	吃了很多次了，鸡丝凉面分量还挺好的，红烧肉还不错，配米饭那真是绝了，土豆丝小孩会喜欢，造型很独特，脆
吃不饱的蛋仔	2020-09-09 11:39	sml-str50	我最近来辛香汇！！真的太好吃了！！周末去的10:50到门口已经有很多人取了第一批大概能排到二十九桌建
匿名用户	2020-09-08 23:37	sml-str50	[薄荷]环境：店内环境还是不错的～一家连锁店生意可以流非火火爆了，到了饭点就需要排队，可以早一点
Gingsue	2020-09-08 16:08	sml-str50	朋友心心念念带我到这里之后，我真是要吃爆辛香汇菜品和coco的百香果双响炮！！钵钵鸡肉质鲜嫩紧实，汤汁
Discesbaby_1471	2020-09-04 22:03	sml-str50	环境真的很不错，辛香汇真的超超好吃，感谢服务了，点赞
无限欢乐	2020-09-04 12:17	sml-str50	去徐州苏宁玩，不知道吃什么，打大众点评发现这家评价很高，这家店人多，一定要提前排队，尤其到了周末
我他你个小饼干儿	2020-09-03 17:32	sml-str45	[钵钵鸡] 钵钵鸡中不太一样，还偏以，可能因为店大了就可以欺负吧，不懂什么呀，真的不明白，和我们
小杰不丢ouli	2020-09-01 08:42	sml-str50	因为快到而去的时候上午饭刚到了下午马路的小钵钵鸡真是好喝吃太喜欢了还给我们一个火鸡特别好吃
好好吃啊	2020-09-01 09:27	sml-str50	还是本味钵钵鸡[钵钵鸡] 每次必点[钵钵鸡] 之前怎么没有发现这个这么好吃的，钵钵鸡真好吃(椒盐平菇) 辣
蟹蟹空城	2020-08-31 23:07	sml-str45	打卡辛香汇，点了金手捞，花雕巴鱼，小笼汤包，提里炸藕圆，份量充足，和朋友二人吃的很饱，口味上
陈姐	2020-08-30 更新于2020	sml-str50	太贴心啦，朋友过生日送了满满一餐长寿面，菜品一如既往的优秀啊，就爱吃的小哥态度和态度都很好，
陈可可	2020-08-28 18:47	sml-str50	啊，所有的菜感觉是差不多的调料，就很好吃，♡♡♡♡最爱欢嘴嘴！真的很好吃！冰冰的凉适合夏天！建

5.3 可视化分析

(1) 某店铺综合评分分布情况



(2) 某店铺评论发布的时间特征



(3) 某店铺评论长度和综合评分之间的关系。

