1. 读入文件
2. 用默认方式分词cut(str, cut\_all=False, HMM=True)顺便大幅度消除中文歧义
3. 使用停用词表
4. 添加自定义词库

**先看以下前两个**

**代码组成简介**

1. 代码部分来源于其他人的博客,但是因为bug或者运行效率的原因,我对代码进行了较大的改变
2. 代码第一部分,设置代码运行需要的大部分参数,你可以方便的直接使用该代码而不需要进行过多的修改
3. 第二部分为jieba的一些设置,当然你也可以利用isCN参数取消中文分词
4. 第三部分,wordcloud的设置,包括图片展示与保存

如果你想用该代码生成英文词云,那么你需要将isCN参数设置为0,并且提供英文的停用词表,但是我更推荐你使用[Python词云 worldcloud 十五分钟入门与进阶](http://blog.csdn.net/fontthrone/article/details/72775865" \t "_blank)这篇文章中的代码,因为它更简洁,更有利于使用’

**Use the code by comment**

关于该程序的使用,你可以直接读注释在数分钟内学会如何使用它

# - \* - coding: utf - 8 -\*-

#

# 作者：田丰(FontTian)

# 创建时间:'2017/5/23'

# 邮箱：fonttian@163.com

# CSDN：http://blog.csdn.net/fontthrone

from os import path

from scipy.misc import imread

import matplotlib.pyplot as plt

import jieba

# jieba.load\_userdict("txt\userdict.txt")

# 添加用户词库为主词典,原词典变为非主词典

from wordcloud import WordCloud, ImageColorGenerator

# 获取当前文件路径

# \_\_file\_\_ 为当前文件, 在ide中运行此行会报错,可改为

# d = path.dirname('.')

d = path.dirname(\_\_file\_\_)

stopwords = {}

isCN = 1 #默认启用中文分词

back\_coloring\_path = "img/lz1.jpg" # 设置背景图片路径

text\_path = 'txt/lz.txt' #设置要分析的文本路径

font\_path = 'D:\Fonts\simkai.ttf' # 为matplotlib设置中文字体路径没

stopwords\_path = 'stopwords\stopwords1893.txt' # 停用词词表

imgname1 = "WordCloudDefautColors.png" # 保存的图片名字1(只按照背景图片形状)

imgname2 = "WordCloudColorsByImg.png"# 保存的图片名字2(颜色按照背景图片颜色布局生成)

my\_words\_list = ['路明非'] # 在结巴的词库中添加新词

back\_coloring = imread(path.join(d, back\_coloring\_path))# 设置背景图片

# 设置词云属性

wc = WordCloud(font\_path=font\_path, # 设置字体

background\_color="white", # 背景颜色

max\_words=2000, # 词云显示的最大词数

mask=back\_coloring, # 设置背景图片

max\_font\_size=100, # 字体最大值

random\_state=42,

width=1000, height=860, margin=2,# 设置图片默认的大小,但是如果使用背景图片的话,那么保存的图片大小将会按照其大小保存,margin为词语边缘距离

)

# 添加自己的词库分词

def add\_word(list):

for items in list:

jieba.add\_word(items)

add\_word(my\_words\_list)

text = open(path.join(d, text\_path)).read()

def jiebaclearText(text):

mywordlist = []

seg\_list = jieba.cut(text, cut\_all=False)

liststr="/ ".join(seg\_list)

f\_stop = open(stopwords\_path)

try:

f\_stop\_text = f\_stop.read( )

f\_stop\_text=unicode(f\_stop\_text,'utf-8')

finally:

f\_stop.close( )

f\_stop\_seg\_list=f\_stop\_text.split('\n')

for myword in liststr.split('/'):

if not(myword.strip() in f\_stop\_seg\_list) and len(myword.strip())>1:

mywordlist.append(myword)

return ''.join(mywordlist)

if isCN:

text = jiebaclearText(text)

# 生成词云, 可以用generate输入全部文本(wordcloud对中文分词支持不好,建议启用中文分词),也可以我们计算好词频后使用generate\_from\_frequencies函数

wc.generate(text)

# wc.generate\_from\_frequencies(txt\_freq)

# txt\_freq例子为[('词a', 100),('词b', 90),('词c', 80)]

# 从背景图片生成颜色值

image\_colors = ImageColorGenerator(back\_coloring)

plt.figure()

# 以下代码显示图片

plt.imshow(wc)

plt.axis("off")

plt.show()

# 绘制词云

# 保存图片

wc.to\_file(path.join(d, imgname1))

image\_colors = ImageColorGenerator(back\_coloring)

plt.imshow(wc.recolor(color\_func=image\_colors))

plt.axis("off")

# 绘制背景图片为颜色的图片

plt.figure()

plt.imshow(back\_coloring, cmap=plt.cm.gray)

plt.axis("off")

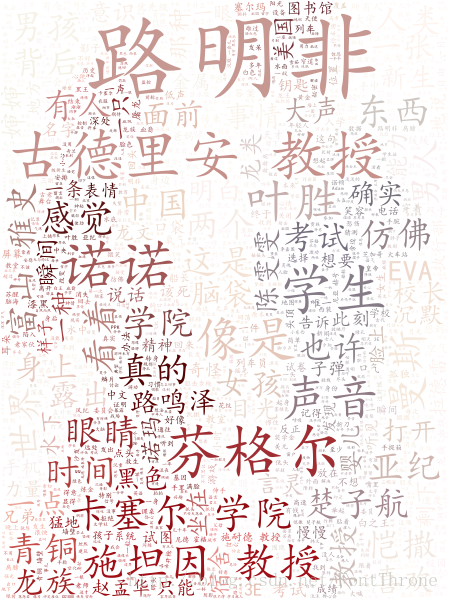
plt.show()

# 保存图片

wc.to\_file(path.join(d, imgname2))







## 关于显示不了中文的问题解决方案

ps.中文文本时： C:\Python3\Lib\site-packages\wordcloud\   注：我的是anaconda,所以路径为：F:\SoftwareInstall\anaconda3\Lib\site-packages\wordcloud

在 wordcloud.py 中修改 FONT\_PATH 为相应字体。（注意是大写，大写的字体路径是默认路径，小写的字体是后面使用时可以再次指定的，会覆盖默认字体）

FONT\_PATH = os.environ.get("FONT\_PATH", os.path.join(os.path.dirname(\_\_file\_\_), "simhei.ttf"))

## 入门

from wordcloud import WordCloud

f = open('E:/a.txt','r').read()

wordcloud = WordCloud(background\_color="white",width=1000, height=860, margin=2).generate(f)

# width,height,margin可以设置图片属性

# width,height,margin可以设置图片属性

# 你可以通过font\_path参数来设置字体集

import matplotlib.pyplot as plt

plt.imshow(wordcloud)

plt.axis("off")

plt.show()

wordcloud.to\_file('test.png')

seg\_list = jieba.cut(str, cut\_all=True, HMM=False) #分词

cut\_all=True ## 全模式

cut\_all=False# 默认模式 在默认模式下有对中文歧义有较好的分类方式

seg\_list = jieba.cut\_for\_search(str, HMM=False) # 搜索引擎模式

# jieba.cut的默认参数只有三个,jieba源码如下

# cut(self, sentence, cut\_all=False, HMM=True)

# 分别为:输入文本 是否为全模式分词 与是否开启HMM进行中文分词

[Python词云 wordcloud 十五分钟入门与进阶](http://blog.csdn.net/fontthrone/article/details/72775865)   
[Python中文分词 jieba 十五分钟入门与进阶](http://blog.csdn.net/fontthrone/article/details/72782499)

**关键词提取**

from os import path

import jieba.analyse as analyse

d = path.dirname(\_\_file\_\_)

text\_path = 'txt/lz.txt' #设置要分析的文本路径

text = open(path.join(d, text\_path)).read()

for key in analyse.extract\_tags(text,50, withWeight=False):

# 使用jieba.analyse.extract\_tags()参数提取关键字,默认参数为50

print (key.encode('utf-8'))

# 设置输出编码为utf-8不然在因为win下控制台默认中文字符集为gbk,所以会出现乱码

# 当withWeight=True时,将会返回number类型的一个权重值(TF-IDF)

## 中文歧义测试与去除停用词

去除文本中的停用词

# - \* - coding: utf - 8 -\*-

#

# 作者：田丰(FontTian)

# 创建时间:'2017/5/27'

# 邮箱：fonttian@163.com

# CSDN：http://blog.csdn.net/fontthrone

import sys

import jieba

from os import path

d = path.dirname(\_\_file\_\_)

stopwords\_path = 'stopwords\stopwords1893.txt' # 停用词词表

text\_path = 'txt/lz.txt' #设置要分析的文本路径

text = open(path.join(d, text\_path)).read()

def jiebaclearText(text):

mywordlist = []

seg\_list = jieba.cut(text, cut\_all=False)

liststr="/ ".join(seg\_list)

f\_stop = open(stopwords\_path)

try:

f\_stop\_text = f\_stop.read( )

f\_stop\_text=unicode(f\_stop\_text,'utf-8')

finally:

f\_stop.close( )

f\_stop\_seg\_list=f\_stop\_text.split('\n')

for myword in liststr.split('/'):

if not(myword.strip() in f\_stop\_seg\_list) and len(myword.strip())>1:

mywordlist.append(myword)

return ''.join(mywordlist)

text1 = jiebaclearText(text)

print text1

添加自定义词库:   
下面的代码主要来自于jieba的github源码,你可以在github下载该例子

#encoding=utf-8

from \_\_future\_\_ import print\_function, unicode\_literals

import sys

sys.path.append("../")

import jieba

jieba.load\_userdict("userdict.txt")

# jieba采用延迟加载，"import jieba"不会立即触发词典的加载，一旦有必要才开始加载词典构建trie。如果你想手工初始jieba，也可以手动初始化。示例如下:

# import jieba

# jieba.initialize() #手动初始化（可选）

# 在0.28之前的版本是不能指定主词典的路径的，有了延迟加载机制后，你可以改变主词典的路径:

# 注意用户词典为主词典即优先考虑的词典,原词典此时变为非主词典

# jieba.set\_dictionary('data/dict.txt.big')

import jieba.posseg as pseg

test\_sent = (

"李小福是创新办主任也是云计算方面的专家; 什么是八一双鹿\n"

"例如我输入一个带“韩玉赏鉴”的标题，在自定义词库中也增加了此词为N类\n"

"「台中」正確應該不會被切開。mac上可分出「石墨烯」；此時又可以分出來凱特琳了。"

)

words = jieba.cut(test\_sent)

print('/'.join(words))

print("="\*40)

result = pseg.cut(test\_sent)

# pseg.cut 切分,并显示词性

# 下面是userdict.txt的内容,如果不加入这个词库,那么在运行结果中,云计算,创新办等词都将无法识别

'''

云计算 5

李小福 2 nr

创新办 3 i

easy\_install 3 eng

好用 300

韩玉赏鉴 3 nz

八一双鹿 3 nz

台中

凱特琳 nz

Edu Trust认证 2000

'''

## 利用背景图片生成词云,设置停用词词集

该段代码主要来自于wordcloud的github,你同样可以在github下载该例子以及原图片与效果图

#!/usr/bin/env python

"""

Image-colored wordcloud

=======================

You can color a word-cloud by using an image-based coloring strategy

implemented in ImageColorGenerator. It uses the average color of the region

occupied by the word in a source image. You can combine this with masking -

pure-white will be interpreted as 'don't occupy' by the WordCloud object when

passed as mask.

If you want white as a legal color, you can just pass a different image to

"mask", but make sure the image shapes line up.

"""

from os import path

from PIL import Image

import numpy as np

import matplotlib.pyplot as plt

from wordcloud import WordCloud, STOPWORDS, ImageColorGenerator

d = path.dirname(\_\_file\_\_)

# Read the whole text.

text = open(path.join(d, 'alice.txt')).read()

# read the mask / color image taken from

# http://jirkavinse.deviantart.com/art/quot-Real-Life-quot-Alice-282261010

alice\_coloring = np.array(Image.open(path.join(d, "alice\_color.png")))

# 设置停用词

stopwords = set(STOPWORDS)

stopwords.add("said")

# 你可以通过 mask 参数 来设置词云形状

wc = WordCloud(background\_color="white", max\_words=2000, mask=alice\_coloring,

stopwords=stopwords, max\_font\_size=40, random\_state=42)

# generate word cloud

wc.generate(text)

# create coloring from image

image\_colors = ImageColorGenerator(alice\_coloring)

# show

# 在只设置mask的情况下,你将会得到一个拥有图片形状的词云

plt.imshow(wc, interpolation="bilinear")

plt.axis("off")

plt.figure()

# recolor wordcloud and show

# we could also give color\_func=image\_colors directly in the constructor

# 我们还可以直接在构造函数中直接给颜色

# 通过这种方式词云将会按照给定的图片颜色布局生成字体颜色策略

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

plt.axis("off")

plt.figure()

plt.imshow(alice\_coloring, cmap=plt.cm.gray, interpolation="bilinear")

plt.axis("off")

plt.show()