|  |  |
| --- | --- |
|  | **第十五届全国大学生软件创新大赛** |
| **文档编号：SWC2022-356-秃头预备役** |

****

**魔法笔记**

**Magic Notes**

**项目开发文档**

**Version: [Version Number]**



**秃头预备役**

**2023-2-28**

**All Rights Reserved**

目录

[1 项目概述 1](#_Toc21360)

[1.1 项目背景 1](#_Toc6210)

[1.2 项目定位 2](#_Toc11437)

[1.2.1 应用场景 2](#_Toc30843)

[1.2.2 目标人群 3](#_Toc9430)

[1.3 项目方案 4](#_Toc12194)

[1.4 项目目标 4](#_Toc25108)

[1.5 项目价值 4](#_Toc2794)

[2 开发计划 6](#_Toc26066)

[2.1 最终呈现形式 6](#_Toc21260)

[2.2 主要功能描述 7](#_Toc22263)

[2.2.1 信息格式转换模块 7](#_Toc22558)

[2.2.2 要点总结模块 7](#_Toc7906)

[2.2.3 搜索推荐模块 7](#_Toc22810)

[2.3 运行环境 8](#_Toc30551)

[2.4 验收标准 8](#_Toc3230)

[2.5 关键问题 9](#_Toc14527)

[2.6 进度安排 9](#_Toc31918)

[2.7 开发预算 10](#_Toc29536)

[3 可行性分析 11](#_Toc13692)

[3.1 技术可行性分析 11](#_Toc29572)

[3.1.1 信息多模态转化 11](#_Toc1926)

[3.1.2 文本摘要 11](#_Toc4005)

[3.1.3 要点结构可视化 12](#_Toc24455)

[3.1.4 混合加权推荐策略 12](#_Toc22801)

[3.2 资源可行性分析 13](#_Toc2832)

[3.2.1 软硬件资源 13](#_Toc19116)

[3.2.2 人力资源 13](#_Toc14017)

[3.2.3 其他资源 13](#_Toc11954)

[3.3 市场可行性分析 13](#_Toc7039)

[3.3.1 市场需求分析 13](#_Toc31661)

[3.3.2 竞品分析 13](#_Toc14868)

[3.3.3 自我分析 15](#_Toc10748)

[4 需求分析 16](#_Toc16670)

[4.1 数据需求 16](#_Toc20119)

[4.1.1 静态数据 16](#_Toc22336)

[4.1.2 动态数据 16](#_Toc27896)

[4.1.3 数据词典 17](#_Toc5092)

[4.1.4 数据采集 20](#_Toc16724)

[4.2 功能需求 20](#_Toc792)

[4.2.1 信息格式转换模块 21](#_Toc3460)

[4.2.2 笔记总结模块 22](#_Toc13641)

[4.2.3 搜索推荐模块 24](#_Toc1614)

[4.3 性能需求 25](#_Toc8686)

[4.3.1 时间特性 25](#_Toc11253)

[4.3.2 适应性 26](#_Toc7257)

[4.4 界面需求 27](#_Toc28722)

[4.5 接口需求 27](#_Toc21580)

[4.5.1 硬件接口 27](#_Toc23809)

[4.5.2 软件接口 27](#_Toc26631)

[4.6 其他需求 27](#_Toc19188)

[4.6.1 可使用性 27](#_Toc31717)

[4.6.2 安全性 27](#_Toc27357)

[4.6.3 可维护性 28](#_Toc11203)

[4.6.4 可移植性 28](#_Toc17939)

[5 概要设计 30](#_Toc3274)

[5.1 处理流程 30](#_Toc12706)

[5.2 总体结构设计 30](#_Toc27215)

[5.3 功能设计 30](#_Toc21534)

[5.4 数据流转设计 30](#_Toc32481)

[5.5 用户界面设计 30](#_Toc6901)

[5.6 数据结构设计 30](#_Toc13006)

[5.7 接口设计 30](#_Toc9079)

[5.7.1 外部接口 30](#_Toc29588)

[5.7.2 内部接口 30](#_Toc31975)

[5.8 错误/异常处理设计 30](#_Toc5150)

[5.8.1 错误/异常输出信息 30](#_Toc5207)

[5.8.2 错误/异常处理对策 30](#_Toc25542)

[5.9 系统配置策略 30](#_Toc24602)

[5.10 系统部署方案 30](#_Toc23143)

[5.11 跨端应用架构设计 30](#_Toc20498)

[5.12 其他相关技术与方案 30](#_Toc27147)

[6 数据库设计 32](#_Toc28966)

[7 手机端侧部署设计 33](#_Toc26012)

[7.1 手机环境需求 33](#_Toc32024)

[8 详细设计 34](#_Toc8506)

[8.1 \*\*功能模块 34](#_Toc21058)

[8.1.1 功能描述 34](#_Toc3916)

[8.1.2 性能描述 34](#_Toc11896)

[8.1.3 输入 34](#_Toc14855)

[8.1.4 输出 34](#_Toc22615)

[8.1.5 程序逻辑 34](#_Toc1505)

[8.1.6 限制条件 34](#_Toc16769)

[8.2 \*\*功能模块 34](#_Toc26841)

文档修订历史

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **序号** | **修订原因** | **版本号** | **作者** | **修订日期** | **备注** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

# 项目概述

## 项目背景

1. **爆炸式增长的资源**

近年来，随着固网宽带、移动互联网、物联网等的快速发展，数据正呈爆炸式增长。相应人们对于资源处理的需求越发强烈，需要更加便捷、兼容的途径来接收源源不断涌来的大量资源信息。面对这样爆炸式增长的各种形式的资源，我们提出一种全新的应对不同形式资源的新资源整理共享平台，为用户发掘、处理、储备多格式资源提供一个便捷的路径。

1. **多模态信息整合困境**

当前人们面对的资源非常多样，丰富了人们的资源库的同时也造成了一个资源整理整合的困境，通常我们所见的资源整理都是针对某一项资源类型的，进行查找搜索也大多根据标题、部分描述信息进行搜索而难以对文字以外的内容信息进行关联。

1. **结构性知识的价值**

信息繁杂无结构，结构性知识有助于以下几点：

(a) 方便记忆：知识点之间的连接可以促进学习和记忆，提高学习者的记忆能力，使接收者更容易理解概念之间的联系。

(b) 提高学习效率：将分散的知识串联起来，使学习更有效率;，巩固学习质量。

(c) 针对性：知识结构使学习（接收信息）更具针对性。

(d) 教育方面：学生通过结构性知识学习，可以培养学生的发散思维和逻辑思维能力，通过从简单到复杂的深入学习，提高学生的自主学习能力，提高学生解决问题的能力。

1. **已有的资源整理共享app难以满足用户需求**

当前的资源整理共享app在资源搜索、资源整理、资源查看、资源定位等方面仍存在缺陷。首先，资源搜索时多数app都只能根据标题和部分内容描述信息来进行搜索，对于资源的核心内容难以完全关联；其次，资源整理的格式单一难以一体化；最后，资源查看定位方面大多市场对象只能让用户进行全文全资源浏览而不能快速定位到用户所希望关注的部分，例如视频只能不断调动进度来寻找定位，而部分快速分段定位的功能也只能由发布资源的用户手动进行、造成极大不便。

1. **覆盖多类型用户群体需求**

对于视障人士群体，在学习的过程中他们无法借助图文来进行知识的接收基本上只能通过听觉和触觉进行学习，而大量书籍一是没有盲文版且不同语言的盲文也是不同的，所以首先限制了从触觉上学习的资源，相对来说现在的听书越来越多、或许听觉学习将成为盲人学习的主流方式，也就需要更多针对听觉资源、提供无障碍模式的app。

1. 我们的解决方案

针对以上的现状困境，我们提出一个全新的解决方案——自动生成结构要点的资源整理共享平台。

用户只需要输入实时或非实时资源，我们即可对各种格式的资源进行整合处理、利用多模态信息转换技术**打破了多模态信息整合困境**。

通过深度学习、端云结合等使用自动生成结构性要点算法，为用户查看、理解、记忆资源提供了一个更好的途径，帮助用户建立结构性知识、进一步提炼了资源利用效率、**更充分地发挥甚至提升了资源价值**。

针对提炼后的资源我们更是提供了资源快速定位服务，让用户享受**更丝滑更便捷的体验**。

生成资源要点后，还可以利用要点进行关联搜索，充分关联核心内容提高信息搜索的准确率；同时，在相关信息领域更精准地找到更多资源、**扩展用户的资源发掘空间**。

我们对拥有特殊需求的视障人士用户群体提供了更便捷的多模态转换和听觉资源快速定位。听觉无法像视觉一样快速定位，如果可以提取要点就可以通过要点进一步加速实现快速定位；并且，含有大量图片信息的资源也可以通过多模态信息的转换变成视障人士可以接收的形式，拓宽了视障人士群体的学习渠道；同时实时的图片信息转换还可以提供出行、日常生活接收外界信息的便利，进一步**覆盖视障人士群体的需求**。

## 项目定位

### 应用场景

场景一：实时线上课堂笔记

在线上课堂时老师的讲解、ppt、板书等等无法回看，也常常有听不清看不清记不清的三不清情况出现，针对以上情况当前大部分解决方案是应用录屏软件进行录屏、我们提出自动实时记录并整理笔记的功能，流程如下：

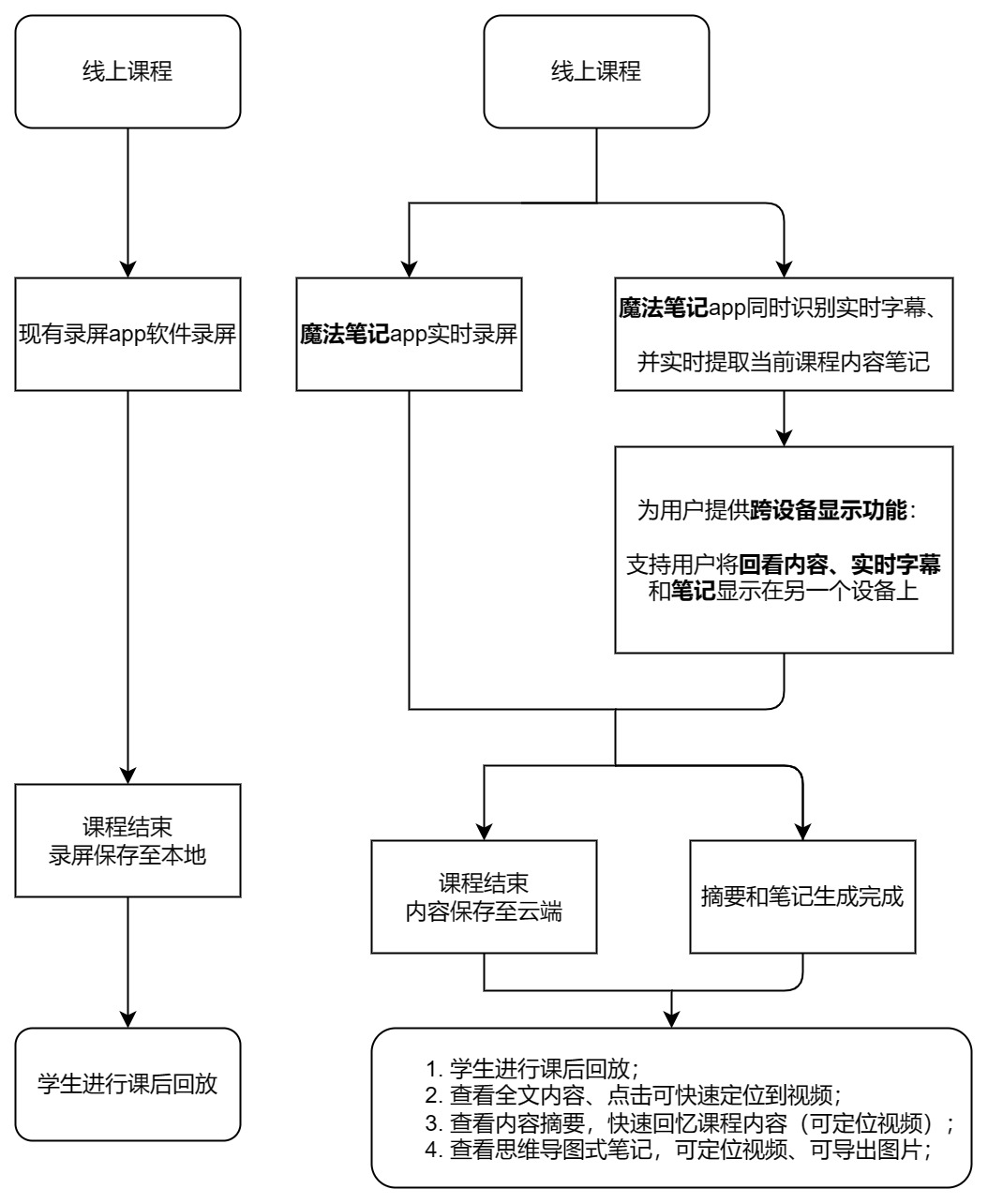


图1.2.1.1 自动实时记录并整理笔记的功能流程

场景二：复习重点利用结构加深记忆。

场景三：大型会议纪要总结。

场景四：大量多模态资源保存和搜索。

### 目标人群

1. 学生群体，为其提供结构化知识服务，像课代表一样为用户生成学习要点和笔记，并为学生用户推荐大量相关音视频及文档学习资源；
2. 职场人士，提供边界优质的
3. 拥有大量资源、需要进行资源整理的用户
4. 学习某类资源并且需要相关音视频、文档资源推荐的用户

## 项目方案

对于**多模态资源统一的问题**，该项目通过智能实时语音识别、图片描述技术将实时或非实时输入的视觉、听觉资源转换为统一的形式，即将视频、音频、图片、文档转换为文字描述，从而后续对文本资源进行整理。

对于**资源过滤的问题**，通过NLP资源过滤提取算法，将统一后的资源中必然存在的不必要干扰信息，去除杂质、提取出有效内容，从而提升整体效率和最终结果准确度。

对于**资源分解的问题**，通过语义分割等资源分解处理技术将大段资源进行分解后处理，一是提高性能效率，二是对大量无结构信息初步总结分类。

对于**资源摘要的问题**，该项目使用抽取与生成结合的文本摘要算法，对处理后文本进行要点提取，得到整个资源的核心内容。

对于**碎片知识结构化的问题**，通过句法语义结合的依存分析技术对碎片信息进行关联性链接，最终得到结构化的知识图谱。

对于**结构性信息可视化的问题**，使用自动优化的可视化算法进行图谱绘制。

对于**相关资源检索、资源推荐的问题**，使用我们所提取的知识图谱关键词检索和混合加权推荐策略，首先将大量可公开资源向用户群体共享，并且通过要点进行更精准的搜索；其次，对用户浏览资源进行相关资源推荐，从而为用户拓展资源渠道、为用户提供更便利的资源服务。

## 项目目标

实现可实时的便捷笔记生成功能，让“结构性知识触手可及”，使得人们能更加从容自在地应对洪流式资源，为人们减轻繁复的信息重担。

为职场人士、学生等所有渴望学习、探究的用户提供更好的学习资源整理平台让学习探究更高效，培养结构化记忆方式，让“智慧无处不在”。

## 项目价值

用户只需要输入实时或非实时资源，我们即可对各种格式的资源进行整合处理、利用多模态信息转换技术打破了多模态信息整合困境。通过深度学习、端云结合等使用自动生成结构性要点算法，为用户查看、理解、记忆资源提供了一个更好的途径，帮助用户建立结构性知识、进一步提炼了资源利用效率、更充分地发挥甚至提升了资源价值。针对提炼后的资源我们更是提供了资源快速定位服务，让用户享受更丝滑更便捷的体验。生成资源要点后，还可以利用要点进行关联搜索，充分关联核心内容提高信息搜索的准确率；同时，在相关信息领域更精准地找到更多资源、扩展用户的资源发掘空间。我们对拥有特殊需求的视障人士用户群体提供了更便捷的多模态转换和听觉资源快速定位。听觉无法像视觉一样快速定位，如果可以提取要点就可以通过要点进一步加速实现快速定位；并且，含有大量图片信息的资源也可以通过多模态信息的转换变成视障人士可以接收的形式，拓宽了视障人士群体的学习渠道；同时实时的图片信息转换还可以提供出行、日常生活接收外界信息的便利，进一步覆盖视障人士群体的需求。

# 开发计划

## 最终呈现形式

1. **统一用户的多模态资源**

帮助用户实现多模态的资源的统一整合和阅读，由于日常生活中视频、音频、文档的阅读、编辑都需要在不同的app上进行，所以我们提供一个可以整合音视频及文档的平台，实现资源的进一步便捷利用、简化了用户对不同模态资源的利用过程。

1. **文字符号性学习提高获取信息的效率和质量**

相比音视频学习，文本学习“阅读速度”高的多。相比视频学习，文本学习“阅读速度”高的多视频的本质是“以时间换取空间”，播放视频时，海量的画面需得一帧接着一帧地连续播放，由于视频的前后帧之间存在强叙事关联，如果我们想确保内容连贯，就必须完全顺序播放。所以在学习时音视频的学习速度远低于文本，所以用户很**自然地需要将音视频中的主要信息以文本的方式提取出来**，故此我们提供给用户一个以文本形式提取资源关键信息的服务，从而让用户更加高效地获取信息。

同时通过音视频进行知识学习受到资源播放速度的限制不具有自主性，文本学习让学习者主动掌控学习节奏，化被动学习为主动学习，加深学习深度、提高学习质量。

1. **快速精确定位提高非线性学习效率**

用户在获取信息尤其是时间有限时往往需要非线性学习，而音视频需要顺序播放且无法进行快速的定位，往往需要用户频繁调试后才可以相对找到需求位置，但是文字可以快速扫描定位。通过快速扫描关键性文字从而实现快速定位需求，同时我们能给用户提供更加精准的定位服务，极大地便利了用户快速定位的需求、提高了精准定位的质量。

1. **结构化信息帮助用户理解、记忆、回顾**

知识点之间的连接可以促进理解和记忆，提高用户的记忆能力，使接收者更容易理解概念之间的联系，同时使学习（接收信息）更具针对性。而结构性信息由于其精简的特点也提高了用户的发散性思维，更有助于通过重点脉络结构帮助用户快速回顾资源内容。

1. **为用户提供更精确的多模态信息检索平台**

当前的信息检索常常只关注资源的文本描述信息，对多模态尤其是有音视频图片资源的内容检索往往不准确，所以该项目为用户提供了针对多模态信息的内容更精确的检索服务，通过对多模态资源的内容分析提取来进行检索。

## 主要功能描述

### 信息格式转换模块

通过信息格式转换，我们将视频、音频、文档等各种格式的资源进行统一化，转换成文字格式，用户对不同信息格式资源的整理更加一体化；在转换后，资源的处理也更加便捷，通过文字将信息进行分割、提取、定位等处理，为后续服务提供基础；在此模块中主要又分成了三大模块：

1. 文档转换：提取文档中的文字，对文档中的图片等信息进行图片识别和描述；
2. 音频转换：从音频中去除环境噪声、进行语音识别和翻译；
3. 视频转换：视频的声音进行去噪声、语音识别、翻译，而视频的画面进行图像识别和描述；

### 要点总结模块

不同的人对笔记风格的喜好不同，用户可根据需求生成对应类型的笔记，如大纲式笔记、树形笔记。大纲笔记可以帮助思考学习的主题。树形笔记可以看出笔记间的参次关系。不同的笔记风格便于提升不同用户的使用体验，在学习的同时满足自己的观感，提高视觉舒适度。

音视频需要顺序播放且无法进行快速的定位，往往需要用户频繁调试后才可以相对找到需求位置，但是文字可以快速扫描定位。通过快速扫描总结性要点从而实现快速定位需求，同时我们能给用户提供更加精准的定位服务，极大地便利了用户快速定位的需求、提高了精准定位的质量。

### 搜索推荐模块

用户常常需要从大量公开资源中找到自己所需的文档。同时用户更倾向于阅览自己更为偏爱的文档。所以我们推出了搜索推荐模块，能够根据用户需求精准查找资源并主动为用户推荐其可能喜欢的资源。

（1）搜索资源

搜索系统理解用户输入的查询语句，根据提取的查询语句关键词在公开资源中搜索最符合用户需求的资源。

（2）推荐资源

推荐系统包括两部分，一部分是基于用户平常的上传点击资源类型推荐内容相似的资源。另一部分为根据用户平常的行为偏好推荐与其同好用户的喜好资源。通过推荐系统向用户提供个性化的资源分享平台。

## 运行环境

|  |  |  |
| --- | --- | --- |
| 类别 | 环境 | 用途 |
| 服务器 | 交互服务器 | 完成用户与后端的交互，并将任务 只能地分配给服务器 |
| GPU服务器 | 完成多模态信息的转换、文本摘要、要点可视化等任务 |
| 文件服务器 | 存储涉及到的所有文件。 |

## 验收标准

验收标准，即每一级别的错误量的可接受范围，下面根据错误等级给出本项

目的验收标准：

1. 不存在错误等级为 1 的错误；
2. 不存在错误等级为 2 的错误；
3. 错误等级为 3 的错误数量小于 5 个；
4. 所有提交的错误都已得到更正；

软件错误的严重等级由重到轻，如下：

1. 一级错误

这一级别的错误包括以下内容：没有实现或错误地实现重要的功能；业务流 程存在重大隐患；软件在操作过程中由于软件自身的原因自动退出系统或出现死 机的情况；软件在操作过程中由于软件自身的原因对系统或数据造成破坏；在现有的软、硬件建设环境下不能实现应有的功能；软、硬件在操作过程中可能危及系统和人身安全。

1. 二级错误

这一级别的错误包括：没有实现基本功能，并且不存在替代办法；没有实现重要功能中的部分功能，并且不存在替代办法；业务流程衔接错误；密钥以明文方式存储；用户权限分配不合理；在现有环境下，不能实现部分功能且没有替代方案；没有满足系统的性能要求。

1. 三级错误

这一级的错误与二级错误对应，但是三级错误存在替代方法；对误操作或错误操作没有提示，导致非法进入数据库。软件不能判断数据的有效性, 屏蔽用户的错误输入, 识别非法值, 并有相应的错误提示。

1. 四级错误

这一级别的错误往往为易用性方面的错误，如界面不友好、前后风格不一、查询结果输出不直观等。

1. 五级错误

一般为文档方面的错误，如安装手册、描述手册、维护手册中的描述错误。

## 关键问题

1. 自动生成笔记的准确性、全面性

自动生成笔记是为了帮助学者更好地提炼知识点，使学习更加高效。本项目实现的自动生成笔记既需要对用户提供的学习资料全面覆盖，又不能改变资料上知识点的含义。

1. 搜索推荐的精确性

在搜索学习资料时，信息过载会影响学习效率，推荐系统会预测用户需求并推荐给用户其可能最喜欢的内容。推荐搜索的精确有助于缓解人们从海量信息中作出选择的烦恼。

1. 语音识别的准确性

在本项目中，语音识别是最基础的一步，也是准确率要求最高的一步，提高语音识别的准确性，对后续笔记定位、自动生成笔记功能有很大助力。

## 进度安排

|  |  |  |
| --- | --- | --- |
| 子项 | 计划完成时间 | 备注 |
| 初步完成文本摘要 | 2023.3.10 | 软件功能的核心模块 |
| 完成信息多模态转化 | 2023.3.15 | 无 |
| 完成资源分解、过滤、处理 | 2023.3.15 | 与上一模块同时进行 |
| 测试改进上述模块 | 2023.3.17 | 无 |
| 初步完成要点结构可视化 | 2023.3.24 | 无 |
| 制作大部分页面 | 2023.4.5 | 与下面部分内容同步配合进行 |
| 完成社区模块 | 2023.3.31 | 无 |
| 完成前后端对接 | 2023.4.8 | 协同完成 |
| 完成推荐系统并整合入APP | 2023.4.14 | 无 |
| APP测试 | 2023.4.17 | 测试性能、收集问题 |
| 优化解决测试过程中的问题 | 2023.4.20 | 无 |
| 美化前段、简化操作 | 4.25 | 无 |
| 完结 | 4.25 | 无 |

## 开发预算

|  |  |  |
| --- | --- | --- |
| 类别 | 名称 | 单价 |
| 服务器 | Web服务器 | 1300/年 |
| GPU服务器 | 10000/年 |
| 文件服务器 | 1100/年 |

# 可行性分析

## 技术可行性分析

### 信息多模态转化

对于视频、音频需要进行一定的信息转化得到其文本内容即得到我们模型可处理的形式。在我们的项目中，选择使用openAI发布的whisper算法，将有监督的语音识别数据集规模从千级提升到了68万小时，在大规模的弱监督数据集上进行预训练，可以适用于各种数据。它的弱监督预训练的范围包含多语言和多任务，通过多语言多任务的Prompt学习，模型无需微调就可以直接应用于各种下游任务。此外，其音频转写的速度与准确性也是我们可以用来调节的，十分契合我们对于信息多模态转化的要求。

### 文本摘要

**资源分解处理技术**

主流的分词算法包括基于词典匹配的方法和基于统计机器学习的方法，在我们的项目中中文分词需要机器学习较好的消歧能力也需要灵活地引入词典等外部资源。综合来看，我们选择的 LTP分词技术能够很好的符合我们地要求，LTP分词模块使用的算法将两种方法进行了融合，LTP使用隐马尔可夫模型，因此拥有不错的消歧能力；维护词典使其在切分语句能够能灵活地进行匹配，十分满足我们项目的需求。

**资源过滤提取**

我们项目需要将文本中没有多少实际意义的词语如助词、连词、副词、语气词等过滤后不影响语句理解的词给过滤掉。我们选择使用HanLP进行停用词过滤。HanLP使用双数字典树对停用词进行过滤，能够达到很好的效果，满足我们项目对于停用词过滤的要求。

**文本摘要**

文本摘要分为抽取式摘要与生成式摘要两种。对于总结性较强的音视频，我们需要抽取式地提取总结式的文本信息，以保证提取内容的准确无误。其次，在内容总结性语句较少的音视频中，我们需要生成式地总结要点。我们的项目对于抽取式与生成式要点都有一定的需求，并且对获取的要点要有较强的可解释性。

经过我们的选择与测试，在抽取式摘要中，Summarunner模型在能够准确的进行文本摘要的基础上，其抽取句子还拥有极高的可解释性，十分满足我们项目的需求，有利于提高我们项目文本摘要的效果。

在生成式摘要上，pointer-generator network模型以及其所采用的覆盖机制能够很好的对文章内容进行总结，能够处理oov问题，并一定程度减少生成式摘要中内容的重复度，提高了生成摘要的质量，满足项目需求。

### 要点结构可视化

对于可视化的需求，我们的项目需要可视化形式条理且多样，在此我们选择的D3.js是一个可以基于数据来操作文档的 JavaScript 库。可以使用 HTML, CSS, SVG 以及 Canvas 来展示数据。D3 遵循现有的 Web 标准，可以不需要其他任何框架独立运行在现代浏览器中，能够结合强大的可视化组件来驱动 DOM 操作。并且D3.js作为偏低层的可视化工具，比其他可视化语言有着更高的灵活性，能够根据需求生成如网状关系图、和弦图、树状图等不同形式的要点图示。能够解决我们对于要点展示多样、条理等的需求。

### 混合加权推荐策略

考虑到推荐系统的准确性与适应性，结合同类视频的推荐特点，本项目最终采用针对视频优化的智能推荐系统，实现对视频与用户动态的精准推荐。

在进行推荐系统的特征提取时，将用户资源生成的关键词作为视频的推荐特征，除此之外，也会选择让用户自行为自己的资源进行命名，从而将这二者结合作为资源的总体特征，以保证特征信息提取的准确性。

在推荐系统算法上，采用基于内容的推荐算法和基于用户的协同过滤算法加权混合的推荐策略，在解决冷启动问题的同时提升推荐算法的个性化程度与准确性。

基于内容的推荐，主要以用户上传或用户历史点击的文档的关键词为基础，生成隐式的用户画像，基于该画像计算特征向量的相似度来对内容进行推荐。

基于协同过滤的推荐，使用基于多层感知机的神经协同过滤算法进行。

最终我们基于现有的用户数量与交互数量，决定两种推荐算法的权重。

## 资源可行性分析

### 软硬件资源

1. 团队已安装 Android Studio，并拥有一台 OPPO 手机。
2. 团队拥有 PC、笔记本等硬件设备，且拥有 Windows10、Linux 操作系统平台，并安装有 WebStorm、SQL server 等必要开发工具。
3. 团队现拥有云服务器。学校提供高性能数据处理服务器一台供使用。学校支持借用高性能 GPU 服务器，可供深度学习建模使用。

### 人力资源

本团队四名成员全部来自计算机科学与技术专业，计算机软件技术基础扎实，且都参加过软件应用项目的开发，项目经验较为丰富。所有成员都有国家级或省级的软创或科创大赛经历，具有很强的团队协作意识和学习能力。

本团队的指导老师教学经验丰富，技术能力全面，可以为我们提供各方面的 指导和支持。

### 其他资源

因特网和图书馆拥有丰富的关于深度学习、安卓开发等领域的前沿学科资料 和学习资料，能提供给我们充足的学习和查询途径。

## 市场可行性分析

### 市场需求分析

随着信息技术的发展，信息的生产和流通速度不断提高，人们迎来了信息爆炸时代。面对大量的信息，人们难以在理想时间内找到自己所想要的内容。尽管当前音视频平台如bilibili、中国大学MOOC的每一音视频资源都有其相应概括性的标题或简介，但其标题与简介往往又局限于音视频整体内容的概括，而并非对于视频中内容要点的详细阐述或展示。

经调查统计，搜索学习资源时人们往往需要的只是某一个细节的讲解，体现在音视频中往往只是很小的一部分，这就导致人们搜寻信息不便、需要花大量时间去审阅音视频、文档去寻找自己所需要的内容。因此bilibili、中国大学MOOC提供给我们的是学习交流的资源平台，但并没有为我们提供完善的服务。因此，如果引入一个能够为音视频、文档提供要点总结定位的软件，那么它势必得到每一个追求高效学习的人的认可，为人们搜索学习过程中提供便利的服务，更好地满足当前人们学习的需求与方式。

### 竞品分析

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 产品名称 | 产品定位 | 功能模块 | | | | | | |  | 优缺点分析 | |
|  |  | 资源交流共享 | 要点总结 | 多元笔记 | 多  情景应用 | 快速定位 | 支持无障碍模式 | 优点 | | | 缺点 |
| 1672728117479  中国大学MOOC | 国家精品课程的在线学习平台 | √√ | × | × | × | × | × | 1.课堂和课后作业测试相结合，有助于知识学习  2.和大学合作绑定，用户群体稳定 | | | 1.课程建设环节和要求过于繁琐  2.缺少交互式视频学习环境。  3.知识沉淀的需求尚未满足。 |
| 有道云笔记 | 多平台记录工具 | √ | × | √ | × | × | × | 1. 支持5种文稿类型，记录更加便捷 | | | 1. 容量少，只能免费使用3G 2. 表格打开速度慢 3. 无法对视频做交互式笔记 |
| 石墨文档 | 可多人实时协作的云端文档 | √√ | × | √ | × | × | × | 1. 发行早，用户群体大 2. 可以生成思维导图等多元笔记类型 | | | 1. 思维导图只能在电脑端操作，功能局限 2. 个人免费版容量只有1G 3. 无法对视频做交互式笔记 |
| 1675413574842  讯飞文档 | 支持多人多端同时编辑的在线文档 | √ | × | × | √ | √ | × | 1. 支持多端同步 2. 拥有在线表格 | | | 1. 无法设置字体大 小 2. 笔记呈现形式不多元 |
| 2023-01-02 (3)  DeepL | 实时机器翻译工具 | × | × | × | √ | √ | × | 1. 翻译结果较为精准 | | | 1. 没有图片翻译功能 2. DeepL有收费升级 |

### 自我分析

“魔法笔记”项目采用基于深度学习算法，解决了普通学习app无法自动根据学习资料自动生成笔记的问题，并大纲笔记、矩阵笔记和传统的线性笔记等多种笔记呈现形式，帮助用户便捷高效地学习；通过多模态信息转换技术，满足用户视听转化等多种需求；通过字幕跳转视频、自适应视频控制、多颜色高亮字幕等技术为用户提供交互式视频学习环境；搭建相关知识的个性化推荐系统，支持了相关知识的智能推荐；提供视频创造者、学习者的交流平台，支持平台用户分享交流、讨论答疑。

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 产品名称 | 产品定位 | 功能模块 | | | | | |  | 优缺点分析 | |
|  |  | 资源交流共享 | 要点总结 | 多元笔记 | 多  情景应用 | 快速定位 | 交互式视频学习环境 | 支持无障碍模式 | 优点 | 缺点 |
| 魔法笔记 | 自动生成结构要点的资源整理共享平台 | √ | √ | √ | √ | √ | √ | √ | 1. 对视频播放精准定位 2. 可自动生成笔记 3. 提供交互式视频学习环境 4. 可对视频做笔记 | 1. 知名度不高 |

# 需求分析

## 数据需求

### 静态数据

表 4.1.1.1 常量信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **数据**  **类别** | **名称** | **值** | **注解** | **类型** |
| 服务  器配  置 | Computin  g  Server | {} | 计算服务器相关  配置信息 | dict |
| Storage  Server | {} | 存储服务器相关  配置信息 | dict |
| … | … | … | … |
| 前端 | Record  Quality |  | 录屏码率 | float |
| Record  fps |  | 录屏帧率 | Int |

### 动态数据

表 4.1.2.1 变量信息表

|  |  |  |  |
| --- | --- | --- | --- |
| **数据类别** | **名称** | **定义** | **类型** |
| **输入** | Resource | 资源类型 | Int |
| Word\_Content | 文档内容 | String |
| Video\_Content | 视频链接url | String |
| Audio\_Content | 音频链接url | String |
| Resource\_att | 资源权限 | Int |
| …… | …… | …… |
| **处理过程** | Sentence | 转换分句 | String |
| Key\_word | 关键词 | String |
| Abstract | 摘要 | String |
| Time\_node | 时间点 | Int |
| …… | …… | …… |
| **输出** | Note\_bstract | 笔记摘要 | String |
| Map | 结构导图url | String |
| Article | 转换全文 | String |
| …… | …… | …… |

### 数据词典

表 4.1.3.1 用户表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **字段** | **注释** | **类型** | **长度** | **非空** | **键** |
| i\_id | 用户id | varchar | 255 | TRUE | 主键 |
| i\_password | 用户密码 | varchar | 255 | TRUE |  |
| i\_name | 用户名 | varchar | 255 | TRUE |  |
| i\_time | 用户注册时间 | datetime | 255 | TRUE |  |
| i\_like | 用户喜好 | varchar | 255 | TRUE |  |

表 4.1.3.2 （本地）搜索历史表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **字段** | **注释** | **类型** | **长度** | **非空** | **键** |
| s\_id | 这条搜索记录的id | varchar | 255 | TRUE | 主键 |
| i\_id | 搜索用户的id | varchar | 255 | TRUE |  |
| s\_content | 搜索内容 | varchar | 255 | TRUE |  |
| s\_time | 搜索时间 | datetime | 255 | TRUE |  |
| i\_like | 用户喜好 | varchar | 255 | TRUE |  |

表 4.1.3.3 资源表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **字段** | **注释** | **类型** | **长度** | **非空** | **键** |
| r\_id | 这条资源的id | varchar | 255 | TRUE | 主键 |
| i\_id | 发布资源的用户的id | varchar | 255 | TRUE |  |
| r\_content | 资源原内容 | varchar |  | TRUE |  |
| r\_limit | 资源是否公开 | varchar | 255 | TRUE |  |
| r\_type | 资源类型 | enum |  | TRUE |  |
| r\_daft | 资源文稿 | varchar |  | False |  |
| r\_d\_loc\_time | 文稿定位时间 | datetime |  | False |  |
| r\_d\_loc\_position | 文稿定位位置 |  |  | False |  |
| r\_point | 资源要点 | varchar |  | False |  |
| r\_p\_loc\_time | 要点定位时间 | datetime |  | False |  |
| r\_p\_loc\_position | 要点定位位置 |  |  | False |  |
| r\_keyword | 资源关键词 | varchar | 255 | False |  |

表 4.1.3.4 最近浏览表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **字段** | **注释** | **类型** | **长度** | **非空** | **键** |
| w\_id | 这条浏览记录的id | varchar | 255 | TRUE | 主键 |
| r\_id | 浏览的资源id | varchar | 255 | TRUE |  |
| i\_id | 浏览的用户id | varchar | 255 | TRUE |  |
| w\_time | 浏览的时间 | datetime | 255 | TRUE |  |

表 4.1.3.5 最近收藏表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **字段** | **注释** | **类型** | **长度** | **非空** | **键** |
| c\_id | 这次收藏行为的id | varchar | 255 | TRUE | 主键 |
| r\_id | 收藏的资源id | varchar | 255 | TRUE |  |
| i\_id | 收藏的用户id | varchar | 255 | TRUE |  |
| w\_time | 收藏的时间 | datetime | 255 | TRUE |  |

表 4.1.3.6 评论表

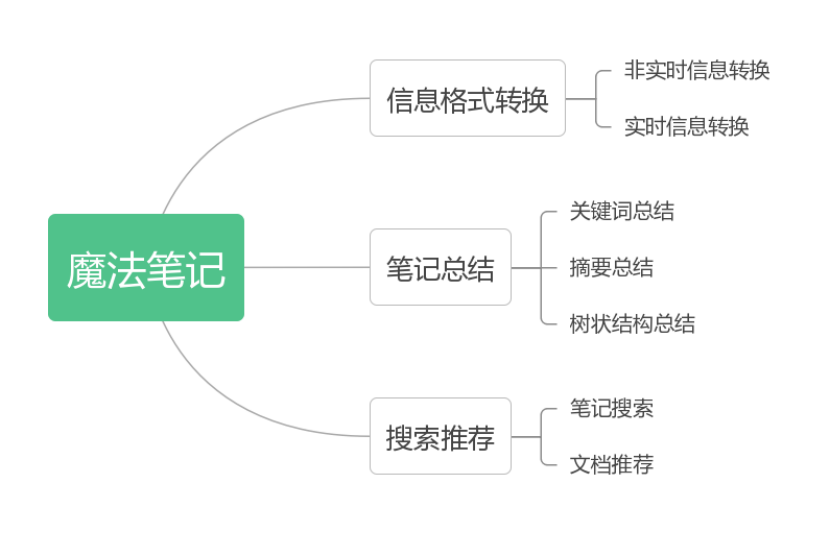
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **字段** | **注释** | **类型** | **长度** | **非空** | **键** |
| d\_id | 这条评论的id | varchar | 255 | TRUE | 主键 |
| df\_id | 父级评论的id | varchar | 255 | FALSE |  |
| i\_id | 评论的用户id | varchar | 255 | TRUE |  |
| d\_time | 评论的时间 | datetime | 255 | TRUE |  |
| d\_text | 评论的内容 | varchar | 2047 | TRUE |  |
| s\_id | 被评论资源的id | varchar | 255 | TRUE |  |
| d\_state | 0 待审核，1  待重审，2  正常，3 屏  蔽 | int |  | TRUE |  |

### 数据采集

表 4.1.4.1 数据采集

|  |  |  |  |
| --- | --- | --- | --- |
| **数据类别** | **数据名称** | **采集方式** | **用途** |
| **静态数据** |  |  |  |
| **动态数据** | 用户选择的资源 | 用户上传或  从资源库中  选择 | 用户学习、做笔记的资料  作为笔记自动生成的原材料 |

## 功能需求



### 信息格式转换模块

表1信息格式转换核心功能模块描述

|  |  |  |  |
| --- | --- | --- | --- |
| 功能模块 | 功能 | 功能描述 | 优先级 |
| 信息格式转换 | 非实时信息格式转换 | 将完整的图片、音频、视频信息转化为文字信息 | 2 |
| 实时信息格式转换 | 将实时输入的音频或录屏信息转化为文字 | 1 |
|  |  |  |

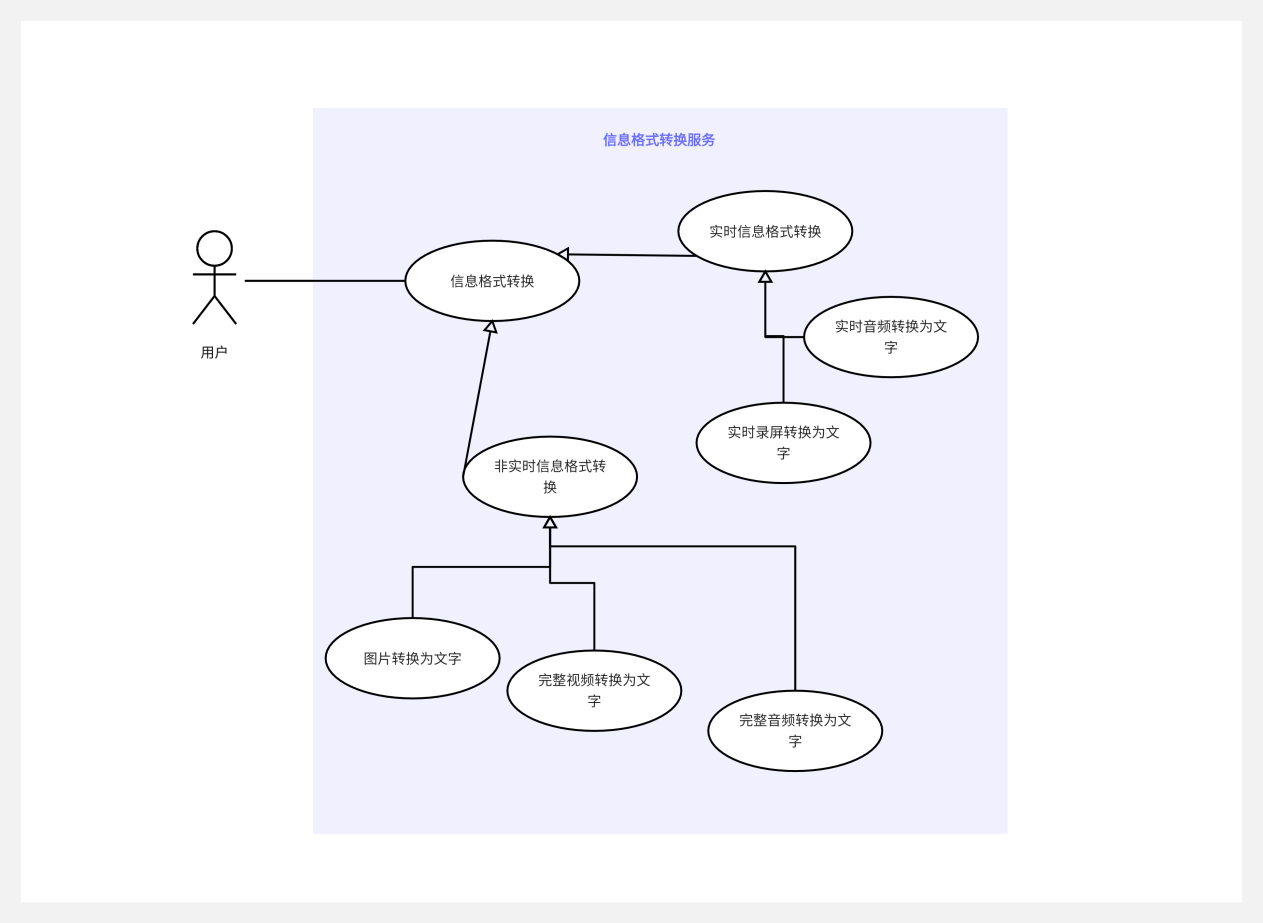


图4.2 信息格式转换功能用例图

表2 信息格式转换用例规约

|  |  |
| --- | --- |
| 用例名称 | 信息格式转换 |
| 功能简述 | 将多模态信息转化为文字格式 |
| 用例编号 | F0001 |
| 执行者 | Sever，Client |
| 前置条件 | 1. 对于非实时信息转换，用户完整完整上传需要转换的信息 2. 对于实时信息转换，用户授予开启麦克风和录屏权限 |
| 后置条件 | 用户得到转换成文字形式的信息 |
| 涉众利益 | 用户：希望能以文字的形式阅读、保存或分享信息  信息格式转换：希望能准确、完整地将不同模态信息转化为文字格式 |
| 基本路径 | 1. 用户根据输入信息的格式选择相应的信息转换功能 2. 用户将需要转换的信息上传到服务器 3. 服务器将信息转换为文字格式 4. 服务器将文档传回用户手机 |
| 扩展路径 | 无 |
| 字段列表 | 输入信息，信息格式 |
| 设计规则 | 单独模块 |
| 未解决的问题 | 无 |
| 备注 | 无 |

### 笔记总结模块

表3笔记总结核心功能模块描述

|  |  |  |  |
| --- | --- | --- | --- |
| 功能模块 | 功能 | 功能描述 | 优先级 |
| 要点总结 | 关键词总结 | 提取输入信息中的关键词 | 1 |
| 摘要总结 | 提取输入信息的要点并组合为摘要 | 1 |
| 树状结构总结 | 提取输入信息的要点并转化为树状结构总结 | 1 |

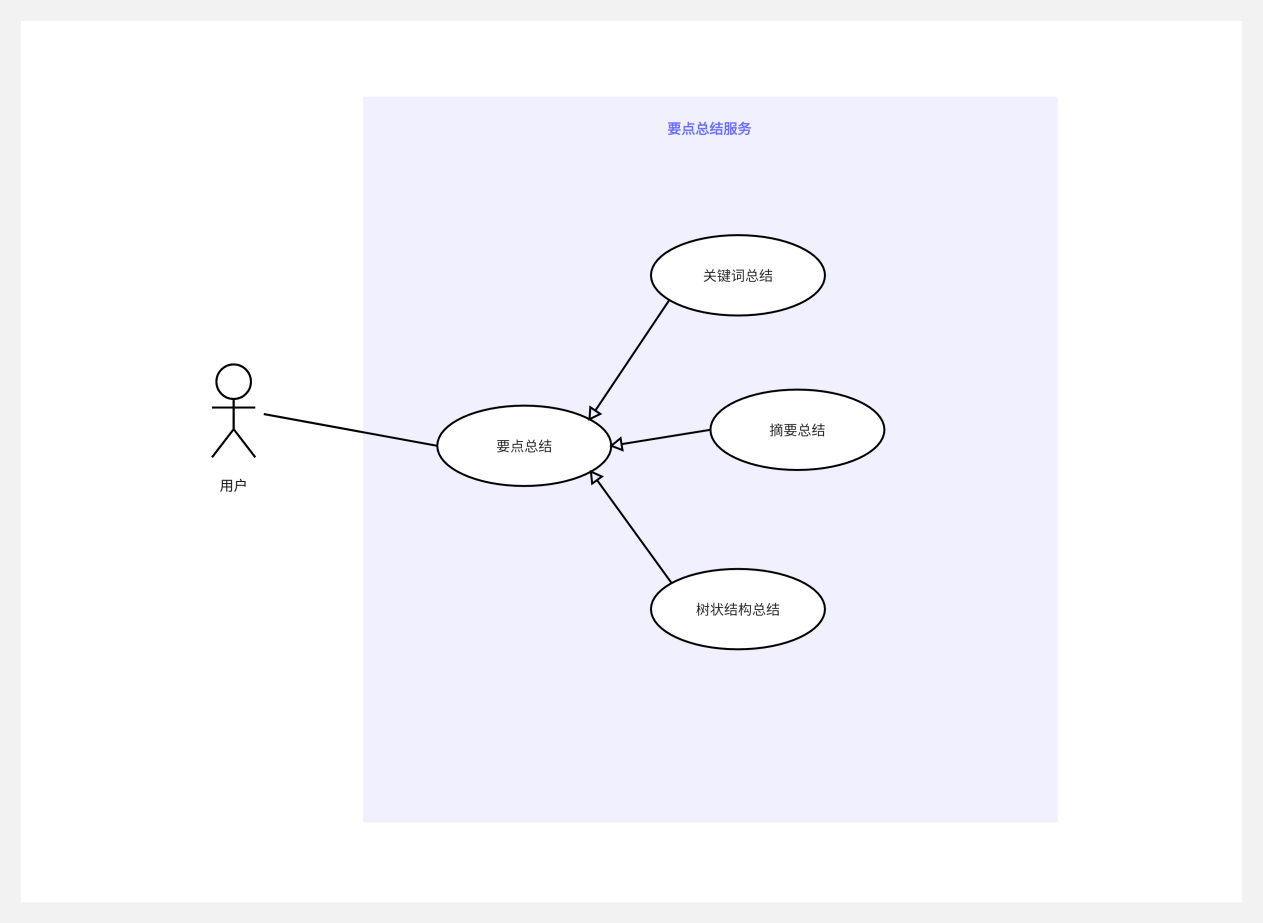


图4.3 要点总结功能用例图

表3 要点总结用例规约

|  |  |
| --- | --- |
| 用例名称 | 要点总结 |
| 功能简述 | 能以不同格式显示笔记的主要内容并提供点击定位功能 |
| 用例编号 | F0002 |
| 执行者 | Sever，Client |
| 前置条件 | 1. 用户上传需要总结的信息 2. 上传的信息符合APP要求 |
| 后置条件 | 根据选择的显示格式，用户可以得到笔记的关键词或摘要或树状结构总结，点击总结信息的语句能够跳转到信息相应地方 |
| 涉众利益 | 用户：希望得到以不同形式总结的信息  笔记总结：能够精准总结信息并以不同形式展示给用户，当用户点击总结语句时能够准确跳转 |
| 基本路径 | 1. 用户上传需要总结的信息 2. 服务器将用户上传的信息转化为文字格式 3. 服务器根据内容对文档进行分割，对分割的每一部分提取关键词、提取摘要并根据提取的摘要总结树状结构 4. 根据用户不同需求显示不同格式的总结 |
| 扩展路径 | 无 |
| 字段列表 | 文档资源，定位分割表，要点总结 |
| 设计规则 | 单独模块 |
| 未解决的问题 | 无 |
| 备注 | 无 |

### 搜索推荐模块

表4搜索推荐核心功能模块描述

|  |  |  |  |
| --- | --- | --- | --- |
| 功能模块 | 功能 | 功能描述 | 优先级 |
| 搜索推荐 | 笔记搜索 | 根据用户输入的关键词搜索相关笔记 | 1 |
| 文档推荐 | 根据用户的喜好，推荐用户可能喜欢的笔记 | 1 |
|  |  |  |

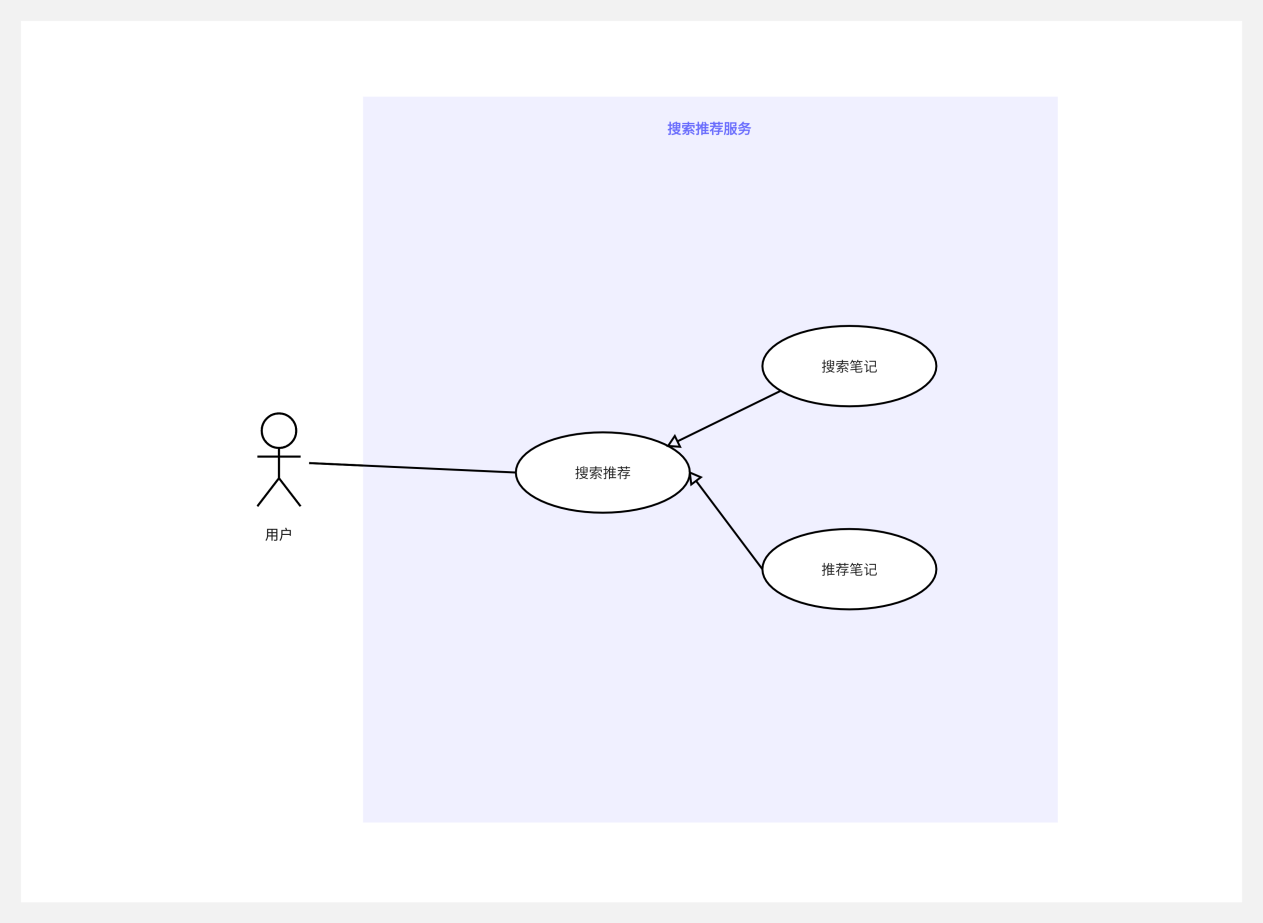


图4.4 搜索推荐模块用例图

表5 搜索推荐用例规约

|  |  |
| --- | --- |
| 用例名称 | 搜索推荐 |
| 功能简述 | 根据用户输入的信息搜索笔记，或推荐用户可能喜欢的笔记 |
| 用例编号 | F0003 |
| 执行者 | Sever，Client |
| 前置条件 | 1.搜索时有用户输入的信息  2.有一些用户交互数据，以便学习用户的喜好 |
| 后置条件 | 1. 用户得到输入信息相关的搜索结果 2. 用户得到喜欢的推荐 |
| 涉众利益 | 用户：希望能找到需要的笔记或得到符合自己喜好的推荐  搜索推荐：希望能满足用户的不同需求和喜好 |
| 基本路径 | 1.搜索系统接收用户输入信息后对信息进行理解，寻找相关文档并对其排序，将最后结果返回给用户  2.推荐系统根据用户当前的喜好等推荐用户可能喜欢的内容 |
| 扩展路径 | 将会在用户使用过程中及时修改喜好，能更好地匹配用户 |
| 字段列表 | 资源列表，推荐动态 |
| 设计规则 | 单独模块 |
| 未解决的问题 | 无 |
| 备注 | 无 |

## 性能需求

### 时间特性

表 4.20 响应时间表

|  |  |  |
| --- | --- | --- |
| **页面** | **时间需求** | **备注** |
| 登录 | 小于 100ms | 前端界面，向后端发送登录请求 |
| 注册 | 小于 100ms | 响应前端界面，后端写入数据库 |
| 搜索 | 小于 100ms | 根据知识名称搜索需要的学习材料 |
| 展示搜索结果 | 小于 300ms | 展示搜索结果 |
| 具体展示页 | 小于 200ms | 展示具体的知识内容 |
| 我的 | 小于 200ms | 展示账户信息、最近浏览、收藏和最近消息 |
| 自动生成的笔记导出 | 小于 300ms | 针对学习材料自动生成笔记 |

表 4.21 更新处理时间表

|  |  |  |
| --- | --- | --- |
| 操作 | 时间需求 | 备注 |
| **更新用户设置信息** | 小于 100ms | 用户更改自己的信息 |
| **更新用户喜好信息** | 小于 200ms | 系统根据用户操作自动更新用户喜  好信息 |
| **用户更改笔记** | 小于 50ms | 用户更改笔记信息 |

表 4.22 数据转换与传输时间表

|  |  |  |
| --- | --- | --- |
| 操作 | 时间需求 | 备注 |
| **音频转文字** | 小于 10s | 将音频转成文字以便自动提取笔记和跳转功能 |
| **传输原视频** | 小于 1min | 用户上传一些知识学习视频供大家学习 |
| **时间戳列表转标志点** | 小于 30ms | 在视频转文字，点击相应文字跳转至对应视频 |

表 4.23 运行时间表

|  |  |  |
| --- | --- | --- |
| 操作 | 时间需求 | 备注 |
| **前端** | 20% | 基本不需要进行运算，只需要完成一部分数据可视化展示的功能 |
| **后端** | 80% | 服务器需要完成音频转文字，数据处理等多种功能 |

### 适应性

**（1）操作方式**

我们尽可能让操作尽可能简单而易于理解，并设定一个统一的设计逻辑，尽 可能不让其变化，当因不可抗力出现整体操作方式的改变，我们将会给出新的指导。

**（2）运行环境**

我们将会考虑不同的手机尺寸和屏幕类型，当显示方式不同时，我们会尽可能让 APP 能正常运行，并且操作方式不会出现任何很大的改变。APP 将会支持绝大部分安卓系统。

**（3）其他软件的接口**

我们将会把接口以参数方式进行传递，当接口发生变化，我们只需管理员重新传入新的参数即可。不会对正常运行产生巨大影响。

## 界面需求

## 接口需求

### 硬件接口

无

### 软件接口

无

## 其他需求

### 可使用性

在端侧，基于移动端实现整个app，实现用户使用的便捷性。应用可以实现输入信息的正确格式转换，对输入信息进行精准概括，并采用多样化方式（例如：树状结构、摘要等）向用户进行展示。保证搜索推荐系统的准确性，能够向用户展示他们需要或偏爱的资源。

在云侧，基于分布式计算核心，提高对单用户的响应时间，防止用户在单步卡顿时间过长。分布式业务响应服务器，实现大量用户的无卡顿服务。

### 安全性

**（1）用户隐私**

未来保护用户隐私安全，传输用户数据时要使用加密算法，且密码等保密级别极高的数据不在数据库中明文存储。

**（2）验证系统**

由于不同身份的用户对于不同的接口使用权限不同，应有完善的验证系统对不同用户的身份进行检查。保证普通用户，管理员用户可以分别执行对应的操作，确保系统的有序性。

**（3）SQL注入**

SQL 注入会对后台数据库造成很大的威胁，也增大了用户信息泄露的危险。因此拥有防止 SQL 注入的机制是必要的。该项目通过采用ORM等机制来解决SQL注入的威胁，保证数据库的安全性。

**（4）XSS保护**

本项目中有很多需要提交的表单，如果这些地方被植入脚本，很可能会对系统造成威胁，所以需要有 XSS 防范机制。

需要使用 XSS 防护机制来对用户可操纵的数据进行 HTML 转义，之后再呈现给用户。如果用户可操纵的内容是 HTML 的标签属性，那么还要进行关键词的黑名单/白名单的过滤。

### 可维护性

**（1）完善的日志系统**

项目建立实时记录应用行为的运行日志，建立完善的日志系统，当应用出现问题时能够帮助我们发现、解决问题。

**（2）数据库的可恢复性**

需要定期为数据库进行备份，将数据库文件数据备份到另一台服务器。

**（3）微体系架构**

项目的各个功能模块都具有独立性。每个模块的开发运行是独立的，当一个模块运行出问题时不会影响当前模块的其他进程，或其余功能模块的正常运行。一些功能应该冗余存储，防止因为个别的节点故障导致系统瘫痪，进而保证系统运行的稳定性。

### 可移植性

项目为模块化实现。每个模块相对独立，模块之间通过接口进行数据交换。其中的重点功能应该为多个高层的功能提供服务。

# 概要设计

## 处理流程

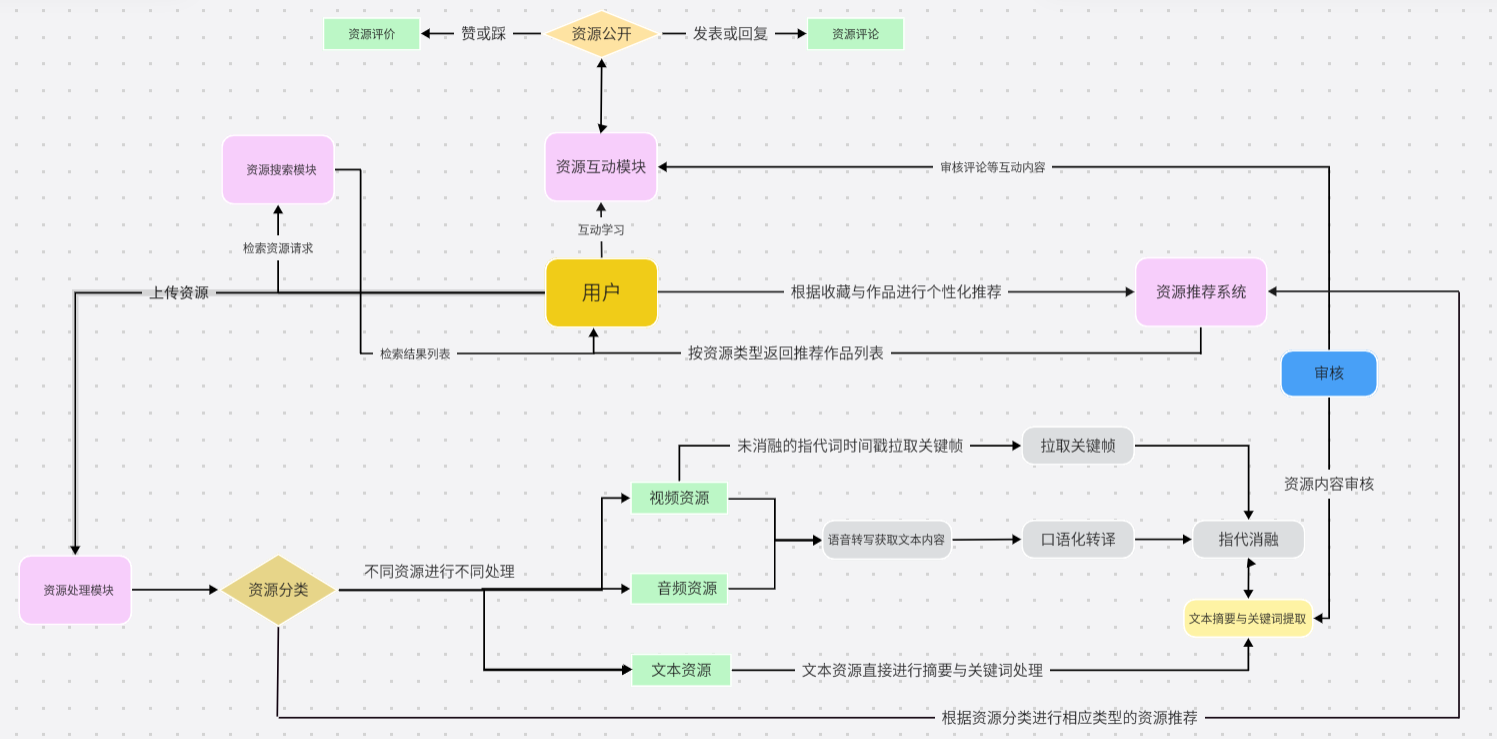


图5.1 逻辑流程图

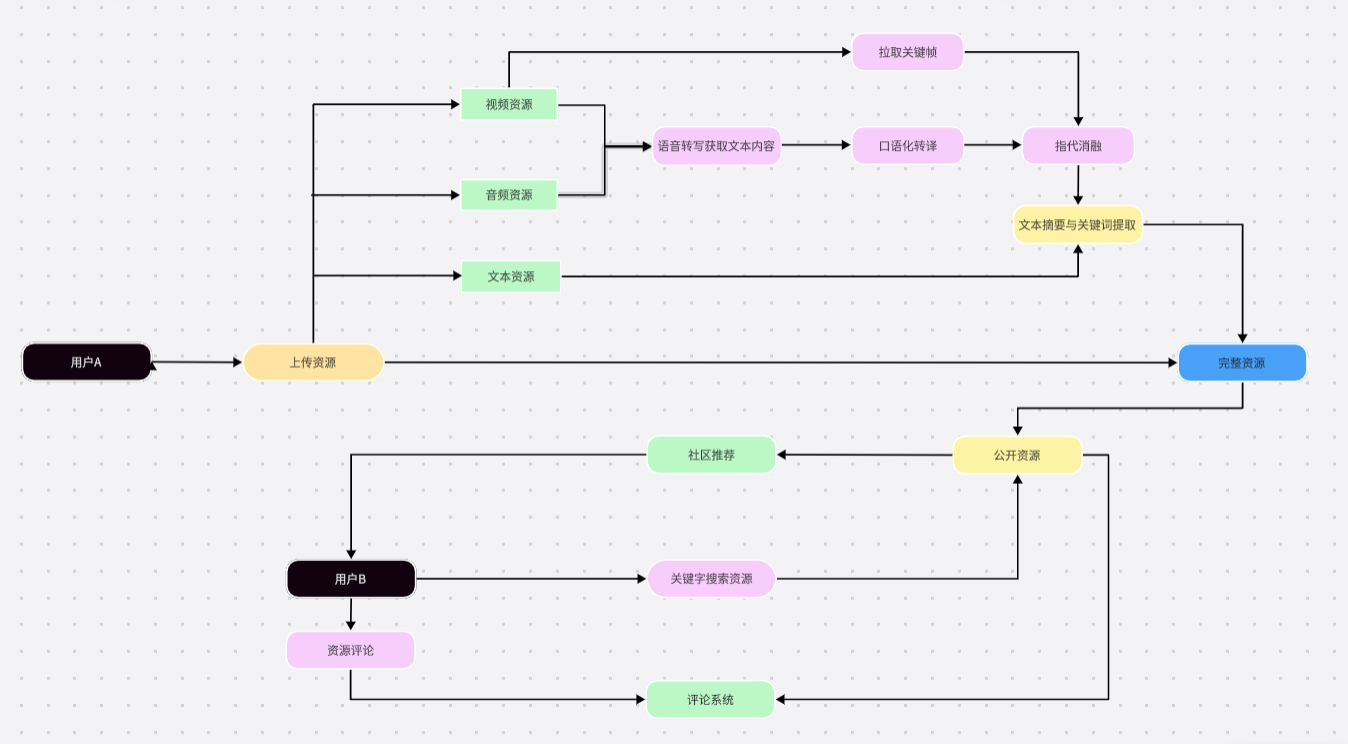


图5.2 数据流程图

## 总体结构设计

## 功能设计

## 数据流转设计

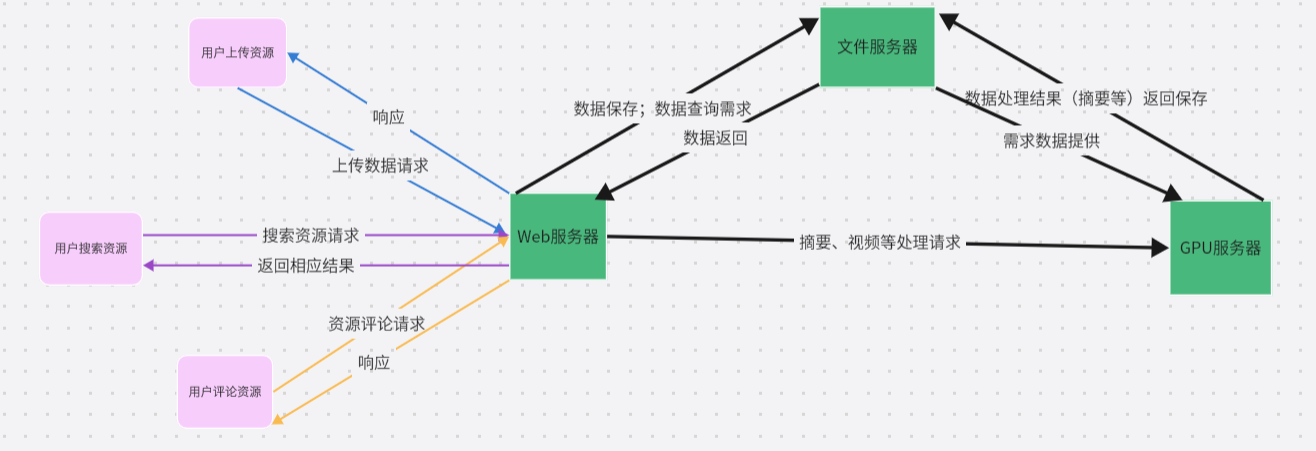


图5.3 数据流转流程图

用户上传资源，数据通过Web服务器发送到文件服务器；同时Web服务器向GPU服务器发送相应的请求以获取摘要等数据；GPU服务器需要的数据由文件服务器提供，GPU处理结果的数据返回至文件服务器中保存。

用户搜索资源、评论摘要通过Web服务器向文件服务器进行查询、保存，文件服务器返回请求的数据或保存传输的评论等数据。

## 用户界面设计

## 数据结构设计

5.6.1 逻辑结构设计

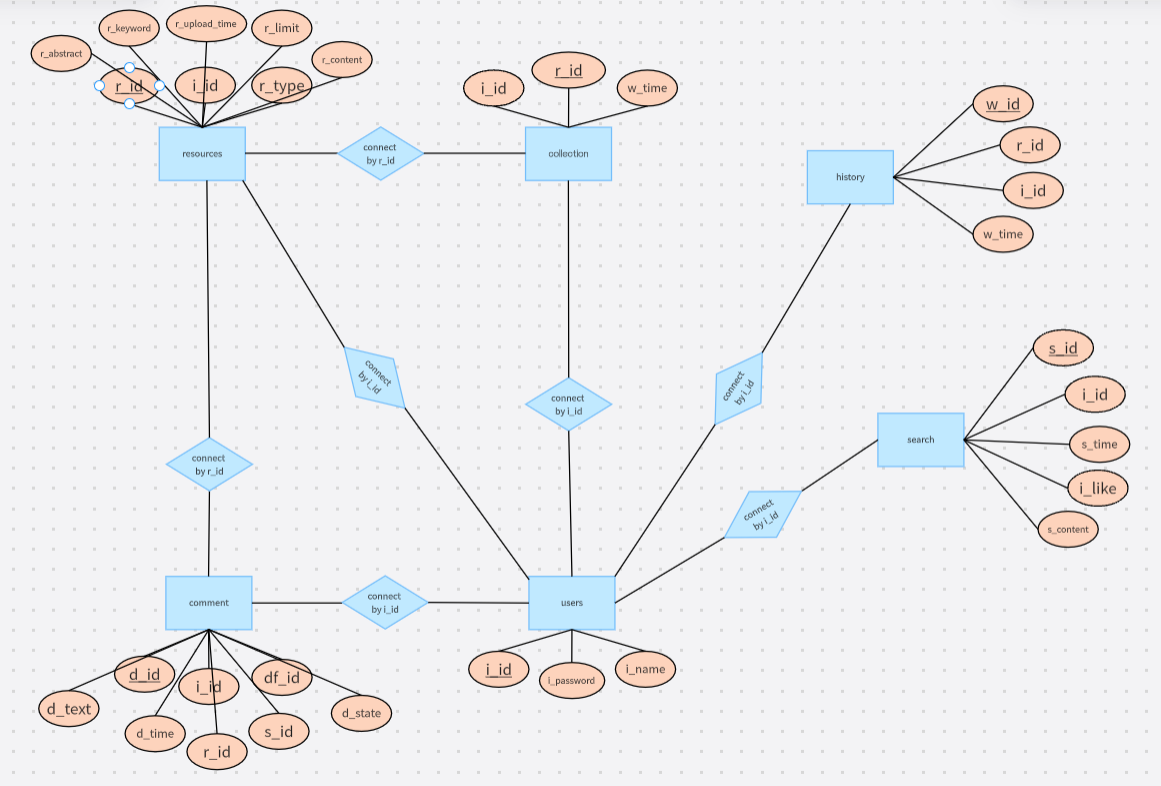


图5.4 逻辑结构图

### 5.6.2 物理结构设计

魔法笔记项目在数据库物理结构上主要做了以下的设计：

（1）对于数据库中大部分表的主键建立稠密索引，索引使用了 BTree 结构， 可以有效减少对于磁盘的访问次数，加快查询速度。

（2）对于数据库中更新操作较多的表，建立稀疏索引，由于稀疏索引所占用的空间相对稠密索引来说更小，并且插入和删除时所需的维护开销也较小，因此可以降低建立稠密索引所带来的空间开销和维护开销。

（3）对于频繁做连接操作的表格，对其进行了聚集操作，这样可以有效地减少连接操作的时间开销。

（4）对于频繁进行联合查询的字段，对相关字段建立联合索引，这样可以有效地提高多字段查询的效率。

## 接口设计

### 外部接口

表5.x 软件接口

|  |  |
| --- | --- |
| 接口 | 功能 |
| 科大讯飞语音转写API | 通过该接口实现音视频文件的语音转写 |
| HanLP指代词消解API | 通过该接口消解文本中的指代词 |
| 腾讯云短信 API | 通过该接口向用户发送验证码 |
| QQ 邮箱邮件 API | 通过该接口向用户发送邮件 |
| 百度智能云文本审核 API | 通过该接口实现文本内容的审核 |
| 百度智能云视频审核 API | 通过该接口实现视频内容的审核 |

### 内部接口

表5.x函数接口

|  |  |  |
| --- | --- | --- |
| 功能模块 | 函数名称 | 接口用途 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

表5.x Restful接口

|  |  |  |  |
| --- | --- | --- | --- |
| 功能模块 | URL路径 | 请求方式 | 接口说明 |
| 用户模块 | /user/register/ | POST | 新增一名注册用户，此时用户处于未认证状态，无法进行登录 |
| / user/register/ | PUT | 更新注册用户的相关信息 |
| /user/login/ | POST | 用户登录接口，返回登录认证令牌（7天有效期） |
| /user/refresh/ | POST | 使用现有的有效令牌申请有效期更长的令牌，超过一个月未登录将无法续期 |
| /user/password/ | PUT | 获取修改密码认证后，即可调用该接口更新密码 |
| /user/verify/<vtool:int>/<vtype:int> | POST | 用户选择 vtool（认证工具）与 vtype（认证方式）对已注册账 号进行身份认证，接口返回认证 验证码 |
| /user/verify/<vtool:int>/<vtype:int> | GET | 发送认证验证码实现注册账号 的身份验证 |
| /user/info/ | GET | 获取用户个人账号信息 |
| /user/info/ | PUT | 修改用户个人账号信息 |
| /user/info/<uid:int>/<itype:string> | GET | 根据 uid（用户 id）以 itype（信 息获取方式）获取指定用户信息 |
| /user/follow/ | POST | 添加用户收藏信息 |
| /user/follow/ | GET | 获取用户收藏信息 |
| /user/follow/ | DELETE | 删除用户收藏信息 |
| 文件模块 | /file/upload/verify/ | POST | 获取文件上传认证 |
| /file/upload/recourse/ | POST | 该接口用于文件上传 |
| 任务模块 | /resources/<uid:int>/<rid:int>/summary/ | GET | 根据uid（用户id）和rid（资源id）生成资源的摘要 |
| /resources/<uid:int>/<rid:int>/keyword/ | GET | 根据uid（用户id）和rid（资源id）生成资源的关键词与树状结构 |
| /resources/<uid:int>/<rid:int>/content/ | GET | 根据uid（用户id）与rid（资源id）生成资源的文本内容 |
| /resources/<uid:int>/<rid:int>/getall | GET | 根据uid（用户id）与rid（资源id）获取资源以及其文本内容、摘要、关键词与树状结构等 |
| 评论模块 | /comment/<ctype:int>/<id:int>/ | POST | 用于在编号为 id 的 ctype（动态 或作品）发表评论 |
| /comment/<ctype:int>/<id:int>/ | GET | 用于动态获取编号为 id的 ctype （动态或作品）部分评论 |
| /comment/<cid:int>/ | GET | 获取编号为 cid 的一级评论下的 全部二级评论 |
| 推荐模块 | /recommendation/w ork/ | GET | 基于协同过滤与网络模型推荐， 获取推荐作品 |
| 搜索模块 | /search/work/ | POST | 根据关键词进行作品检索 |

## 错误/异常处理设计

### 错误/异常输出信息

### 错误/异常处理对策

## 系统配置策略

考虑到不同的业务需要的硬件不用，我们把服务器分为 5 类：

1. 业务服务器：使用 Springboot 框架，搭建基本符合 RESTful 规范的接口，为用户提供常规运行逻辑，其内使用多个应用模块，每个应用可以看作一个数据主体的实现，各个模块之间有较高的耦合度。使用该框架可以基本做到数据校验稳定可靠，接口安全且易拓展。此类服务器需要有较多核心数的CPU，较大的内存，从而高并发的实现业务逻辑，该服务器部署时数量可以按照业务需求量的不同进行调整，单个服务器内运行的进程数也可以按照硬件条件进行配置。多个业务服务器通过网关进行负载均衡，通过数据服务器的Redis进行缓存加速

2. 数据服务器：数据服务器主要由四部分组成，分别为：消息队列，缓存数据库，关系型数据库与注册中心，在本项目中分别使用 RabbitMQ，Redis，MySQL，Consul。该服务器需要支持高效的 IO 操作，拥有较大的内存（进行缓存与维护 网络连接），与业务服务器与 GPU 服务器之间有良好的网络连接，以便于高速的数据交换。

3. 文件服务器：文件服务器的主要功能是进行文件的分发与处理，例如文 件的高性能上传，普通文件的快速下载，音视频文件的点播服务（DASH，HLS）， 文件的转码与审核等。该服务器需要有较大的硬盘，用来存储文件数据。通过配 置注册中心，可以进行分布式多节点设置。

4. 网关服务器：主要负责负载均衡，鉴权等业务，需要有较大的带宽应对巨大的流量。

5. AI 服务器：主要运行各种 AI 算法，需要有显卡进行运算支持，通过消息队列与主机进行通信。

## 跨端应用架构设计

## 其他相关技术与方案

项目实现过程中的其他算法与技术详情请参考项目技术研究文档。

# 数据库设计

6.1 ER图

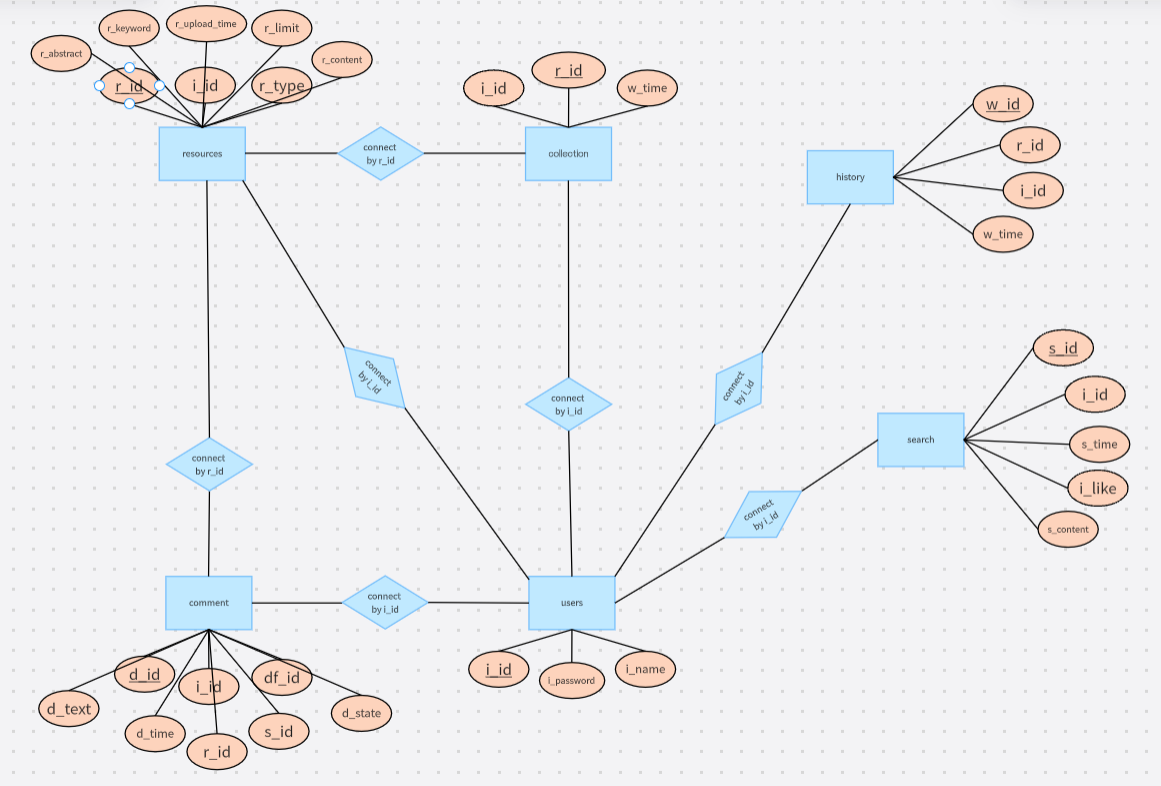


图 6.1 ER图

6.2表结构与相关字段说明

表6.2 User用户表

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 字段 | 数据类型 | 解释 | 最大长度 | 可空 | 主键 | 默认 | 唯一 | 备注 | 校验 |
| i\_id | Int | 用户ID | 40 |  | TRUE |  |  |  |  |
| i\_password | Char | 用户密码 | 40 |  |  |  | True |  |  |
| i\_name | Char | 用户名 | 32 |  |  |  |  |  |  |

表5.3 Search搜索表

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 字段 | 数据类型 | 解释 | 最大长度 | 可空 | 主键 | 默认 | 唯一 | 备注 | 校验 |
| s\_id | Int | 搜索标号 | 40 | FALSE | TRUE |  |  |  |  |
| i\_id | Int | 用户ID | 40 | FALSE |  |  |  |  |  |
| s\_time | Datetime | 搜索时间 |  | FALSE |  |  |  |  |  |
| i\_like | Char | 用户名 | 255 | FALSE |  |  |  |  |  |
| s\_content | Char | 搜索内容 | 255 | FALSE |  |  |  |  |  |

表5.4 comment评论表

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 字段 | 数据类型 | 解释 | 最大长度 | 可空 | 主键 | 默认 | 唯一 | 备注 | 校验 |
| d\_id | Int | 评论ID | 40 | FALSE | TRUE |  |  |  |  |
| df\_id | Int | 父级评论id | 40 | FALSE |  |  | True |  |  |
| i\_id | Int | 评论用户id | 40 | FALSE |  |  |  |  |  |
| d\_time | Datetime | 评论时间 |  | FALSE |  |  |  |  |  |
| d\_text | Char | 评论内容 | 255 | FALSE |  |  |  |  |  |
| s\_id | Int | 评论资源id | 40 | FALSE |  |  |  |  |  |
| d\_state | Int | 审核状态 | 3 | FALSE |  |  |  | 0 待审核  1待重审  2正常  3屏蔽 |  |

表5.5 resources资源表

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 字段 | 数据类型 | 解释 | 最大长度 | 可空 | 主键 | 默认 | 唯一 | 备注 | 校验 |
| r\_id | Int | 资源ID | 40 | FALSE | TRUE |  |  |  |  |
| i\_id | Int | 用户ID | 40 | FALSE |  |  |  |  |  |
| R\_content | Char | 资源文本内容地址 | 255 | FALSE |  |  |  |  |  |
| R\_limit | Int | 资源是否公开 | 2 | FALSE |  |  |  |  |  |
| R\_type | Int | 资源类型 | 3 | FALSE |  |  |  |  |  |
| R\_abstract | Char | 摘要地址 | 255 | TRUE |  |  |  |  |  |
| R\_keyword | Char | 关键词地址 | 255 | TRUE |  |  |  |  |  |
| R\_uploadtime | Date | 文件上传时间 |  | FALSE |  |  |  |  |  |

表5.6 collection收藏表

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 字段 | 数据类型 | 解释 | 最大长度 | 可空 | 主键 | 默认 | 唯一 | 备注 | 校验 |
| r\_id | Int | 资源id | 40 | FALSE | TRUE |  |  |  |  |
| i\_id | Int | 用户id | 40 | FALSE |  |  |  |  |  |
| w\_time | Datetime | 收藏时间 | 32 | FALSE |  |  |  |  |  |

表5.7 history 浏览历史表

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 字段 | 数据类型 | 解释 | 最大长度 | 可空 | 主键 | 默认 | 唯一 | 备注 | 校验 |
| w\_id | Int | 浏览历史条目id | 40 | FALSE | TRUE |  |  |  |  |
| r\_id | Int | 资源id | 40 | FALSE |  |  |  |  |  |
| i\_id | Int | 用户id | 40 | FALSE |  |  |  |  |  |
| w\_time | datetime | 浏览时间 |  | FALSE |  |  |  |  |  |

# 手机端侧部署设计

## 手机环境需求

# 详细设计

## 信息格式转换功能模块

### 功能描述

用户上传的资源分为文本、音频与视频三类，对于音频与视频的处理需要首先将其转为可处理的文本从而进行摘要，返回音视频对应的文本内容。

### 性能描述

前端上传资源后后端进行信息格式的转换，转换结束后对应的文本内容存放至文件服务器，待后续用户需要时直接根据索引展现。

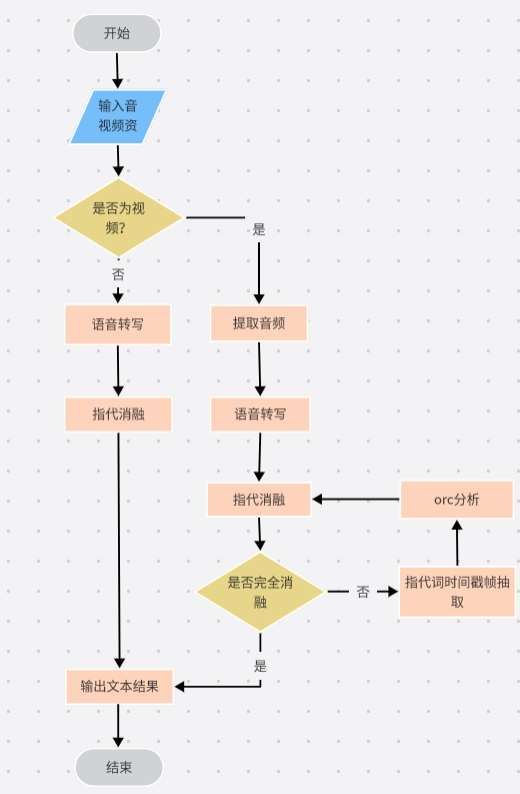
### 输入

用户上传的音、视频资源。

### 输出

音视频经算法转换后的文本内容文件。

### 程序逻辑



### 限制条件

需是音视频文件，文本文件无需信息格式转换。

## 笔记总结功能模块

### 功能描述

根据资源的文本内容，对文本内容进行摘要提取、关键词提取。

### 性能描述

上传资源后信息格式转换结束自动进行笔记总结，在用户点击查看之前基本可以完成，直接展现摘要笔记即可。

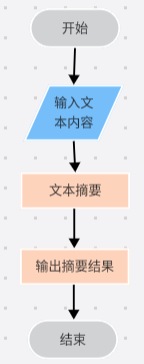
### 输入

要查看的资源r\_id。

### 输出

资源文本内容、摘要、关键词、笔记。

### 程序逻辑



### 限制条件

无。

## 搜索推荐功能模块

### 功能描述

### 性能描述

### 输入

### 输出

### 程序逻辑

### 限制条件