信息检索课程设计（Comprehensive Practice on Information Retrieval) Course Resources

作业8：爬虫和某机构主页检索系统（综合项目）

- 运行环境：windows11 21H2
- 处理器：AMD Ryzen 7 5800H with Radeon Graphics 3.20 GHz
- python版本:python 3.10
- python导入的package:

```
import requests
import re
from bs4 import BeautifulSoup
import ssl
import urllib3
import urllib.request
import jieba
import os
import numpy.lib.npyio
from sklearn.feature_extraction.text import TfidfVectorizer
import warnings
```

其中，外部库为(这些库需要自行安装,若不安装，程序将无法正确运行)

- request
- BeautifulSoup
- jieba
- Scikit-Learning

本程序使用后产生的文档的保存位置如下：

文档	文档保存地址
网页源文件(html文件)	text/web
网页的title和body	text/txt
每个网站的TF-IDF值	text/tfidf值
查询结果	text/查询结果

使用方法:

该程序分为两个py文件，分别为提取网页.py和检索系统.py

其中，提取网页.py的用处是爬取网站，检索系统.py的作用为处理爬取的所有网站，并且针对这些网站建设一个根据**TF-IDF**的检索系统。

第一步：选择需要爬取的网站

打开提取网页.py，需要爬取的总网页可在

```
encoding = "UTF-8"
page = "http://scst.suda.edu.cn/"  # 该网址为需要爬取的总网页，可进行修改
web_list = [page]  # 该列表用于存放网站
web_queue = [page]  # 该队列用于存放网站
```

这一部分中进行修改，该文档默认爬取的网站为**http://scst.suda.edu.cn/**(苏州大学计算机科学与技术学院网站)

第二步：开始提取网页内容

运行程序，程序运行结束后,所有的网页源文件被保存至text/web目录下,所有网页的title和body将被保存至text/txt目录下,格式如下：

```
title:
(网页的标题)
body:
(网页的正文部分)
```

第三步：开始计算对各网站进行倒排索引，并且计算TF-IDF值，并将计算得到的TF-IDF值进行保存

打开检索系统.py,并且运行程序，运行后，程序将会显示

```
C:\Users\FHN\Desktop\信息检索\作业8：爬虫和某机构主页检索系统（综合项目）
\venv\Scripts\python.exe C:/Users/FHN/Desktop/信息检索/作业8：爬虫和某机构主页检索系统
（综合项目）/检索系统.py
Building prefix dict from the default dictionary ...
Loading model from cache C:\Users\FHN\AppData\Local\Temp\jieba.cache
Loading model cost 0.429 seconds.
Prefix dict has been built successfully.
请输入需要查询的句子(输出后请按下Enter键):
```

显示出这段文字时，表明程序已经对text/txt中所保存的网站进行了倒排索引，并且计算了TF-IDF值，TF-IDF值被保存在text/tfidf值中。保存格式为

```
(某个词语) (该词语在该文档中的TF-IDF值)
```

举例，**scst.suda.edu.cn**的TF_IDF值被保存至文档scst.suda.edu.cn.txt的TF_IDF值中。
该文档中的一部分内容为

```
党支部 0.07581148918510222
党校 0.011338650701056222
党组织 0.01695901598207448
党群 0.011338650701056222
入选 0.018952872296275555
全体 0.018952872296275555
```

前面的为词语，后面的为该词语在该文档中的TF-IDf值。若为0，则表示该文档中未出现这个词。

第四步：进行查询，并且进行网页排序，得到相似度最高的网站

接下来输入需要查询的句子，程序将自动对句子进行分词，并且在分词之后对词语进行网页排序，并且将排序结果保存至text/查询结果

举例，当我们想查询**苏州大学的毕业生**这一句子时

```
C:\Users\FHN\Desktop\信息检索\作业8：爬虫和某机构主页检索系统（综合项目）
\venv\Scripts\python.exe C:/Users/FHN/Desktop/信息检索/作业8：爬虫和某机构主页检索系统
（综合项目）/检索系统.py
Building prefix dict from the default dictionary ...
Loading model from cache C:\Users\FHN\AppData\Local\Temp\jieba.cache
Loading model cost 0.429 seconds.
Prefix dict has been built successfully.
请输入需要查询的句子(输出后请按下Enter键):苏州大学的毕业生
```

然后程序将会运行，将**苏州大学的毕业生**这一句子通过分词得到的**苏州大学**和**毕业生**这两个词分别进行搜索，最终找出TF-IDF值最高的文档，把该文档中的关键词标注出后，输入至目录text/查询结果，并且生产两个文档，分别为'苏州大学'的查询结果.txt和'毕业生'的查询结果.txt