Retrieving high-frequency word of book comments

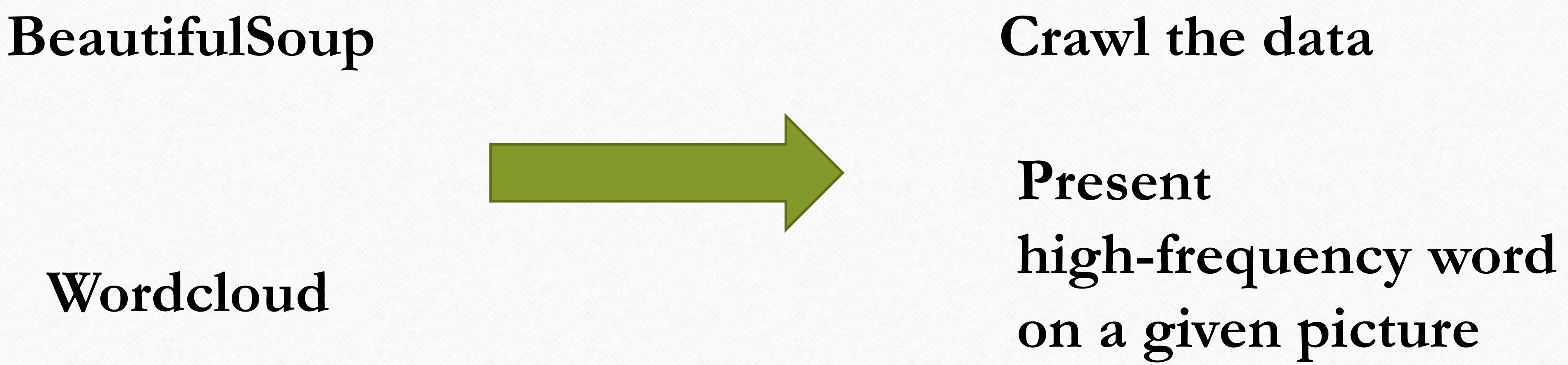
team name:cloud9

Part 0: Prologue

- **Questions**
- Do you use Douban（豆瓣） to get comments of a book？
- Do you ever feel bothered to get the point while getting through pages of comments in Douban？
- Do you want a program which can screen tons of comments and take out of the keywords of a book？

Part 1:Introductions to our project

- Functions



BeautifulSoup

Wordcloud

Crawl the data

Present high-frequency word on a given picture

Part 1: Introductions to our project

- **Practical use**

A general understanding of the content ○
Key words of comments

🚫 **Browsing the comments page by page**
Failing to catch the main point

The output can be a highlight ★

Part 1: Introductions to our project

- Methods

BeautifulSoup → Parse web page source code

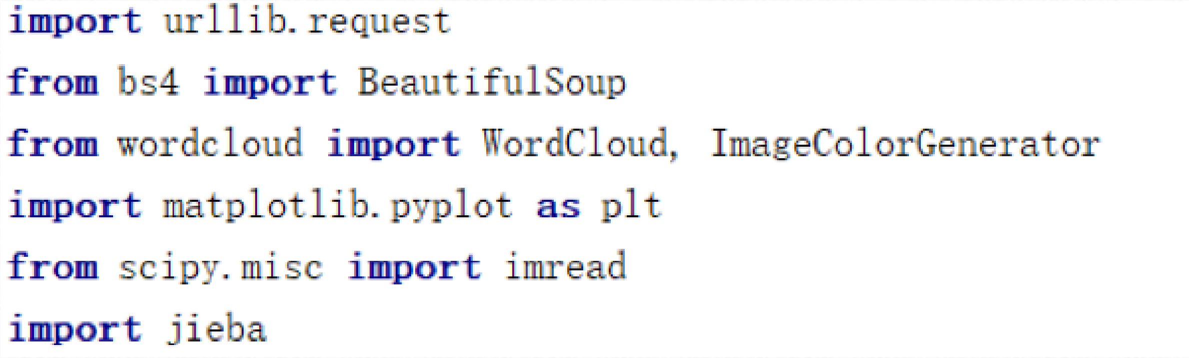
jieba → Divide lines into phrases

comments → A txt document

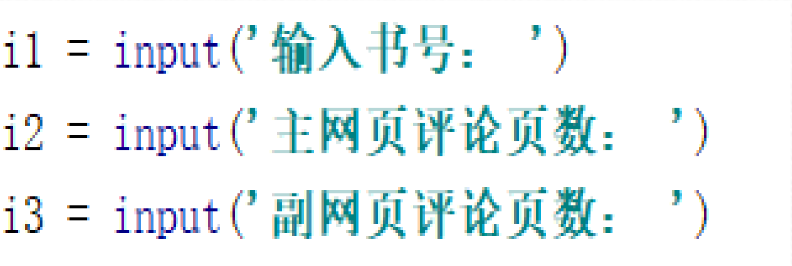
Wordcloud → Generate the ultimate image

Part 2: Algorithm Descriptions

- 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
```

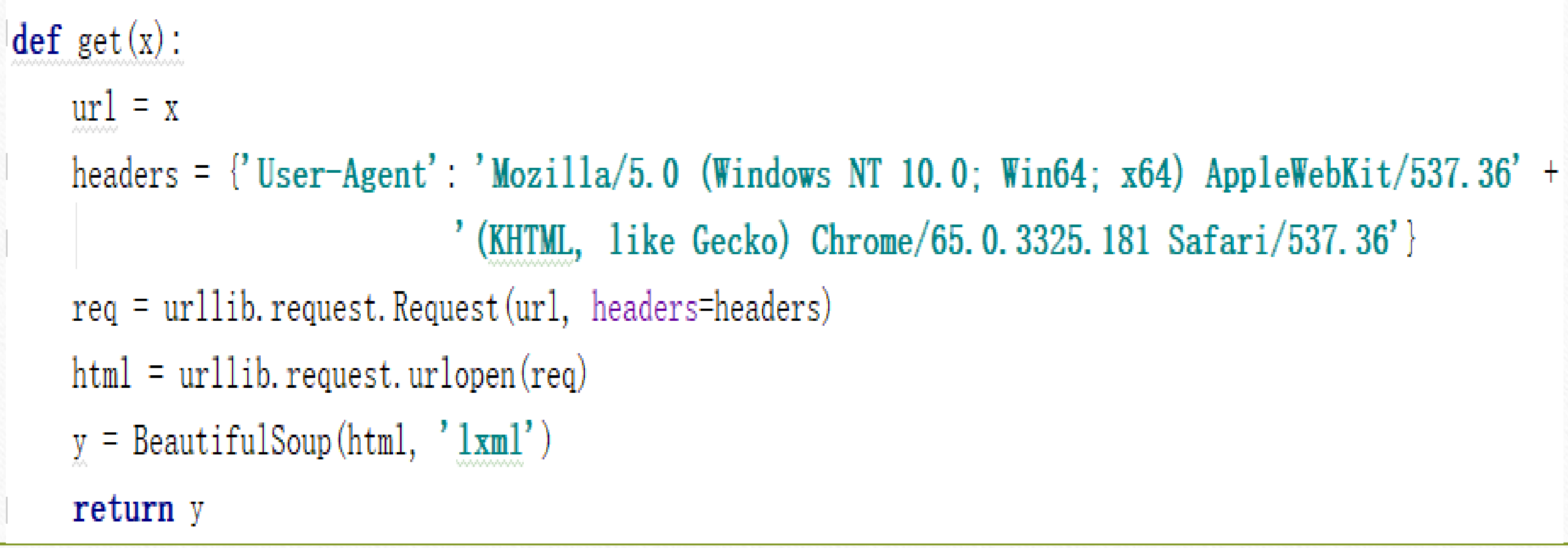


```
i1 = input(’输入书号： ’)
i2 = input(’主网页评论页数： ’)
i3 = input(’副网页评论页数： ’)
```

- Ask the user to input basic information like book number and requested number of web pages they’d like to browse.

Part 2: Algorithm Descriptions

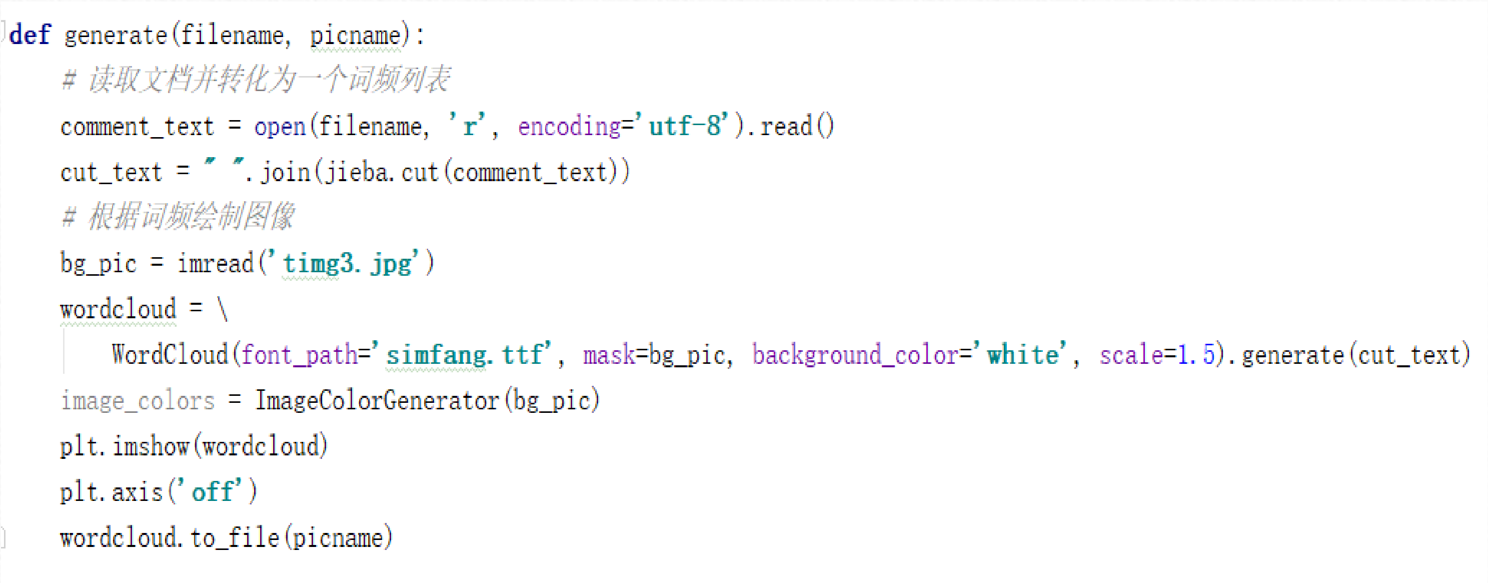
- Define ‘get’ function which can get the source code of targeted web page.



```
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
```

Part 2: Algorithm Descriptions

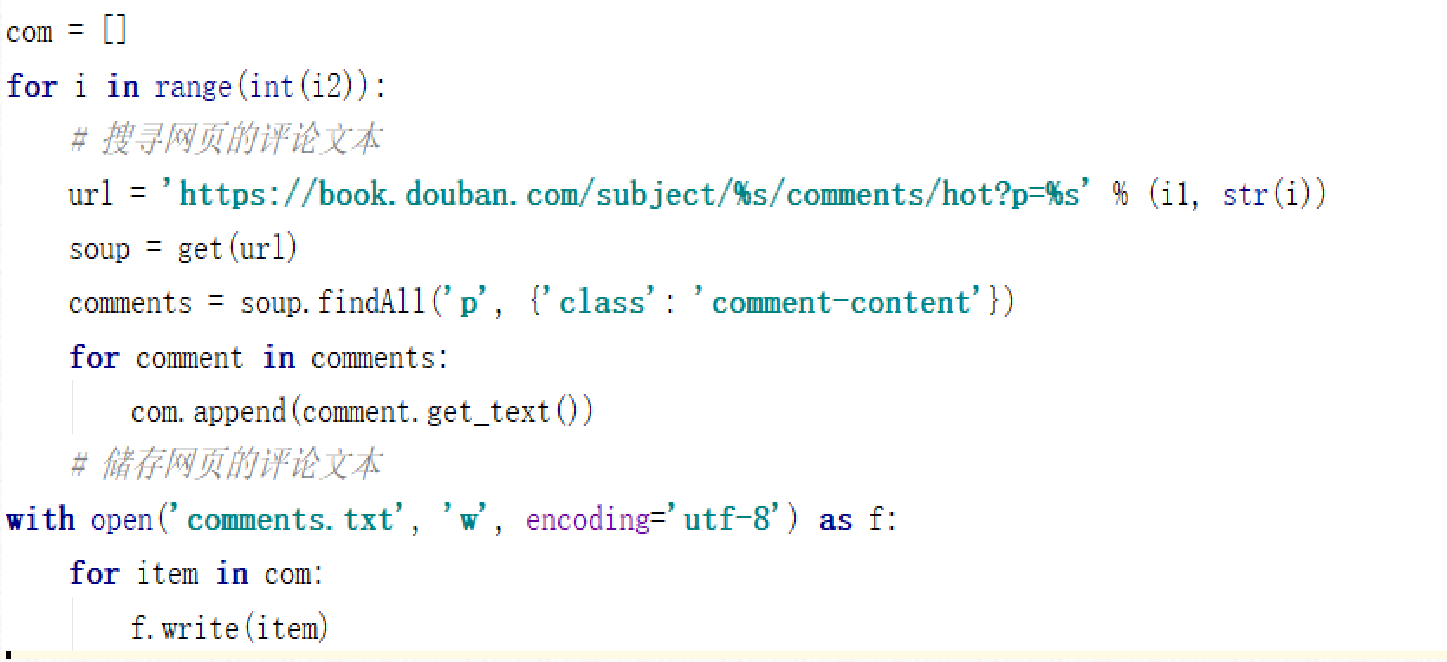
- Define ‘generate’ function which can generate the final image with input of picture and txt.



```
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)
```

Part 2: Algorithm Descriptions

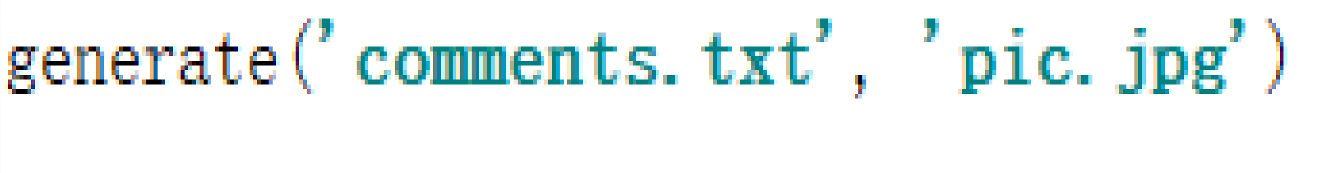
- ‘Crawl’ the data from the home page and store it into ‘comments.txt’.



```
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)
```

Part 2: Algorithm Descriptions

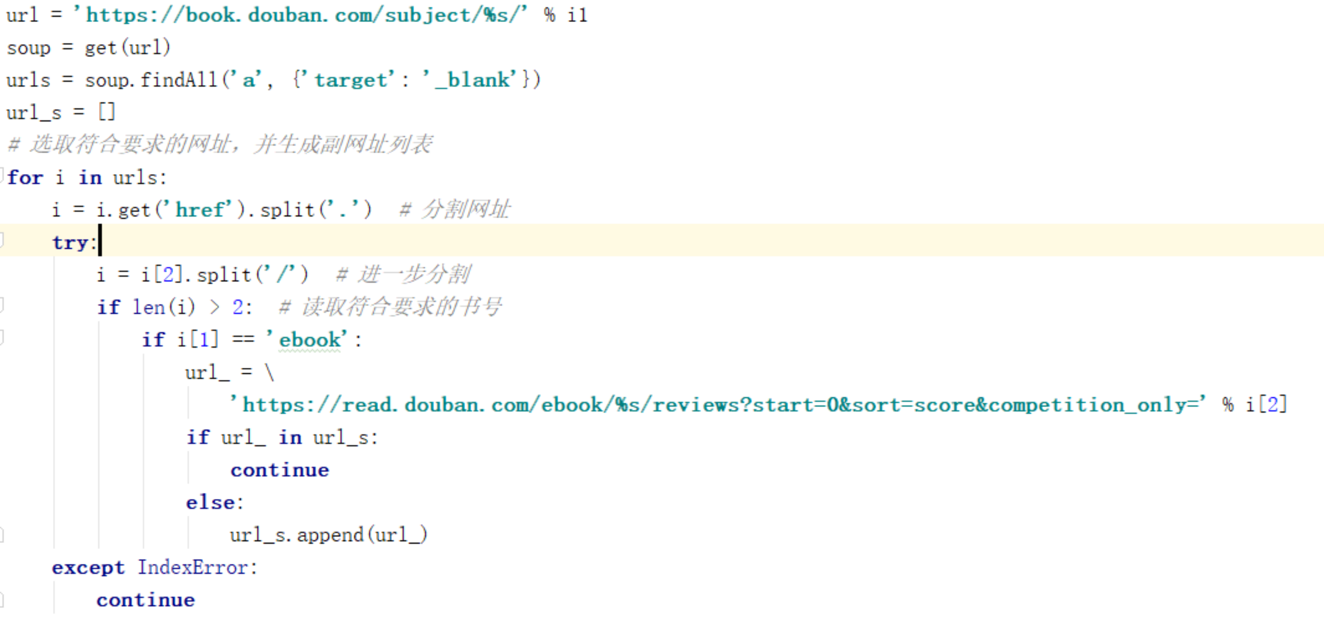
- Generate the worcloud image of the home page.



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

Part 2: Algorithm Descriptions

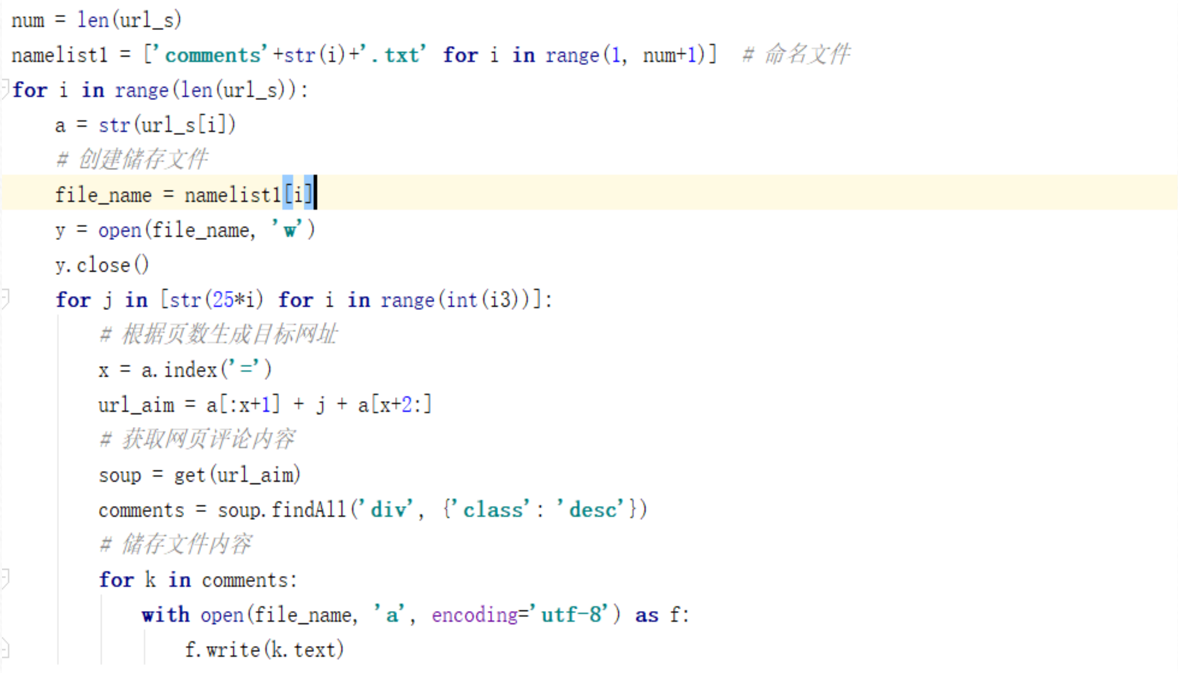
- Search the URL of appendant web page and generate a list of URL of appendant web page.



```python
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
```

Part 2: Algorithm Descriptions

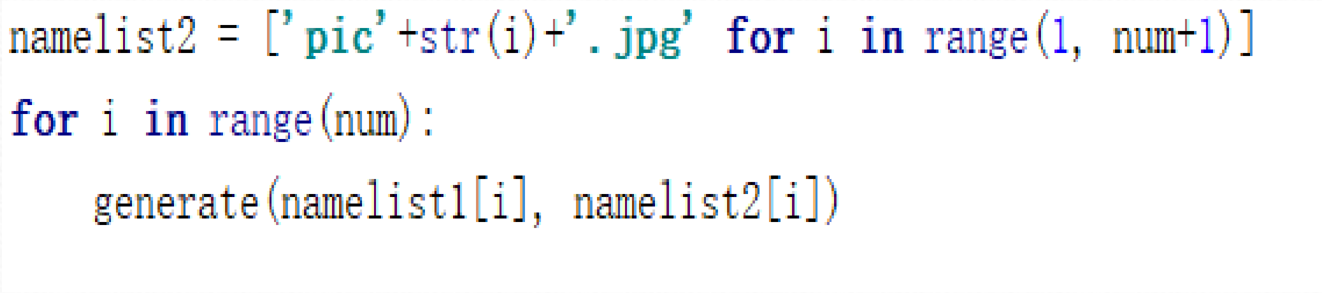
- ‘Crawl’ the data from the appendant web page list and store it into a list of txt.



```python
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)
```

Part 2: Algorithm Descriptions

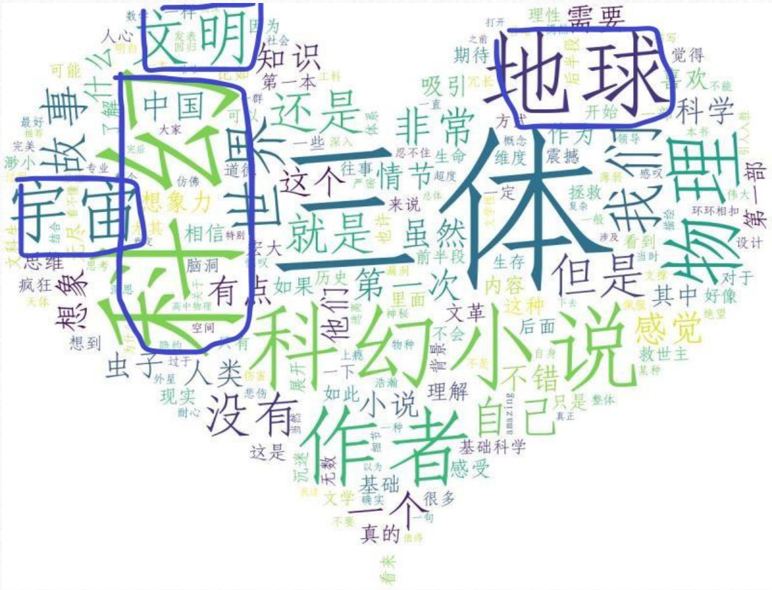
- Generate the image of comments from the appendant web page.

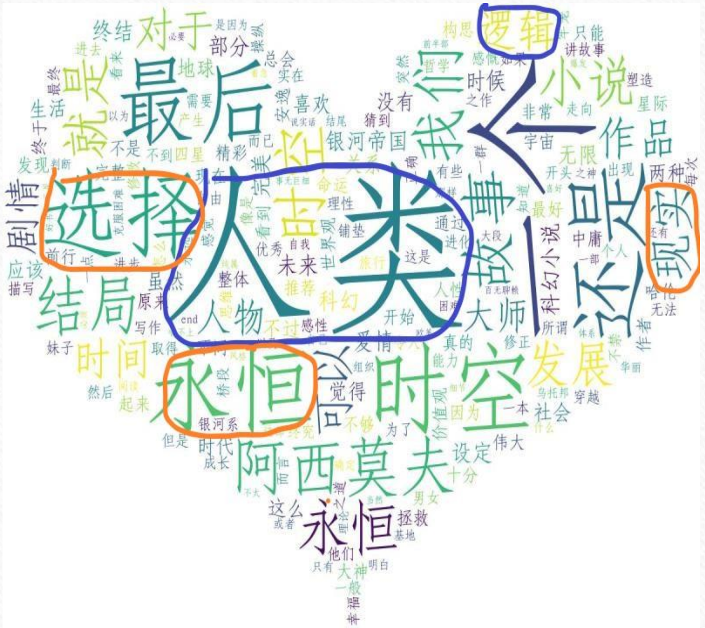


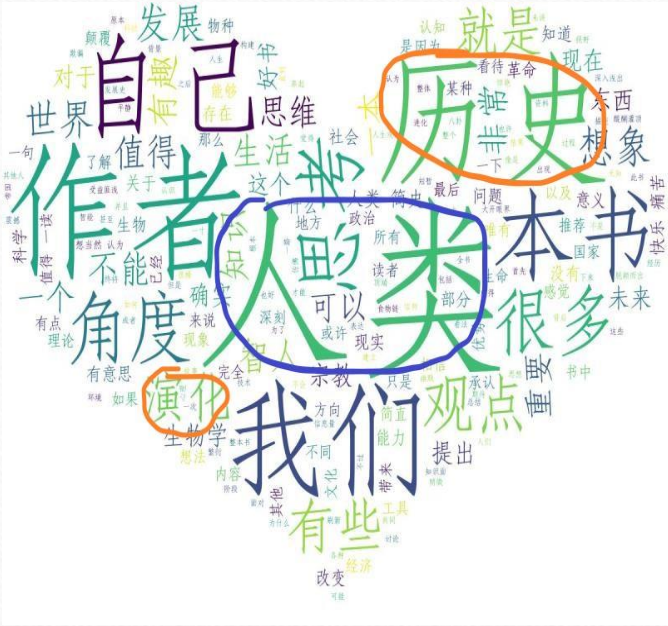
```
namelist2 = ['pic'+str(i)+'.jpg' for i in range(1, num+1)]
for i in range(num):
    generate(namelist1[i], namelist2[i])
```

Part 2: Algorithm Descriptions

- **Testing results**

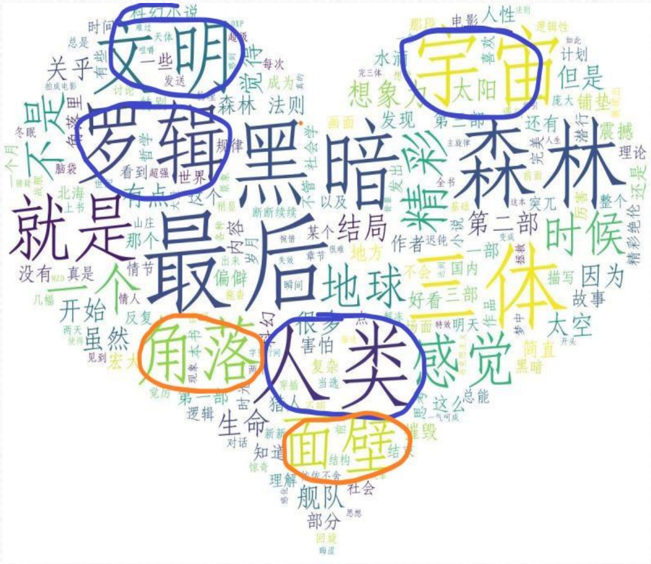


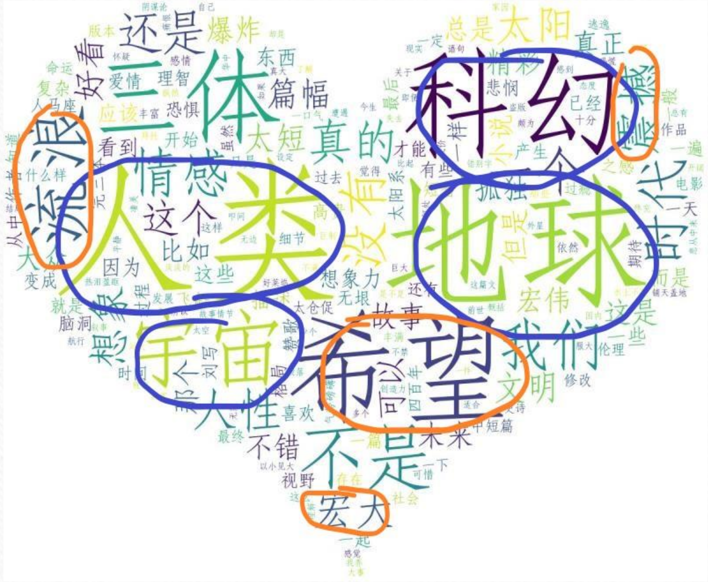


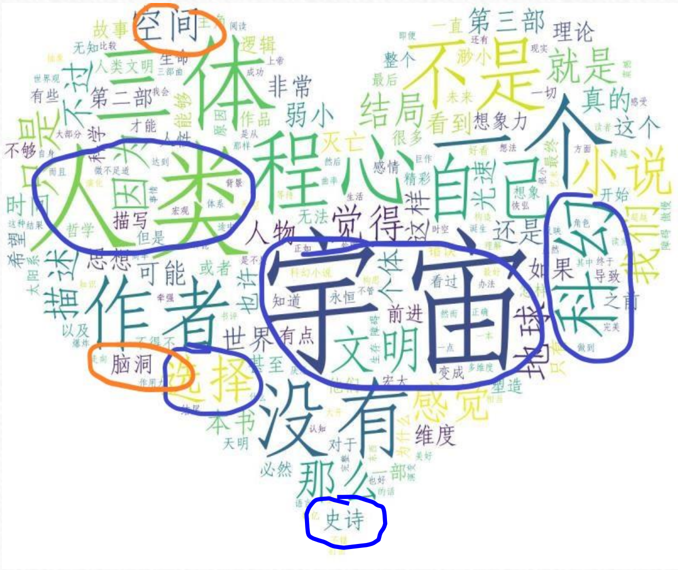


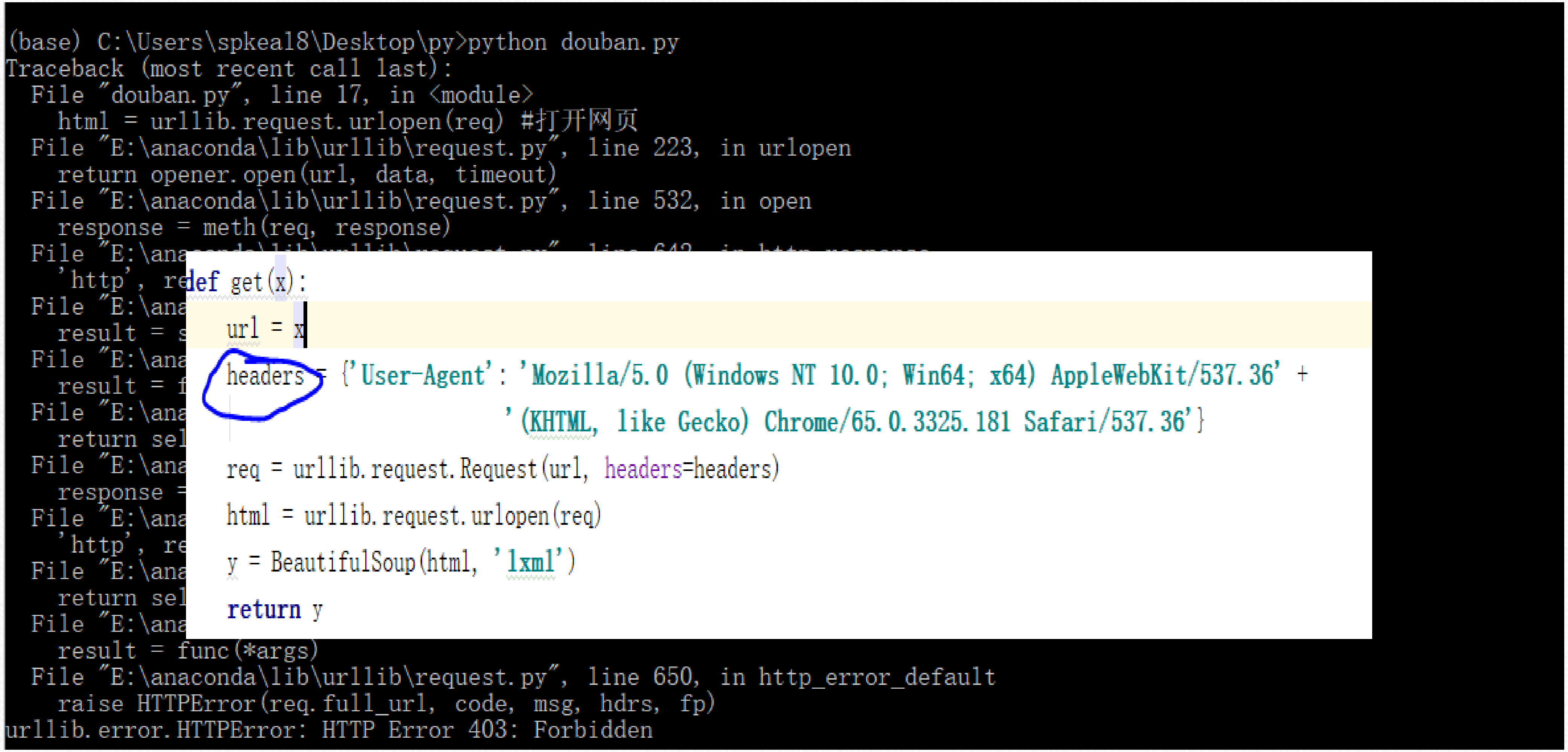
Part 2: Algorithm Descriptions

- **Testing results**







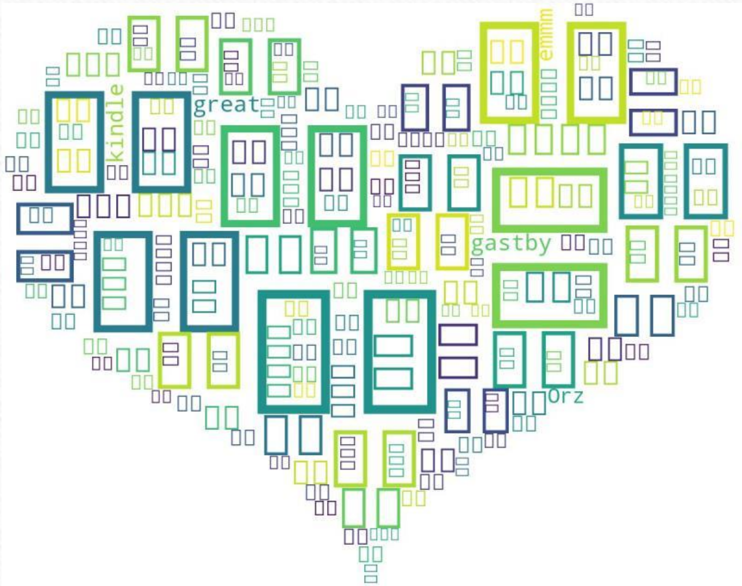


```
(base) C:\Users\spkea18\Desktop\py>python douban.py
Traceback (most recent call last):
  File "douban.py", line 17, in <module>
    html = urllib.request.urlopen(req) #打开网页
  File "E:\anaconda\lib\urllib\request.py", line 223, in urlopen
    return opener.open(url, data, timeout)
  File "E:\anaconda\lib\urllib\request.py", line 532, in open
    response = meth(req, response)
  File "E:\ana
    'http', re
  File "E:\ana
    result = s
  File "E:\ana
    result = f
  File "E:\ana
    return sel
  File "E:\ana
    response =
  File "E:\ana
    'http', re
  File "E:\ana
    return sel
  File "E:\ana
    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
```

```
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
```

Part 3: Problems

- **Problem 2: The wordcloud cannot identify Chinese characters.**



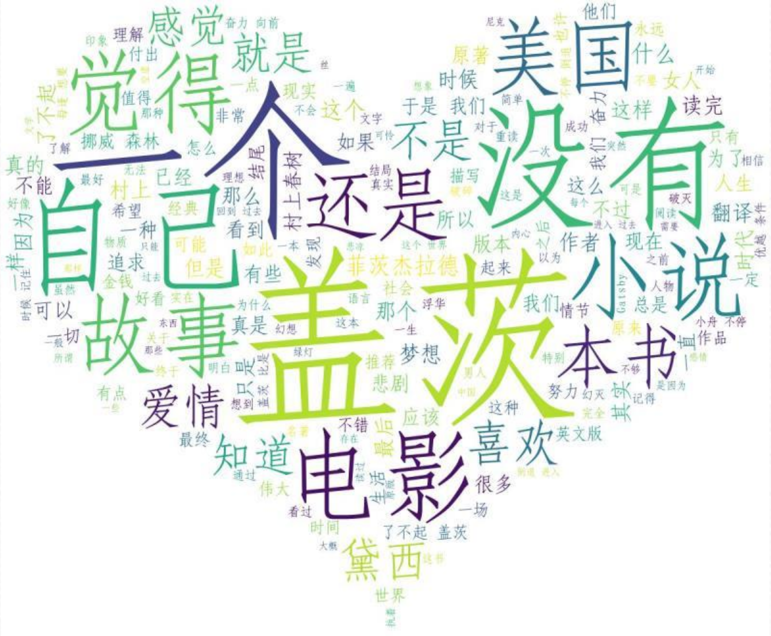
Part 4: Conclusions

- Powerful Python libraries
- Further problems to solved

a)BeautifulSoup

b)trial and error

c)jieba


盖茨
自己
一个
没有
美国
小说
电影
故事
觉得
感觉
就是
还是
不是
本书
爱情
知道
喜欢
黛西
村上春树
菲茨杰拉德
时候
原著
什么
这样
读完
翻译
人生
作者
版本
所以
梦想
悲剧
推荐
应该
最后
世界
很多
一场
了不起
伟大
现在
时代
作品
情节
其实
努力
英文版
这种
不错
最终
生活
时间
可以
一切
追求
金钱
可能
但是
经典
那么
已经
村上
希望
一种
挪威
森林
真的
不能
因为
理解
值得
如果
这个
于是
我们
他们
永远
女人
只有
为了
这么
不过
社会
那个
真是
有些
看到
非常
现实
一点
描写