**数 学 与 人 工 智 能 学 院**

**二零二五年五月**

目录

[1、 摘要 1](#_Toc12264)

[2、 项目概述 1](#_Toc15982)

[2.1项目背景： 1](#_Toc3686)

[2.2项目目标： 1](#_Toc16787)

[2.3项目范围： 2](#_Toc15811)

[3、 需求分析 2](#_Toc19262)

[3.1 功能需求: 2](#_Toc4892)

[3.2 性能需求: 3](#_Toc18243)

[3.3 用户需求: 3](#_Toc12664)

[3.4 其他需求: 4](#_Toc8456)

[4、总体设计 4](#_Toc16697)

[4.1 系统架构设计: 4](#_Toc26547)

[4.2 数据库设计: 5](#_Toc9233)

[4.3 总体模块设计: 5](#_Toc20679)

[4.4具体模块相应功能： 6](#_Toc4516)

[5、核心算法原理 12](#_Toc29888)

[5.1 算法选择: 12](#_Toc10818)

[5.2 算法详细描述： 13](#_Toc15723)

[6、详细设计与实现 18](#_Toc14425)

[6.1 模块详细设计： 18](#_Toc31748)

[7、 测试与性能评价 19](#_Toc24209)

[7.1 测试： 19](#_Toc3440)

[7.2 性能评价指标： 20](#_Toc8112)

[7.3 具体测试示例： 20](#_Toc25861)

[7.4测试结果： 21](#_Toc22201)

[7.5 可优化措施： 21](#_Toc23221)

[8、 总结与展望 22](#_Toc10466)

[8.1 项目总结： 22](#_Toc16299)

[8.2 未来展望： 22](#_Toc2129)

# 摘要

本项目是一个基于 Python 和 PyQt5 开发的综合检索与评估系统，旨在为用户提供便捷的文件检索服务，并对检索性能进行有效评估。系统具备用户认证功能，通过 SQLite轻量级数据库管理用户信息，确保系统安全性。支持多种检索算法，如关键词搜索、TF - IDF 搜索、布尔模型搜索等，能够满足不同用户的检索需求。同时，系统提供了检索性能评估模块，可计算精确率、召回率、F1 值等关键评价指标，帮助用户了解检索效果。

# 项目概述

## 2.1项目背景：

在信息爆炸的时代，快速准确地从大量文件中检索所需信息变得至关重要。现有的检索工具在功能和性能上存在一定的局限性，无法满足多样化的检索需求和对检索性能的有效评估。因此，开发一个功能丰富、性能优良的综合检索与评估系统具有重要的现实意义。

## 2.2项目目标：

·开发一个具有用户认证功能的综合检索与评估系统，确保系统的安全性和用户数据的保密性。

·实现多种检索算法，提高检索的准确性和效率，满足不同用户的检索需求。

·提供检索性能评估功能，计算精确率、召回率、F1 值的关键评价指标，帮助用户了解检索效果。

·设计友好的用户界面，提高用户体验。

## 2.3项目范围：

·涵盖用户认证模块，包括用户登录、注册、管理等功能。

·实现文件检索模块，支持多种检索算法。

·提供检索性能评估模块，计算和展示评估指标。

·设计系统主界面和各个功能页面，实现页面之间的切换和交互。

# **需求分析**

## 3.1 功能需求:

用户认证功能:

·用户可以进行登录和注册操作，系统验证用户凭据的有效性。

管理员可以对用户信息进行管理，包括添加、删除、更新用户信息和重置用户密码。

文件检索功能:

·用户可以输入关键词，选择检索算法，对指定文件列表进行检索。

·支持多种检索算法，如关键词搜索、TF - IDF 搜索、布尔模型搜索。

·显示检索结果，包括文件名、行号、关键词上下文(前后10个字符，包括空格等占位符)等信息。

检索性能评估功能:

·用户可以选择真实样本集和预测样本集文件，设置置信度阈值，进行检索性能评估。

·计算精确率、召回率、F1 值、平均响应时间和误检率等评估指标。

·显示评估指标结果和详细评估结果表格。

·支持评估结果的导出和可视化展示。

## 3.2 性能需求:

·应具备较快的响应速度，在合理的时间内完成检索和评估任务。

·系统涉及多文件io，应该考虑读写分离的io方式，避免出现数据幻读。

## 3.3 用户需求:

·用户希望系统界面简洁、操作方便，能够快速上手，使用了第三方GUI框架，实现了简单的界面交互程序。

·需要系统提供详细的检索结果和评估指标，以便了解检索效果。

# **总体设计**

## 4.1 系统架构设计:

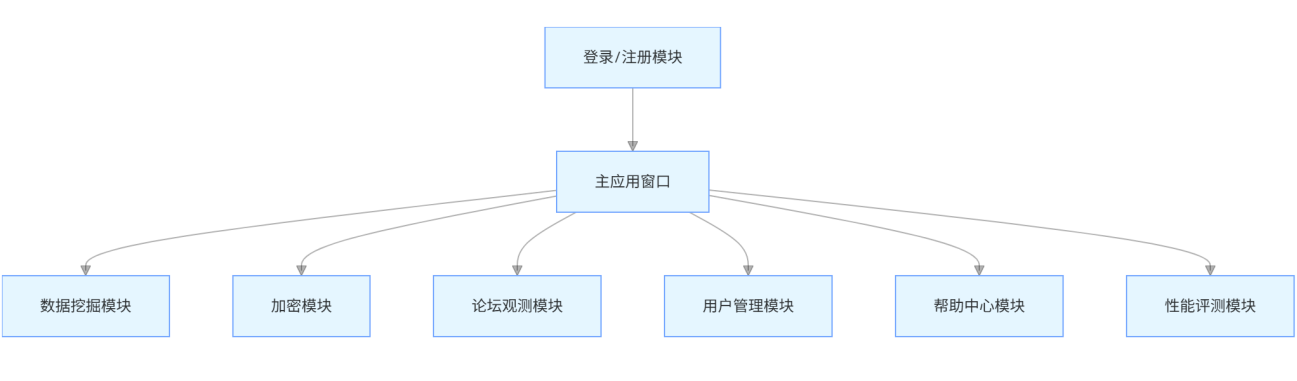
系统采用分层架构设计，主要分为表示层、业务逻辑层和数据访问层。

·表示层：使用 PyQt5 实现用户界面，包括登录页面、系统主界面、各个功能页面等，负责与用户进行交互。

·业务逻辑层：实现系统的核心业务逻辑，如用户认证、文件检索、检索性能评估等，处理用户请求并调用数据访问层进行数据操作。

·数据访问层：使用 SQLite 数据库存储用户信息，通过DatabaseManager类实现对数据库的增删改查操作。

## 4.2总体模块设计:



·用户认证模块：负责用户的登录、注册和管理，调用DatabaseManager类进行用户信息的验证和操作。

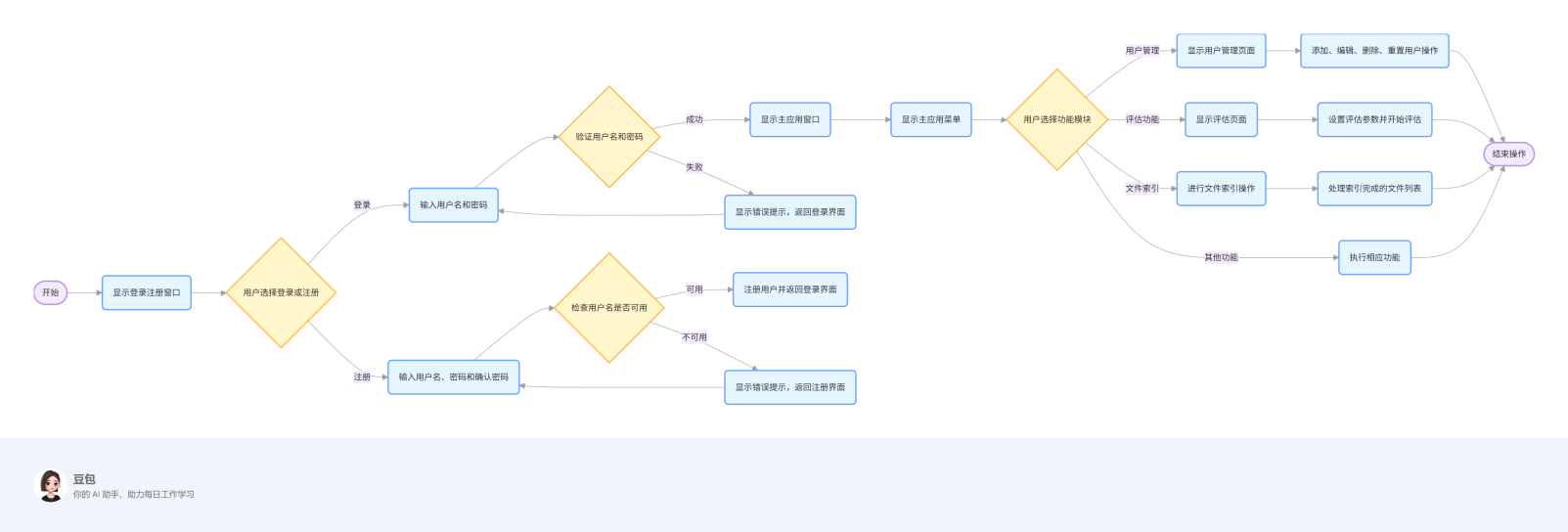
·文件检索模块：根据用户输入的关键词和选择的检索算法，对指定文件列表进行检索，使用SearchWorker类在后台线程中执行检索任务。

· 索引性能评估模块：根据用户选择的真实样本集和预测样本集文件计算评估指标，使用EvaluationWorker类在后台线程中执行评估任务。

·界面模块：设计和实现系统的各个页面，包括登录页面、系统主界面、检索页面、评估页面等，处理用户的界面交互事件。

## 4.3具体模块相应功能：

双击查看总体流程

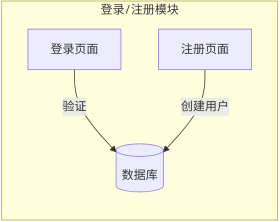


**4.4.1登录/注册模块**

用户认证系统

数据库用户管理

登录/注册界面切换



**4.4.2数据源管理模块**

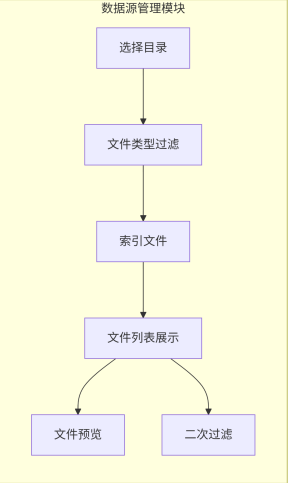
目录选择与浏览

文件类型过滤

文件索引功能

文件预览（文本/信息）

二次过滤功能



**4.4.3检索模块**

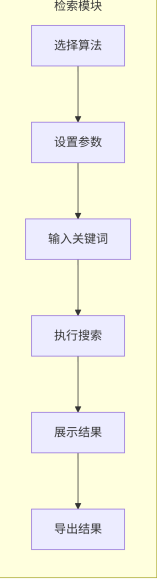
多算法支持（关键词/TF-IDF/布尔/向量空间）

参数配置（阈值/最大结果数）

关键词搜索

结果表格展示

CSV导出功能



**4.4.4动态观察模块**

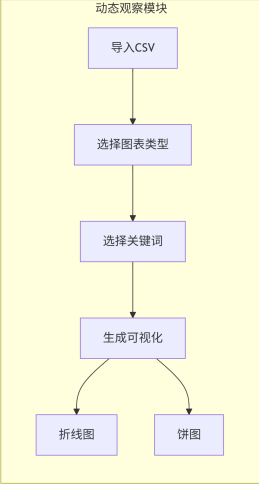
CSV数据导入

图表类型选择（折线图/饼图）

关键词分析

交互式可视化

图表保存功能



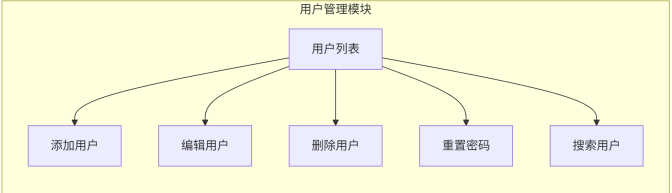
**4.4.5用户管理模块**

用户列表展示

用户增删改查

密码重置

用户搜索功能

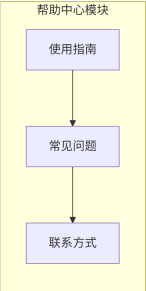


**4.4.6帮助中心模块**

使用指南文档

常见问题解答

开发者联系方式



**4.4.7性能评估模块**

样本集导入（真实/预测）

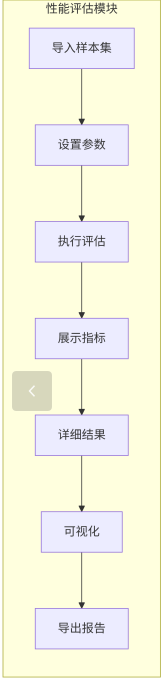
评估参数设置

核心指标计算（精确率/召回率/F1值）

响应时间分析

可视化报告

评估结果导出



# 5、核心算法原理

## 5.1 算法选择:

·关键词搜索算法：简单直接，适用于快速查找包含特定关键词的文本，能够满足基本的检索需求。

·TF - IDF 搜索算法：考虑了词在文档中的重要性，能够提高检索的准确性，适用于对检索结果质量要求较高的场景。

·布尔模型搜索算法：支持复杂的逻辑查询，能够满足用户对检索结果的精确筛选需求。

·向量空间模型搜索算法：读取文档并进行中英文分词，构建基于 TF-IDF 的文档向量和关键词权重查询向量，计算余弦相似度评估匹配度，最后提取关键词上下文作为匹配结果。

## 5.2 算法详细描述：

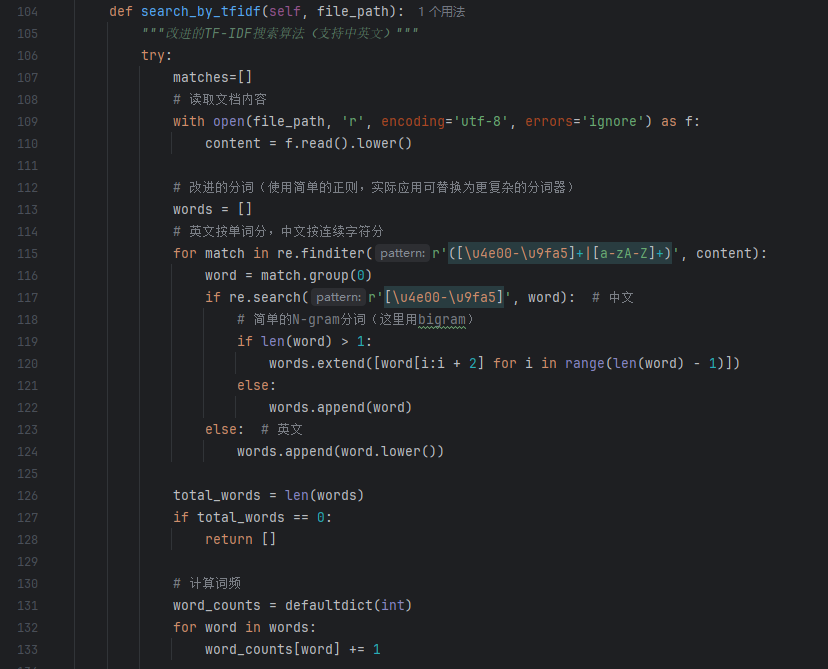
关键词搜索算法：通过正则表达式匹配文本中的关键词，支持中英文单词边界，对匹配结果进行高亮显示和上下文提取。

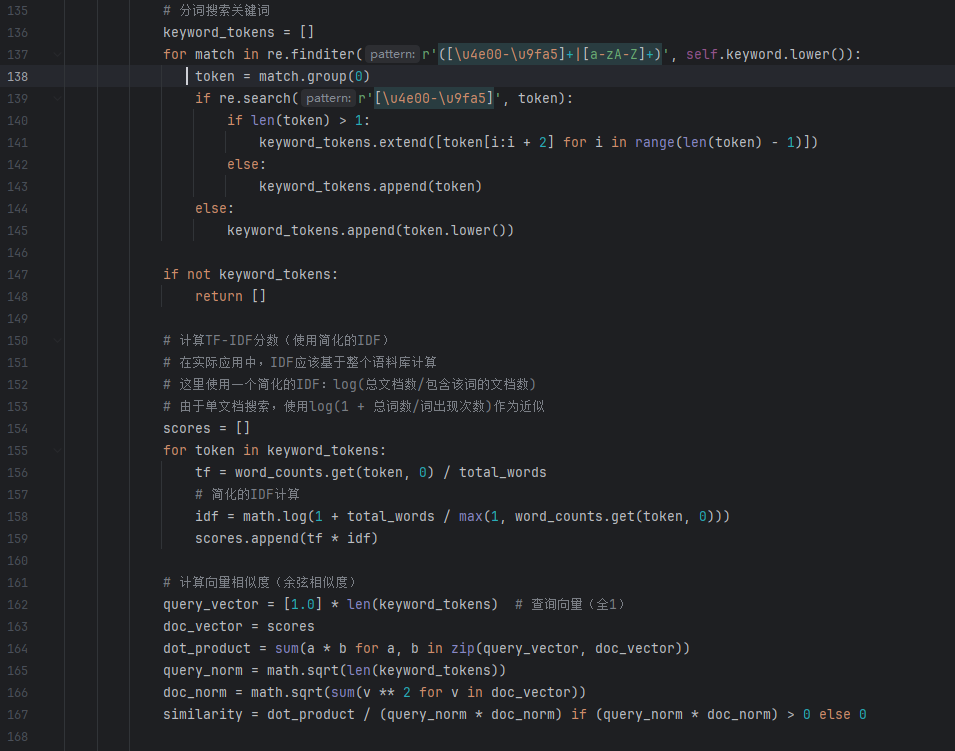
关键词搜索算法代码：

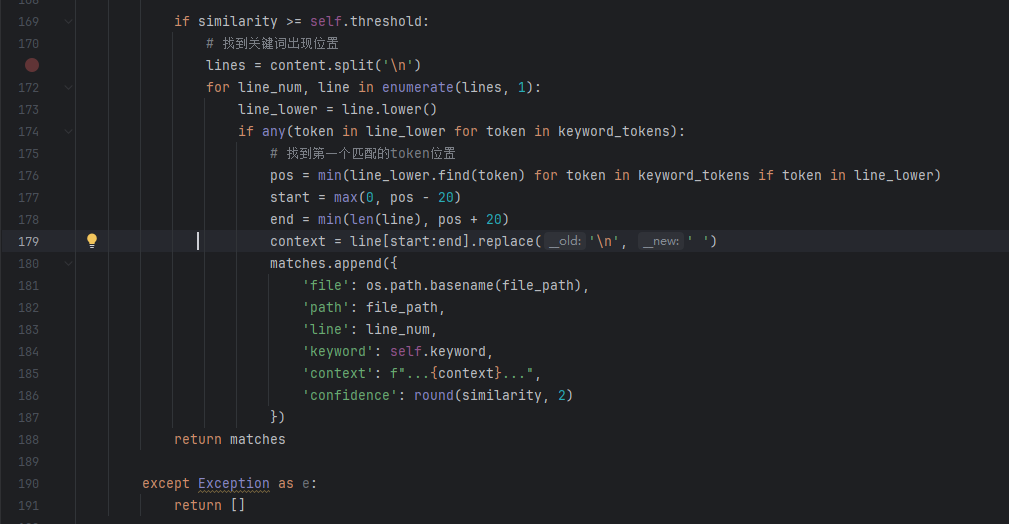


TF - IDF 搜索算法：计算词的 TF - IDF 分数，通过向量相似度（余弦相似度）判断文档与查询的相关性，对相关性较高的文档进行匹配结果提取。

TF - IDF 搜索算法代码：

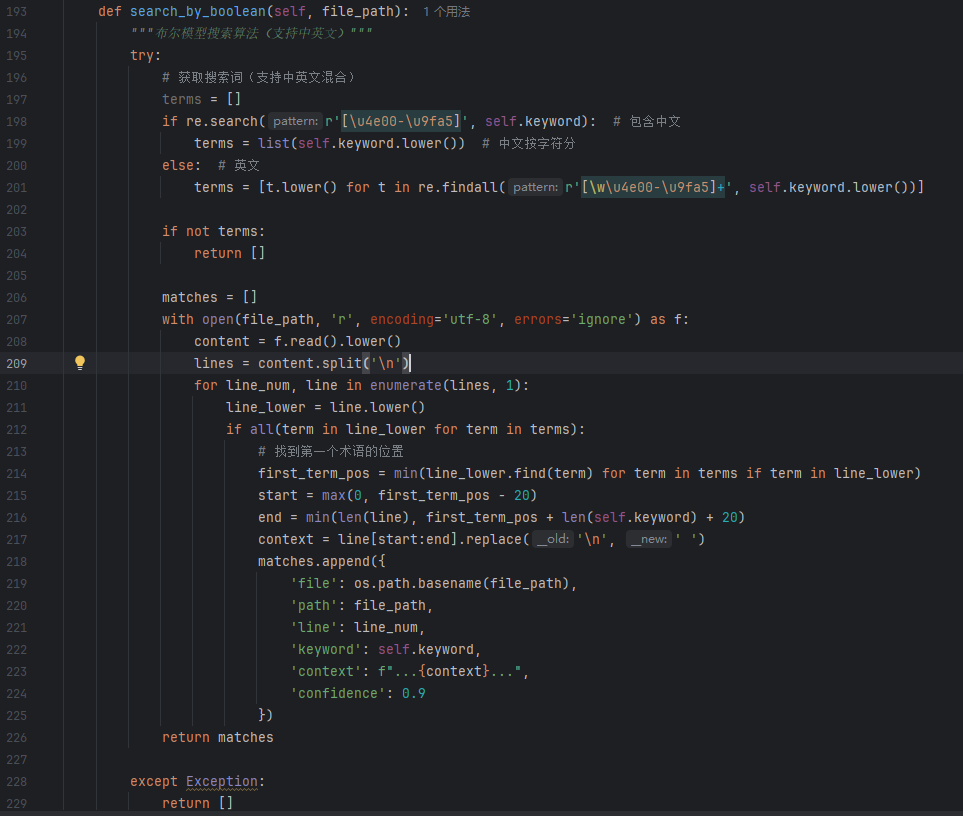






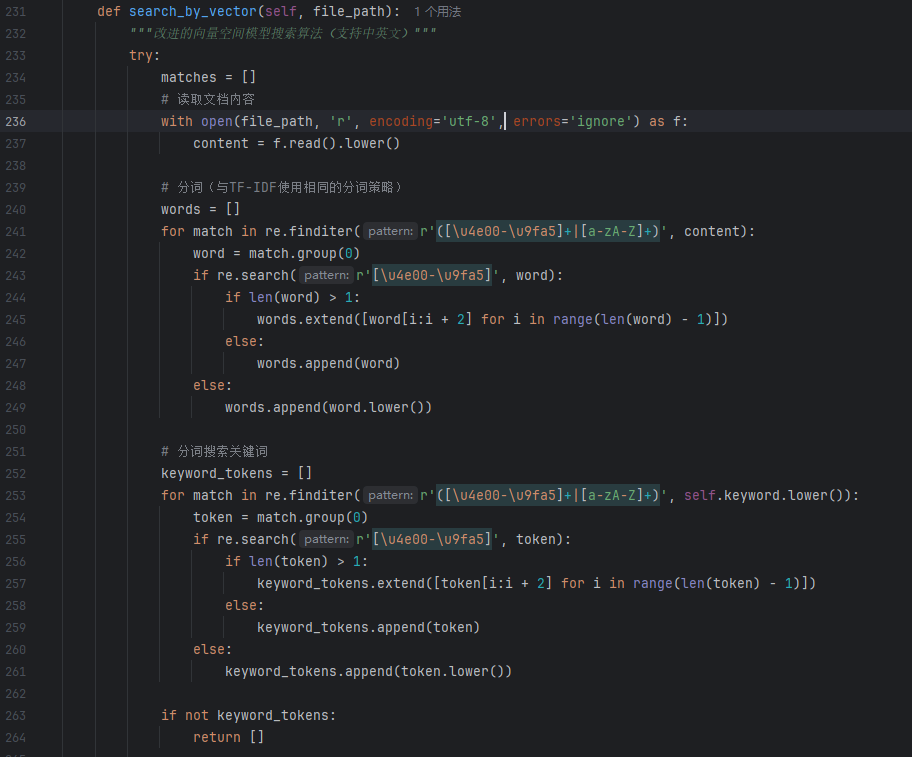
布尔模型搜索算法：根据用户输入的关键词构建布尔表达式，对文本进行匹配，返回满足布尔条件的匹配结果。

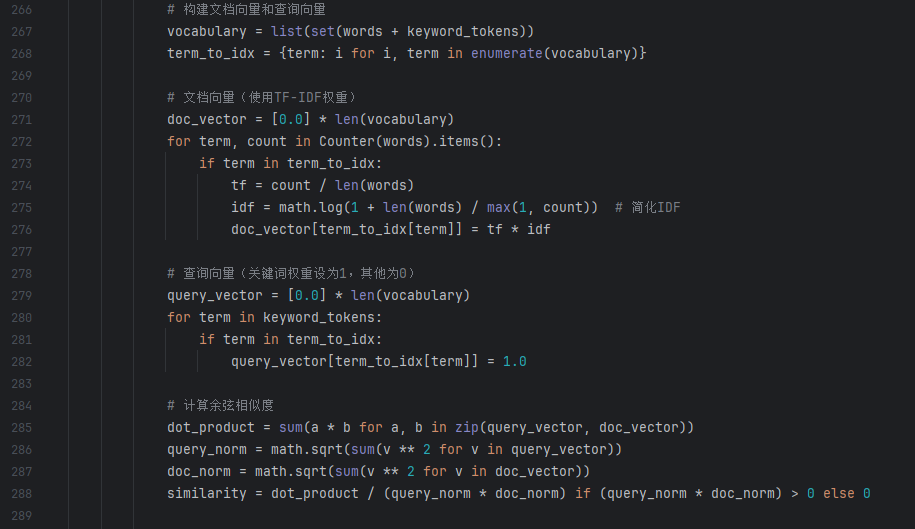
布尔模型搜索算法代码：

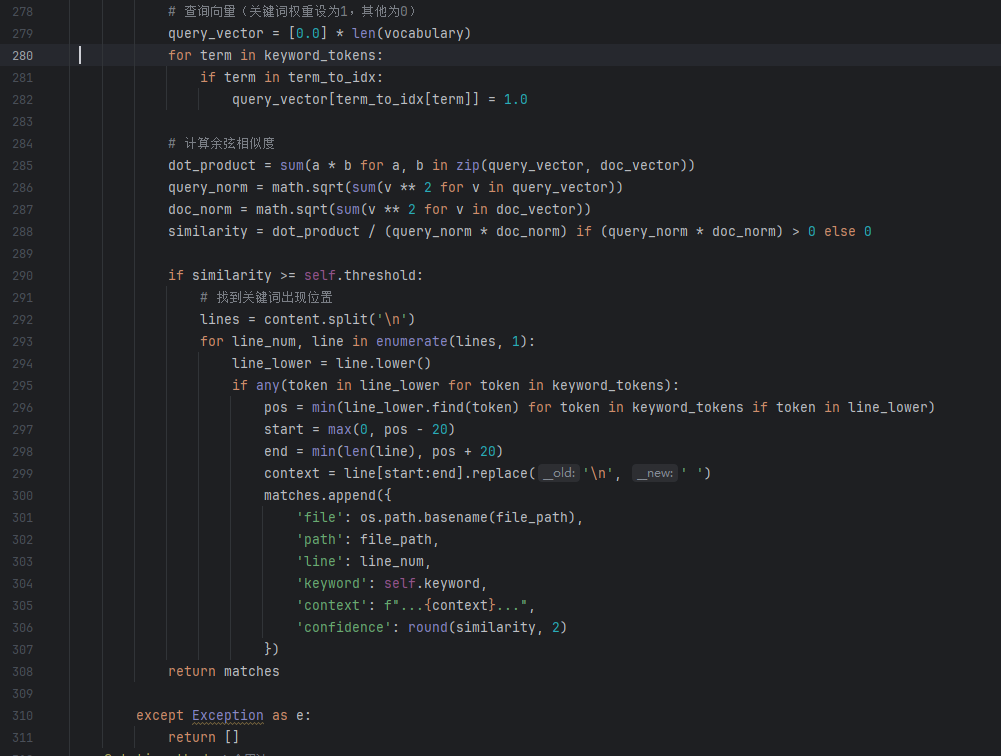


向量空间模型搜索算法：读取文档并进行中英文分词，构建基于 TF-IDF 的文档向量和关键词权重查询向量，计算余弦相似度评估匹配度，最后提取关键词上下文作为匹配结果。

向量空间模型搜索算法代码：







# **6、详细设计与实现**

## 6.1用户认证模块：

LoginPage类：实现用户登录界面，处理用户登录请求，调用DatabaseManager类的validate\_user方法验证用户凭据。

RegisterPage类：实现用户注册界面，处理用户注册请求，调用DatabaseManager类的register\_user方法注册新用户。

DatabaseManager类：负责与 SQLite 数据库进行交互，实现用户信息的增删改查操作。

## 6.2文件检索模块：

SearchWorker类：在后台线程中执行文件检索任务，根据用户选择的检索算法对指定文件列表进行检索，发出检索进度和结果信号。

SearchPage类：实现文件检索界面，接收用户输入的关键词和选择的检索算法，启动SearchWorker线程进行检索，显示检索结果。

## 6.3检索性能评估模块：

EvaluationWorker类：在后台线程中执行检索性能评估任务，加载真实样本集和预测样本集文件，计算评估指标，发出评估进度和结果信号。

EvaluationPage类：实现检索性能评估界面，接收用户选择的真实样本集和预测样本集文件，设置置信度阈值，启动EvaluationWorker线程进行评估，显示评估指标结果和详细评估结果表格。

# **测试与性能评价**

## 7.1 测试模块：

**功能测试：**对系统的各个功能模块进行测试，包括用户认证、文件检索、检索性能评估等功能，确保功能的正确性和稳定性。

**性能测试：**测试系统在不同负载情况下的响应时间和吞吐量，评估系统的性能指标是否满足需求。

**安全测试：**对系统的安全性进行测试，包括用户信息的加密存储和传输、防止 SQL 注入等，确保系统的安全性。

## 7.2 性能评价指标：

精确率：反映检索结果中正确结果的比例，精确率越高，说明检索结果的准确性越好。

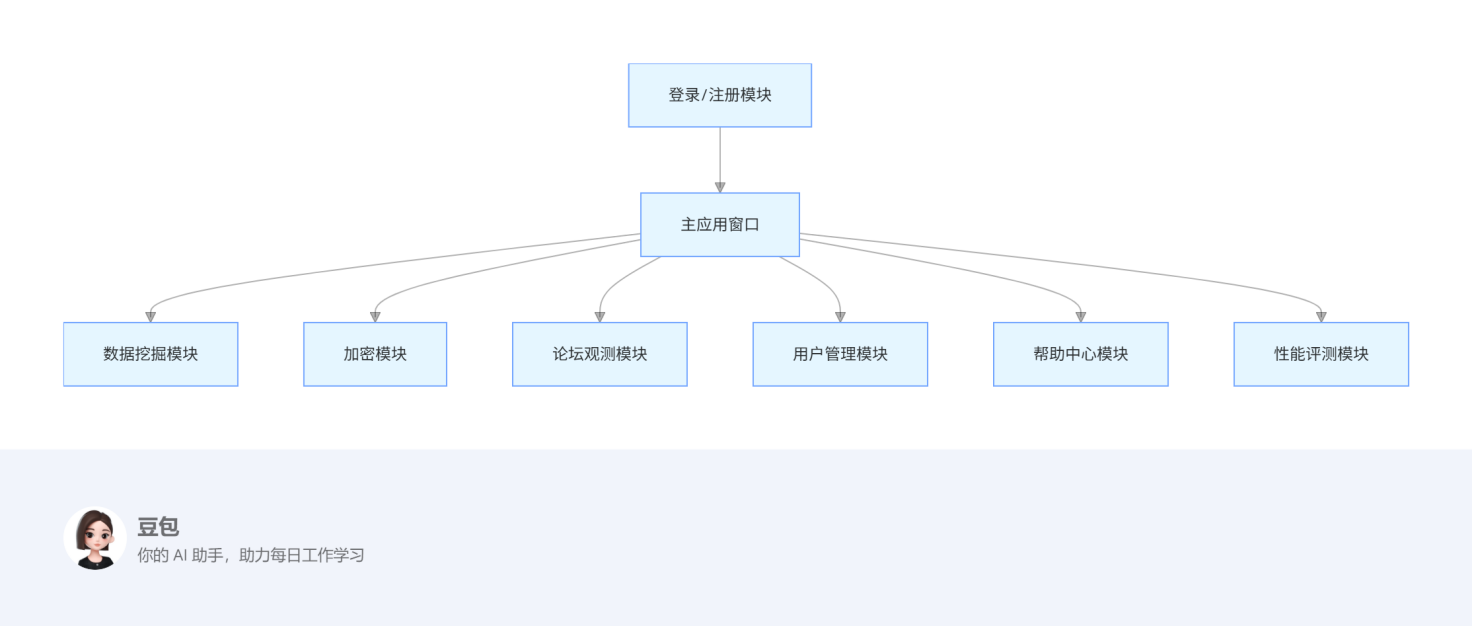
召回率：反映检索结果中包含的相关结果的比例，召回率越高，说明检索系统能够找到更多的相关结果。

F1 值：综合考虑精确率和召回率的指标，F1 值越高，说明检索系统的性能越好。

平均响应时间：反映系统处理检索请求的平均时间，平均响应时间越短，说明系统的响应速度越快。

误检率：反映检索结果中误检结果的比例，误检率越低，说明检索系统的准确性越好。

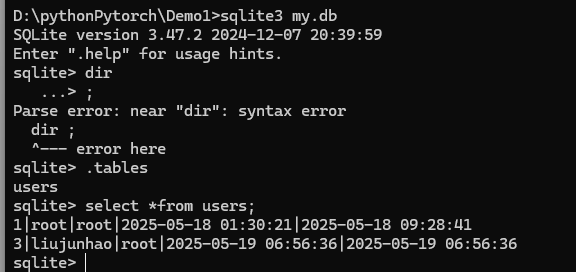
## 7.3 具体测试示例：



**7.3.1功能测试：**

### 登录模块测试：

数据库user表中默认有两个用户：



启动程序登录，输入账号root，密码root（这里为了方便，设置了的字符大小写转换，任意输入大小写字符都会按照小写录入数据库）

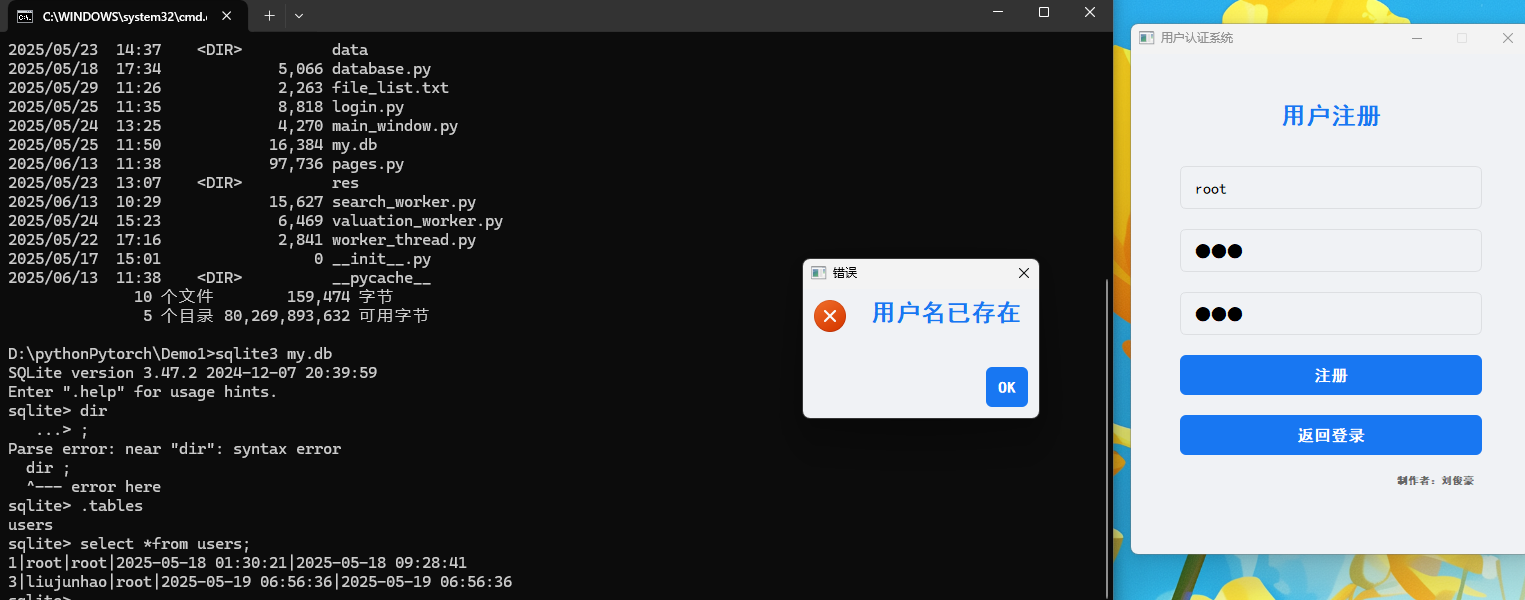


登录成功后跳转到第一个模块

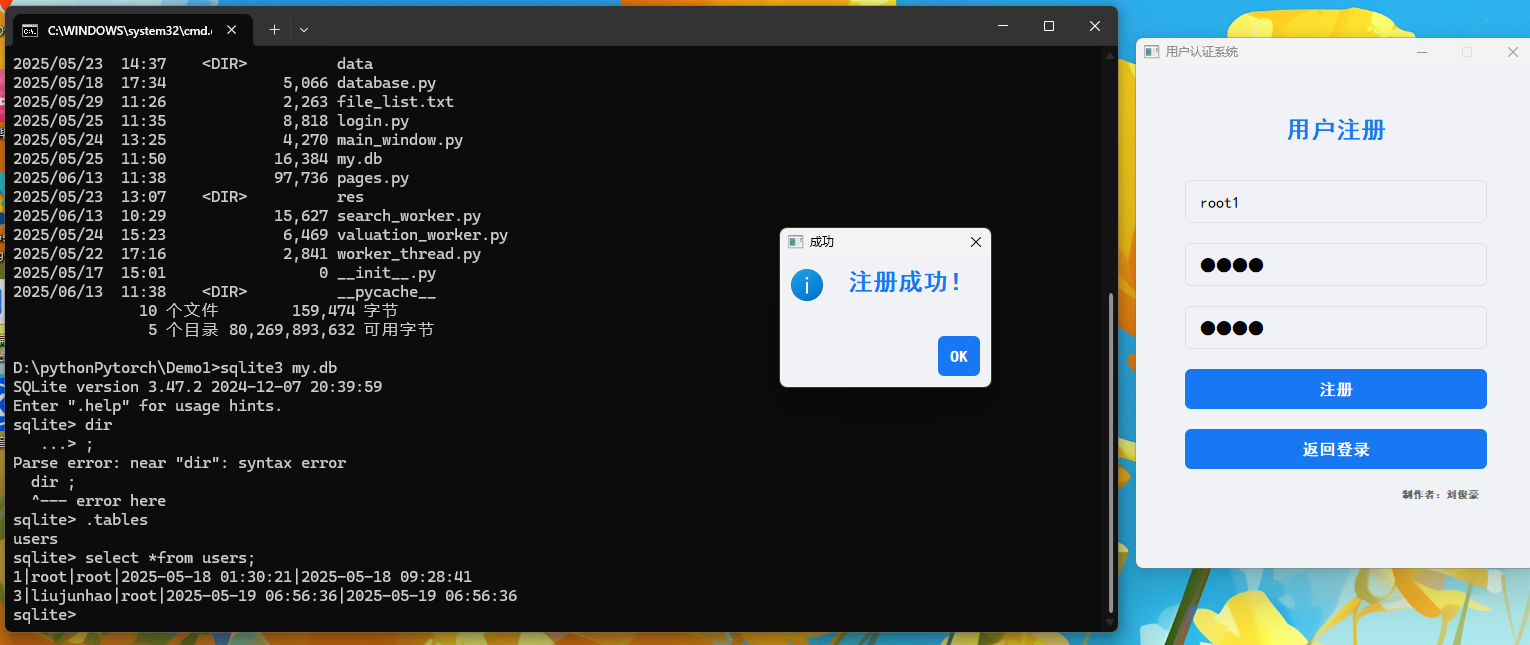


### 注册模块测试：

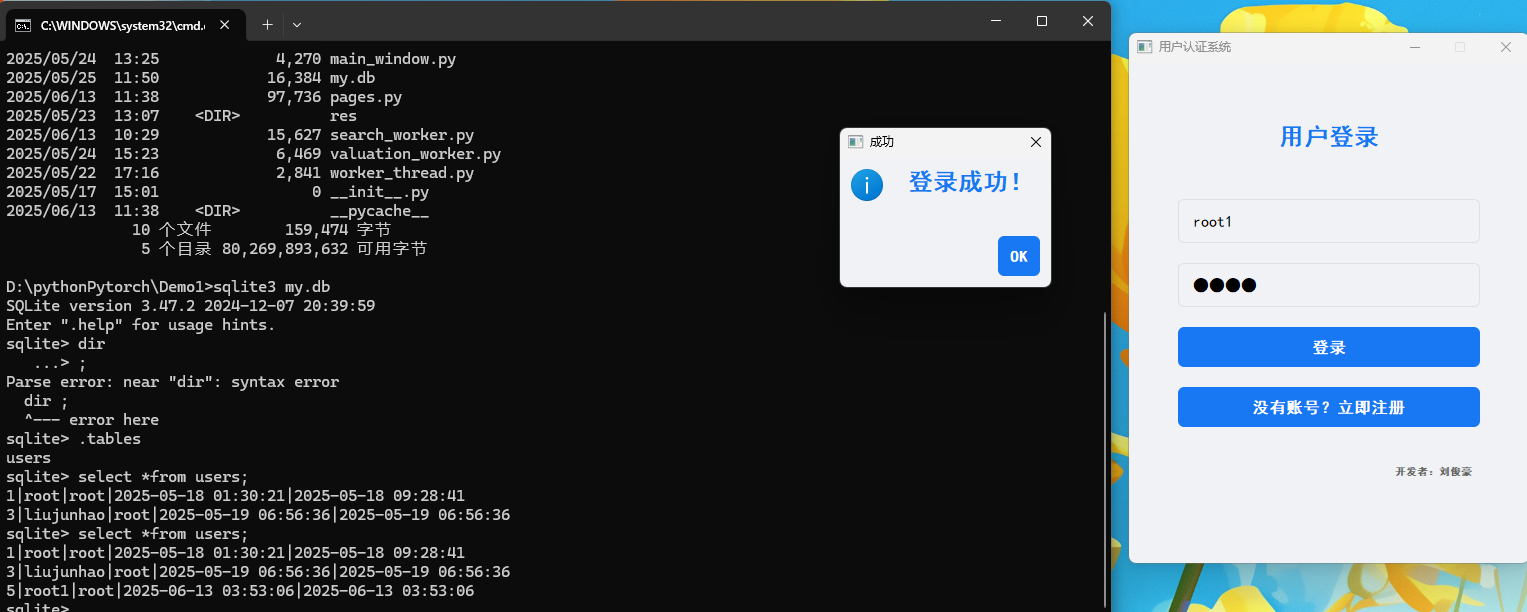
用户名唯一性测试，表中已经存在，无法创建同为root的新用户



限制注册的约束条件只有unique，因此新建表中没有的root1，可以成功创建



使用root1用户登录

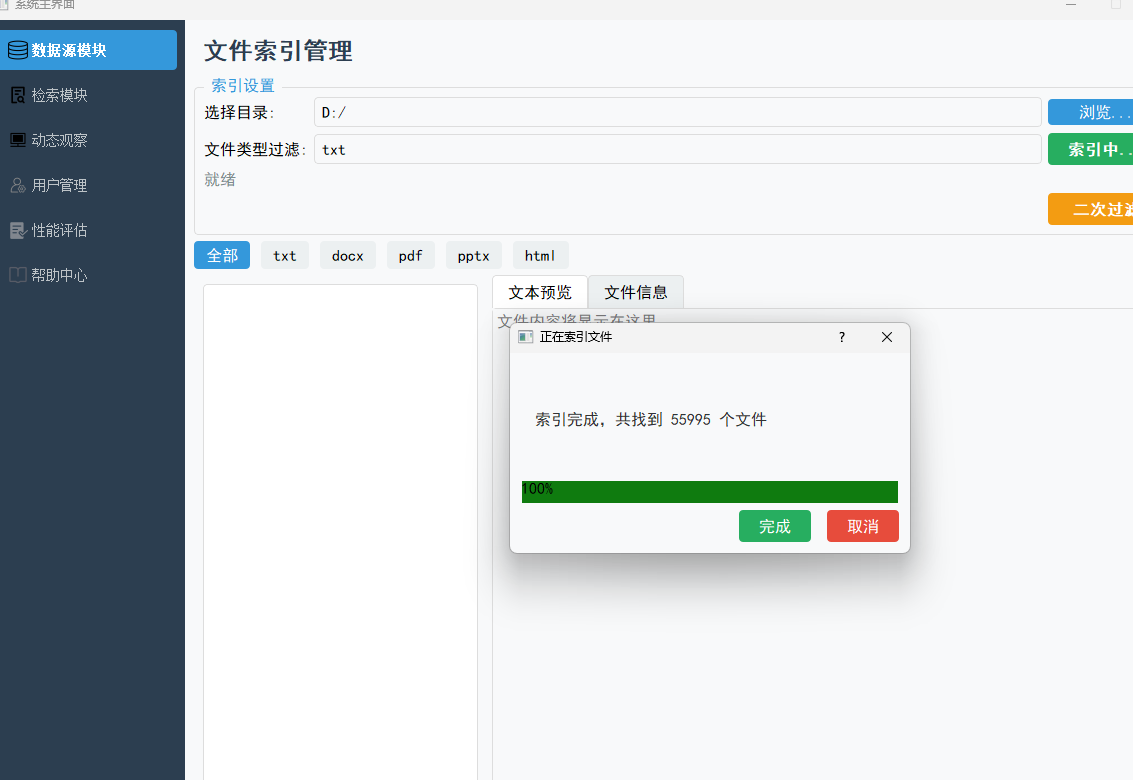


查看表中数据是否一致

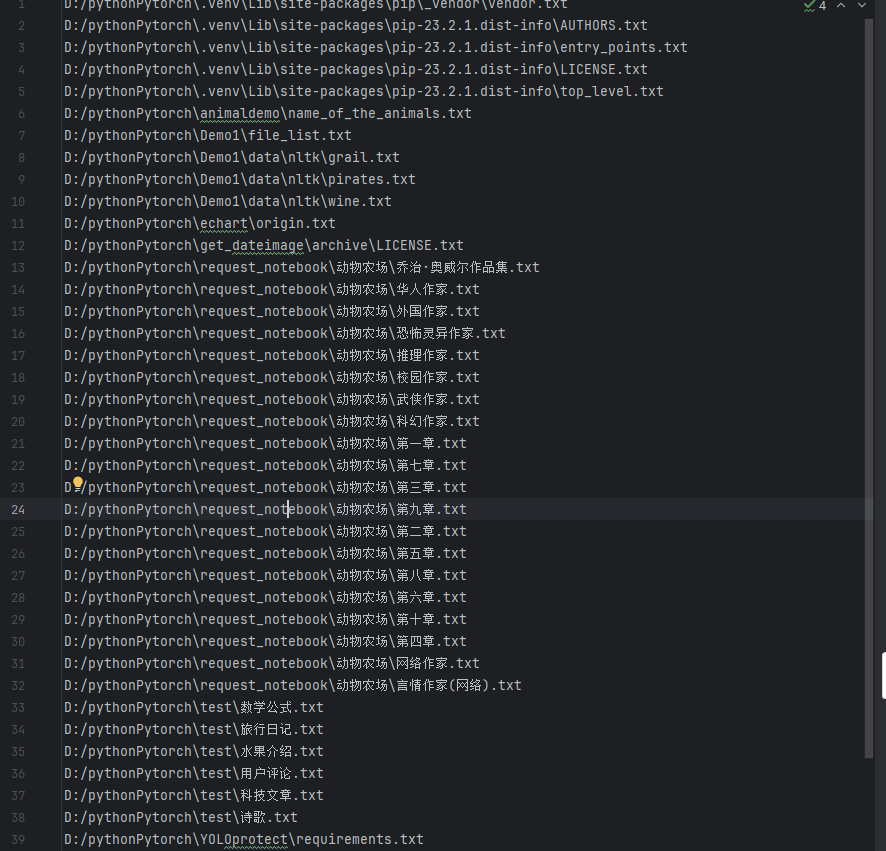


### 数据源模块测试：

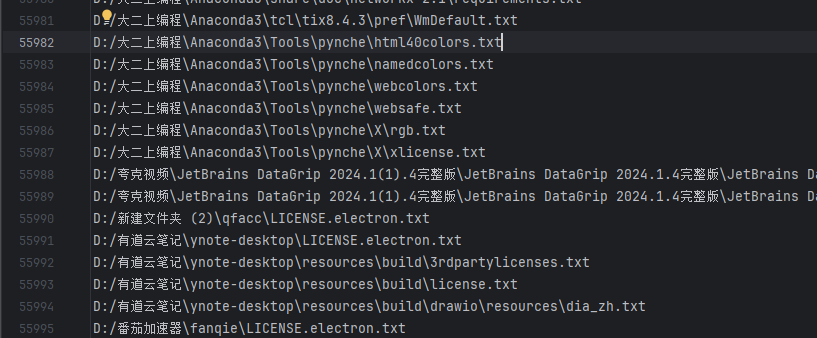
过滤索引D盘下的txt文件,子线程正常执行没有影响到主线程运行

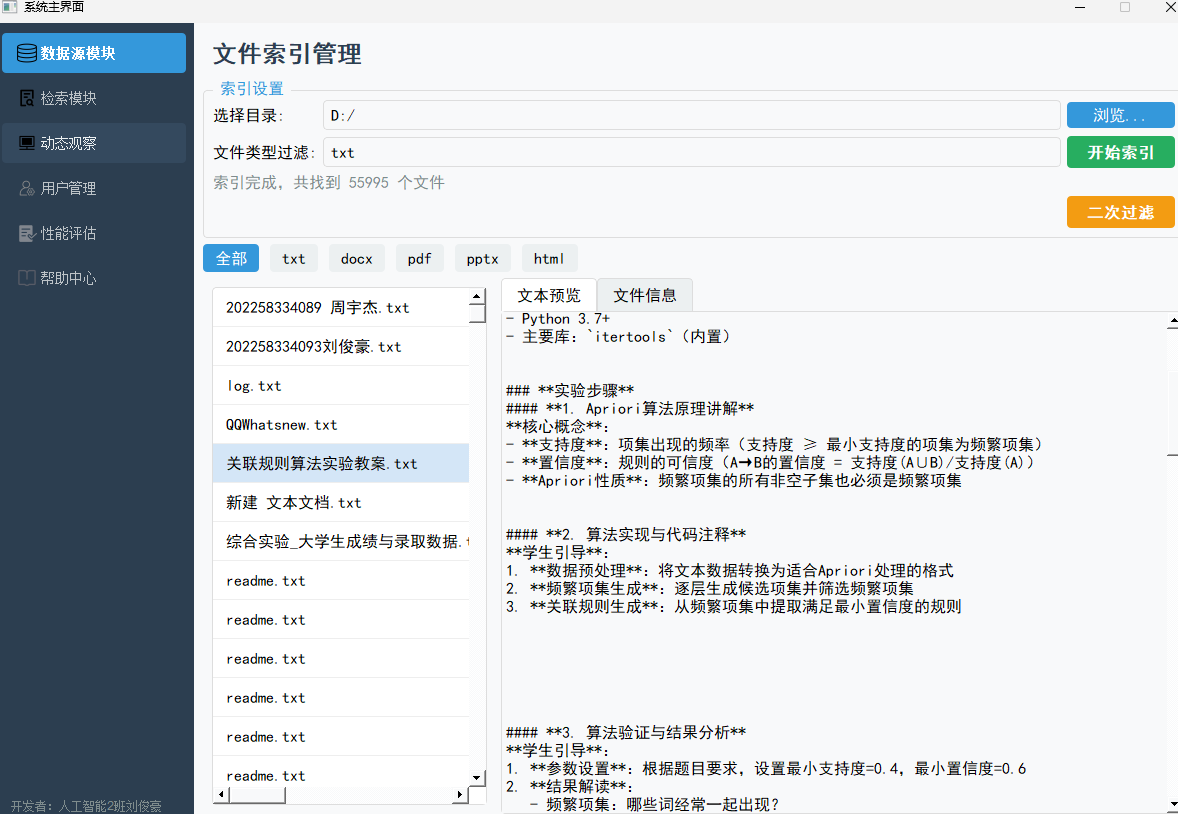


完成索引后的txt文件路径会写入到file\_list.txt中，这里索引完成后，file\_list.txt中只有40多个文件，是因为我为了保护数据一致性，设置了必须点击《完成》按钮才会开始写入上面的55995个txt文件，如果点击取消不会改变file\_list.txt的内容，类似数据库的回滚，我只有提交事务和不提交事务，。



点击确定后写入

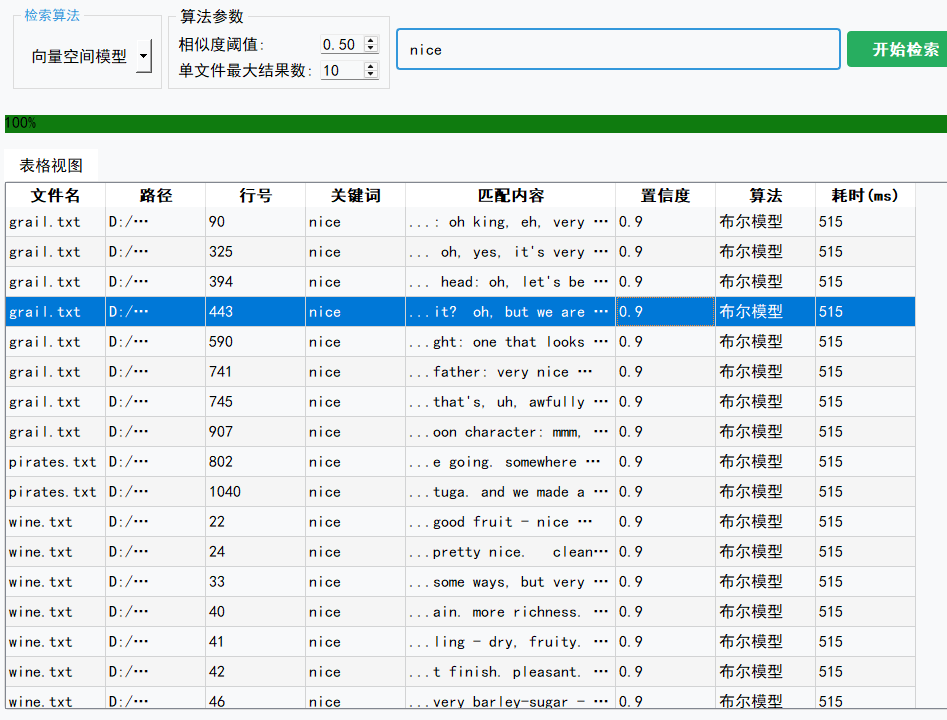




### 检索模块测试：









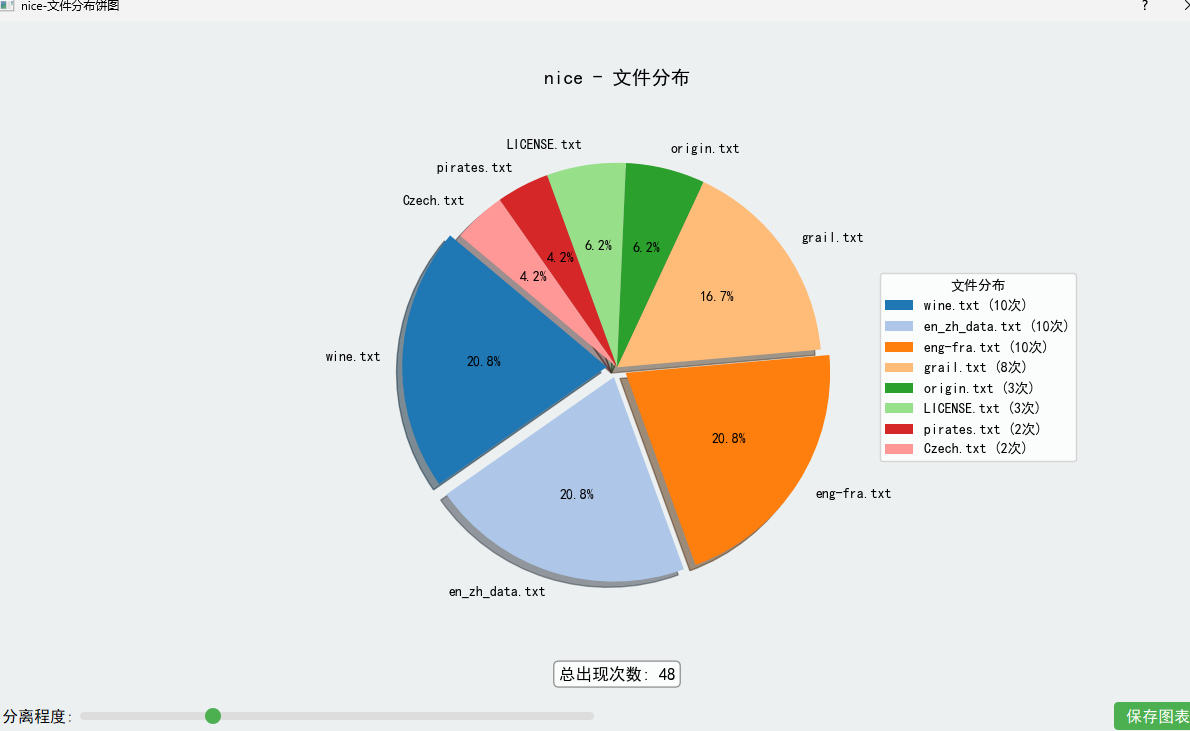
### 可视化模块测试：



动态折线图：序号为x轴，置信度为y轴，采用字典提取出csv文件中字段，建立坐标x，y轴



饼图：读取csv文件中的字段，统计对应文件数量，



### 评估模块测试：

## 7.4测试结果：

功能测试：系统的各个功能模块均能正常工作，未发现明显问题。

性能测试：在合理的负载情况下，系统的响应时间和吞吐量满足需求，但在高负载情况下，系统的性能有所下降，需要进一步优化。

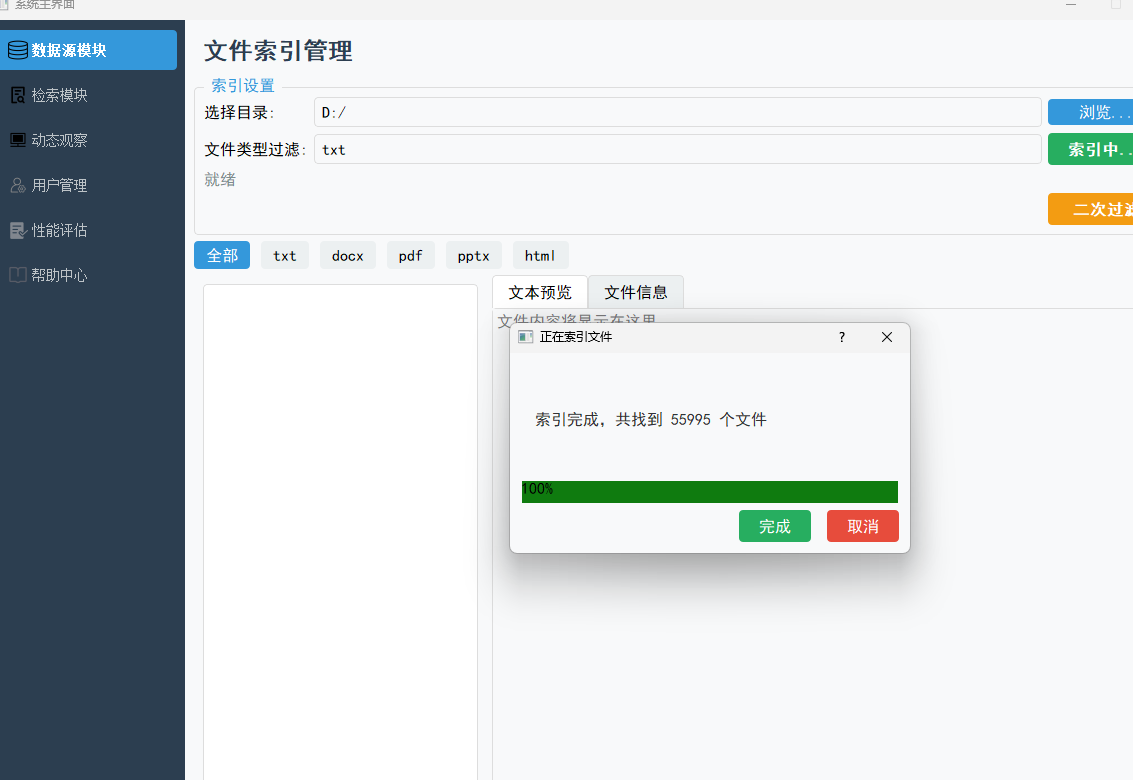
# **总结与展望**

## 8.1 项目总结：

本项目成功开发了一个基于 Python 和 PyQt5 的综合检索与评估系统，实现了用户认证、文件检索、检索性能评估等核心功能。系统采用分层架构设计，具有良好的可扩展性和可维护性。通过测试和性能评价，系统的功能和性能基本满足需求，但在高负载情况下仍存在一定的性能瓶颈，需要进一步优化。

## 8.2 未来展望：

**性能优化：**检索海量数据文件耗时问题，索引D盘时使用的只用了一个子线程Indexwoker，如果采用线程池轮询处理文本io操作，如将这55595个文件分摊给线程池（4-8个），能大幅提高效率。



**检索功能拓展：**如模糊检索、语义检索等，提高检索的准确性和智能化水平。

**用户信息安全加密：**用户信息的加密使用，对称加密（例如AES）对用户信息进行加密存储。

**修改权限设置：**可以在用户之间进行加密隔离，为每个数据请求和操作设置权限检查，只有经过授权的用户或管理员（如root）才能对敏感数据进行修改或访问

**不一致同名文件：**同名文件（我写在界面的同名文件，如readme，其实是存在不同路径的文件）可能会影响用户的交互，待优化。

