回望与前行

1.探索新知

1.1 处理文本数据

wordcloud库是专门根据文本生成词云的Python第三方库。

而wordcloud默认会以空格或标点为分隔符对目标文本进行分词处理。

因此，对于

- 英文文本：无需分词，可直接调用wordcloud库函数

例如：Good job!

- 中文分词：分词处理需要由用户自主完成

例如：电风扇真是人类最好的朋友我只是问它我是不是长得丑它就很认真地对我摇了一晚上头。

中文文本处理一般步骤为：

分词处理 → 空格拼接 → 调用wordcloud库函数

在这里，我们将使用jieba库对文本进行分词处理，将文本文件中的长句子切割成单个的词。

```
import jieba
text = "有人说 大年三十儿不一定是个不眠夜 开学前一夜才是真正的万家灯火"
lst = jieba.lcut(text)
print(lst)
```

控制台 清空 缩小

```
Building prefix dict from the default dictionary ...
Loading model from cache C:\Users\WUGEDI~1\AppData\Local\Temp\jieba.cache
Loading model cost 0.564 seconds.
Prefix dict has been built successfully.
['有人', '说', ' ', '大年三十', '儿', '不', '一定', '是', '个', '不眠夜', ' ', '开学', '前', '一夜', '才', '是', '"', '真正',
 '的', '万家灯火', '"']
```

1.1 处理文本数据

简单的字符串分词没问题，但我想将存储在txt文件中的内容进行分词怎么办呢？

开学前一夜.txt

1. 读取文本数据

```
with open('开学前一夜.txt', 'r', encoding='utf-8') as f:
    text = f.read()
```

2. 分词处理

```
import jieba
with open('开学前一夜.txt', 'r', encoding='utf-8') as f:
    text = f.read()
lst = jieba.lcut(text)
m = ' '.join(lst)
```

任务一新增代码：

```
import jieba
with open('开学前一夜.txt', 'r', encoding='utf-8') as f:
    text = f.read()
lst = jieba.lcut(text)
m = ' '.join(lst)
```

1.2 生成普通词云

在这一步，我们需要使用wordcloud库来生成词云。我们会设定词云图大小、字体、停用词（我们不想在词云中显示的词，比如 '的' 这类无意义的词）以及背景图片等

1. 导入wordcloud库

```
import wordcloud
```

2. 输出“开始制作词云...”的提示

```
print('开始制作词云...')
```

创建一个wordcloud对象，常用的参数有：

- width，height：画布的宽度和高度，单位为像素。若没设置mask值，才会使用此默认值400*200。
- font_path：字体路径。默认指向一个英文字体。
- mask：用于设定绘制模板，需要是一个nd-array（多维数组）。
- background_color：词云图背景色，默认为黑色。可根据需要调整。
- mode：当设置为"RGBA" 且background_color设置为"None"时可产生透明背景。
- stopwords：被排除词列表，排除词不在词云中显示。

wordcloud对象，常用的方法有：

方法	功能
generate(text)	由text文本生成词云
to_file(filename)	将词云图保存为名为filename的文件

3. 先创建一个最最普通的词云对象

```
w = wordcloud.WordCloud( )
```

1.2 生成普通词云

4. 由文本m生成词云

```
w.generate(m)
```

词云图保存好了吗？
存到了哪里呢？

5. 保存词云图

```
w.to_file('词云.png')
```

词云图会默认存在.py文件所在的目录下。

我们需要给用户一个保存完毕的提醒。而且最好将生成的词云图直接展示给用户。

6. 保存完毕的提醒

```
print('词云图片已生成,开始展示图片')
```

7. 展示词云图

想要打开图像，可以使用pillow库中的Image模块

- **首先，导入Image模块**

```
from PIL import Image
```

- **接着，打开图像**

```
p = Image.open('词云.png')
p.show()
```

8. 展示完成的提示语

```
print('图片展示完成')
```

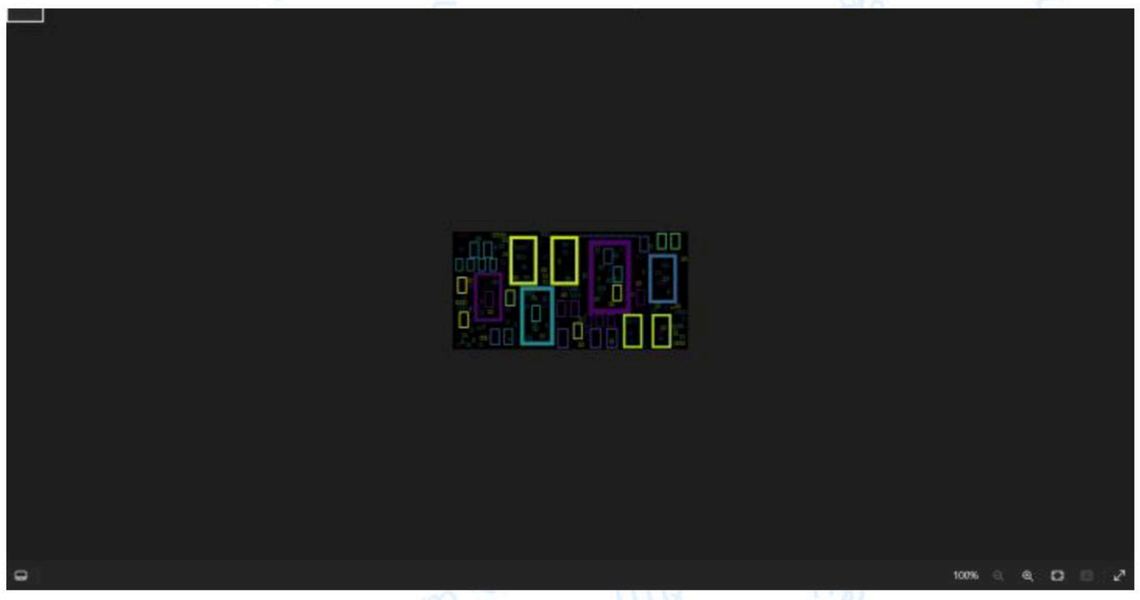
1.2 生成普通词云

任务二新增代码：

```
import wordcloud
from PIL import Image

print('开始制作词云...')
w = wordcloud.WordCloud()
w.generate(m)
w.to_file('词云.png')
print('词云图片已生成,开始展示图片')
p = Image.open('词云.png')
p.show()
print('图片展示完成')
```

1.3 词云效果优化



不填入任何参数的词云如图所示，可以发现，文字都没有出现，并且词云尺寸太小，接下来我们先优化这两个问题。

1.3 词云效果优化

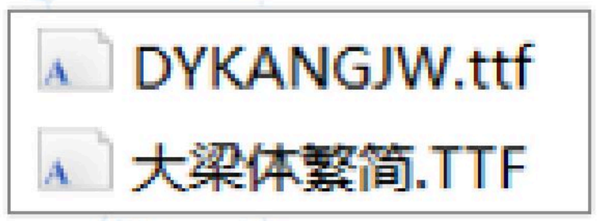
字体的选择

处理中文时还需要指定中文字体，若未指定，生成的图片可能是这样的

注意：

如果处理的是英文，则可以使用默认的英文字体正常输出。

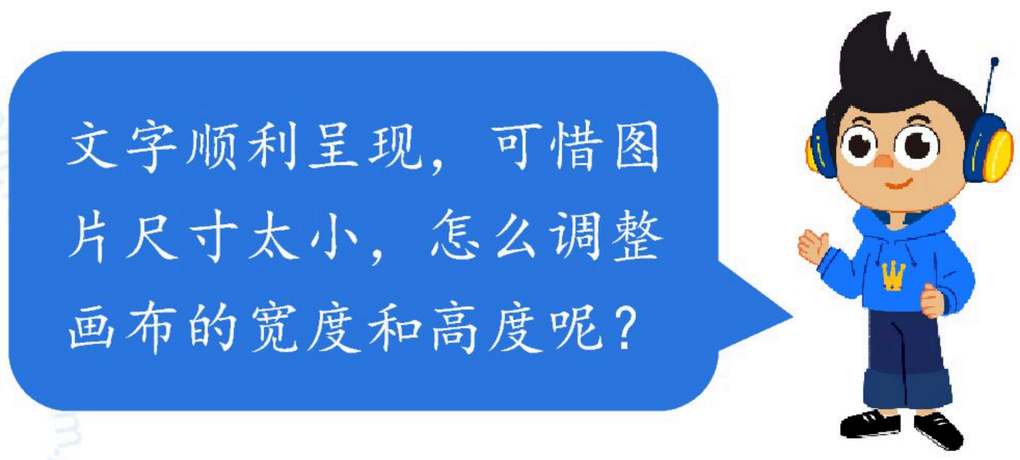
1. 设置w的 font_path 参数



字体准备好啦，
选择一个喜欢的吧！

```
w = wordcloud.WordCloud(font_path = '大梁体繁简.TTF')
```

效果如下



2. 设置w的 width 、height 参数

```
w = wordcloud.WordCloud(
    font_path = '大梁体繁简.TTF'
    width = 1200,
    height = 800
)
```

自行设置画布的宽度和高度吧！

1.3 词云效果优化

现在的效果已经很不错了，接下来我们再整点细节

- **去除无价值的高频词**

文本中出现最多的词，也就是词云图中最大的词语，是“你”、“我”、“的”、“是”，这些词并没有太大的价值。所以我们可以把这些词排除，让他们不在词云图中显示。怎么实现呢？

```
w = wordcloud.WordCloud(
    ... ,
    stopwords = ['你',"我","是",'的']
)
```

- **设置词云的形状**

wordcloud可以生成任何形状的词云，为了获取形状，需要提供一张形状的图片。

1.3 词云效果优化

- **设置词云的形状**

词云图是可以自定义背景图的，但不是随便拿一张图用都可以。
需要注意词频背景图中想要的形状的背景需要是白色的，否则可能得不到想要的词云图形状。

mask: nd-array or None (默认为None),
用于设定绘制模板，需要是一个 nd-array（多维数组），所以在用Image.open()读取图片后，需要用 np.array 转换成数组。
当mask不为0时，那么之前依据height和width设置的画布则作废，此时“画布”形状大小由mask决定。默认None，即方形图。

```
import numpy as np
# 加载背景图片
mask_image = np.array(Image.open("蝴蝶.png"))
w = wordcloud.WordCloud(font_path = '大梁体繁简.TTF',
    # width = 1200, height = 800,
    stopwords = ['你',"我","是",'的'],
    mask = mask_image
)
```

mask不为0，所以设定的宽高就不起作用了，可以删掉

1.3 词云效果优化

- **设置词云图背景色**

词云图背景色默认为黑色，如何将它设置为其他颜色或者透明色？

```
background_color = 'pink'
```

```
background_color = None,
mode = 'RGBA'
```

任务三新增代码：

```
import numpy as np
# 加载背景图片
mask_image = np.array(Image.open("蝴蝶.png"))
w = wordcloud.WordCloud(
    font_path = '大梁体繁简.TTF',
    width = 1200,
    height = 800
    stopwords = ['你',"我","是",'的'],
    background_color = None,
    mode = 'RGBA',
    mask = mask_image
)
```

完整代码

```
import jieba
import wordcloud
from PIL import Image
import numpy as np


# 加载背景图片
mask_image = np.array(Image.open("蝴蝶.png"))

with open('开学前一夜.txt', 'r', encoding='utf-8') as f:
    text = f.read()
lst = jieba.lcut(text)
m = ' '.join(lst)

print('开始制作词云...')

w = wordcloud.WordCloud(
    font_path = '大梁体繁简.TTF',
    stopwords = ['你',"我","是",'的'],
    # 使用参数background_color设置图片背景
    background_color = 'None',
    mode = 'RGBA',
    mask = mask_image
)
w.generate(m)
w.to_file('词云.png')

print('词云图片已生成,开始展示图片')
p = Image.open('词云.png')
p.show()
print('图片展示完成')
```

2.强化练习

1. jieba.lcut(text) 在代码中的作用是？（　　）

A. 对文本进行升序排列

B. 对文本进行中文分词

C. 对文本进行编码转换

D. 对文本进行压缩

2. 函数 w.to_file('词云.png') 的作用是？（　　）

A. 打开'词云.png'文件

B. 删除'词云.png'文件

C. 重命名'词云.png'文件

D. 保存词云图像为'词云.png'文件

3. ' '.join(lst) 的作用是？（　　）

A. 将lst列表中的元素用空格连接

B. 将lst列表转为元组

C. 将lst列表中所有元素替换为' '

D. 将lst列表中的元素以' '分隔开来

2.强化练习

4．词云图的背景色默认什么颜色？（　　）

A．白色

B．透明色

C．黑色

D．没有默认颜色

5．想要给词云图设置被排除词列表，该使用下面的哪个参数？（　　）

A. font_path

B. mask

C. background_color

D. stopwords

3.术语箱

text	文本	mode	模式
cut	切割	generate	生成
path	路径	image	图像
mask	面具	array	数组
background	背景		

4.课后挑战

你的词云图

要求：

1. 处理文本数据（春.txt）

2. 导入喜欢的背景模板

3. 生成词云图