统计词频制作词云图

一、应用场景

利用Python我们可以轻松完成词频统计，找到出现频率最高的一组形容词和名词。并按照频率高低将这些形容词和名词绘制成一张词云图，帮你轻松分析一篇文章。

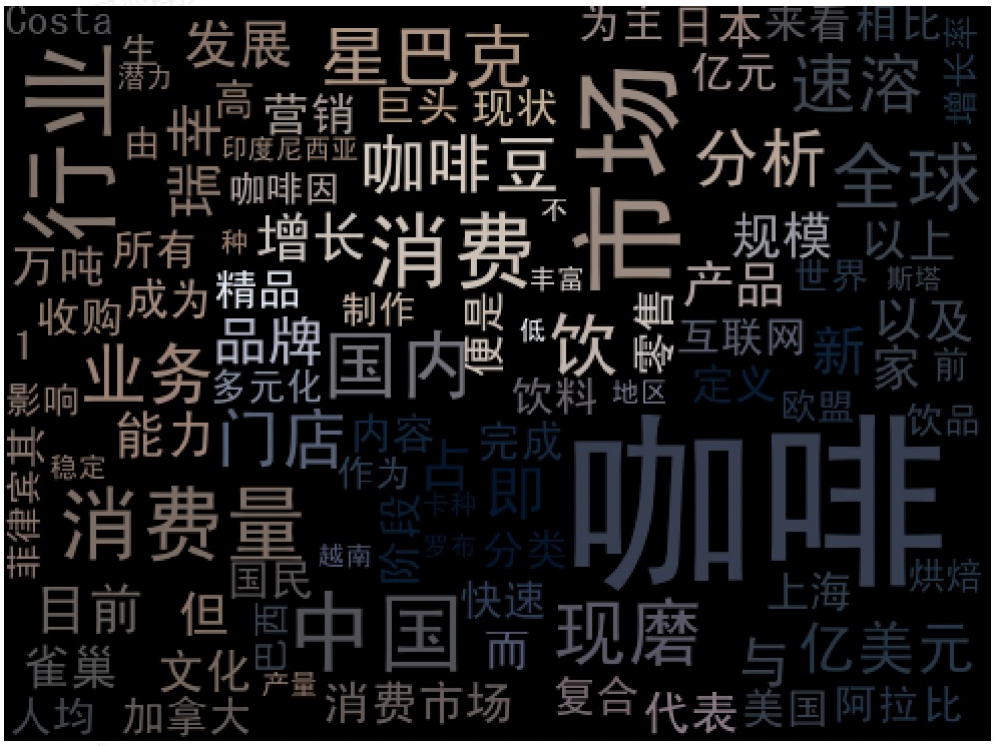
二、项目代码及结果展示

2.1 项目代码

```
import re,collections,wordcloud,jieba
import matplotlib.pyplot as plt
from scipy.misc import imread

file = open('咖啡行业定义与分类.txt')
data = file.read()
file.close()

pattern = re.compile(u'\t|\n|\.|-|:|;|\)|\(|\?|"|“|：|；|”')
clear_data = re.sub(pattern, '', data)

clear_data = jieba.cut(clear_data, cut_all = False)
object_list = []
remove_words = [u'的', u'，',u'和', u'是', u'随着', u'对于', u'对',u'等',u'能',u'都',
                u'。',u' ',u'、',u'中',u'在',u'了',
                u'通常',u'我们',u'需要',u'具有',u'不同',u'用',u'年',u'月',u'日',
                u'日常',u'主要',u'同时',u'指出',u'我国',u'及',u'2019',u'达到',u'为',
                u'占据',u'分别',u'2018',u'从',u'概括',u'较',u'将',u'为',u'受',
                u'被',u'以',u'比',u'上',u'2',u'也',u'到',u'已']

for word in clear_data:
    if word not in remove_words:
        object_list.append(word)

word_counts = collections.Counter(object_list)
top_word_counts = word_counts.most_common(10)
```

```
27
28 mk=imread("咖啡.jpg")
29 word_cloud = wordcloud.WordCloud(
30     font_path='simhei.ttf',
31     mask=mk,
32     max_words=100,
33     max_font_size=100
34 )
35
36 word_cloud.generate_from_frequencies(word_counts)
37 image_colors = wordcloud.ImageColorGenerator(mk)
38 word_cloud.recolor(color_func=image_colors)
39
40 plt.imshow(word_cloud) # 显示词云
41 plt.axis('off') # 关闭坐标轴
42 plt.show() # 显示图像
```

2.2 结果展示



三、项目操作及解析

3.1 素材准备

在这个项目中我们需要新建一个文件夹。文件夹中需要放我们分析的txt文件、生成词云的背景图以及字体文件。这里我们选用黑体字体，直接在网页上搜索simhei.ttf下载即可。同时在这个文件夹中新建一个python文件。

上述操作完成之后，在vs code中打开这个文件夹，在新建的python文件中编写代码。

3.2 安装、导入第三方库

我们需要使用wordcloud、matplotlib、jieba和scipy四个第三方库，先用pip install xxx（xxx为库的名称）去安装这些库，安装成功后导入这些库。

```
import re,collections,wordcloud,jieba
import matplotlib.pyplot as plt
from scipy.misc import imread
```

3.3 过程讲解

- **读取文本文件**

接下来我们要读取需要的文本内容并赋值给一个变量方便后续使用。我们选择分析的文本是中国报告大厅的《咖啡行业定义与分类》。

```
file = open('咖啡行业定义与分类.txt')    # 此处替换成你自己txt文档
data = file.read()
file.close()
```

- **去除文本中的标点**

我们可以编写一个正则表达式的匹配模式，匹配到常用的标点符号。然后用re.sub()函数从文本中筛选出文本中的标点符号并删除掉，下面的代码可以不用更改直接使用。

```
pattern = re.compile(u'\t|\n|\.|-|:|;|\)|\(|\?|"|“|: |; |”') #这里可以在|之后添加你需要的标点
clear_data = re.sub(pattern, '', data)
```

- **分词后去除文本中的干扰词**

我们可以用jieba.cut()函数来把文本切成词语组成的列表，这个函数第一个参数是我们需要的文本，第二个参数cut_all=False表示使用精准模式切分（一般场景选用精准模式即可）。

我们需要分析的对象为文本中的形容词和名词，而文本中的连词和语气词并无实际意义，会干扰我们的分析，因此我们要去除掉文本中的连词和语气词。

```
clear_data = jieba.cut(clear_data, cut_all = False)
object_list = []
remove_words = [u'的', u', ',u'和', u'是', u'随着', u'对于', u'对',u'等',u'能',u'都',
                u'。',u' ',u'、',u'中',u'在',u'了',
                u'通常',u'我们',u'需要',u'具有',u'不同',u'用',u'年',u'月',u'日',
                u'日常',u'主要',u'同时',u'指出',u'我国',u'及',u'2019',u'达到',u'为',
                u'占据',u'分别',u'2018',u'从',u'概括',u'较',u'将',u'为',u'受',
                u'被',u'以',u'比',u'上',u'2',u'也',u'到',u'已']
# remove_words里紧跟着u的引号里面可以写你要去除的词组或字符
for word in clear_data:
    if word not in remove_words:
        object_list.append(word)
```

- **统计词频**

制作一个词云图核心的步骤就是统计词频，我们可以用collections.Counter()函数来得到一个字典，字典的键是词语，字典的值是词语出现的次数。

同时我们可以选用出现次数最高的10个词来打印看看，使用word_counts.most_common()函数即可实现，函数的参数是选用词的数量。

```
word_counts = collections.Counter(object_list)
top_word_counts = word_counts.most_common(10)
print(top_word_counts)
```

- **创建词云对象**

我们可以使用wordcloud.WordCloud()来创建一个词云对象，其中font_path是字体的路径，你需要自己换成自己的字体文件路径。mask是词云的背景图，我们可以用imread()函数读取一张图片并传递给mask参数，其中的参数是图片的路径。max_words参数是最词云图所能显示的词数。max_font_size是词云图的显示大小。

```
# 下面的内容你可以自己修改
mk=imread("咖啡.jpg")
word_cloud = wordcloud.WordCloud(
    font_path='simhei.ttf',
    mask=mk,
```

```
    max_words=100,
    max_font_size=100
)
```

- **生成词云并显示**

我们用词云对象.generate_from_frequencies()方法就可以导入上面的词频字典，参函数所需的参数是一个字典。词云字典.ImageColorGenerator()方法则可以建立词云的颜色方案，参数为读取的一张图片。词云对象.recolor()方法用来重新设定词云的颜色，这个方法的参数就是我们上面建立的颜色方案。

最后我们用matplotlib库中pyplot类中的一系列函数展示词云。

```
word_cloud.generate_from_frequencies(word_counts)
image_colors = wordcloud.ImageColorGenerator(mk)
word_cloud.recolor(color_func=image_colors)

plt.imshow(word_cloud)
plt.axis('off')
plt.show()
```