黄冈师範学院
HUANGGANG NORMAL UNIVERSITY

人工智能与机器学习

Artificial Intelligence and Machine Learning

章节：实验二 使用Python制作中文词云

教师：刘重

学院：计算机学院

厚德 博学 力行 致远

一、实验目的

- 1） 掌握wordcloud (词云制作)、jieba (中文分词)的安装、配置方法
- 2） 掌握wordcloud (词云制作)、jieba (中文分词)、numpy (数组处理)、matplotlib (基础画图)、PIL (读取图片)库的使用方法
- 3） 使用Python制作中文词云

二、实验内容

- 1、准备工作
- 1.1. 安装并引入必要的函数库
- 1.2. 设置文件路径
- 2、文本处理: 分词, 过滤, 词频计算
- 3、词云生成, 画图

- 1、处理文本数据

在生成词云时，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。对于中文文本,分词处理需要由用户来完成。一般步骤是先将文本分词处理,然后以空格拼接,再调用wordcloud库函数

- 2、产生词云图片

wordcloud库的核心是WordColoud类,所有的功能都封装在WordCloud类中。使用时需要实例 化 一 WordColoud类 的 对 象 , 并 调 用 其generate(text)方法将text文本转化为词云

（1）文本和图片

制作词云之前, 我们需要事先准备以下素材:

✓一篇文章, 以文本文件 (.txt) 格式保存;

✓停用词表 (英文非必要, 中文需要自己准备).

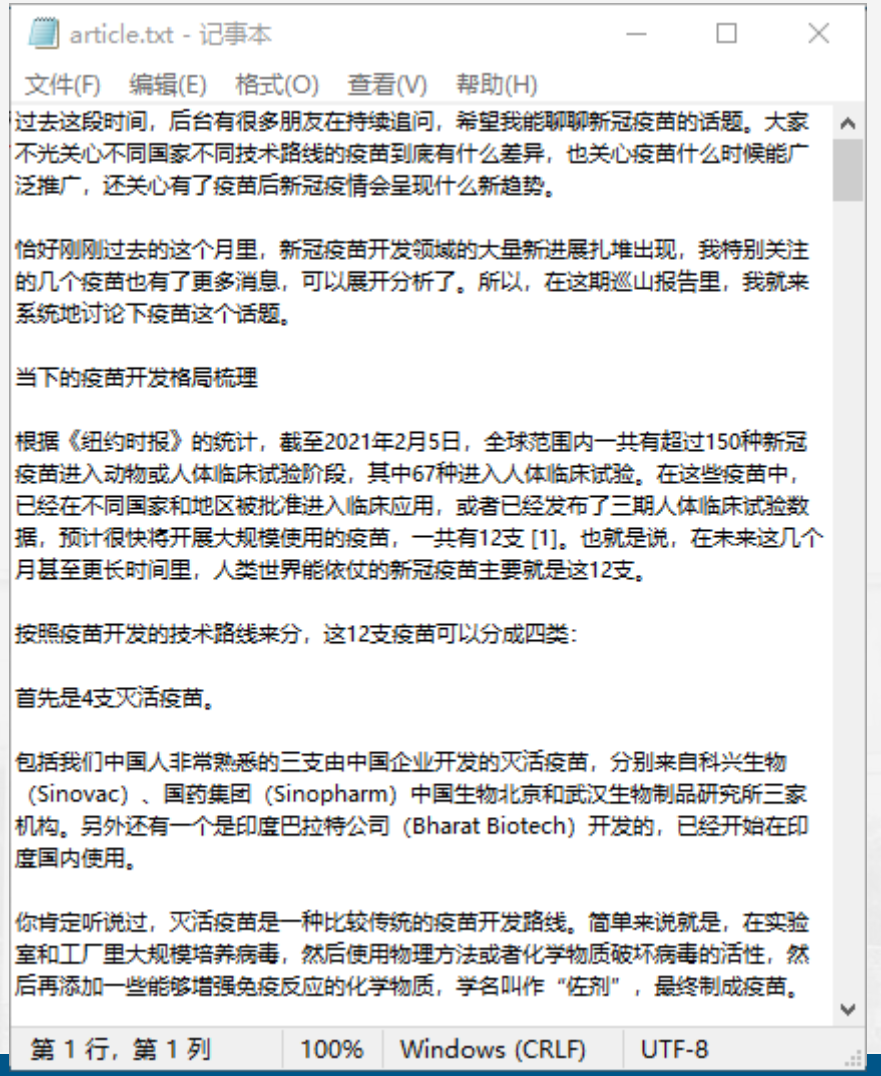
✓避免中文乱码的字体文件 (英文非必要);

✓一张你喜欢的图片(为词云上色, 或者制作剪影, 非必要)。

从素材准备可以看出来, 相比英文, 制作中文词云会稍微麻烦一点, 因为需要解决额外的两个问题:

- 使用 jieba分词, 将连续的中文句子切割成单个词语.
- 设置字体, 避免中文乱码.

①被识别文章的路径：“Chinese_word_cloud\texts\ article.txt”


article.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

过去这段时间，后台有很多朋友在持续追问，希望我能聊聊新冠疫苗的话题。大家不光关心不同国家不同技术路线的疫苗到底有什么差异，也关心疫苗什么时候能广泛推广，还关心有了疫苗后新冠疫情会呈现什么新趋势。

恰好刚刚过去的这个月里，新冠疫苗开发领域的大量新进展扎堆出现，我特别关注的几个疫苗也有了更多消息，可以展开分析了。所以，在这期巡山报告里，我就来系统地讨论下疫苗这个话题。

当下的疫苗开发格局梳理

根据《纽约时报》的统计，截至2021年2月5日，全球范围内一共有超过150种新冠疫苗进入动物或人体临床试验阶段，其中67种进入人体临床试验。在这些疫苗中，已经在不同国家和地区被批准进入临床应用，或者已经发布了三期人体临床试验数据，预计很快将开展大规模使用的疫苗，一共有12支 [1]。也就是说，在未来这几个月甚至更长时间里，人类世界能依仗的新冠疫苗主要就是这12支。

按照疫苗开发的技术路线来分，这12支疫苗可以分成四类：

首先是4支灭活疫苗。

包括我们中国人非常熟悉的三支由中国企业开发的灭活疫苗，分别来自科兴生物（Sinovac）、国药集团（Sinopharm）中国生物北京和武汉生物制品研究所三家机构。另外还有一个是印度巴拉特公司（Bharat Biotech）开发的，已经开始在印度国内使用。

你肯定听说过，灭活疫苗是一种比较传统的疫苗开发路线。简单来说就是，在实验室和工厂里大规模培养病毒，然后使用物理方法或者化学物质破坏病毒的活性，然后再添加一些能够增强免疫反应的化学物质，学名叫作“佐剂”，最终制成疫苗。

第 1 行，第 1 列 100% Windows (CRLF) UTF-8

②停用词表：“Chinese_word_cloud\stopwords”

README.md - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

中文常用停用词表

词表名	词表文件
中文停用词表	cn_stopwords.txt
哈工大停用词表	hit_stopwords.txt
百度停用词表	baidu_stopwords.txt
四川大学机器智能实验室停用词库	scu_stopwords.txt

第 8 行，第 43 列 220% Unix (LF) UTF-8

英文

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

--
?
“
”
》
- -
able
about
above
according
accordingly
across
actually
after
afterwards
again
against
ain't
all
allow
allows
almost
alone
along
already
also

UTF-8

中文

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

$
0
1
2
3
4
5
6
7
8
9
?
–
“
”
、
。
《
》
—
一些
一何
一切
一则
一方面
一旦

UTF-8

③字体文件：“Chinese_word_cloud\ SourceHanSerifK-Light.otf”

④一张你喜欢的图片“Chinese_word_cloud\pictures”

（2） 库的安装

主要涉及以下Python库:

wordcloud (词云制作)

jieba (中文分词)

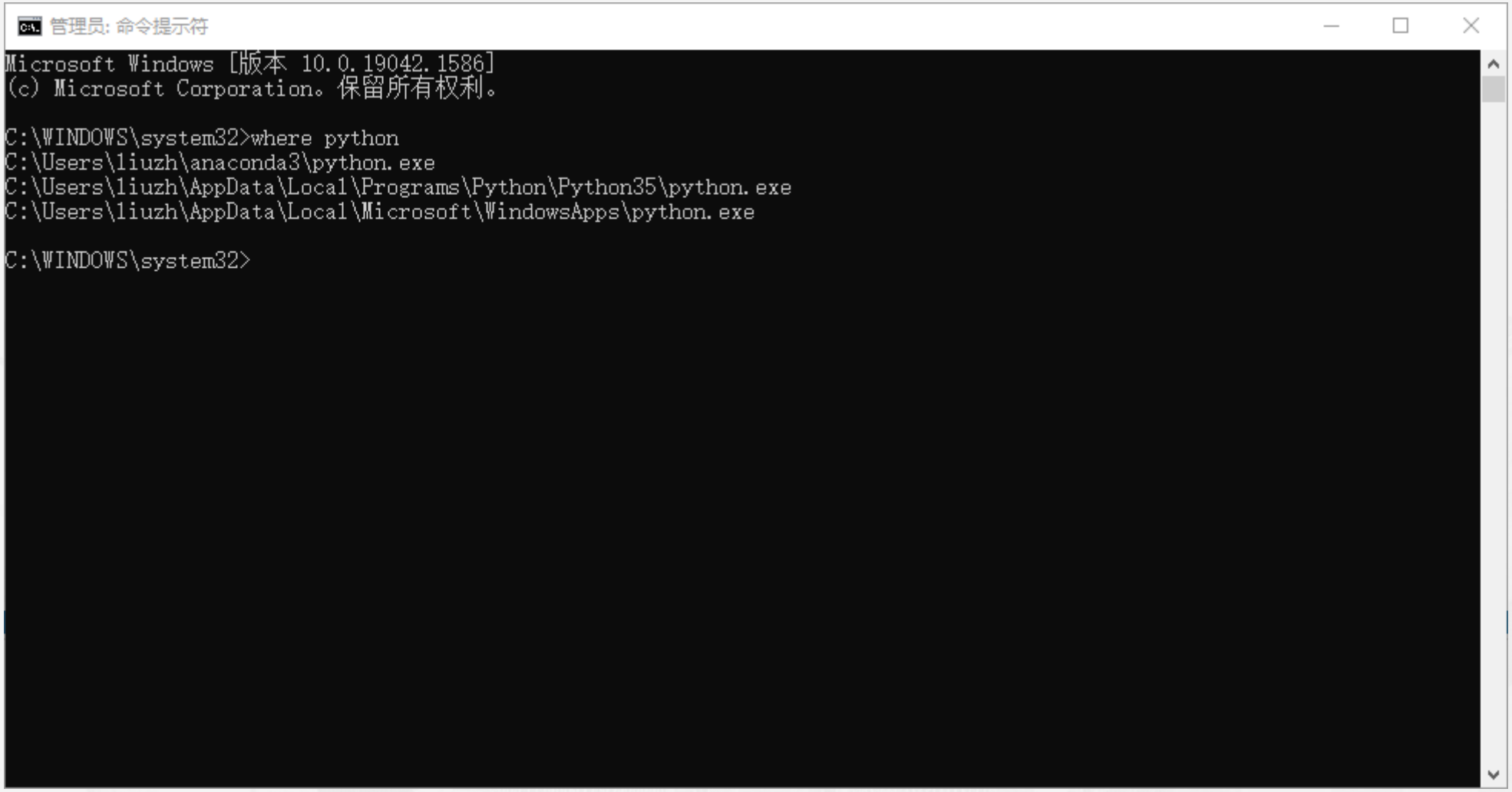
numpy (数组处理)

matplotlib (基础画图)

PIL (读取图片)

四、实验步骤

- 打开命令行 (cmd)，输入“where python”查询python的安装路径



- 输入“cd C:\Users\liuzh\anaconda3\”,进入相对应的路径(视具体情况而定)

- 制作词云的库:pip install wordcloud
- 用于中文分词的库:pip install jieba

- 代码详解：
- 第1步：准备工作
- 1.1：引入库

```
import numpy as np
from wordcloud import WordCloud, ImageColorGenerator#, STOPWORDS
import matplotlib.pyplot as plt
from PIL import Image
import jieba # cutting Chinese sentences into words
```

- 1.2：设置文件路径

```
# setting paths
fname_text = 'texts/article.txt'
fname_stop = 'stopwords/hit_stopwords.txt'
fname_mask = 'pictures/owl.jpeg'
fname_font = 'SourceHanSerifK-Light.otf'
```

- 第2步：文本处理。这一步的主要目的是将一篇文章转化为一个"词频"表 (字典 dict).

- 2.1 读取文本

- 首先, 我们得读取一篇文章, 以及停用词表:

```
# read in texts (an article)
text = open(fname_text, encoding='utf8').read()
# Chinese stop words
STOPWORDS_CH = open(fname_stop, encoding='utf8').read().split()
```

```
# read in texts (an article)
text = open(fname_text, encoding='utf8').read()
# Chinese stop words
STOPWORDS_CH = open(fname_stop, encoding='utf8').read().split()
```

- 需要注意以下几点:
- open(filename, encoding='utf8') 命令打开一个文件, 且 encoding='utf8' 告诉计算机该文件的编码方式是 ‘utf-8’, 如果没有这个设定, 会导致中文字符乱码.
- 对打开的文件, .read() 操作会返回一个字符串. 因此代码中的 text 是字符串类型.
- 最后一行中的 .split() 操作将字符串 (按照空格, tab\t, 换行符 \n) 分割成了一系列字符串, 因此STOPWORDS_CH 是一个由字符串组成的列表 list.

- 2.2 分词和过滤
- 首先用 jieba.cut(text) 函数将字符串 text 分割成一个个词或词组 (该函数返回的是一个’生成器 generator), 然后对里面的每一个词, 过滤掉没有意义的 ‘停用词’ (w not in STOPWORDS_CH), 最后只保留长度大于1的词组 (len(w) > 1).

```
# processing texts: cutting words, removing stop-words and single-charactors
word_list = [
        w for w in jieba.cut(text)
        if w not in STOPWORDS_CH and len(w) > 1
        ]
freq = count_frequencies(word_list)
```

- 2.3 统计词频。下面代码定义了一个函数, 输入一个词语列表, 输出保存每个词语出现频率的字典.

```
def count_frequencies(word_list):
    freq = dict()
    for w in word_list:
        if w not in freq.keys():
            freq[w] = 1
        else:
            freq[w] += 1
    return freq
```

- 当然也可以利用 colections.Counter() 或 pandas.value_count() 函数来计算, 以下代码与上面等价:

```
from collections import Counter
freq = dict(Counter(word_list))

import pandas as pd
freq = pd.value_counts(word_list).to_dict()
```

- 第3步：制作并画出词云
- 3.1词云的对象的创建
- 首先用 WordCloud() 建立一个词云的对象, 并设置好初始参数 (字体的路径). 然后基于刚刚建立的词频生成词云.

```
# generate word cloud
wcd = WordCloud(font_path=fname_font, # font for Chinese charactors
                background_color='white',
                mode="RGBA",
                mask=im_mask,
                )
```

- 3.2从图片提取颜色。选择自己喜欢的图片作为背景色。
- 首先读取图片, 将其转化为 RGB 数组;
- 然后用 ImageColorGenerator 从中提取颜色, 它会得到一个颜色生成器, 依照每个词所占的矩形区域的颜色平均来确定改词最终的颜色.

```
# processing image
im_mask = np.array(Image.open(fname_mask))
im_colors = ImageColorGenerator(im_mask)
```

- 3.3图片保存。
- 如果想要保存图片:
- ax.figure.savefig(f'conbined_wcd.png', bbox_inches='tight', dpi=150)
- bbox_inches='tight' 可以确保你保存的图片形状合适.

- 当然, 我们还可以来个拼图:
- fig, axs = plt.subplots(1, 2)
- plt_imshow(im_mask, axs[0], show=False)
- plt_imshow(wcd, axs[1])
- fig.savefig(f'conbined_wcd.png', bbox_inches='tight', dpi=150)

五、实验结果

六、实验报告要求

- 1、实验目的
- 2、实验内容
- 3、实验原理
- 4、实验代码
- 5、运行截图
- 6、实验小结

- 说明：每个学生都要交电子版的实验报告，命名格式：
- 01/02-XXXX（学号）-XXX（姓名）

黄冈师範学院
HUANGGANG NORMAL UNIVERSITY

Q&A

厚德 博学 力行 致远