《UDIAB——面向软件开发者的现代化搜索引擎》开题报告

小组成员 ： 张曙 赵宏铎 钱正玭 熊吉思汗
撰写人：赵宏铎

一． 系统开发目的与意义

随着信息化的蓬勃发展，搜索引擎成为了人们日常获取信息的关键渠道。然而占据较高市场份额的搜索引擎绝大多为“百科全书型”搜索引擎——可用于全网搜索任何领域的信息，但是由于缺少定位特定领域高质量信息的能力，导致“百科全书类”搜索引擎在面对特定领域专业问题时，搜索结果信息正确率低、质量低下等问题。本系统的开发目的是，提供“专业技术型”搜索引擎——将文档库定位于软件开发相关专业平台、论坛，为开发者提供更加精准有效的信息支持。能为软件开发者提供优秀的检索体验、更高的开发效率便是这一系统的意义所在。

二． 市场类似系统调研

我们通过对专业性问题的搜索对市场上部分“百科全书型”搜索引擎进行了对比。其中专业性问题搜索包括短语搜索和代码搜索，我们对比搜索引擎的依据是，其反馈的检索结果中前十个条目的正确率。

统计结果展示于下表 2-1。

表 2-1 针对部分专业性问题的高知名度搜索引擎表现对比

	百度	必应	Google
路由器文件系统	6	8	10
panda重放功能	0	3	1
专业性代码1	1	5	4
专业性代码2	0	6	3

由表中可见。较其他专业性问题而言，基本无二义性的专业词汇的搜索，如“路由器文件系统”，搜索引擎表现都相对良好；针对存在二义性的专业短语的搜索，如“panda 重放功能”，搜索引擎表现都普遍较差；针对代码的搜索，搜索引擎表现介于有无二义性词汇、短语搜索之间，不过仍普遍较差。

另外，对搜索引擎进行横向对比发现，综合表现最好的是必应搜索，而最差的是百度搜索。不过，百度搜索正是我们日常搜索专业问题比较倾向的搜索引擎。

三． 系统功能与优势

1. 系统功能
   软件开发专业性短语检索功能，代码检索功能，高级搜索功能。
2. 系统优势

2.1. 提供软件开发领域专业信息检索服务

我们将用户聚焦为软件开发人员，从软件开发人员评价较高的活跃的专业平台、论坛中获得检索文档库，尽可能使检索的正确率较其他搜索引擎有显著的提高。通过智能的分词方式解决专业词汇检索的二义性问题。同时，我们利用更加高效、现代的技术和轻量级的模块尽可能使信息检索的等待时间较其他搜索引擎有提高。

2.2. 支持以代码为主体的搜索形式

UDIAB 鼓励用户通过代码进行搜索，为代码信息设计特殊的分词方式，使通过代码搜索变得更加智能。

四． 开发内容和技术支撑

1. 开发内容

UDIAB 搜索引擎开发包含以下四个组件——WEB 端、后端、爬虫、索引器四部分。下表 4-1 中详细说明了开发所需要实现的各组件的功能和组件内部各模块的功能。

表 4-1 UDIAB 主要组件与组件内模块信息

主要组件	WEB端	后端	爬虫	索引器
组件功能	负责通过网页与用户交互	实现主要功能逻辑，传送用户需要的数据	爬取信息检索所需文档	负责建立和完善索引数据结构，负责根据查询返回检索信息
组件模块功能	**搜索查询：**搜索框接收用户的输入。	**通信模块：**与用户进行数据交互。	**爬取模块：**爬取特定专业网站的内容。	**分词模块：**对爬虫爬取的文档数据进行分词处理。
	搜索建议：根据用户在搜索框中的输入提供包含搜索建议的下拉框。	**分词模块：**将用户在WEB端的输入的搜索信息进行分词。	**定时模块：**定时启动爬取模块以更新文档库。	**索引模块：**对分词模块处理后的数据根据词项建立索引。
	高级搜索：用户可设置“根据热度查询”等高级搜索功能	**搜索查询模块：**利用索引器提供的接口完成信息的检索。	**存储模块：**分析爬取模块得到的文档数据，将结构化的数据存储到硬盘中。	**存储模块：**将索引模块建立的索引用合适方式存储。

2. 技术支撑

2.1. 技术概况

下表 4-2 展示了 UDIAB 搜索引擎的四个组件分别使用的技术。

表 4-2 UDIAB 主要组件涉及技术

主要组件	WEB端	后端	爬虫	索引器
所含技术	TypeScript语言	Rust语言	Python语言	Rust语言
	React前端框架	Actix服务器框架	Scrapy爬虫框架	Tantivy搜索框架
	Ant Design图形库	jieba分词系统		jieba分词系统

2.2. 技术选型

2.2.1. TypeScript

TypeScript 由 Microsoft 主导开发，是 WEB 编程语言也是静态类型语言，在 Github 拥有富有活力的开源社区，被大多 WEB 框架支持。

其优于 JavaScript 的地方在于，TypeScript 的静态类型系统使开发者可以在编译期检查所有类型错误，不会像 JavaScript 一样无法通过静态分析检查类型，使开发和维护变得十分简单同时增加了网页的稳定性和安全性。

2.2.2. React

React 由 Facebook 主导开发，是一个用于构建用户界面的 JavaScript 库，同时支持 TypeScript。

React 框架的优势在于，与传统的 JQuery 不同，React 通过声明式的 JSX 文件，可以极其方便地操纵 DOM 元素；并且 React 运行时通过高效的虚拟 DOM 差异分析算法，极大降低了渲染的成本。

2.2.3. Ant Design

Ant Design 图形库由阿里主导开发，提供大量 WEB 前端组件，与 React 高度集成，支持 TypeScript。与 bootstrap 相比，它的图形更加的美观。

2.2.4. Rust 语言

Rust 语言由 Mozilla 主导开发，是一个静态强类型语言，以安全性、高效性、成熟性著称。Rust 语言的后端为 LLVM，通过零成本抽象，Rust 能够达到极高的效率，在某些测试中其性能可与 C 语言媲美远超 Python。

2.2.5. Actix

Actix 是一个基于 Rust 语言的效率极高的服务器框架。Actix 底层为 Tokio 异步框架，使用了 Rust 提供的协程功能，保证了它的效率性。在 TechEmpower 公布的服务器框架性能排名中一直位列前 5 名。

2.2.6. Jieba

Jieba 分词系统是目前使用最为广泛的分词系统，提供 Rust 语言版本。Jieba 分词具有对中文文本进行分词、词性标注、关键词抽取等功能。

2.2.7. Scrapy

Scrapy 是基于 Python 的用于爬取网站数据、提取结构性数据的应用框架。是目前使用最为广泛的爬虫框架。

2.2.8. Tantivy

Tantivy 是受 Apache Lucene 启发并以 Rust 编写的全文本搜索引擎库。

五． 开发进度安排

1. 爬虫组件。见下表 5-1。

表 5-1 爬虫组件开发进度安排

组件	进度安排		
	10.1 ~ 10.17	10.17 ~ 11.17	11.17 ~ 12.17
爬虫	1. 学习爬虫框架。 2. 对拟定的平台、论坛进行爬取测试并确定爬取文档的平台、论坛。 3. 确定存储数据的格式。	1. 设计实现爬取模块和存储模块。 2. 对爬取模块和存储模块进行单元测试。	1. 设计实现定时模块。 2. 对整个爬虫组件进行功能测试。 3. 完善爬虫组件。

2. 索引器组件。见下表 5-2。

表 5-2 索引器组件开发进度安排

组件	进度安排		
	10.1 ~ 10.17	10.17 ~ 12.1	12.1 ~ 12.17
索引器	1. 学习语言、Tanvity搜索框架和Jieba分词系统的相关知识。 2. 设计针对代码的分词方式。	1. 按照分词模块->索引模块->存储模块的顺序依次进行设计实现并依次完成对应单元测试。	1. 对索引器进行整体功能测试。 2. 进行UDIAB集成测试。

3. WEB 端及后端。见下表 5-3。

表 5-3 WEB 端与后端组件开发进度安排

组件	进度安排		
	10.1 ~ 10.17	10.17 ~ 11.17	11.17 ~ 12.17
WEB端与后端	1. 设计实现WEB端搜索查询模块并进行单元测试。 2. 设计WEB端与后端交互的接口。	1. 按照通信模块->搜索建议模块->分词模块->搜索查询模块->高级搜索模块的顺序依次进行设计实现并依次完成对应单元测试。	1. 对WEB端和后端进行功能测试。 2. 进行UDIAB集成测试。

六． 开发团队

张曙 ：WEB 端与后端开发
赵宏铎 ： 索引器开发与文档编写
钱正玭 ： 爬虫开发
熊吉思汗 ： 集成测试与代码完善