7.2 词云绘制——wordcloud 库

2023 年 9 月 18 日

7.2 词云绘制——wordcloud 库

企业可以使用数据可视化（如图表、图形和信息图）来传达一目了然的基本信息，但如果你的原始数据是基于文本的呢?

字云是一种引人注目的可视化方法，可以突出重要的文本数据点。它可以使沉闷的数据闪闪发光，快速传递关键信息。作为一种视觉表现形式。它是一组以不同大小显示的词：该词越大、越粗，它在文件中出现的频率越高，越重要。文本云包括数据可视化、文本数据、字体颜色、词频分析和特定的单词图形。

通过观察图片，我们可以清楚地看到，这个词云是关于冠状病毒大流行的，因为“冠状病毒”、“covid19”和“病毒”是最常见的词汇。我们还可以看到，这是一个全球性的问题，它强调了个人防护设备和疫苗管理的重要性。

1. 安装 wordcloud 库

如果使用 pip 安装工具，在终端 Terminal 里的命令行模式下，输入以下命令：

```
pip install wordcloud
```

或使用 ipython 的魔术命令 (magic commands)直接运行以下代码，

[3]:
```
%pip install wordcloud
import wordcloud
wordcloud.__version__
```

[3]:
```
'1.9.1.1'
```

[5]:
```
%pip install pillow --upgrade
import PIL
PIL.__version__
```

[5]: '9.2.0'

2. 绘制一个简单的词云图

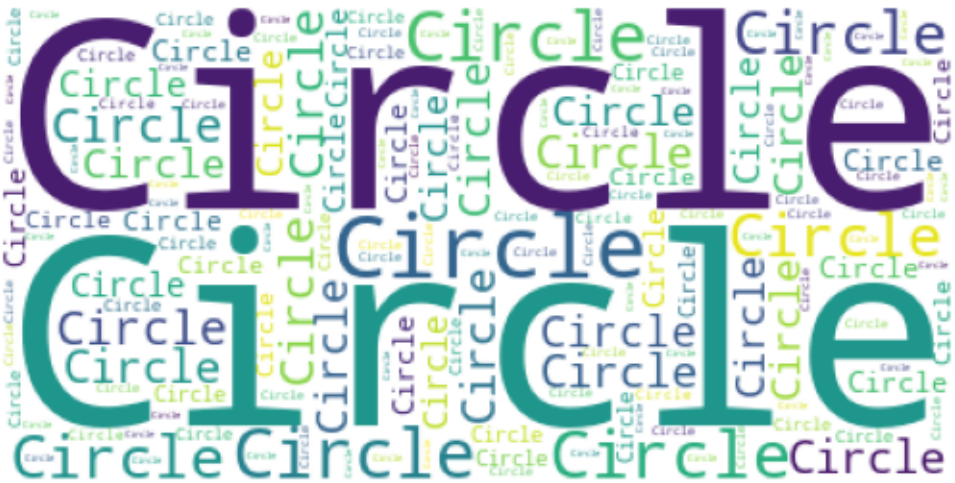
首先，应用 worldcloud 生成词云图片对象

[1]:
```
from wordcloud import WordCloud
text = "Circle"
wc = WordCloud(background_color="white", repeat=True,
               width=400,
               height=200)
wc.generate(text)
```

[1]: <wordcloud.wordcloud.WordCloud at 0x107fcca90>

接着，应用 matplotlib 库，呈现该图片对象

[2]:
```
import matplotlib.pyplot as plt
plt.imshow(wc)
plt.axis("off")
plt.show()
```



练习

观察上图，如果自定义，或者是自己设计一个词云图，考虑设计哪些特性？1. xxx 2. xxx 3. xxx

3. 增加更多的参数

wordcloud 库的 WordCloud 类有以下参数可选：

```
wordcloud.WordCloud(font_path=None, width=400, height=200, margin=2, ranks_only=None, prefer_ho
```

参数	参数类型	解释
font_path	string	将被使用的字体的路径（OTF 或 TTF）
width	int	默认 400，画布的宽度
height	int	默认 200，画布的高度
mask	nd-array	如果不是“无”，则给出一个二进制掩码，用于绘制文字的位置。如果掩码不是无，宽度和高度将被忽略，而将使用掩码的形状。所有白色（#FF 或 #FFFFFF）条目将被视为“屏蔽”，而其他条目可以自由绘制
scale	float	默认为 1，计算和绘图之间的比例。对于大的词云图像，使用比例而不是更大的画布尺寸，明显更快，但可能会导致对单词的粗略拟合
min_font_size	int	默认为 4，使用最小的字体大小。当这个尺寸没有更多的空间时就会停止
max_font_size	int	最大的字的最大字体大小。如果没有，则使用图像的高度
max_words	int	默认 200，最大单词数量
background_color	color vlaue	默认为黑色，词云图像的背景颜色
mode	string	默认为’RGB’，当模式为“RGBA”且 background_color 为 None 时，将生成透明的背景

参数	参数类型	解释
repeat	bool	默认为 False，是否重复单词和短语，直到达到 max_words 或 min_font_size

4. 定制一个英文词云

下面通过一个简单的例子，看看参数是如何使用的。

4.1 准备一个英文文本

第一步，准备一个较长的字符串，包含一个文本。

[3]:
```
text = """
       The text is...
       """
```

或者将新闻文本保存在 TXT(.txt) 文件里，然后使用以下语句读取：

[4]:
```
f = open('datasets/news_sample02.txt', 'r', encoding='utf-8')
text = f.read()
text = text.replace('\n', ' ')  # 删除换行符
```

4.2 准备一个白底的图形

在网上下载图片，例如在搜索引擎上搜索“人民币 + 图标 + 白底”:

使用修改的图片的软件修改图片大小，在保证宽 * 高比不变的情况下，宽改成 1500pixels，避免图片过大或者过小。

读取准备好的图形，作为掩码的形状。然后使用 imageio 库来读取图片。

[5]:
```
import imageio as imageio
mask_img = imageio.imread('image/rmb.png')
print('图片大小: ', mask_img.shape)
```

```
图片大小:  (1500, 1500, 4)

/var/folders/7m/rgg1hhpj3yb1j0cf0c82yq1w0000gn/T/ipykernel_28989/3419627869.py:2
: DeprecationWarning: Starting with ImageIO v3 the behavior of this function
```

```
will switch to that of iio.v3.imread. To keep the current behavior (and make
this warning dissapear) use `import imageio.v2 as imageio` or call
`imageio.v2.imread` directly.
  mask_img = imageio.imread('image/rmb.png')
```

4.3 选择颜色

以 CSS 颜色为例，

选择一个色系，例如'purple'，

[6]:
```
from wordcloud import get_single_color_func
color_func = get_single_color_func('purple')
```

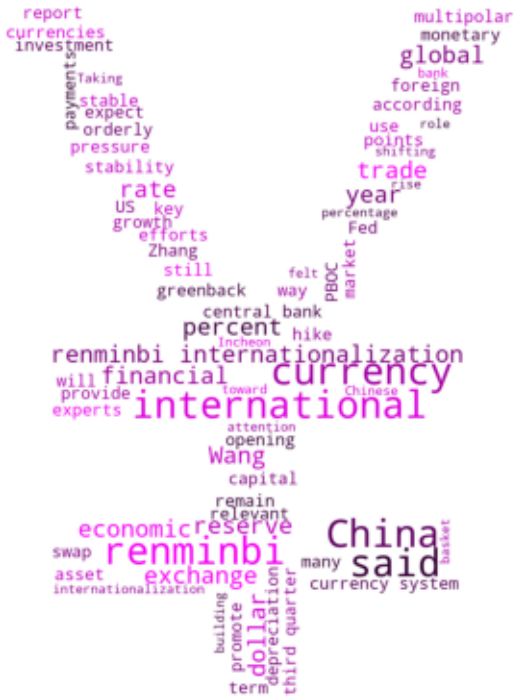
4.3 生成词云

接着，生成词云。

[7]:
```
from wordcloud import WordCloud
wc = WordCloud(mask=mask_img,
               max_words=200,
               max_font_size=80,
               min_font_size=25,
               width=1500,
               height=1500,
               background_color="white",
                color_func=color_func,
               repeat=True)
wc.generate(text)
```

[7]:
```
<wordcloud.wordcloud.WordCloud at 0x10a863d00>
```

[8]:
```
import matplotlib.pyplot as plt
plt.imshow(wc)
plt.axis("off")
plt.savefig("image/ciyun_en.jpg", dpi=500) # 或者保存成.png .svg .pdf .eps 等
plt.show()
```



练习

1. 检索并保存一个较长的文本
2. 在 bing.com 搜索引擎上，输入“符号白底”等关键词
3. 制作相应的词云图

[ ]:

附录

[1] wordcloud 文档: http://amueller.github.io/word_cloud/

[2] wordcloud 的 API Reference:http://amueller.github.io/word_cloud/references.html

[3] imageio:https://imageio.readthedocs.io/en/latest/index.html