Team Project Report

Retrieving high-frequency word of book comments

Team Name: Cloud9
Team Members（names not listed in order）：章腾 雷铭杭 刘佳鑫 娄瑜婕 高雅琪

刘佳鑫：517120910150
章腾：517120910168
雷铭杭：517120910163
娄瑜婕：517120910176
高雅琪：517120910170

1.Introduction

1.1 Functions

Our project can retrieve the high-frequency word of book or film comments on www.douban.com by using BeautifulSoup to crawl the data from the website and using wordcloud to present high-frequency word of the specified book on a given picture. This project can allow users to have a general understanding of the content and key words of comments instead of browsing the comments page by page while failing to catch the main points. Plus, the output of wordcloud, if present in the image of the book's or film's character, can be a highlight of the website, attracting wider attention.

1.2 Method

We use Beautifulsoup to parse web page source code, jieba to divide lines into phrases and after storing comments of the specified book into a txt document, we use wordcloud to generate the ultimate image, and the same procedure is used to generate the image of appendant web page.

2. Task Allocation

娄瑜婕:presentation

章腾:code

雷铭杭:code

刘佳鑫:team project report

高雅琪:PPT

3. Algorithm Description

3.1 import necessary Python library

```
import urllib.request
from bs4 import BeautifulSoup
from wordcloud import WordCloud, ImageColorGenerator
import matplotlib.pyplot as plt
from scipy.misc import imread
import jieba
```

3.2 ask the user to input basic information like book number and requested number of web pages they'd like to browse.

```
i1 = input('输入书号： ')
i2 = input('主网页评论页数： ')
i3 = input('副网页评论页数： ')
```

3.3 define 'get' function which can get the source code of targeted web page

```
def get(x):
    url = x
    headers = {'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36' +
                             '(KHTML, like Gecko) Chrome/65.0.3325.181 Safari/537.36'}
    req = urllib.request.Request(url, headers=headers)
    html = urllib.request.urlopen(req)
    y = BeautifulSoup(html, 'lxml')
    return y
```

3.4 define 'generate' function which can generate the final image with input of picture and txt.

```
def generate(filename, picname):
    # 读取文档并转化为一个词频列表
    comment_text = open(filename, 'r', encoding='utf-8').read()
    cut_text = " ".join(jieba.cut(comment_text))
    # 根据词频绘制图像
    bg_pic = imread('timg3.jpg')
    wordcloud = \
        WordCloud(font_path='simfang.ttf', mask=bg_pic, background_color='white', scale=1.5).generate(cut_text)
    image_colors = ImageColorGenerator(bg_pic)
    plt.imshow(wordcloud)
    plt.axis('off')
    wordcloud.to_file(picname)
```

3.5 'crawl' the data from the home page and store it into 'comments.txt'

```
com = []
for i in range(int(i2)):
    # 搜寻网页的评论文本
    url = 'https://book.douban.com/subject/%s/comments/hot?p=%s' % (i1, str(i))
    soup = get(url)
    comments = soup.findAll('p', {'class': 'comment-content'})
    for comment in comments:
        com.append(comment.get_text())
    # 储存网页的评论文本
with open('comments.txt', 'w', encoding='utf-8') as f:
    for item in com:
        f.write(item)
```

3.6 generate the worcloud image of the home page

```
generate('comments.txt', 'pic.jpg')
```

3.7search the URL of appendant web page and generate a list of URL of sppendant web page

```
url = 'https://book.douban.com/subject/%s/' % i1
soup = get(url)
urls = soup.findAll('a', {'target': '_blank'})
url_s = []
# 选取符合要求的网址，并生成副网址列表
for i in urls:
    i = i.get('href').split('.')  # 分割网址
    try:
        i = i[2].split('/')  # 进一步分割
        if len(i) > 2:  # 读取符合要求的书号
            if i[1] == 'ebook':
                url_ = \
                    'https://read.douban.com/ebook/%s/reviews?start=0&sort=score&competition_only=' % i[2]
                if url_ in url_s:
                    continue
                else:
                    url_s.append(url_)
    except IndexError:
        continue
```

3.8 ‘crawl’ the data from the appendant web page list and store it into a list of txt

```
num = len(url_s)
namelist1 = ['comments'+str(i)+'.txt' for i in range(1, num+1)]  # 命名文件
for i in range(len(url_s)):
    a = str(url_s[i])
    # 创建储存文件
    file_name = namelist1[i]
    y = open(file_name, 'w')
    y.close()
    for j in [str(25*i) for i in range(int(i3))]:
        # 根据页数生成目标网址
        x = a.index('=')
        url_aim = a[:x+1] + j + a[x+2:]
        # 获取网页评论内容
        soup = get(url_aim)
        comments = soup.findAll('div', {'class': 'desc'})
        # 储存文件内容
        for k in comments:
            with open(file_name, 'a', encoding='utf-8') as f:
                f.write(k.text)
```

3.9 generate the image of comments from the appendant web page

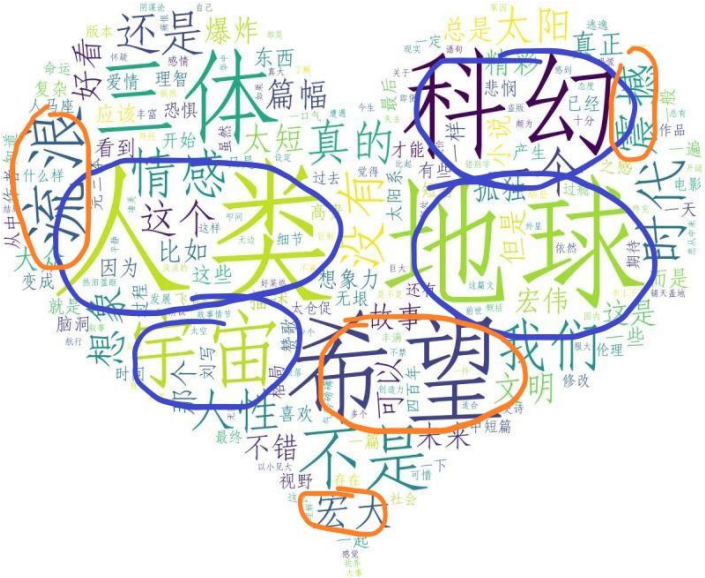
```
namelist2 = ['pic'+str(i)+'.jpg' for i in range(1, num+1)]
for i in range(num):
    generate(namelist1[i], namelist2[i])
```

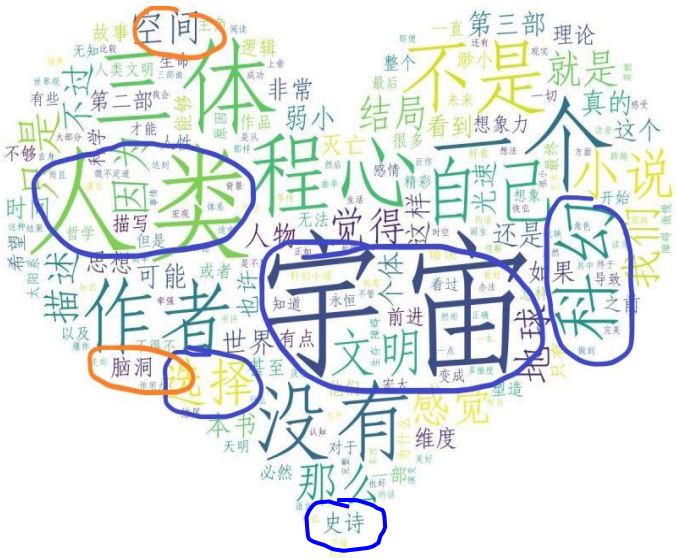
4. testing results

We use part one of ”三体” as our specified book in the process of test, which means that we will retrieve the high frequency word of the comments on the home page of part one of “三体“ and that of its related book.

Home page image

Appendant web page image





Just as highlighted in blue circles, there are some key words which repeat in home page comments wordcloud and appendant web page comments wordcloud, like “宇宙”、”人类”、”科幻”、”逻辑”、”宏大”、”震撼”，which indicate the main theme relevance and similar definition and comments between these books, however there’s also some seemingly contradicted key words in two alleged relevant books, like “历史”and ”科幻”，”希望” and “黑暗”，which implies some theme difference.

5.Conclusion

The recommending system of douban website is precise and reasonable in general, and the relevant book recommended by the website is similar to the targeted book in main theme, writing style and mainstream comments, and the wordcloud form of presentation can generalize the link and difference in a creative way.

6. problems faced in the process of coding

First, we failed to disguise our code as browser and during the debugging process, In the course of trial and error of the code, due to excessive visits on the website, our web scraper was intercepted from further searching data as an IP error had been detected，and refuse our access to the website.

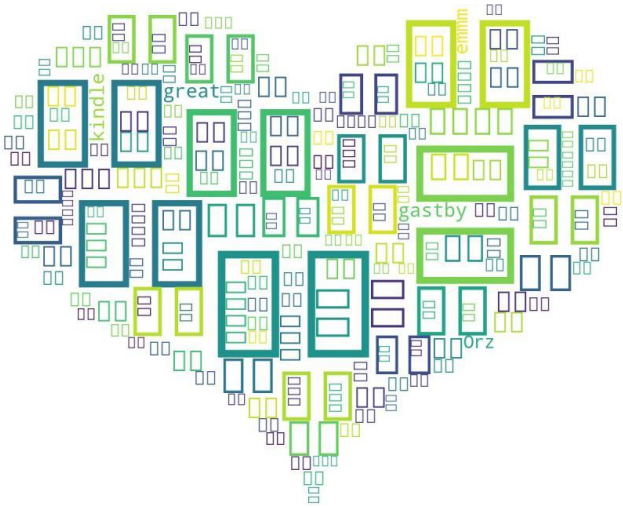
```
(base) C:\Users\spkeal8\Desktop\py>python douban.py
Traceback (most recent call last):
  File "douban.py", line 17, in <module>
    html = urllib.request.urlopen(req) #打开网页
  File "E:\anaconda\lib\urllib\request.py", line 223, in urlopen
    return opener.open(url, data, timeout)
  File "E:\anaconda\lib\urllib\request.py", line 532, in open
    response = meth(req, response)
  File "E:\anaconda\lib\urllib\request.py", line 642, in http_response
    'http', request, response, code, msg, hdrs)
  File "E:\anaconda\lib\urllib\request.py", line 564, in error
    result = self._call_chain(*args)
  File "E:\anaconda\lib\urllib\request.py", line 504, in _call_chain
    result = func(*args)
  File "E:\anaconda\lib\urllib\request.py", line 756, in http_error_302
    return self.parent.open(new, timeout=req.timeout)
  File "E:\anaconda\lib\urllib\request.py", line 532, in open
    response = meth(req, response)
  File "E:\anaconda\lib\urllib\request.py", line 642, in http_response
    'http', request, response, code, msg, hdrs)
  File "E:\anaconda\lib\urllib\request.py", line 570, in error
    return self._call_chain(*args)
  File "E:\anaconda\lib\urllib\request.py", line 504, in _call_chain
    result = func(*args)
  File "E:\anaconda\lib\urllib\request.py", line 650, in http_error_default
    raise HTTPError(req.full_url, code, msg, hdrs, fp)
urllib.error.HTTPError: HTTP Error 403: Forbidden
```

检测到有异常请求从你的 IP 发出，请登录 使用豆瓣。

Therefore it's necessary to disguise the code as Browser, like "headers" in the previous code.

```
def get(x):
    url = x
    headers = {'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36' +
                             '(KHTML, like Gecko) Chrome/65.0.3325.181 Safari/537.36'}
    req = urllib.request.Request(url, headers=headers)
    html = urllib.request.urlopen(req)
    y = BeautifulSoup(html, 'lxml')
    return y
```

Second, the first time we run the code, it presented image like this:



This is because the wordcloud cannot identify Chinese characters, therefore we install a ttf document under the file of the code, which contains the font we need.

7. Reflection

Fundamentally speaking, the process has shown the power of Python library like Beautifulsoup and wordcloud, providing us with valuable learning experience of "crawling" data from website. However, there is still something we can improve.

First, actually jieba cannot identify some specific character name;

Second, Beautifulsoup is, to some extent, a shortcut to parser website source code, and it is of great importance to learn about regular expression if we want to crawl data more accurately;

Finally, trial and error is the imperative part in programming, especially in the field of web scraper, which poses great challenges upon coders, as they have to think considerately, and prepare for any circumstance that will happen. For the time being, the grammar of try…except… is very helpful.

Actually, the targeted book is called “了不起的盖茨比”, but jieba just split “盖茨比” into two words because its database has not such character names in literary.

Second, some of the words collected in the wordcloud is meaningless, like “一个”、“没有”、“自己”, these words are commonly used in daily conversation, but not necessary when analyzing key words of comments.