Nanako Search Engine

Linjiang Li

July 26, 2020

1 Introduction

This program is for website search using website spider,TF-IDF and inverted index to achieve.

You can crawl all the websites which can be reached from the URL you input.And run a small server on your PC. Then you can use browser to query the websites.

It also supports the search for pdf files.

2 Usage

(The usage is for Windows OS, but you can still do similar operations on Linux)

2.1

Python3($\geq$3.8.4) is recommended.

2.2

To make sure you have the packages that will be used, run following command in your *cmd* or *bash*

```
pip install flask
pip install jieba
pip install BeautifulSoup
pip install pdfminer
```

2.3

This step is to make a database for following search

You should run *crawler.py*, *parser.py*, *index.py* one by one.

(To avoid being banned by the website server,*crawler.py* may take a lot of time.the rate of progress will be shown on the terminal)

Here is an example for Windows: run following command in your *cmd*

```
crwaler.py
https://info.ruc.edu.cn (input the domain name you want to crawl)
parser.py
index.py
```

2.4

This step is about how to use the search engine.
Run *routes.py*, then use your browser access *https://127.0.0.1:5000/*
And you will see the homepage

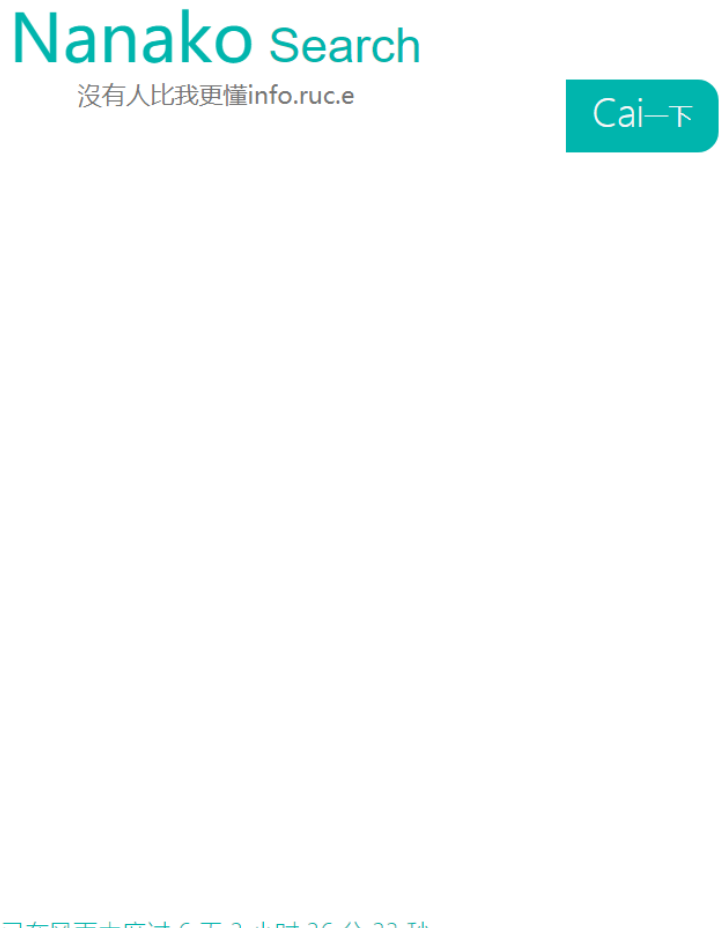


Figure 1: Nanako search: Homepage

Then you can input your queries in the search bar, and access the result page.

Figure 2: Nanako search: Resultpage

3 Details

3.1 Crawler

The crawler is implement with BeautifulSoup to get the URLs in label *<herf>*

Every accessable website will be downloaded in a file folder and numbered from 1 to n.

content.xxx(file type,like php or pdf), url.txt,suf.txt will be in the file folder.

Some urls will be ignored: the url with *download?*, the url with *mailto*, reduce

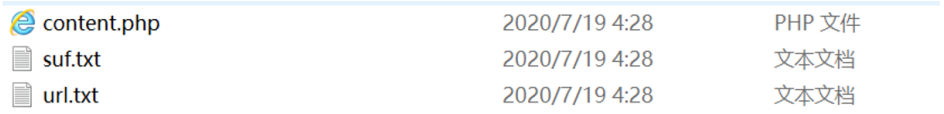


Figure 3: Nanako search: Crawler

index.php in urls.

3.2 Parser

The crawler is implement with BeautifulSoup, pdfminer and jieba.

The website content will be divided in to 3 fields: content,title and subtitle.

For html, content.txt includes the content with $<class{=}para>$, and $<li>$. subtitle.txt includes the content with $<h1>$, $<h2>$. title.txt includes the content with $<title>$.

For html, it is difficult to divided the content into different fields.So all the content will be in content.txt

After that, each field will be divded into single words by jieba.

3.3 Index

Undoubtedly, different fields enjoy different importance.So I use Fieldweight to measure its importance.(title = 10, subtitle = 2, content =1)

$tf(word, document) = 1{+}log(\sum_{word\ in\ document} appeartimes{\times}Fieldweight)$

$idf(word) = 1{+}log(\frac{N}{\sum_{each\ \ document} tf(word,document)})\quad N = max(\sum_{each\ \ document} tf(word, document))$

$score(query, document) = \sum_{word\ in\ query} tf(word, document) \times idf(word)$

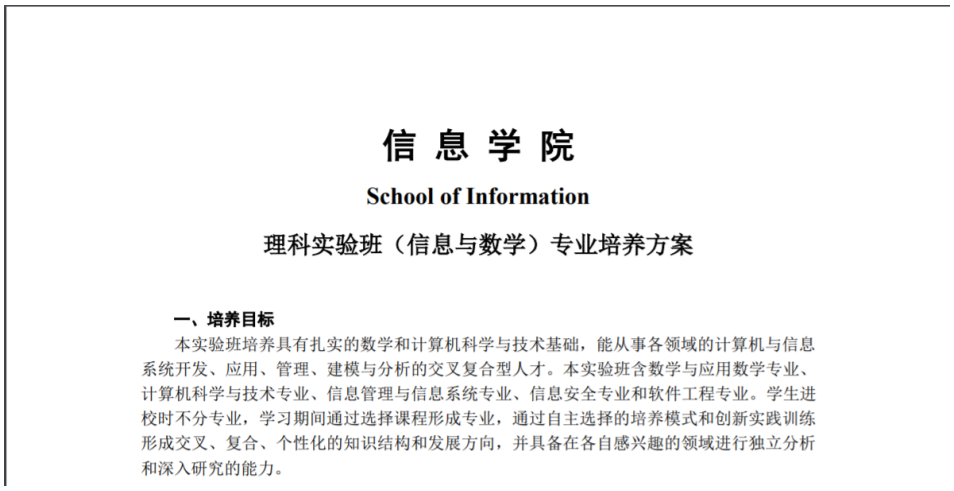
index.py will return the highest 100 documents, and this step is implemented with muti-threads.

4 Example

Here shows some queries.

Figure 4: Example 1

Figure 5: Example 2



信 息 学 院

School of Information

理科实验班（信息与数学）专业培养方案

一、培养目标

本实验班培养具有扎实的数学和计算机科学与技术基础，能从事各领域的计算机与信息系统开发、应用、管理、建模与分析的交叉复合型人才。本实验班含数学与应用数学专业、计算机科学与技术专业、信息管理与信息系统专业、信息安全专业和软件工程专业。学生进校时不分专业，学习期间通过选择课程形成专业，通过自主选择的培养模式和创新实践训练形成交叉、复合、个性化的知识结构和发展方向，并具备在各自感兴趣的领域进行独立分析和深入研究的能力。

Figure 6: pdf in Example 2

Figure 7: Example 3

Figure 8: Example 3 detail