LeavesSearch Report

Shu Wentong (2019201418)

July 21, 2020

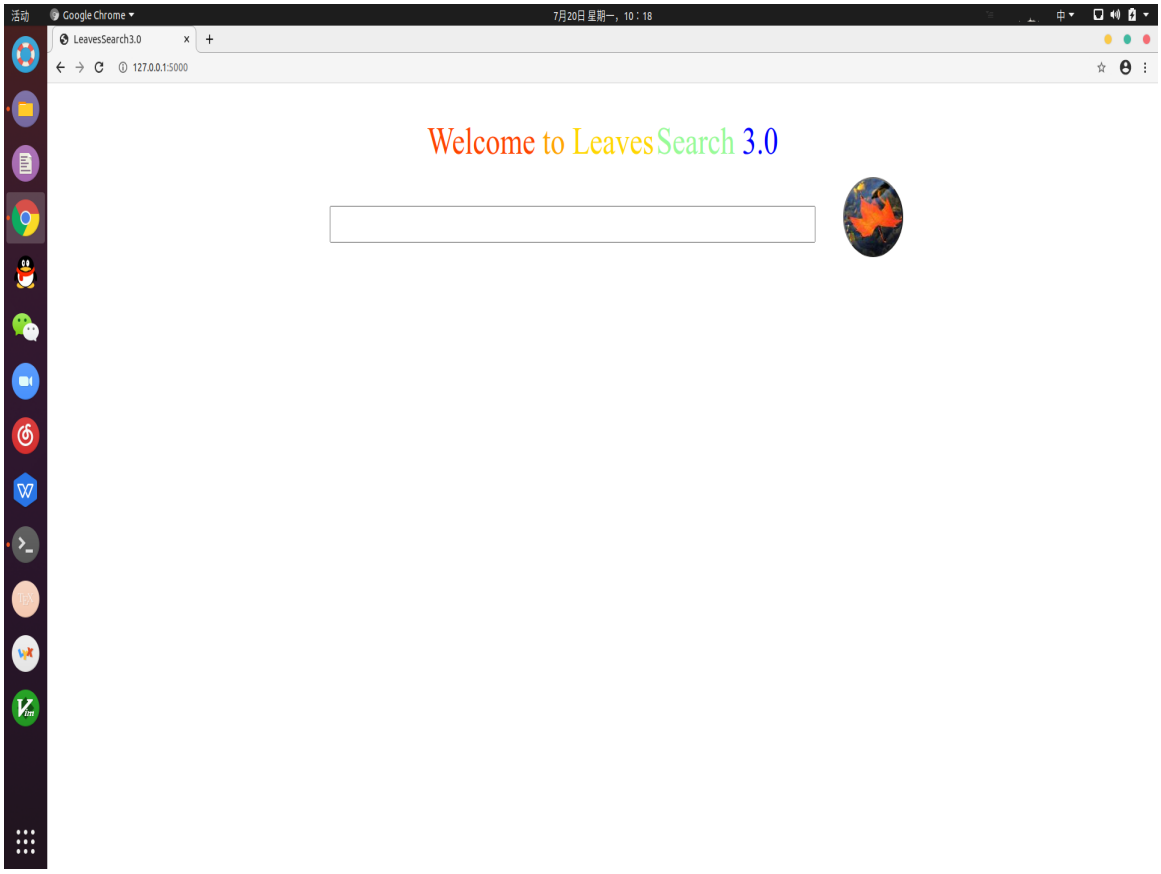
Abstract

This is a report for LeavesSearch — a simple search engine writen by Leaves(Shu Wentong).In this essay,I will introdue the introduction of the Function,thinking of design,ideas for programing,examples of queries and the way to use and update it.

1 Introduction

This is a simple Search Engine for RUC Information School.

Like google,you can submit your query words into this system,then you will get the search answers for the query in RUCINFO.(Search results are available until July 20.)



2 Thinking of Desgin

2.1 How to Get the Url?

The root of url is http://info.ruc.edu.cn/.

We start with this one,using “wget” to download the html test of this url.Then read every possible url in the html,and do normalizing,delete some illegal url like ‘.doc’ ‘.pdf’ ‘.xls’ and so on.Push legal url into the Queue.And do the same thing to others.

2.2 How to Get the Words and Cut them?

For each url,we read the title and tests in its html.Like title or p ,we read the words after these label.

Use THULAC to cut words,and ignore some words' characteristic like 'w' or 'h'.

2.3 How to Search?

We use tf-idf and VSM.

First,we cut the query words.Then calculate the tf-idf for the query and each document.

We have a vector for each document now.

Calculate the cos between the query's vector and document's.Choose the best,the biggest.

Sort them by cos.Take K url and give them to the query.

2.4 Language

We use C++ to write urlget,cut ,pre and Search.

Use python to write appweb.(use flask)

Use html to write UI.

3 Ideas for Programing

3.1 How to Get tf-idf?

We can't wget every html and calculate tf-idf at the beginning of the Search.It costs too much time.We need to do preparation.

We write a file of tf for every document and write a file for word's IDF.When we need them,we just read the file and get the informations.

If you just calculate tf-idf and cos for tests,you will get many problems.

For example,we use this formula to calculate:

$$V_j = (1 + \log_{10}(tf_j)) \times \log_{10}(\frac{N}{idf_j}))$$

if we have: (word,times)

query: (A,1) (B,1)
doc 1: (A,1) (B,1)
doc 2: (A,1) (B,5)

Although we know,B is more important than A(like A='RUC',B='ACM'),but IDF is same to every doc and query,so as to the cos,the answer will choose doc1 instead of doc2.

And one more things,when we are Searching,the title is often more important than the contents.So we need to take titles into consideration.

So I find my way to calculate it.

First,we calculate tf-idf for titles.

$$V_j = (1 + \log_{10}(Times_j)) \times \log_{10}(\frac{2\times N - idf_j}{2\times idf_j}))$$

($Times_j$: the times of j in this title)

Do the same thing to query.Get the cos and this cos is the T_i — the first value.

Second,we calculate tf-idf for contents.

$$V_j = (1 + \log_{10}(Times2_j) + 0.25 \times \frac{Times_j}{TotalT} + \frac{Times2_j}{TotalC}) \times \log_{10}(\frac{2\times N - idf_j}{2\times idf_j}))$$

($Times_j$: the times of j in this title)

($Times2_j$: the times of j in this test)

($TotalT$: the times of words in this title)

($TotalC$: the times of words in this test)

Do the same thing to query.Get the cos and this cos is the S_i — the second value.

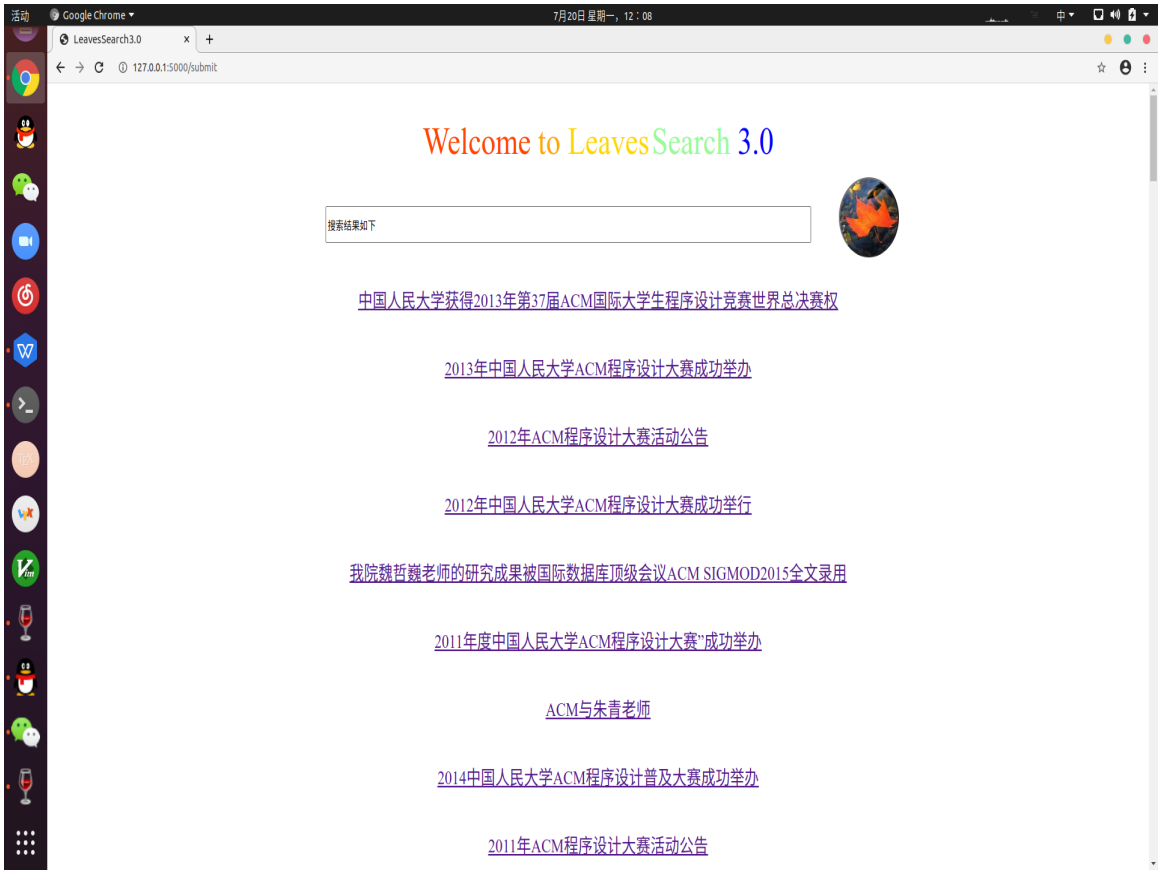
Then sort url by first value.If first value is same,then sort by second value.

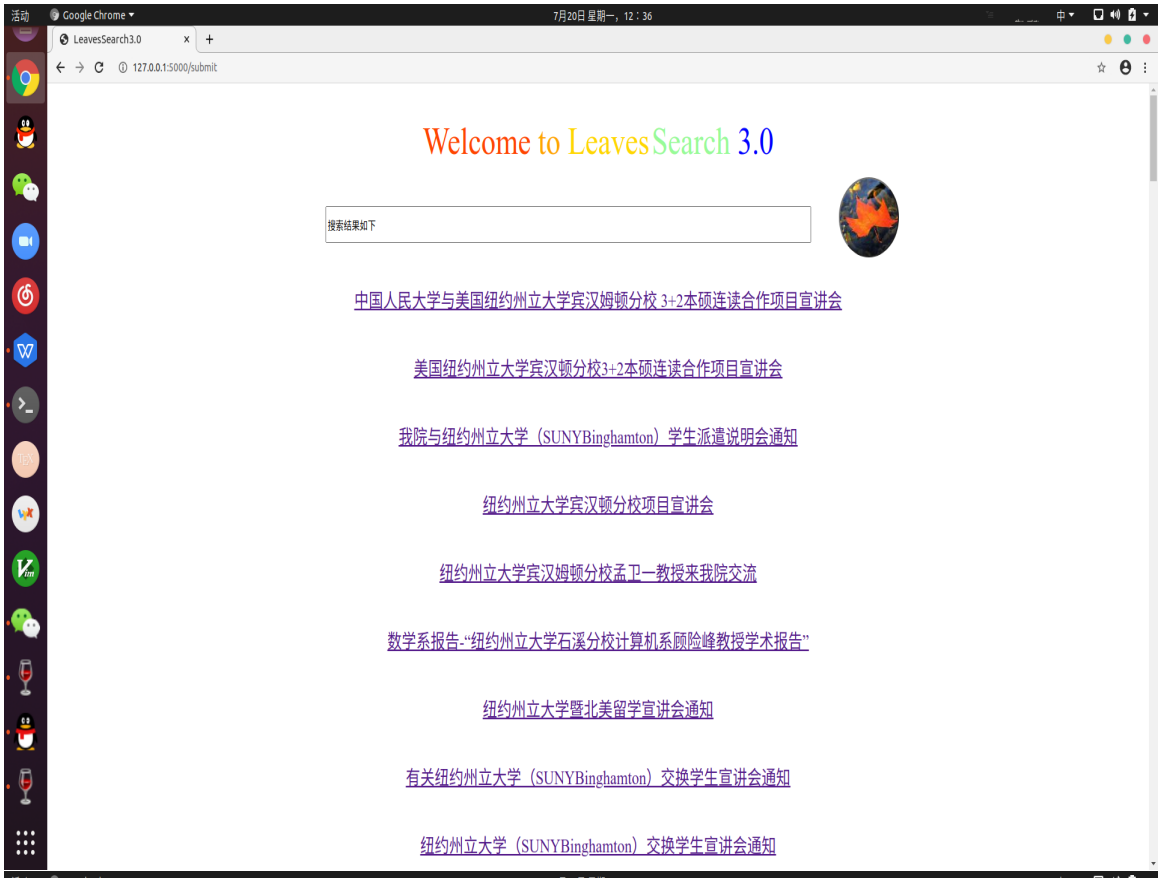
In this way,We can keep the importance of titles by first value.And when titels value is same,we can compare the test by second value.In second value,we use $\log 10$,word frequency for title and tests.And use coefficient to decrease the effect from titles.(another way is to add two values and sort)

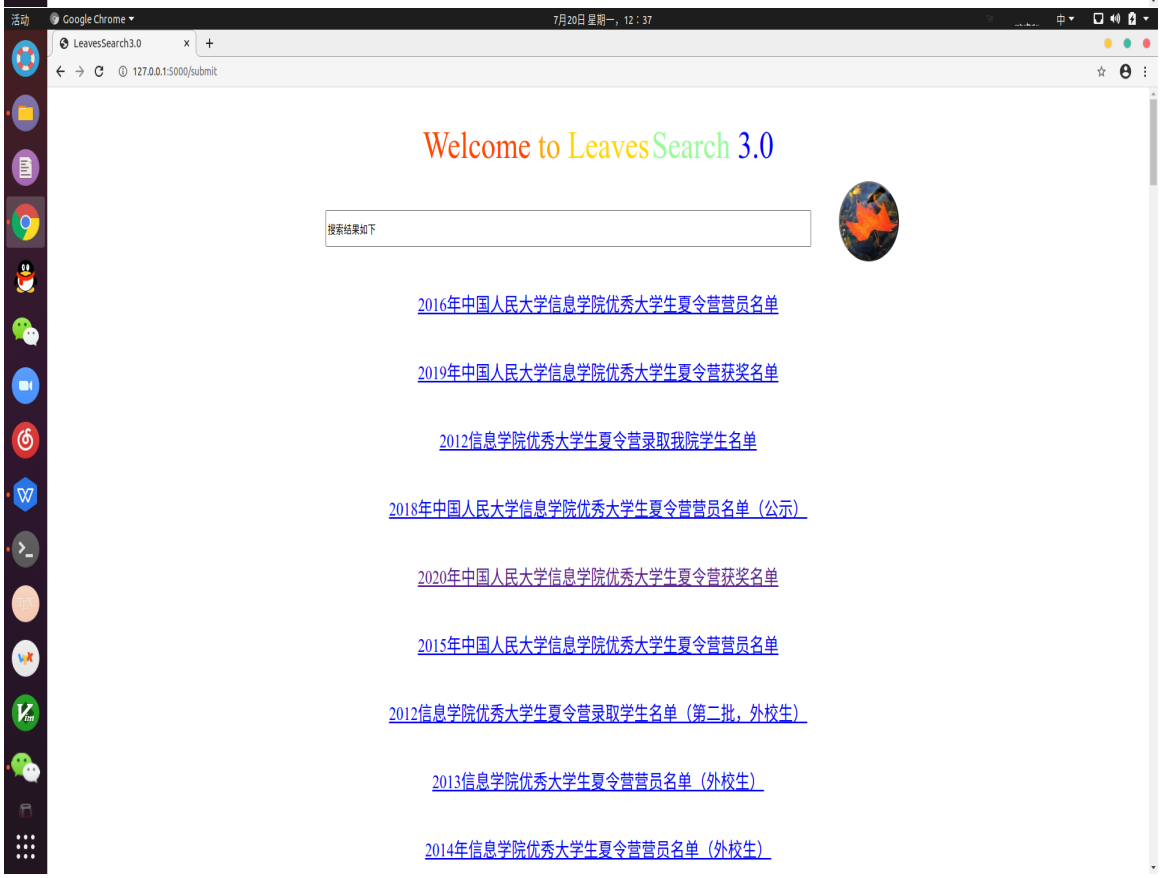
And the answer is not so bad.I am one of five people for Best Human Evaluation.And after updating my data and use the right new program to retest,I also get a good score for Automatic Evalution.

4 Examples of Query

Let's see some examples of query.




活动
Google Chrome
7月20日 星期一，12：36
LeavesSearch3.0
127.0.0.1:5000/submit
Welcome to Leaves Search 3.0
搜索结果如下
中国人民大学与美国纽约州立大学宾汉姆顿分校 3+2本硕连读合作项目宣讲会
美国纽约州立大学宾汉顿分校3+2本硕连读合作项目宣讲会
我院与纽约州立大学（SUNYBinghamton）学生派遣说明会通知
纽约州立大学宾汉顿分校项目宣讲会
纽约州立大学宾汉姆顿分校孟卫一教授来我院交流
数学系报告-“纽约州立大学石溪分校计算机系顾险峰教授学术报告”
纽约州立大学暨北美留学宣讲会通知
有关纽约州立大学（SUNYBinghamton）交换学生宣讲会通知
纽约州立大学（SUNYBinghamton）交换学生宣讲会通知


活动
Google Chrome
7月20日 星期一，12：37
LeavesSearch3.0
127.0.0.1:5000/submit
Welcome to Leaves Search 3.0
搜索结果如下
2016年中国人民大学信息学院优秀大学生夏令营营员名单
2019年中国人民大学信息学院优秀大学生夏令营获奖名单
2012信息学院优秀大学生夏令营录取我院学生名单
2018年中国人民大学信息学院优秀大学生夏令营营员名单（公示）
2020年中国人民大学信息学院优秀大学生夏令营获奖名单
2015年中国人民大学信息学院优秀大学生夏令营营员名单
2012信息学院优秀大学生夏令营录取学生名单（第二批，外校生）
2013信息学院优秀大学生夏令营营员名单（外校生）
2014年信息学院优秀大学生夏令营营员名单（外校生）

5 The Way to Use and Update

5.1 Use

Cd LeavesSearch
* flask run
* go to http://127.0.0.1:5000/

5.2 Update

Cd app/THULAC-master
* g++ -o urlget urlget.cpp
* ./urlget
* g++ -o CutforTF CutforTF.cpp -lpthread
* ./CutforTF
* g++ -o CutforIDF CutforIDF.cpp -lpthread
* ./CutforIDF

(They take about 25min)
Then cd app
* g++ -o pre pre.cpp -lpthread

Check app/THULAC-master/url ,get the total of URL — N
change the N in Search.cpp
* g++ -o Search Search.cpp

Then you will have new data.

6 End

That's all.Thank for my Teacher,Zhao Xin.I have learnt a lot of things in this programing training.I have learnt Pyhton,HTML , web skills and many knowledges during this week.And also thank for assistants,especially Menci(Huang Haorui).Without your help,I can't achieve these things.

Thank you all.