**用WordCloud词云 + LDA主题模型，带你读一读《芳华》(python实现)**

**修改部分代码 python2实现**

https://www.jianshu.com/p/e14111d9de51

何小嫚&刘峰原图.jpg



人物词云效果.jpg

电影《芳华》在春节重映了一波，加上之前的热映，最终取得了14亿票房的好成绩。严歌苓的原著也因此被更多的人细细品读。用文本分析的一些技术肢解小说向来是自然语言处理领域的一大噱头，这次当然也不能放过，本篇达成的成就有：  
1、提取两大主角刘峰和何小嫚（萍）的关键词并绘制好看的人物词云；  
2、以章节为单位探索小说的主题分布并画图展示。

**主要功能包：**

jieba

lda

wordcloud

seaborn

安装命令： pip install \*\*\*

**需要的外部文件：**

1、小说全文， 芳华-严歌苓.txt

2、中文停用词，stopwords.txt

3、小说人物名称，person.txt，作为jieba的用户自定义词典

4、两个人物的png图片

5、你喜欢的中文字体的ttf文件，我用的楷体



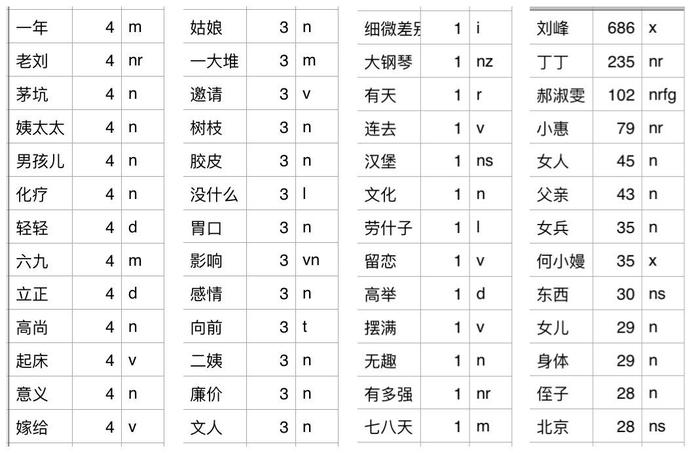
人物名和停用词文件示例.jpg

**一、文本预处理**

**1、分词，并过滤无意义词**

文本挖掘的必备步骤，毕竟理解中文的最小单位是词汇。这里没有使用简单的jieba.cut进行分词，因为我们需要知道单词的词性，便于稍后根据词性过滤不重要的词。

采用jieba.posseg.cut分词可以输出词性。我们并不能拍脑门决定是要动词还是名词等等，词性有非常多个，我把全部分词结果按照词性分好类，看了一下每个词性对应哪些词，最后决定保留词性为["a", "v", "x", "n", "an", "vn", "nz", "nt", "nr"]的词，例如图中，m代表量词，这是对语义没有帮助的词，应该舍弃。



词性示例.jpg

import jieba.posseg

jieba.load\_userdict("data/person.txt")

STOP\_WORDS = set([w.strip() for w in open("data/stopwords.txt").readlines()])

def cut\_words\_with\_pos(text):

seg = jieba.posseg.cut(text)

res = []

for i in seg:

if i.flag in ["a", "v", "x", "n", "an", "vn", "nz", "nt", "nr"] and is\_fine\_word(i.word):

res.append(i.word)

return list(res)

# 过滤词长，过滤停用词，只保留中文

def is\_fine\_word(word, min\_length=2):

rule = re.compile(r"^[\u4e00-\u9fa5]+$")

if len(word) >= min\_length and word not in STOP\_WORDS and re.search(rule, word):

return True

else:

return False

**2、划分章节**

我们按照“第\*章”这样的字眼将小说的不同章节分割开来，作为独立的文档，用于之后的主题分析。定义了一个名为MyChapters的生成器，存储每章分好的词汇，是为了避免章节过多带来的一些程序运行问题。其实《芳华》仅有15章，用一个简单的列表也是可以的。

class MyChapters(object):

def \_\_init\_\_(self, chapter\_list):

self.chapter\_list = chapter\_list

def \_\_iter\_\_(self):

for chapter in self.chapter\_list:

yield cut\_words\_with\_pos(chapter)

def split\_by\_chapter(filepath):

text = open(filepath).read()

chapter\_list = re.split(r'第.{1,3}章\n', text)[1:]

return chapter\_list

**二、人物关键词提取**

要提取人物关键词，首先要解决的问题是，在不借助外部的人物描述（比如百度百科和豆瓣电影上的角色介绍）的情况下，如何确定跟这个人物相关的内容。这里采用的比较简单的策略是，对小说文件中的每一行，如果该人物的名称存在，则将该行加入到此人的相关语料中去。再以此为基础统计词频，结果大致ok，为了人物词云更精确的展示，我将词频输出到了文件，手动删除了一些词，并简单调整了一些词的词频，下图是调整过后的词和词频，左为何小嫚，右为刘峰。

import pandas as pd

def person\_word(name):

lines = open("data/芳华-严歌苓.txt", "r").readlines()

word\_list = []

for line in lines:

if name in line:

words = cut\_words\_with\_pos(line)

word\_list += words

# 统计词频并按照词频由大到小排序，取top500

cnt = pd.Series(word\_list).value\_counts().head(500)

# 可以把结果输出到文件，进行一些手动调整

# cnt.to\_csv("data/cntliu.csv")

# 返回字典格式

return cnt.to\_dict()



人物关键词提取结果示例.jpg

**三、词云绘制**

python有wordcloud包可以用于词云绘制，在使用过程中需要注意：

1、用于定义形状的外部图片必须是png格式，默认纯白色部分为非图像区域；  
2、中文词云必须载入一个字体文件；  
3、字的颜色可以自己定义，也可以使用图片本身的底色。本例中何小嫚的图片 底色很鲜艳明晰，可以用本身的底色（ImageColorGenerator）；而刘峰的图片是单色，且色浅，我使用了自定义颜色（my\_color\_func）；  
4、绘制词云需要用到的数据格式为dict，key为词，value为词频，词频越大，在图片中的字体越大。

import matplotlib.pyplot as plt

from wordcloud import WordCloud, ImageColorGenerator

from scipy.misc import imread

from random import choice

# 定义颜色，方法很多，这里用到的方法是在四个颜色中随机抽取

def my\_color\_func(word, font\_size, position, orientation, random\_state=None, \*\*kwargs):

return choice(["rgb(94,38,18)", "rgb(41,36,33)", "rgb(128,128,105)", "rgb(112,128,105)"])

def draw\_cloud(mask\_path, word\_freq, save\_path):

mask = imread(mask\_path) #读取图片

wc = WordCloud(font\_path='data/kaiti.TTF', # 设置字体

background\_color="white", # 背景颜色

max\_words=500, # 词云显示的最大词数

mask=mask, # 设置背景图片

max\_font\_size=80, # 字体最大值

random\_state=42,

)

# generate\_from\_frequencies方法，从词频产生词云输入

wc.generate\_from\_frequencies(word\_freq)

plt.figure()

# 刘峰， 采用自定义颜色

plt.imshow(wc.recolor(color\_func=my\_color\_func), interpolation='bilinear')

# 何小嫚， 采用图片底色

# image\_colors = ImageColorGenerator(mask)

# plt.imshow(wc.recolor(color\_func=image\_colors), interpolation='bilinear')

plt.axis("off")

wc.to\_file(save\_path)

plt.show()

# 获取关键词及词频

input\_freq = person\_word("刘峰")

# 经过手动调整过的词频文件,供参考

# freq = pd.read\_csv("data/cntliu.csv", header=None, index\_col=0)

# input\_freq = freq[1].to\_dict()

draw\_cloud("data/liu.png", input\_freq, "output/liufeng.png")

对人物进行抠图，背景设置为纯白，存储为png格式。  
为了使形状更鲜明，对小嫚的辫子还有腰的部分做了加白处理，可以对比文章开头原图感受一下。



何小嫚&刘峰用作生成词云的图片.jpg



人物词云效果.jpg

如果你看过这部作品，不知道印象最深的是不是像词云显示的那样？小嫚在精神病院的月下独舞，刘峰对丁丁的深深眷恋，在战争中失去手臂，与不爱的人结婚又离婚，和小嫚以朋友的姿态相伴终老... ...