循环结构进行数据简单整理
基于Python词云可视化《三体》文本

昆明市第一中学西山学校 齐洪

词云可视化《三体》

- 尝试通过Python词云可视化呈现刘慈欣《三体》的三部曲，看看作者在每部作品中的主要人物，以及每部作品的关键词有哪些？通过词云图了解《三体》的轮廓。
- 在词云可视化小说过程中，了解数据分析处理的基本过程，掌握循环结构整理数据的过程与方法。

2023年10月世界科幻大会在中国成都举办
图中刘慈欣为“2023雨果奖”获得者海漄颁奖
（海漄《时空画师》获得最佳短中篇小说奖）

本节课的学习任务

三体1.txt

文件 编辑 查看

边，拨开了上面丛生的藤蔓，露出了斑驳的铁锈，其他人这才发现"岩石"原来是一个巨大的金属基座。

"这是天线的基座。"叶文洁说。地球文明被外星世界听到的第一声呼唤，就是通过这个基座上的天线发向太阳，再由太阳放大后向整个宇宙转发的。

人们在基座旁发现了一块小小的石碑，它几乎被野草完全埋没，上书：

红岸基地原址
（1968~1987）
中国科学院
1989.03.21

碑是那么小，与其说是为了纪念，更像是为了忘却。

叶文洁走到悬崖边，她曾在这里亲手结束了两个军人的生命。她并没有像其他同行的人那样眺望云海，而是把目光集中到一个方向，在那一片云层下面，有一个叫齐家屯的小村庄……

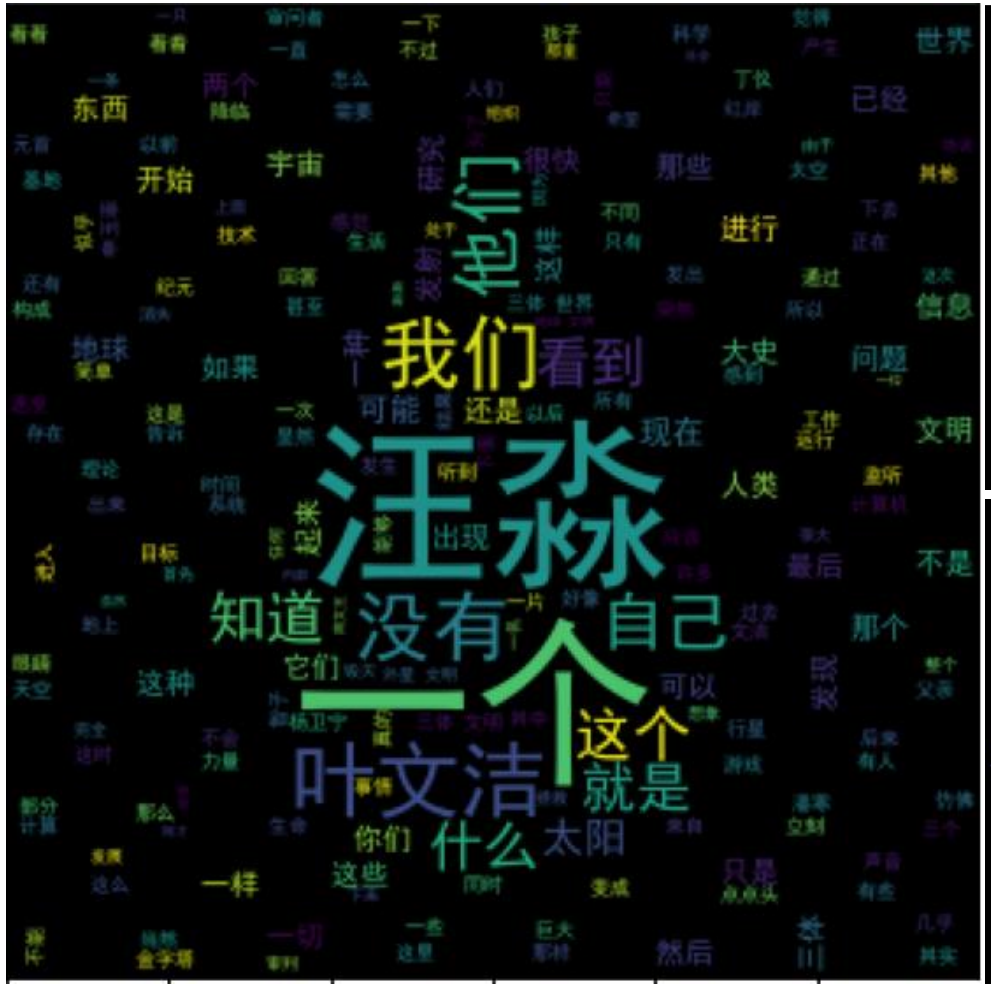
叶文洁的心脏艰难地跳动着，像一根即将断裂的琴弦，黑雾开始在她的眼前出现，她用尽生命的最后能量坚持着，在一切都没入永恒的黑暗之前，她想再看一次红岸基地的日落。

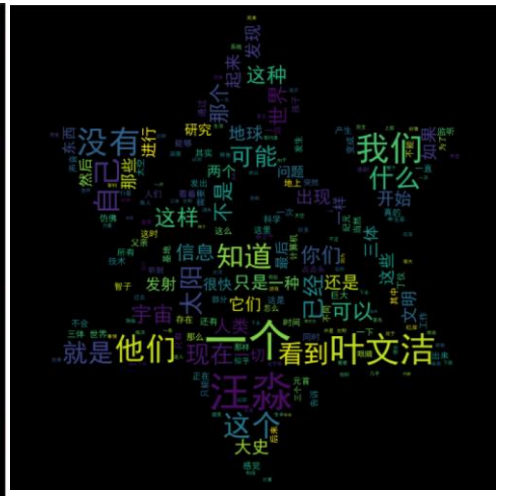
在西方的天际，正在云海中下沉的夕阳仿佛被融化了，太阳的血在云海和天空中弥漫开来，映现出一大片壮丽的血红。

"这是人类的落日……"叶文洁轻轻地说。

（全文完）

行 3687，列 7 100% Windows (CRLF) ANSI





三体小说文本数据的词云可视化

既然要对数据进行分析，首先要……


```
import requests
from bs4 import BeautifulSoup

urls=[]#用来存每个章节的网址
#url="http://www.songyuwenxue.com/322/322700/index.html"
url='https://wap.tianyabook.com/shu/2681.html'
headers = {'User-Agent':'Mozilla/5.0 (Windows NT 10.0; WOW64; Trident/7.0; rv:11.0) like Gecko'}
r=requests.get(url,headers=headers)  #向网站服务器发送请求
r.encoding=r.apparent_encoding
html=r.text
soup=BeautifulSoup(html,'html.parser')  #解析源代码
#ts=soup.select('div[class="clearfix dirconone"] a')
ts=soup.select('div[id="list-chapterAll"] a')
#print(ts)
for t in ts:
    url=t.get('href')#得到网址超链接
    if len(url)==0:continue#如果网址为空，过滤
    #https://wap.tianyabook.com/shu/9442/1109189.html
    if url[0:4]!='http':url='https://wap.tianyabook.com'+url
    urls.append((t.text,url))
```

数据
三体文本
数据哪儿来

数据收集的方式：

- 实验数据采集
- 调查研究采集
- 网络数据采集
- 系统日志采集
- 传感器设备采集
- ……

文本数据词云可视化的基本过程

```
1 import matplotlib.pyplot  as plt
2 import wordcloud as wc
3 import numpy as np
4 from PIL import Image
5
6 f=open("三体1.txt",encoding="ANSI")
7 fs=f.read()
8 f.close()
9
10 bg = np.array(Image.open("1.jpg"))
11 a=wc.WordCloud(font_path="simhei.ttf",mask=bg)
12 a.generate(fs)
13
14 plt.imshow(a)
15 plt.savefig(fname="三体词云图.jpg")
16 plt.show()
```

导入数据分析处理
相应的外部库
matplotlib(绘图库)
wordcloud(词云库)

打开并读取文本数据

通过词云库wordcloud
提取特征生成词云

通过绘图库matplotlib
绘制显示词云图

词云可视化三体文本
初步分析

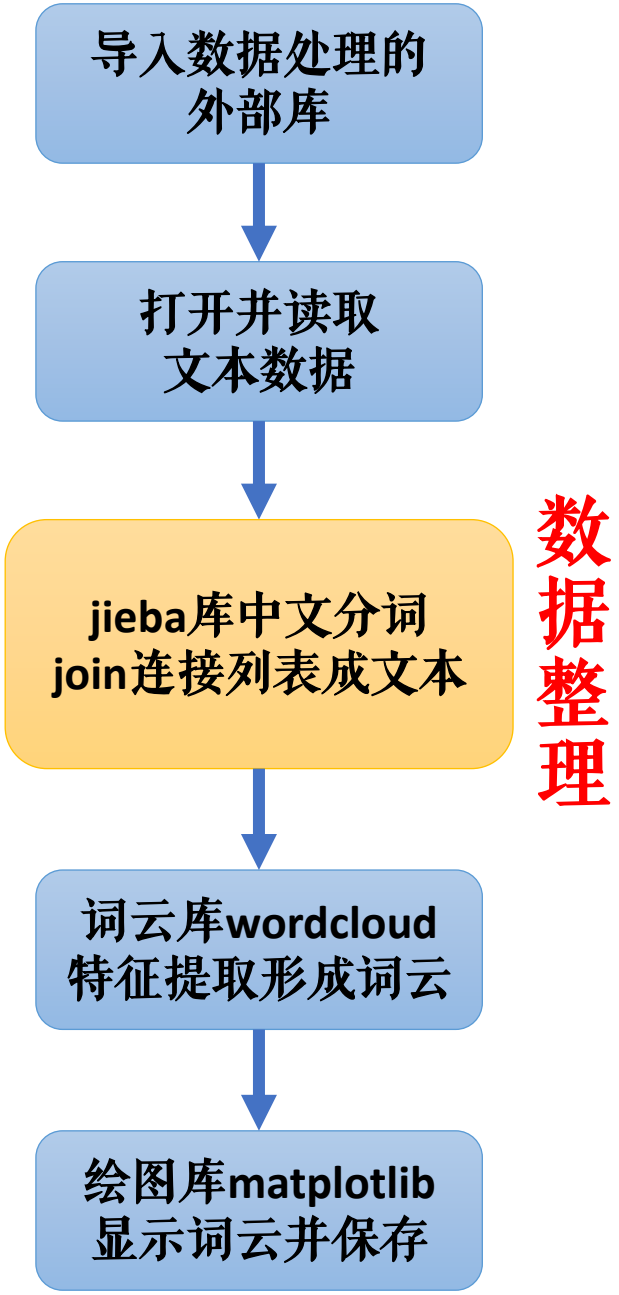
- 结合上一节课对“云南”百度词条分析的思考上面代码的作用？
- 结合现在代码及词云图，思考有什么问题？什么原因导致的？

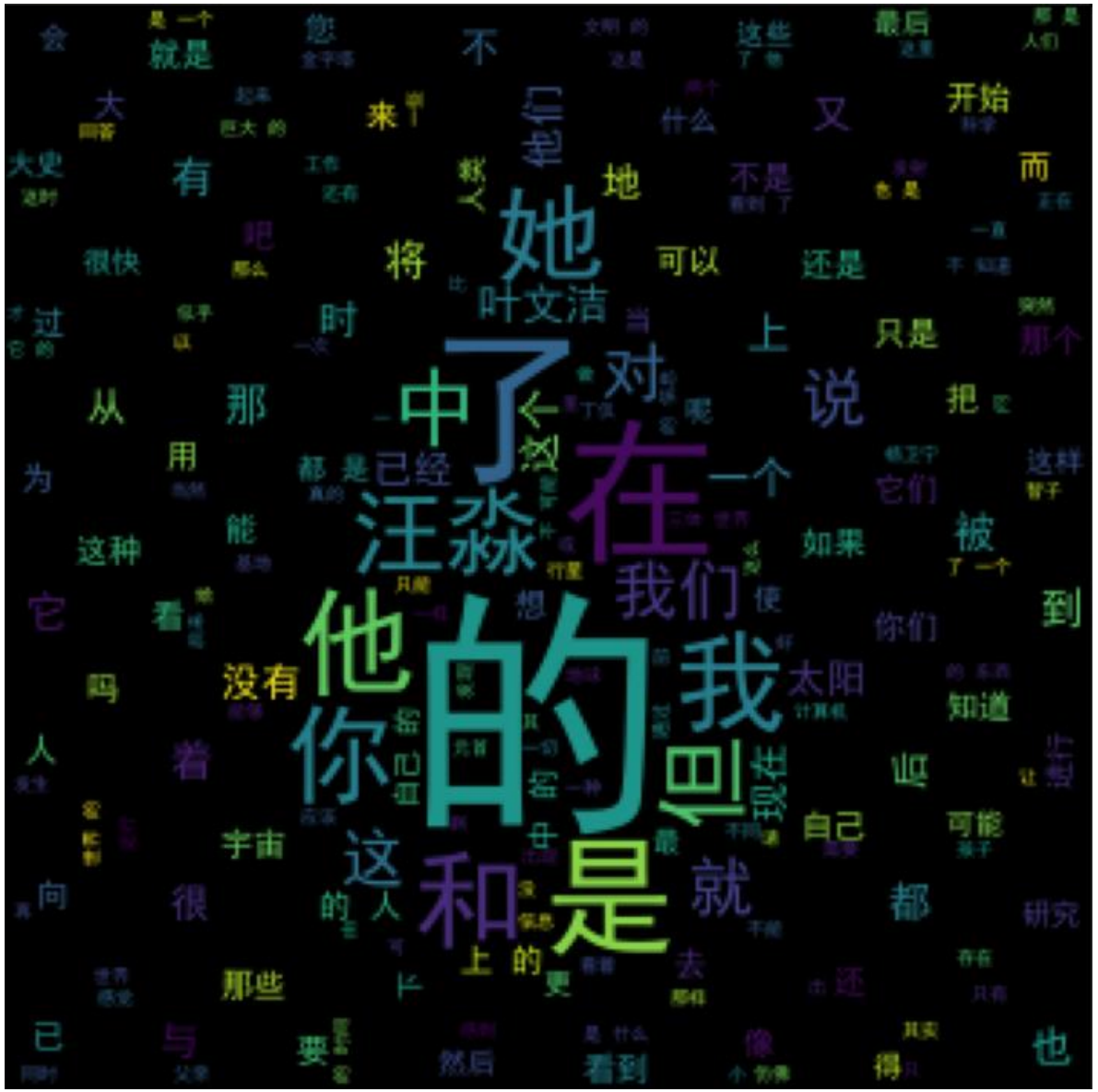
数据整理活动1：jieba库分词

- 在上面已有代码的基础上对三体第一部（三体1）进行中文分词处理，并思考以下问题：
  - 通过jieba.lcut()产生的数据类型是什么？
  - 为什么要使用" ".join()将分词后的文本连接起来？
  - 通过分词库处理后的词云图中，哪些词汇最多，反应了中文文本的什么特点？

先分析文件量较小的文本三体0.txt，代码测试没问题再分析文件量大的数据

三体0.txt	三体1.txt	三体2.txt	三体3.txt
三体第一部第一章 6000字 数据量小 分析快	三体第一部全部 约20万字 数据量大 分析慢	三体第二部全部 约31万字 数据量大 分析慢	三体第三部全部 约36万字 数据量大 分析慢

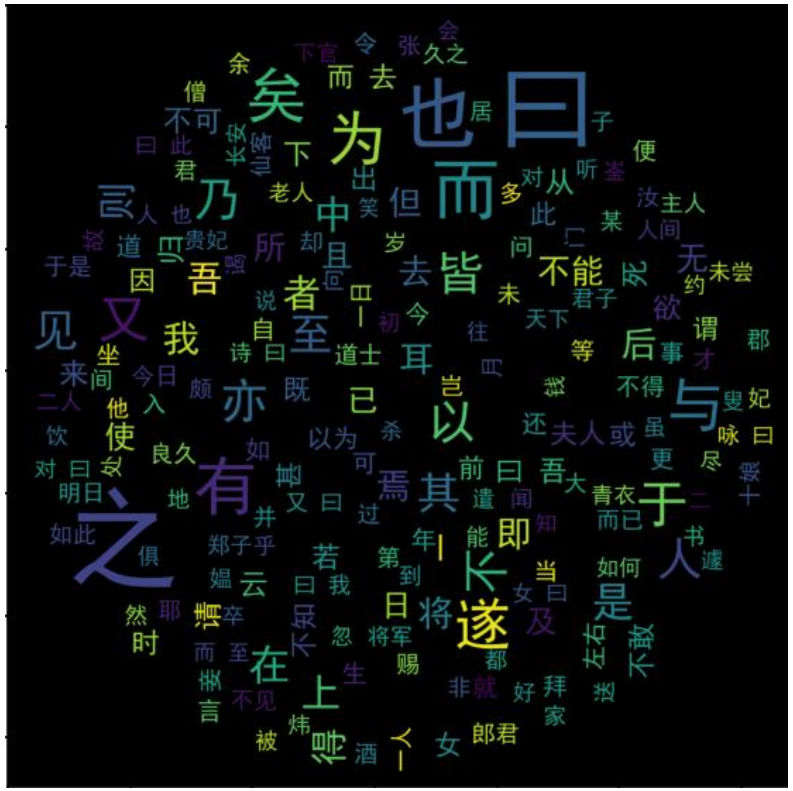




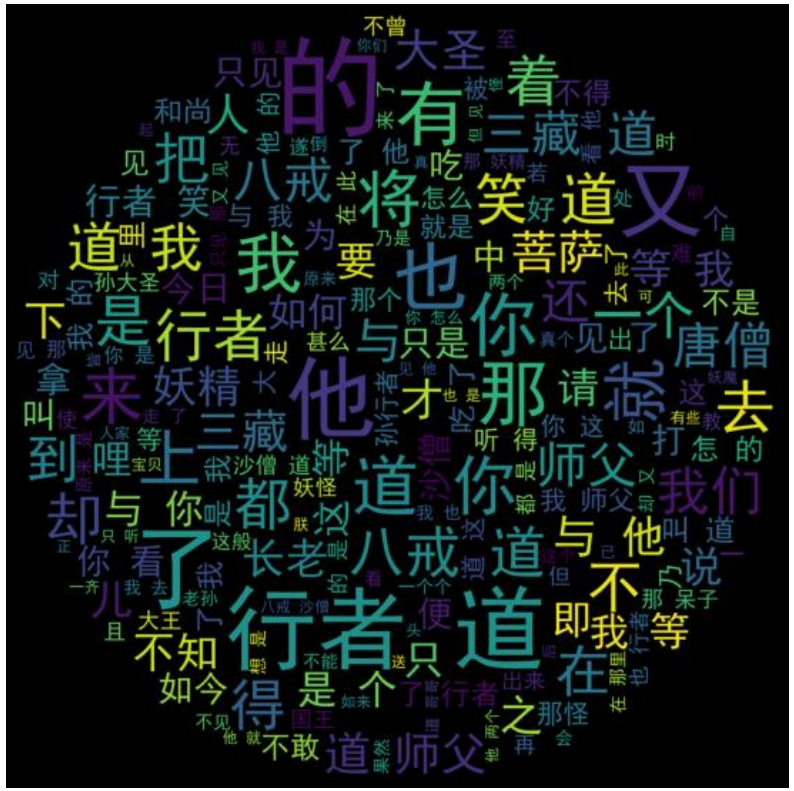
- 通过jieba.lcut()产生的数据类型是什么？
- **列表**

- 为什么要使用" ".join()将分词后的列表文本用空格连接起来？
- **Wordcloud词云库分析的数据是文本字符串，提取词频依赖于文本中的空格及标点符合**

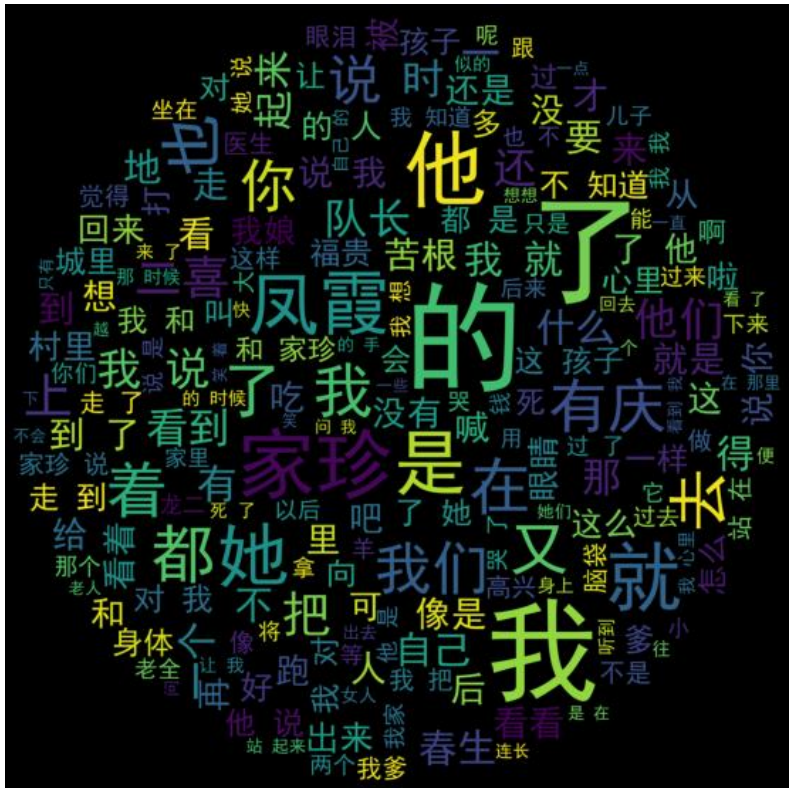
- 通过分词库处理后的词云图中，哪些词汇最多，反应了现代中文文本的什么特点？
- **的、了、在、是、他、我、你等词汇**



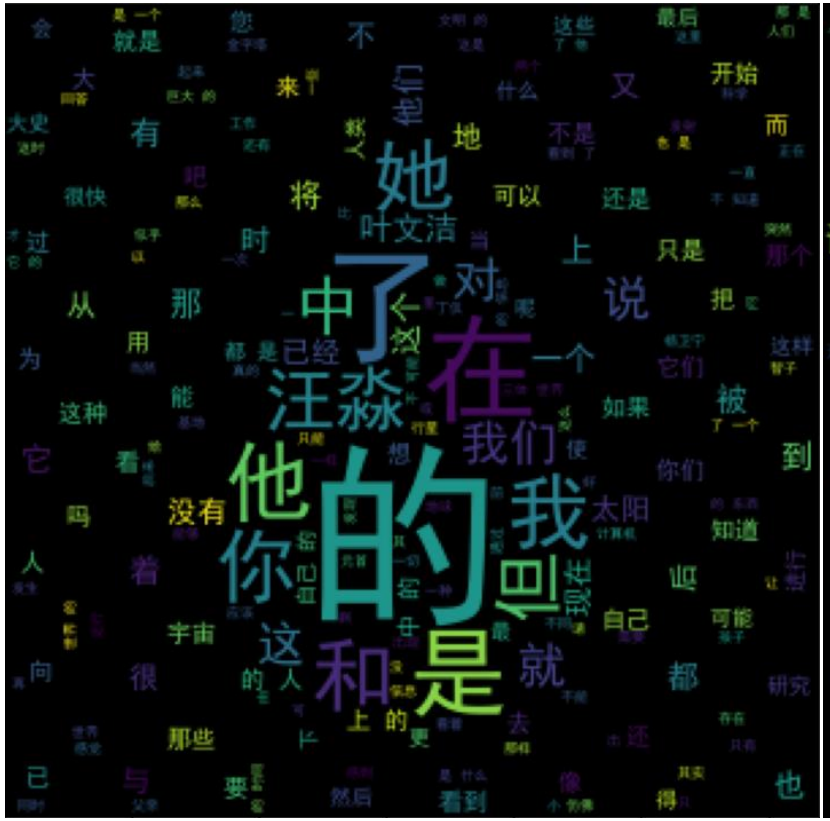
《唐宋传奇》



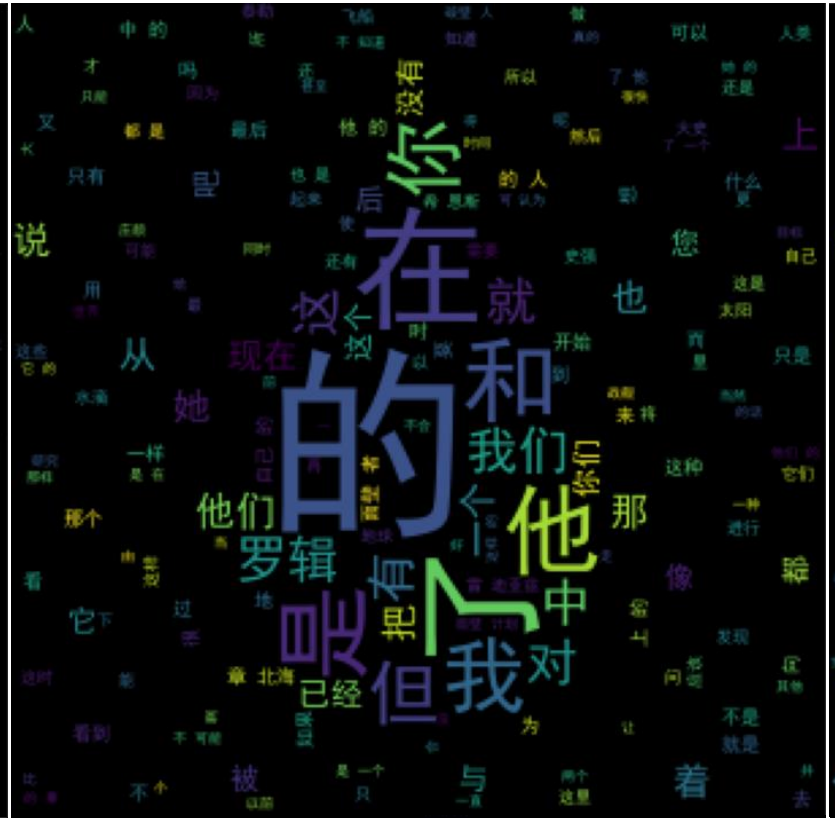
《西游记》



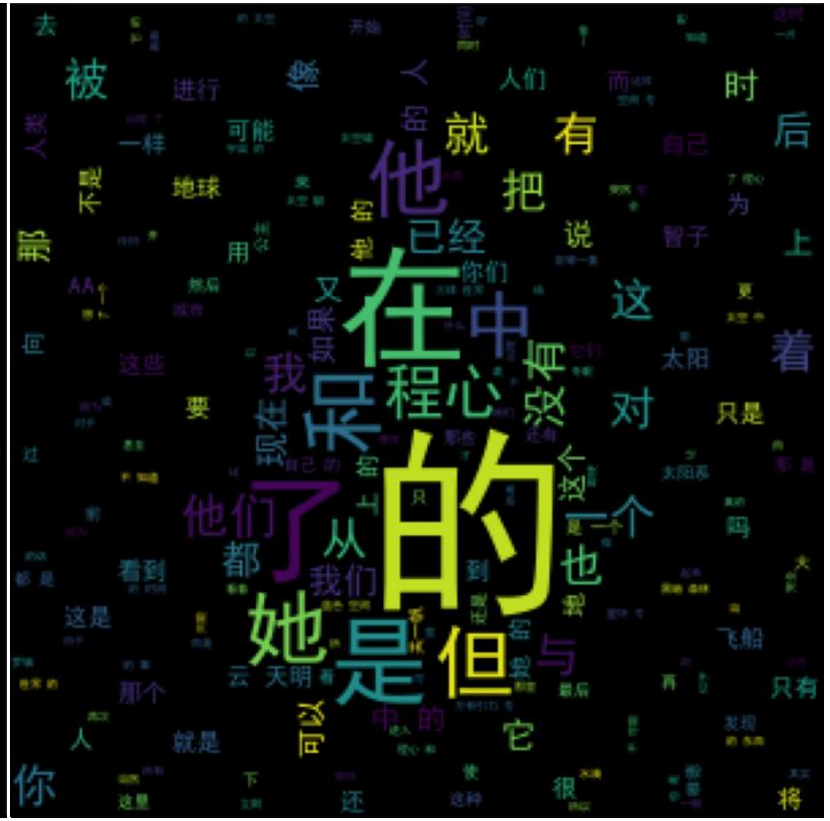
《活着》



三体 I



三体II

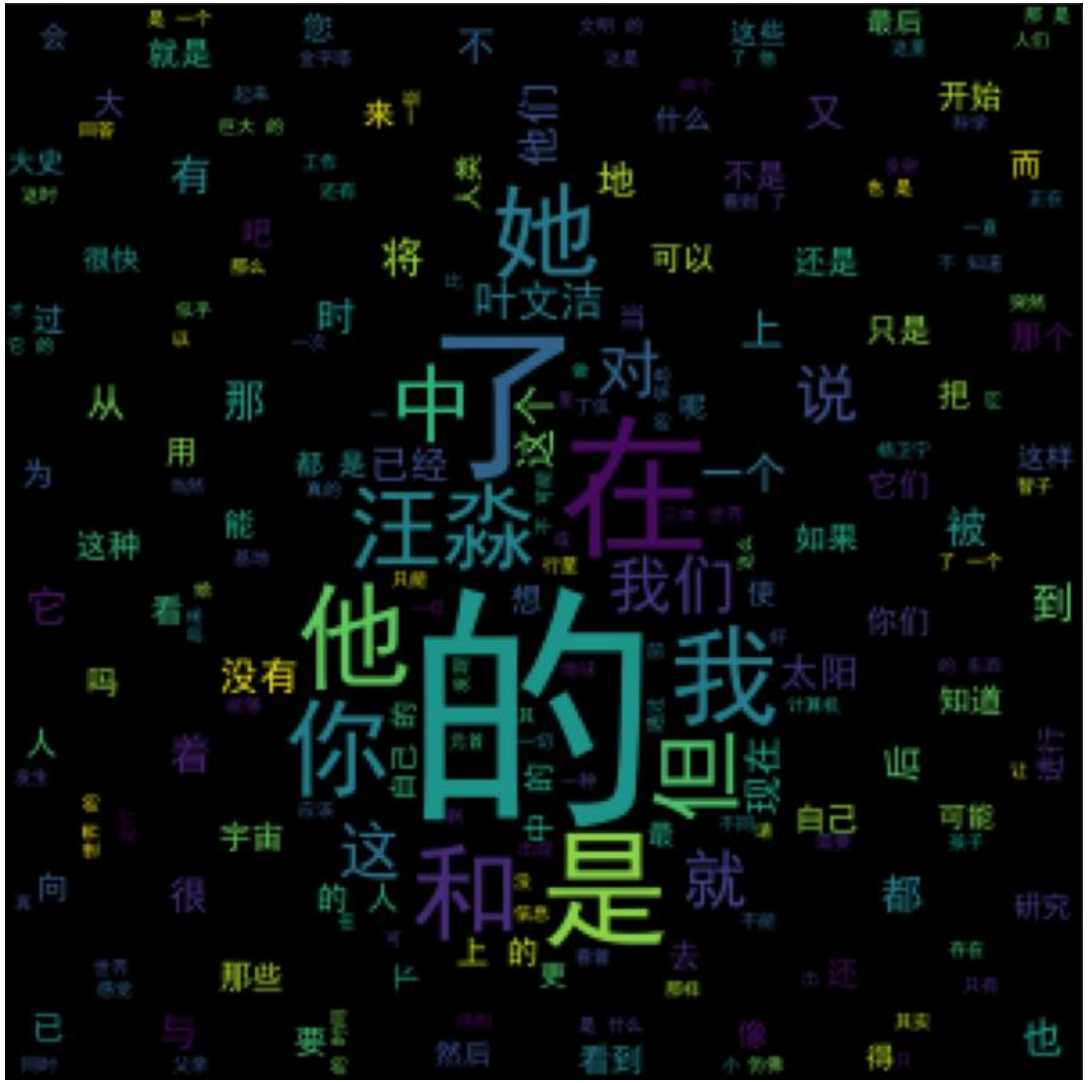


三体III

- 反映了现代小说文本中的一些用词特点，“的”助词使用最为频繁，“的”是现代汉语里定语的标志词，基本上我们每个句子里都会有这个字；
- 你、我、他是常用的代词，介词“在”、助词“了”等这些都是现代汉语句子中常用词语。

```python
import matplotlib.pyplot  as plt
import wordcloud as wc
import numpy as np
from PIL import Image
import jieba

f=open("三体1.txt",encoding="ANSI")
fs=f.read()
f.close()

words=jieba.lcut(fs)
ds=" ".join(words)

bg = np.array(Image.open("1.jpg"))
a=wc.WordCloud(font_path="simhei.ttf",mask=bg)
a.generate(ds)

plt.imshow(a)
plt.savefig(fname="三体词云图.jpg")
plt.show()
```

jieba库进行中文分词



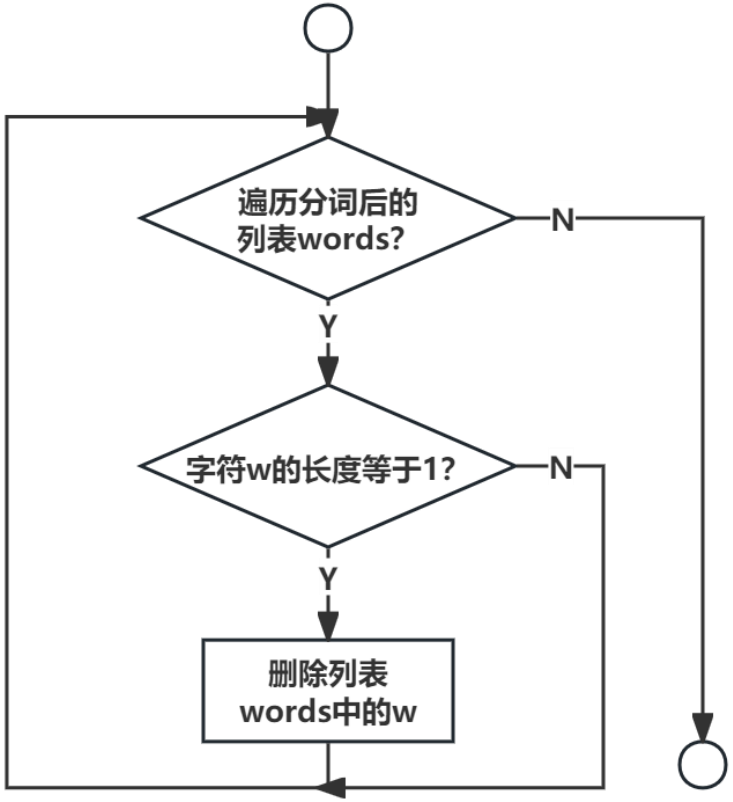
进一步进行数据整理：去除单个词语

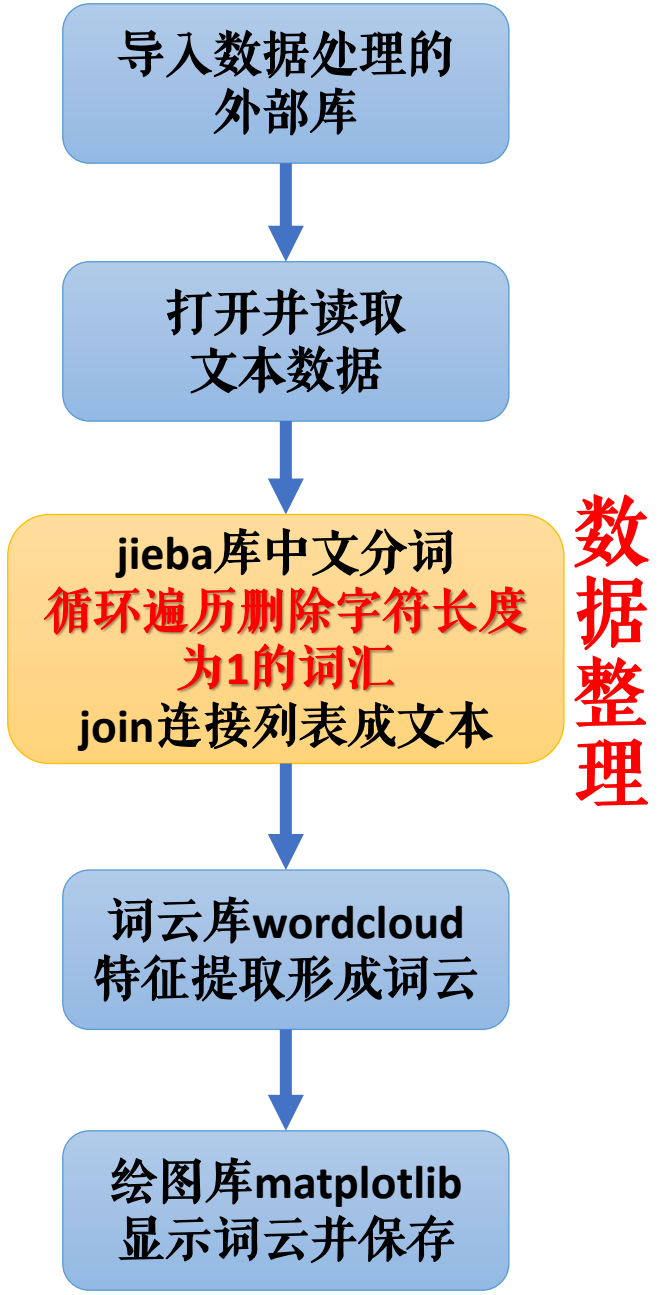
数据整理活动2：
循环结构删除单文本

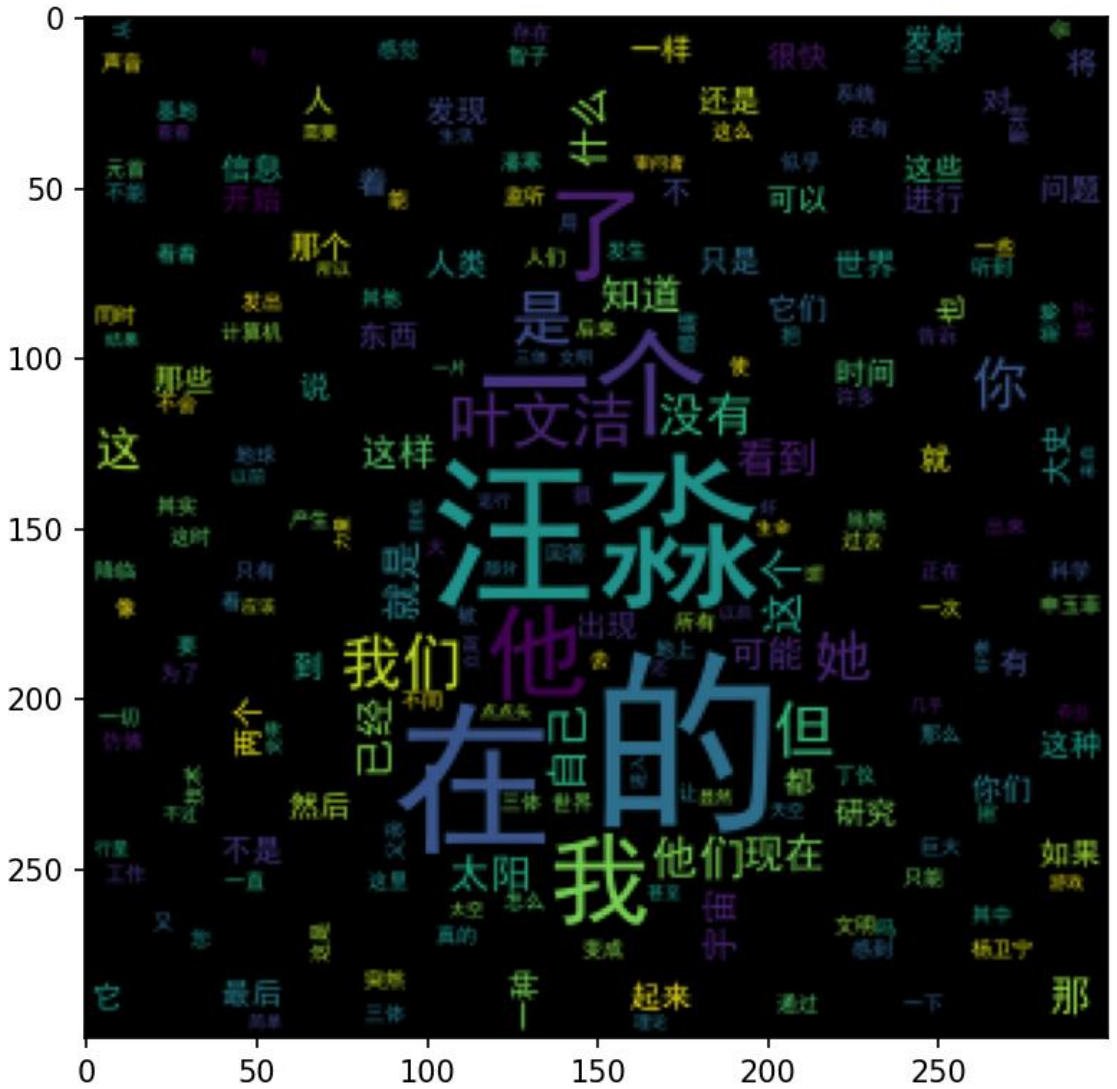
正向思维：

- 遍历分词后words列表
- 删除字符长度为1的字符

- 在列表中删除某个元素的方法
- words.remove(w)







- **为什么还有单个词汇？**

```
1 list1=["三体","的","我","世界"]
2 for w in list1:
3     if len(w)==1:
4         list1.remove(w)
5 print(list1)
```

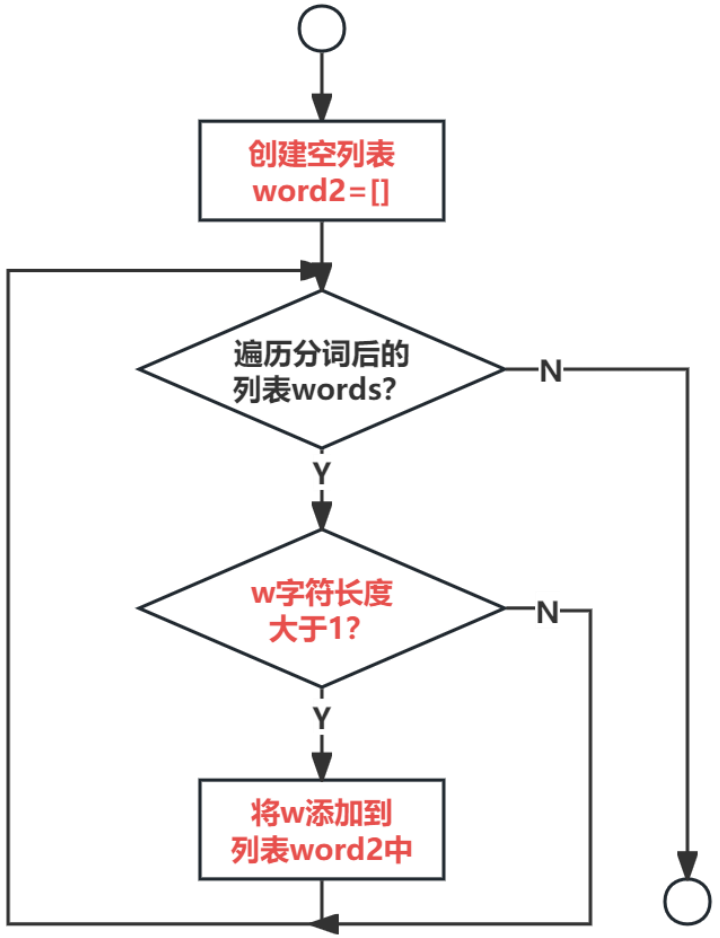
```
['三体', '我', '世界']
```

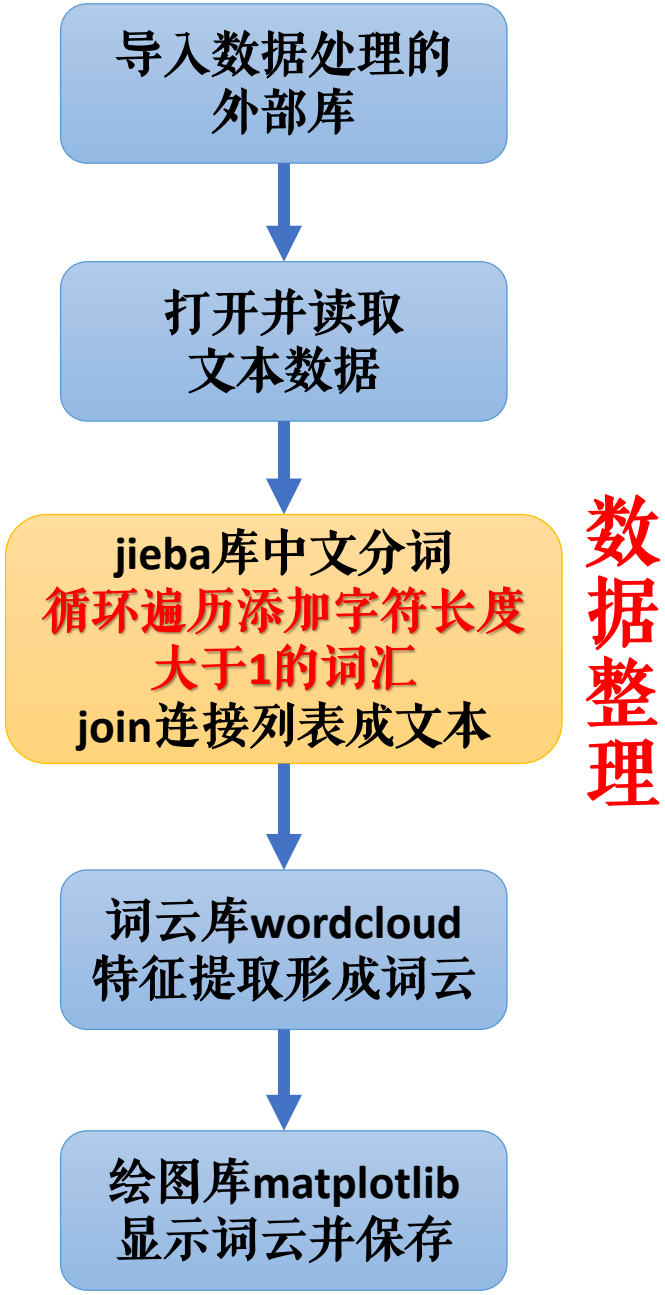
数据整理活动3：
循环结构优化处理单文本

逆向思维：

将字符长度大于等于2的字符，
添加到新的列表word2中，
不修改原有的列表words

- 遍历分词后words列表
- 如果字符w长度大于1，则添加到新的列表word2中。
- word2.**append**(w)





```python
import matplotlib.pyplot  as plt
import wordcloud as wc
import numpy as np
from PIL import Image
import jieba

f=open("三体1.txt",encoding="ANSI")
fs=f.read()
f.close()

words=jieba.lcut(fs)
word2=[]
for w in words:
    if len(w)>1:
        word2.append(w)
ds=" ".join(word2)

bg = np.array(Image.open("1.jpg"))
a=wc.WordCloud(font_path="simhei.ttf",mask=bg)
a.generate(ds)

plt.imshow(a)
plt.savefig(fname="三体词云图.jpg")
plt.show()
```

数据整理：
将源数据整理为有格式的可分析的数据
（循环遍历添加字符长度大于1的词汇）

学习活动3：三体文本初步分析与呈现

三体 I

三体II

三体III

- 三体三部曲，哪些词汇是高频词？每本书的主要人物是谁？
- 去除我们、这个、没有、一个等现代语句常用词汇，这三本书有哪些共同词汇话题？比如地球。
- 这三本书又有哪些不同的话题，比如第三部中的飞船，而第一部、第二部飞船的讨论较少。

三体 I

三体 II

三体 III

- 词云图是一种将文本数据进行可视化表达的方式，即对在大量文本中直观地突出出现频率较高的关键词予以视觉上的突出，从而**过滤掉大量的文本信息**。
- 词云图只是为大家提供了轮廓，这三本书到底讲了什么，**唯有课下深入阅读方能知晓，这是文本阅读的意义所在。**欢迎阅读后深入讨论。

课堂小结：Python处理数据及词云可视化的过程

导入外部处理库

```
import matplotlib.pyplot  as plt
import wordcloud as wc
import numpy as np
from PIL import Image
import jieba
```

文本导入与读取

```
f=open("三体1.txt",encoding="ANSI")
fs=f.read()
f.close()
```

分词与数据整理

```
words=jieba.lcut(fs)
word2=[]
for w in words:
    if len(w)>1:
        word2.append(w)
ds=" ".join(word2)
```

特征提取形成词云

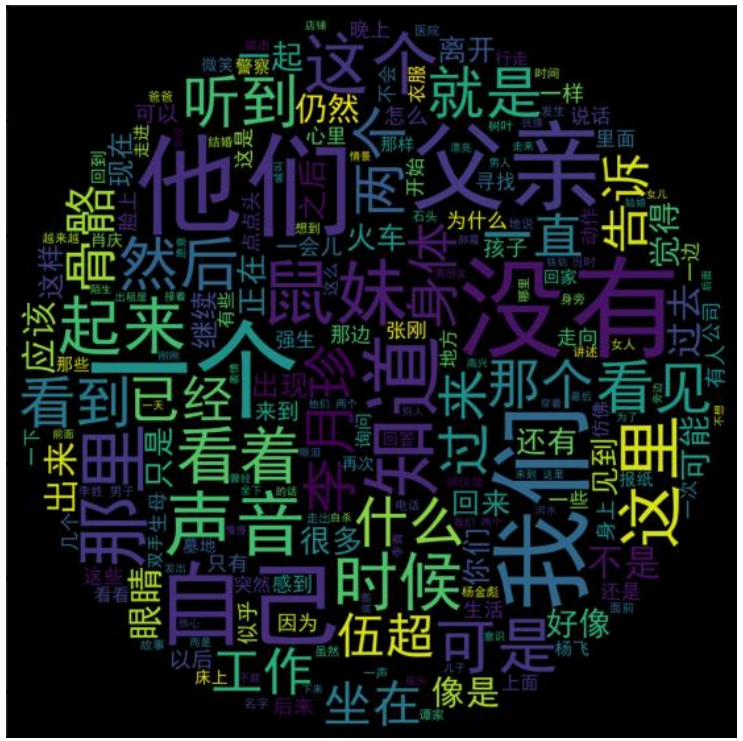
```
bg = np.array(Image.open("1.jpg"))
a=wc.WordCloud(font_path="simhei.ttf",mask=bg)
a.generate(ds)
```

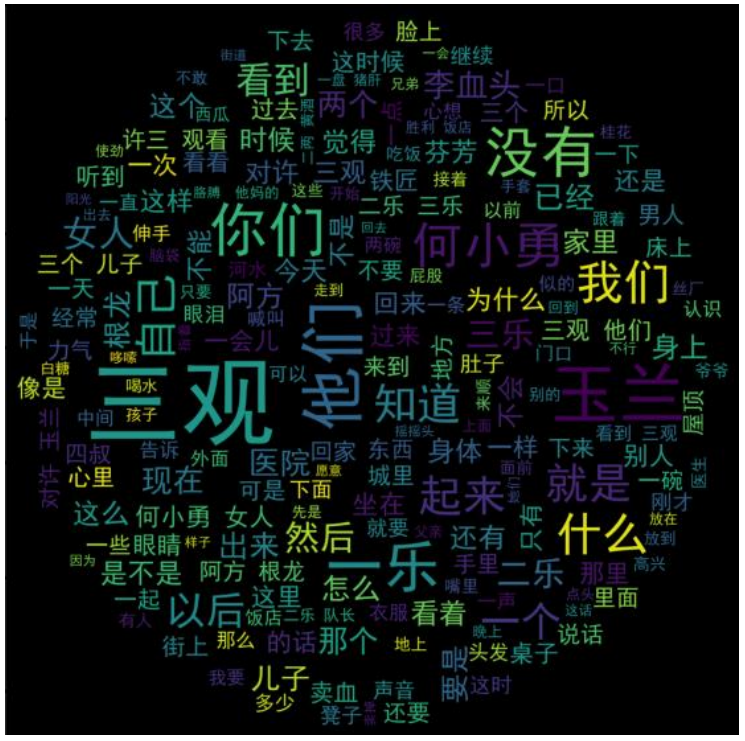
显示词云并保存

```
plt.imshow(a)
plt.savefig(fname="三体词云图.jpg")
plt.show()
```

课后作业：

分析并分享你喜欢的作者或阅读过的小说文本



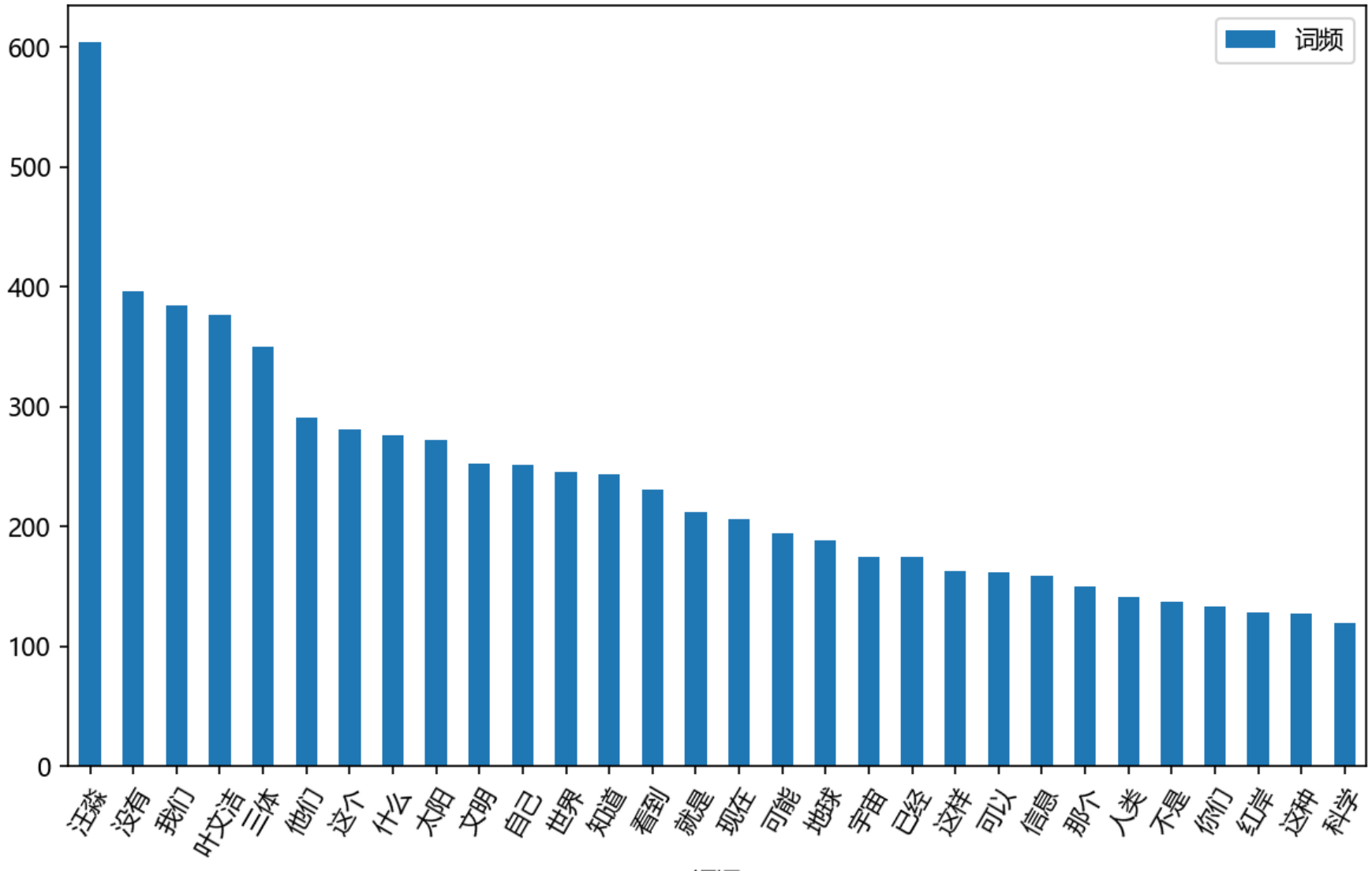


课后思考：如何去除常用双词进一步反应主题

- 设计制作思路，如何去除一个、她们、我们、你们等词汇，建立一个列表word3，将这些词汇加入word3，将不属于word3的添加到要分析的word2中。


```
words=jieba.lcut(fs)
word3=["一个","我们","他们","知道","没有","自己",
word2=[]
for w in words:
    if len(w)>1 and (w not in word3):
        word2.append(w)
ds=" ".join(word2)
```

课后思考：如何精确统计词频并排序

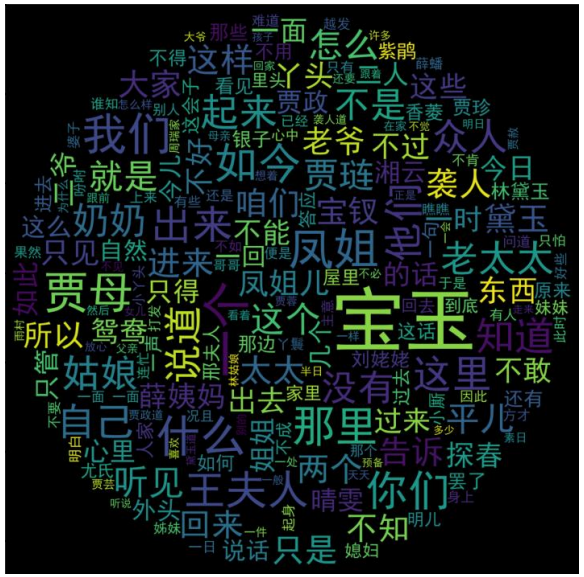


词语	词频
汪淼	604
没有	396
我们	384
叶文洁	376
三体	350
他们	291
这个	281
什么	276
太阳	272
文明	252
自己	251
世界	245
知道	243
看到	231
就是	212
现在	206
可能	194
地球	188

三体 I

活着

乡土中国



红楼梦

课堂结语

每个人都有自己词汇边界，但我们可以通过阅读拓展自己的词汇边界。

生活亦如此，需要不断尝试与行走，拓宽自己的边界。 No boring，不怕出错。