

SHY Search Engine of CC98 Forum

Lu Yifan
Zhejiang University
3110101132
15267001426
maxluyifan@gmail.com

Shi Yi
Zhejiang University
3110102957
13777857181
yishi@zju.edu.cn

Shan Caihua
Zhejiang University
3110000164
15267035890
sxdttg@126.com

ABSTRACT

This paper provides an introduction of our SHY search engine using in CC98 Forum in Zhejiang University. We use typical techniques in information retrieval to create a fast and precise search results for users. Search scope includes title, author and content of posts. We support exact search and fuzzy search. We also invent a demo showing how our search engine works.

General Terms

Algorithms

Keywords

CC98 Forum Search engine

1. INTRODUCTION

CC98 Forum is popular in Zhejiang University, and almost every student has login the forum, post and comment about some interesting issues. But nowadays the forum supports to search in the title and author only, not in the content of posts. And it can only find the text that is an exact match of any characters. It can't support fuzzy search. These shortcomings cause users great inconvenience when they search.

At most cases users don't know his needs exactly, so exact search will miss many related posts. Meanwhile, some posts' title don't summarizes the content successfully, so searching in the title will also lead to miss many related posts.

Thus, it's necessary to invent a new and smarter search engine for CC98 Forum. And our work is just to do it. The main function of our search engine includes as following:

1. Users can input several key words once, then our search engine can search hierarchically. Firstly we search in the title and author, then we search in the main content of posts.

2. Our search engine support fuzzy search. We use \$ to respresent one or more characters. For example, if you input "林*华", you can get some posts related to "林建华" or "林清华". If you input "计算机*技术", you can get equal effect as you input "计算机科学与技术"、"计算机安全技术".

3. We will sort the search results. The main factors include the degree of correlation, the time of Last Post and the number of Replies.

Of course we think students in Zhejiang University will benefit from our search engine. Using our engine, they can save more time and find more useful information.

2. METHOD

2.1 Crawler

Because the structure of CC98 Forum is simple and the content which we should crawl is clear and definite, the crawler is easy to implement.

We crawl the whole CC98 Forum for 4 steps. Firstly we put the homepage www.cc98.org into the waiting Scheduler. Though the homepage, we can get every board's URL.

版面分类		
教师答疑 (76)	版主: Argdyt 发帖: 27389	学习天地 (12) 今日: 63
个性生活 (12)	版主: klgekaka 发帖: 368110	休闲娱乐 (15) 今日: 652
电脑技术 (22)	版主: Auser 发帖: 146390	社科学术 (12) 今日: 22
感性空间 (8)	版主: klgekaka 发帖: 1594022	瞬间永恒 (16) 今日: 291
院系交流 (28)	版主: woai 发帖: 906626	社团风采 (72) 今日: 21

Secondly, we enter into each board and get every small board's URL.

教师答疑	
教师答疑区事务版	
受理开版请求及区下老师要求	
版主: Argdyt	
计算机类通识课程答疑版	
主要面向大一学生, 课程包含: 大学计算机基础、C/C++语言、JAVA、VB语言等通识课程	
版主: chenjh919 jijm_luhq xsy 0092546	
精品课程C语言答疑中心 (5)	
何钦铭老师, 许瑞清老师, 吴春明老师, 吕红兵老师, 应晶老师在此答疑	
版主: 暂无	
陈建海老师答疑 (3)	
大学计算机基础 :C/C++程序设计、软件项目管理、大型机技术	
版主: chenjh919	
陈越老师答疑版	
数据结构, 数值分析, 软件工程 http://cy.cc98.org	
版主: chenye	
方红光老师答疑版	
方老师主讲VB, 网络应用基础课程	
版主: fhg	

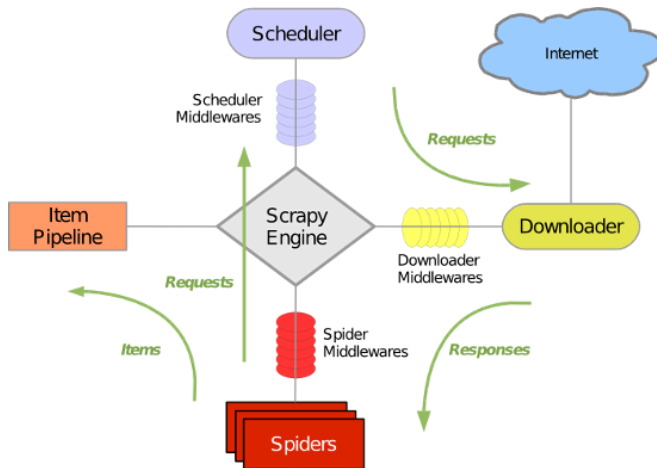
Thirdly, we enter into each small board and get every post's URL.

主题
【公告】CC98 学生网络协会成员长期公开招聘 [0 1 2 3 4 ... 11 12 13]
论坛假期管理办法
教师答疑区开版及版主申请的相关管理办法
申请彭笑刚老师答疑版版主
请辞彭笑刚老师答疑版版主
请辞
有机化学在线答疑
申请沈语冰老师答疑版版主 [0 1 2]
请问外国哲学 应奇老师的联系方式!
请辞魏宝刚老师答疑版版主
强烈建议开启《工程训练》答疑
增加子版块申请 [0 1 2]
申请彭笑刚老师答疑版版主
申请开设王跃明老师答疑版

Finally, we enter into every post. According to tag in the html, we can judge what is title, what is author and what is main content.



We use **Scrapy** to implement the whole procedure. **Scrapy** is an open source web scraping framework for Python. The framework of it is figured as below:



The green line represents the data flow. From initial URL, Scheduler will give it to Downloader. After downloading, data will be transferred to Spider to analyze. The result of analyzing has two situations: one is that it is the next page URL, so it should be transferred back to Scheduler; another is that it should be stored, so it is transferred to Item Pipeline. In Item Pipeline, the data is processed in detail.

2.2 Parser

To every post, we should analyze the title, author and content. For the sake of convenience, we store them separately. It means that we store the title and author in a text, and store the corresponding content in another text. It will be useful in Tiered Indexes.

Because we crawl the Chinese data, we use a Chinese word segmentation system named **jieba** to process the data. It has three models: The first is called precise model which can cut the sentence into several words. The second is called full model which can get all the possible word in the sentence. The third is

called search engine model which can cut the long word and improve recall.

In our search engine, we use full model because it can get as many as possible words in a sentence. The effect of it can cut "计算机" into "计算" and "计算机".

2.3 Indexer

2.3.1 Tiered Indexes

We use two-tier system to search. Tier 1 is the index of all titles and authors in the posts. Tier 2 is the index of all contents in the post. Obviously, posts containing the search words in the title are better hits than posts containing the search words in the content. So it's sensible to search hierarchically.

In the Tire 1, we use Boolean Retrieval. It can improve the speed of searching. So we just use the inverted list to store the information. Dictionary stores all the words appears in the title of the post. Postings lists store which documents it appears.

In the Tire 2, we use vector space model. We use traditional method to calculate the df-idf.

The log frequency weight of term t in d is defined as follows:

$$w_{t,d} = \begin{cases} 1 + \log_{10} \text{tf}_{t,d} & \text{if } \text{tf}_{t,d} > 0 \\ 0 & \text{otherwise} \end{cases}$$

We define the idf weight of term t as follows:

$$\text{idf}_t = \log_{10} \frac{N}{\text{df}_t}$$

So the tf-idf weight of a term is the product of its tf weight and its idf weight:

$$w_{t,d} = (1 + \log \text{tf}_{t,d}) \cdot \log \frac{N}{\text{df}_t}$$

Fianlly we normalize $|\vec{q}|$ and $|\vec{d}|$, using the formula as below to calculate the similarity between $|\vec{q}|$ and $|\vec{d}|$.

$$\cos(\vec{q}, \vec{d}) = \text{SIM}(\vec{q}, \vec{d}) = \frac{\vec{q}}{|\vec{q}|} \cdot \frac{\vec{d}}{|\vec{d}|} = \sum_{i=1}^{|\vec{V}|} \frac{q_i}{\sqrt{\sum_{i=1}^{|\vec{V}|} q_i^2}} \cdot \frac{d_i}{\sqrt{\sum_{i=1}^{|\vec{V}|} d_i^2}}$$

2.3.2 Permuterm Index

To fuzzy search, we use permuterm index to support. Permuterm index rotates wildcard to the right. Like a word "计算机", we store "计算机\$", "算机\$计", "机\$计算", "\$计算机" into the B-tree. So when we search "计*算机", we just rotate wildcard into right, so actually we search "算机\$计*" in the B-tree.

2.4 Scorer

Because we use tiered indexes, we think if we can search something in Tiered 1, it's more important than what we can search in Tiered 2.

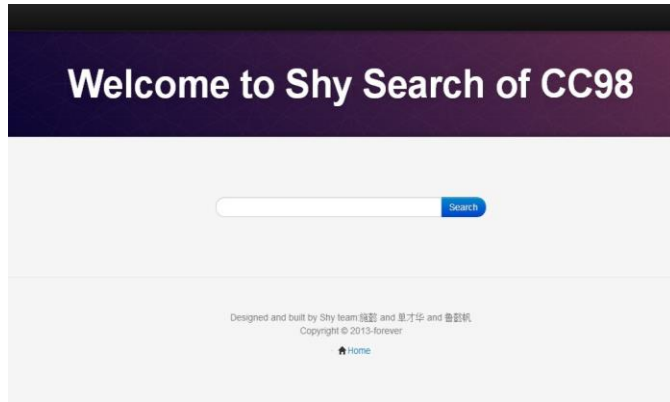
Since in Tiered 1 we use Boolean Retrieval, we just sort the results in the time of Last Post. Then in Tiered 2 we use vector

space model, we sort the results with angle. We use binary min-heap to get Top K results. It will be faster than sort all the results.

To fuzzy search, we also sort the results in the time of Last Post.

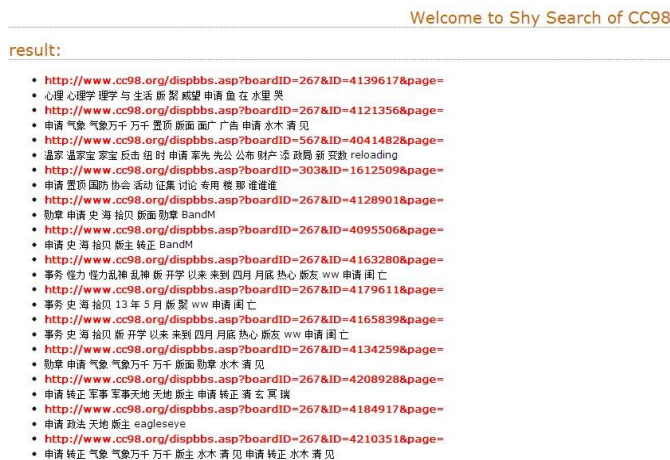
2.5 Search Interface

Our main search interface is like as below:



So users just need to input key words, then click 'search', the result will be shown automatically.

The results will be shown in a list:



It will show the link address and key words of the post, so users can judge whether the post is useful or not easily. If it's interesting, users just click the link address then he can enter that post directly.

3. EVALUATION

3.1 Correction

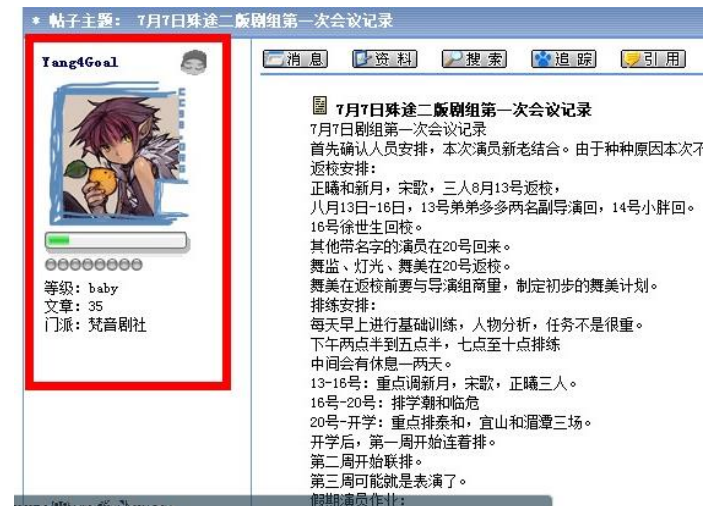
To demonstrate the correction, we just test several words in our search engine.

First example: if we search author name like "Yang4Goal", the results are as below:

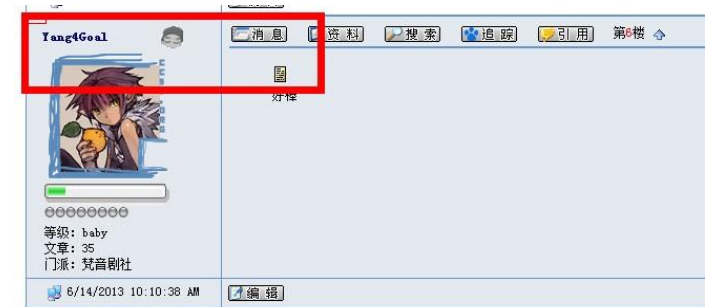


The first red box represents the posts that Yang4Goal posted. And next red box represents the posts that Yang4Goal replied. Of course we should put the theme posts(主题帖) posted by Yang4Goal in the front of the reply posts(回帖).

If we click the theme post posted by Yang4Goal, we can see

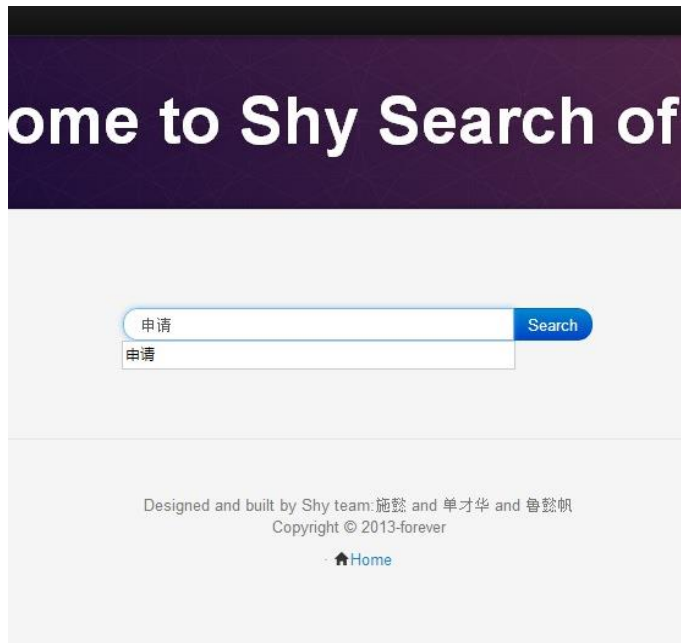


Then if we click the reply post posted by Yang4Goal, we can see



The red box shows where Yang4Goal appears. Obviously in this post Yang4Goal is a replier.

Second example: we search "申请", like the picture as below:



Then the results are as below:

result:

- <http://www.cc98.org/dispbbs.asp?boardID=267&ID=4139617&page>
- 心理 心理学 理学 与生活 版 聚 威望 申请 鱼 在 水里 哭
- <http://www.cc98.org/dispbbs.asp?boardID=267&ID=4121356&page>
- 申请 气象 气象万千 万千 置顶 版面 面 广告 申请 水木 清 见
- <http://www.cc98.org/dispbbs.asp?boardID=567&ID=4041482&page>
- 温家 温家宝 家宝 反击 组 时 申请 率 先 公 布 财 产 添 政 局 新 变 数 reloading
- <http://www.cc98.org/dispbbs.asp?boardID=303&ID=1612509&page>
- 申请 置顶 国防 协会 活动 征集 讨论 专用 楼 那 谁 谁 谁
- <http://www.cc98.org/dispbbs.asp?boardID=267&ID=4128901&page>
- 勋章 申请 史 海 拾 贝 版面 勋章 BandM
- <http://www.cc98.org/dispbbs.asp?boardID=267&ID=4095506&page>
- 申请 史 海 拾 贝 13 年 5 月 版 聚 ww 申请 闰 亡
- <http://www.cc98.org/dispbbs.asp?boardID=267&ID=4163280&page>
- 事务 怪力 怪力 乱 神 乱 神 版 开 学 以 来 到 四 月 月 底 热 心 版 友 ww 申请 闰 亡
- <http://www.cc98.org/dispbbs.asp?boardID=267&ID=4179611&page>
- 事务 史 海 拾 贝 13 年 5 月 版 聚 ww 申请 闰 亡
- <http://www.cc98.org/dispbbs.asp?boardID=267&ID=4165839&page>
- 事务 史 海 拾 贝 版 开 学 以 来 到 四 月 月 底 热 心 版 友 ww 申请 闰 亡
- <http://www.cc98.org/dispbbs.asp?boardID=267&ID=4134259&page>
- 勋章 申请 气象 气象万千 万千 版面 勋章 水木 清 见
- <http://www.cc98.org/dispbbs.asp?boardID=267&ID=4208928&page>
- 申请 转正 军事 军事 天地 天地 版 主 申请 转正 清 玄 冥 瑞
- <http://www.cc98.org/dispbbs.asp?boardID=267&ID=4184917&page>
- 申请 政法 天地 版 主 eagleseye
- <http://www.cc98.org/dispbbs.asp?boardID=267&ID=4210351&page>
- 申请 转正 气象 气象万千 万千 版 主 水木 清 见 申请 转正 水木 清 见

Look the results, we can find these posts' title contains "申请", so we show them. In our search engine, we use Tiered Indexes, which means the post whose title contains key words is more important than the post whose content contains. Since "申请" is usually used in the title of posts, so we only show the post whose title contains "申请".

Third example: we search "施懿" which is our teammate's name, we can get results like this:

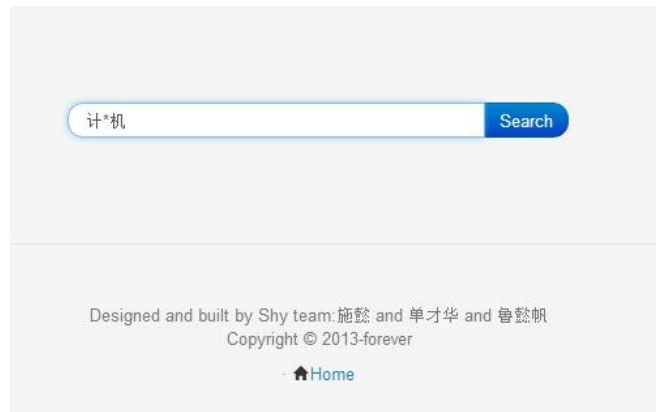


This post is actually related to "施懿". But the post below is not so related.



The reason for searching this post is that there are too many "施" in the title and content. So the angle may be small.

Fourth example: we use wildcard and do fuzzy search, like we input "计*机":



The results will be shown as below:

sult:

- <http://www.cc98.org/dispbbs.asp?boardID=626&ID=3689522&page=>
- 求 问 杨 颖 老师 计算 计算机 计算机辅助 算机 辅助设计 设计 初步 的课程 情况 郁 筱 楠
- <http://www.cc98.org/dispbbs.asp?boardID=574&ID=4007929&page=>
- 老师 大 一 新 生 求 助 关于 计算 计算机 算机 二级 考试 的 根 | 记忆
- <http://www.cc98.org/dispbbs.asp?boardID=183&ID=4187716&page=>
- 关于 计算 计算机 算机 机组 组成 的 第五 第五章 五章 作业 有人 忘记 写 姓名 姓名 学 学 学 学 号 ta
- <http://www.cc98.org/dispbbs.asp?boardID=183&ID=4191460&page=>
- 计算 计算机 算机 机组 组成 成 平 平时 作业 作业 成绩 成绩 如有 疑问 疑问 请 站 短 联系 ta net
- <http://www.cc98.org/dispbbs.asp?boardID=183&ID=4191462&page=>
- 计算 计算机 计算机 网 计算机 网络 算机 网络 基础 平时 作业 作业 成绩 成绩 及 实验 实验 报告 报告 告
- <http://www.cc98.org/dispbbs.asp?boardID=183&ID=4197311&page=>
- 计算 计算机 计算机 网 计算机 网络 算机 网络 基础 第 7 章 的 作业 上交 截至 时间 2013 6 14 日 lu
- <http://www.cc98.org/dispbbs.asp?boardID=183&ID=4202491&page=>
- 计算 计算机 算机 机组 组成 截止 第 7 章 的 平时 作业 作业 成绩 成绩 如有 疑问 疑问 请 站 短 联系
- <http://www.cc98.org/dispbbs.asp?boardID=183&ID=4204556&page=>
- 计算 计算机 算机 机组 组成 第三 第三 版 中文 文 电 电子 电子书 子 书 已 放在 FTP 服务 服务器 器 器
- <http://www.cc98.org/dispbbs.asp?boardID=183&ID=4217992&page=>
- 2013 年 春 夏 春 夏 学 期 计算 计算机 算机 机组 组成 期 中 试 试卷 lukj
- <http://www.cc98.org/dispbbs.asp?boardID=183&ID=4217819&page=>
- 计算 计算机 算机 机组 组成 答疑 安排 排 在 考试 前一 天 一天 的 下午 lukj
- <http://www.cc98.org/dispbbs.asp?boardID=183&ID=4218985&page=>
- 计算 计算机 算机 机组 组成 第五 第五章 五章 有 两份 电子 电子版 作业 未 写 名字 学 学 号 lukj
- <http://www.cc98.org/dispbbs.asp?boardID=183&ID=4016233&page=>
- 计算 计算机 计算机 网 计算机 网络 算机 网络 基础 计算 计算机 算机 机组 组成 汇编 与 接口 等 课程
- <http://www.cc98.org/dispbbs.asp?boardID=183&ID=4022474&page=>
- 计算 计算机 计算机 网 计算机 网络 算机 网络 基础 的 习题 实验 内容 参考 参考资料 资料 请 看 本 站
- <http://www.cc98.org/dispbbs.asp?boardID=640&ID=3749167&page=>
- 有 同学 想买 逻辑 与 计算 计算机 算机 机 设 设计 中文 中文版 文 版 教材 吗 已 出 wangwy1105
- <http://www.cc98.org/dispbbs.asp?boardID=351&ID=4196041&page=>
- 计算 计算机 算机 学院 在 全校 范围 内 选拔 优秀 大学 大学生 学生 赴 灵 隐 街道 开展 挂职 挂职 职 职

The results include "计算机"、"计算机网络"and so on. It means "计*机" can match these key words and search the posts containing these words.

3.2 Efficiency

Our demo is based on CC98 Forum. We crawl 4000 posts which have 4000 titles and 4000 contents. It is estimated that demo has 130,000 words.

Now doing a search in the demo costs 30-40 seconds per query. Opening the Chinese word segmentation system named **jieba** to process the query needs 15 seconds. Searching in two Tiered Indexes and finding relevant post need 15-25 seconds. Because we use vector space model, we should calculate the angle between query and each post. It takes the majority of time. If we have

several servers, we think our algorithm can get the search results in an acceptable time.

4. FURTHER ENHANCEMENT

1. Since we must need more than 20 seconds to complete one query, we think many methods to reduce time.

One way is to open the Chinese word segmentation system named jieba and some necessary information once, make them store in memory. So if we need them later on, we just need to read them from memory, not disk. Another way is to use multi-threading. They will add complexity to search engine execution but improve efficiency dramatically.

2. We want to achieve more complex query. Now we don't support 'and' and 'or' Boolean operator hierarchically. We want to achieve query such as ((1 or 2) and 3) or 4, where 1-4 are any words. We think it will be a novel improvement.

3. We are supposed to add more vocabulary thesaurus to cut sentences more precisely. Then we should also build a better Semantic Model to process Chinese. We think it will provide a better user experience.

4. We plan to combine exact search with fuzzy search. It means users can input several words that can contains wildcard or not, such as "林*华 浙江大学 重庆大学 山东". There are exact words and fuzzy words in a query. From these key words, we think we can find posts better than just a fuzzy word "林*华".

5. REFERENCES

- [1] <http://www.scrapy.org/>
- [2] <http://www.python.org/>
- [3] <https://github.com/fxsjy/jieba>
- [4] <http://www.w3school.com.cn/xpath/>
- [5] <http://www.w3school.com.cn/h.asp>