城市爬虫与接龙

- 运行环境: python3，需要第三方库BeautifulSoup4
- 所有文件都在city_crawler_chains文件夹里
    1. city_crawler.py为爬虫代码（爬取网站为去哪儿网）

```
# -*- coding: utf-8 -*-

import requests
from bs4 import BeautifulSoup
import csv
import random
User_Agent=["Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/51.0.2704.106 Safari/537.36","Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10_6_8; en-us) AppleWebKit/534.50 (KHTML, like Gecko) Version/5.1 Safari/534.50","Mozilla/5.0 (Windows; U; Windows NT 6.1; en-us) AppleWebKit/534.50 (KHTML, like Gecko) Version/5.1 Safari/534.50","Mozilla/5.0 (Macintosh; Intel Mac OS X 10.6; rv:2.0.1) Gecko/20100101 Firefox/4.0.1","Mozilla/5.0 (Windows NT 6.1; rv:2.0.1) Gecko/20100101 Firefox/4.0.1"]

HEADERS = {
    'User-Agent':  User_Agent[random.randint(0,4)],
    # 'User-Agent':'Mozilla/5.0 (Windows NT 6.1; WOW64; rv:55.0) Gecko/201002201 Firefox/55.0',
    'Accept': 'text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8',
    'Accept-Language': 'zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3',
    'Accept-Encoding': 'gzip, deflate, br',
    'Cookie': '',
    'Connection': 'keep-alive',
    'PRagma': 'no-cache',
    'Cache-Control': 'no-cache'
}
csvfile = open('city_where.txt','w',encoding='utf-8', newline='')
writer = csv.writer(csvfile)
writer.writerow(["区域"])

def download_page(url):  # 下载页面
    try:
        data = requests.get(url, headers=HEADERS, allow_redirects=True).content  # 请求页面，获取要爬取的页面内容
        return data
    except:
        pass
```

```
#下载页面如果没法下载就pass
def download_soup_waitting(url):
    try:
        response= requests.get(url,headers=HEADERS,allow_redirects=False,timeout=5)
        if response.status_code==200:
            html=response.content
            html=html.decode("utf-8")
            soup = BeautifulSoup(html, "html.parser")
            return soup
    except:
        pass

def getTypes(url_num):
    types=["文化古迹"]
    for type in types:
        url="http://piao.qunar.com/ticket/list.htm?
keyword=%E7%83%AD%E9%97%A8%E6%99%AF%E7%82%B9&region=&from=mpl_search_suggest&subject="+
type+"&page=1"
        getType(type,url,url_num)

def getType(type,url,url_num):# url_num为要爬取的链接数
    soup=download_soup_waitting(url)
    for i in range(url_num):
        try:
            search_list=soup.find('div', attrs={'id': 'search-list'})
            sight_items=search_list.findAll('div', attrs={'class': 'sight_item'})
            for sight_item in sight_items:
                districts=sight_item['data-districts']
                writer.writerow([districts.replace("\n","")])
            next=soup.find('a',attrs={'class':'next'})
            if next:
                next_url="http://piao.qunar.com"+next['href']
                getType(type,next_url)
        except:
            pass

if __name__ == '__main__':
    #可以设置爬取的链接数
    getTypes(20)
    print('crawler complete!'
```

2. 爬取城市代码运行后会自动在本文件目录下创建city_where.txt，爬取的数据存放在里面

3. District.py文件为提取city_where.txt文件里的城市名代码，主要是对city_where.txt文件的清洗

```
# -*- coding: utf-8 -*-

import csv

path="C:/Users/Wvv/Desktop/city_crawler_chains"
def district():
    with open(path+'/district.txt','w') as d:
        writer = csv.writer(d)
        with open(path+'/city_where.txt','r',encoding='utf-8') as f:
            lines=f.readlines()
            for i in lines[1:]:
                temp=i.split('·')[0]
                writer.writerow([temp])
if __name__=='__main__':
    district()
```

4. District.py文件运行后会自动再当前目录下自动创建district.txt

5. city_chains.py文件为城市接龙代码

```
# -*- coding: utf-8 -*-
import random

path="C:/Users/Wvv/Desktop/city_crawler_chains" #需要改自己本地文件存放的位置
def city_exists(x,flag):
    if flag==0:
        adr='district.txt' #爬取并处理后的数据文件
    else:
        adr='csm_new.txt'  #验证谐音城市接龙的文件
    #判断输入的城市是否在城市库中
    with open(path+'/'+adr,'r') as f:
    #with open(path+'/district.txt','r') as f:
        for i in set(f.readlines()):
            if x==i.strip():
                return True
        return False
#汉字转拼音
def chinese_to_pinyin(x):
    y = ''
    dic = {}
    with open(path+"/unicode_pinyin.txt") as f:
        for i in f.readlines():
            dic[i.split()[0]] = i.split()[1]
    for i in x:
        i = str(i.encode('unicode_escape'))[-5:-1].upper()
        try:
            y += dic[i] + ' '
        except:
            y += 'XXXX ' #非法字符用XXXX代替
    return y
```

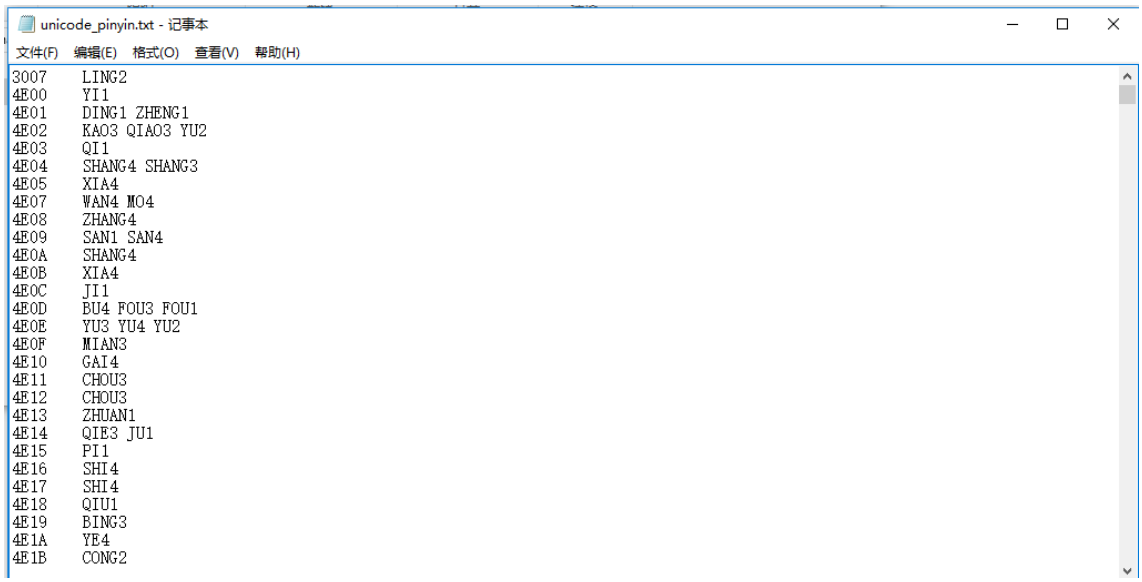
```python
def city_select(x,flag):
    if x==''':
        return 'sorry,i need to have a rest!'
    #参数x为城市名，返回该城市名的接龙匹配城市
    if flag==0:
        adr='district.txt'
    else:
        adr='csm_new.txt'
    t=[]
    with open(path+'/'+adr,'r') as f:
    #with open(path+'/district.txt','r') as f:
        pinyin = chinese_to_pinyin(x[-1])
        base = f.readlines()[:-1]
        random.shuffle(base)
        for i in base:
            if chinese_to_pinyin(i[0])[:-2] == pinyin[:-2]:
                t.append(i)
    if t!=[]:
        return "接龙城市：%s" %random.choice(t)
    else:
        return "很遗憾，词库缺少此城市的接龙城市名！"

def city_start(flag):
    #flag为0时用爬取的数据文件作为词库，flag为其它时，用可以验证谐音城市接龙的文件作为词库
    while True:
        p = input("请输入一个城市名:")
        cycle_num1=0
        cycle_num2=0
        while p.strip() == '':
            p=input("对不起，我不明白，请重新输入：")
            cycle_num1+=1
            if cycle_num1>5:
                break
        while city_exists(p,flag) == False:
            p=input("该城市名不存在，请重新输入：")
            cycle_num2+=1
            if cycle_num2>5:
                break
        t = city_select(p,flag)
        print (t)
        m=input("是否继续（y/n）：")
        if m=='y':
            n=input("请选择接龙词库（0 or else num）：")
            if n=='0':
                city_start(0)
            else:
                city_start(n)
        else:
            print("已退出！")
        break
if __name__ == '__main__':
    #flag,m参数可根据需要修改

    city_start(0)
```

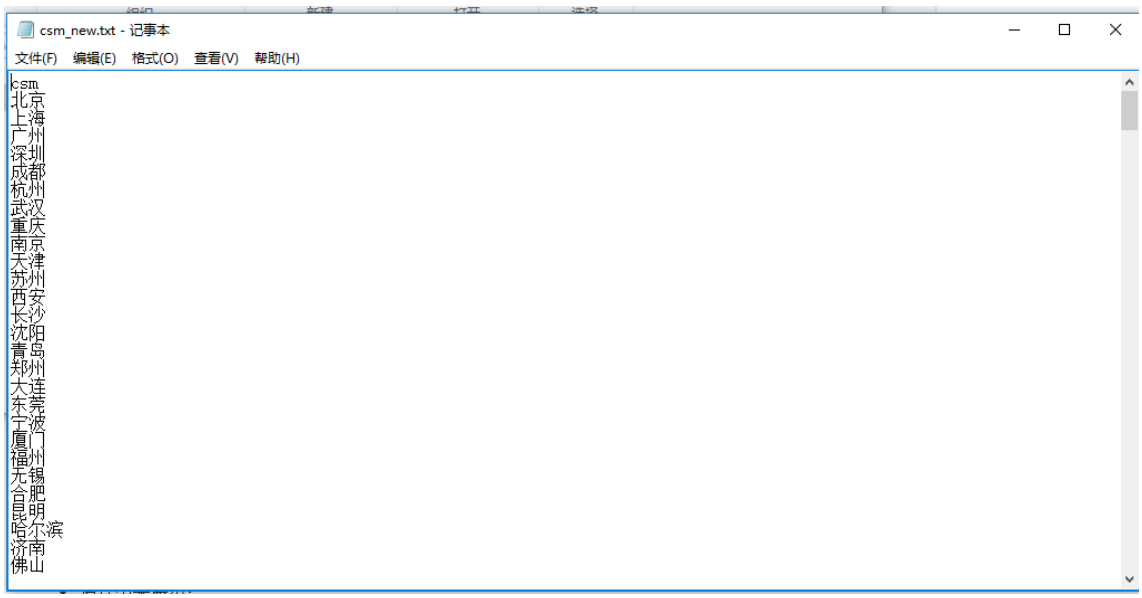
6. 额外数据文件：

- unicode_pinyin.txt文件，文件中列有从4E00-9FA5标准汉字的Unicode编码所对应的拼音，外加一个落单的3007编码的“〇”。（首先将汉字转为Unicode编码，再通过查询这个匹配文件去获得该汉字的拼音）




```
3007    LING2
4E00    YI1
4E01    DING1 ZHENG1
4E02    KAO3 QIAO3 YU2
4E03    QI1
4E04    SHANG4 SHANG3
4E05    XIA4
4E07    WAN4 MO4
4E08    ZHANG4
4E09    SAN1 SAN4
4E0A    SHANG4
4E0B    XIA4
4E0C    JI1
4E0D    BU4 FOU3 FOU1
4E0E    YU3 YU4 YU2
4E0F    MIAN3
4E10    GAI4
4E11    CHOU3
4E12    CHOU3
4E13    ZHUAN1
4E14    QIE3 JU1
4E15    PI1
4E16    SHI4
4E17    SHI4
4E18    QIU1
4E19    BING3
4E1A    YE4
4E1B    CONG2
```

- csm_new.txt文件，由于爬取的城市数据缺少谐音接龙的城市名，这是用于验证谐音城市名接龙的文件




```
csm
北京
上海
广州
深圳
成都
杭州
武汉
重庆
南京
天津
苏州
西安
长沙
沈阳
青岛
郑州
大连
东莞
宁波
厦门
福州
无锡
合肥
昆明
哈尔滨
济南
佛山
```

- 自动运行全部的脚本文件run_all.py：

```python
# -*- coding: utf-8 -*-

import city_chains
import city_crawler
import District

if __name__ == '__main__':
    #可以设置爬取的链接数
    city_crawler.getTypes(20)
    print('crawler complete!')
    #提取爬取数据的城市名，unicode保存
    District.district()
    #0表示为爬取的数据作为词库，其它表示用额外的数据作为词库，验证谐音城市接龙
    p=input("请选择接龙词库（0：表示用爬取的数据作为词库 or else num：用额外的数据作为词库，验证谐音城市接龙）：")
    if p=='0':
        city_chains.city_start(0)
    else:
        city_chains.city_start(p)
```

- 单文件运行顺序：
  - 先运行city_crawler.py文件;接着运行District.py文件;最后运行city_chains.py文件
- 程序测试结果：