"慧阅"智能批注系统 整体技术架构解决方案

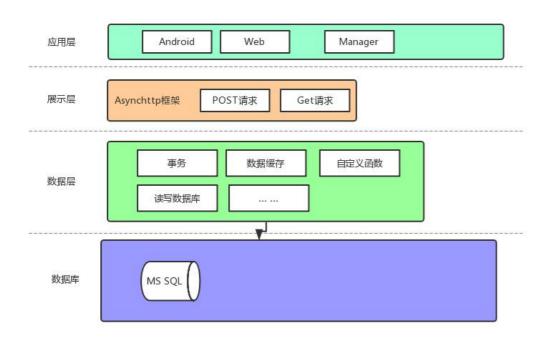
目录

1.	. 前端技术实现	2
	1.1技术路线概述	
	1.2 Android 技术实现	
2.	深度学习技术实现	6
	2.1 环境	
	2.2 技术路线	7
	2.3 技术亮点	7
	2.4 解决方案	7
3.	3. 后端管理技术实现方案	9
	3.1 编写目的	9
	3.2项目整体架构	9
	3.3 项目实现技术架构概要	9
	3.4 详细实现方案	10

1. 前端技术实现

1.1 技术路线概述

- 本项目主要分为 Android, Web 以及后台管理
- Android、Web 通过 WebService 连接数据库(如下图)
- ▶ 通过读取数据库中的信息来获得相应数据从而在移动终端上显示班级信息、作业信息、排行信息等等
- 后台管理模块主要是对数据库中的信息进入录入、更新



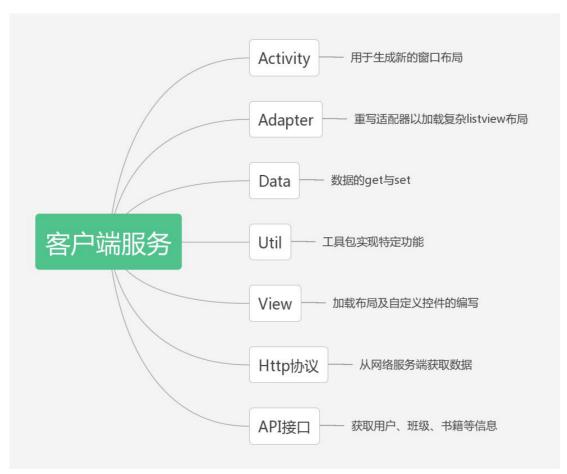
1.2 Android 技术实现

开发环境:



技术路线:

*





技术亮点:

- ▶ 采用 JSON 网络协议,轻量,安全,方便。
- ▶ 所有布局属性采用 Style 方式,便于整体修改。
- ▶ 采用相对布局,以便适应不同分辨率的移动终端
- ▶ 采用 AsyncHttp 框架 结构清晰,功能定义明确,对于多个后台任务时,简单清晰。
 - ✓ 兼容 Android API23 或更高版本

- ✓ 发送异步的 http 请求. 在匿名的回调 callback 中处理响应 response
- ✓ 流式 Json 上传, 无需三方库支持
- ✓ 能处理环形和相对重定向
- ✓ 内置 multiPart file 上传, 无需第三方库支持
- ✓ 相比 app 来说库很小, 仅仅只有 90k
- ✔ 针对移动连接自动智能的请求重试优化机制
- ✓ 自动的 gzip 响应解码
- ✓ 支持字节流响应处理 BinaryHttpResponseHandler
- ✓ 内置 Json 文本响应处理 JsonHttpResponseHandler
- ✓ 通过实现抽象类 BaseJsonHttpResponseHandler 可以无缝对接三方 Json 解析库
- ✓ 支持各种语言和 content 编码, 不止是 UTF-8

➤ 采用 Glide 库

- ✓ Glide 是一个非常成熟的图片加载库,他可以从多个源加载图片,如:网路,本地,Uri等,更重要的是他内部封装了非常好的缓存机制并且在处理图片的时候能保持一个低的内存消耗。
- ✓ 一个明显的优点就是它可以加载 gif 图片
- ✓ 可以有效的防止 00M 错误
- 程序采用 MVC 框架。用一种业务逻辑、数据、界面显示分离的方法组织代码,将业务逻辑聚集到一个部件里面,在改进和个性化定制界面及用户交互的同时,不需要重新编写业务逻辑。
 - ✓ 多个视图可以对应一个模式。按 Mvc 设计模式,一个模型对应多个视图,可以减少代码的复制及代码的维护量,一旦模式发生改变,也易于维护。
 - ✓ 模式返回的数据与显示逻辑分离。模式数据可以应用任何的显示技术,例如,使用 JSP 页面, Velocity 模版或者直接产生 Excel 文档等。
 - ✓ 应用被分隔为三层,降低了各层之间的耦合性,提供了应用的可扩展性。
 - ✓ 控制层的概念也很有效,由于它把不同的视图组合在一起,完成不同的请求。
 因此,控制层可以说是包含了用户请求权限的概念。
 - ✓ 更符合软件工程化管理的精神。不同的层各司其职,每一层的组件具有相同的特征.有利于通过工程化和工具化产生的管理程序代码。

项目架构



2. 深度学习技术实现

2.1 环境

● 操作系统: Windows 10

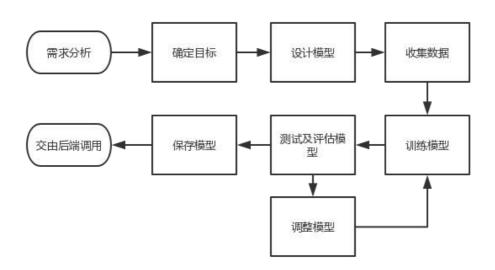
● 开发语言: Python3.5

● 开发环境: Anaconda3; Spyder

● 使用框架: Tensorflow

● 其它依赖: Numpy; Pandas; Matplotlib

2.2 技术路线



2.3 技术亮点

- Python3 是趋势且各主流框架都已支持。
- Tensorflow 是 Google 的深度学习框架,大大降低了深度学习在各个行业中的应用难度。

2.4 解决方案

主要考量了优秀读者的智能评判以及书籍的智能推荐这两个模型。

● 优秀读者

我们的算法采用了线性回归算法,

$y' = b + w_1x_1 + w_2x_2 + w_3x_3$

 $y' = b + w_1 x_1 + w_2 x_2 + w_3 x_3$

我们依据学生阅读的时间、阅读的时长、批注的种类与分布情况、批注字数、以及可选的教师打分等特征来智能判别优秀读者并进行排行。

我们的团队对批注的种类以及用法有着详细而又细致的划分,也因此造成了批注特征的复杂性,因此我们也引入了 L2 正则化来简化我们的模型。

我们依据学生阅读的时间、阅读的时长、批注的种类与分布情况、批注字数、以及可选的教师打分等特征来智能判别优秀读者并进行排行。

 $L = {w 1^2 + w 2^2 + ... + w n^2}$

 L_2 regularization term = $||\boldsymbol{w}||_2^2 = w_1^2 + w_2^2 + ... + w_n^2$

同时,为了使输出结果易于展示与比较,我们采用了 S 型函数进行了处理,将优秀学生的评判标准量化打分。

 $y = \frac{1}{1 + e^{-z}}$

$$y = \frac{1}{1 + e^{-z}}$$

在该模型中,使用均方误差 (MSE) 作为训练与中的损失函数。

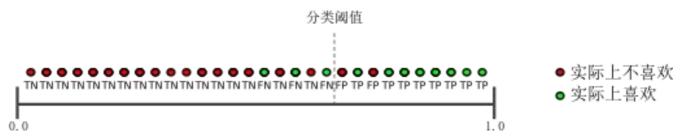
 $MSE = \frac{1}{N} \sum_{(x, y) \in D} (y - prediction(x))^2$

$$MSE = \frac{1}{N} \sum_{(x,y) \in D} (y - prediction(x))^2$$

● 书籍推荐

关于优秀书籍的推荐, 我们认为相较于未能把所有优秀的读者喜欢的书籍推荐给读者, 将读者不喜欢的书籍推荐给读者更影响读者的阅读体验。

所以,我们相对于召回率(真正例占真正例与假负例的比重),更加重视提高精确率(真正例占真正例与假正例)。在尽量减少误推荐的可能状态下再考虑推荐给读者更多的书籍,因此我们提高了逻辑回归的分类阙值。



在该模型中,使用对数损失函数作为训练与中的损失函数。

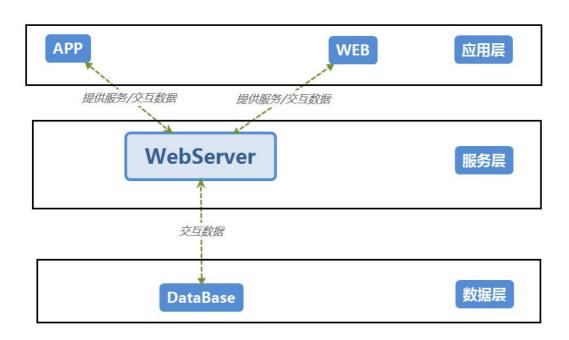
\$Log Loss = \sum_{(x,y)\in D} -ylog(y') - (1 - y)log(1 - y')\$
$$LogLoss = \sum_{(x,y) \in D} -ylog(y') - (1 - y)log(1 - y')$$

3. 后端管理技术实现方案

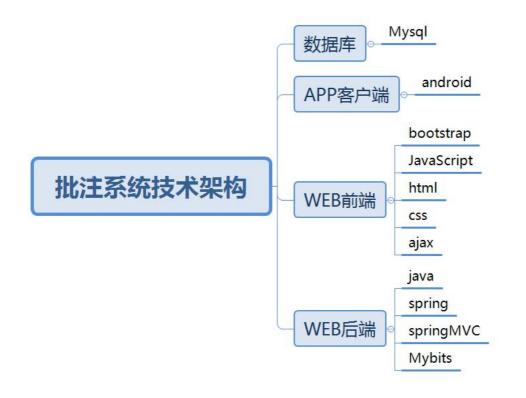
3.1 编写目的

为了给 APP 端和 Web 提供服务以及使管理人员更好的管理批注系统而计划开发的系统

3.2项目整体架构



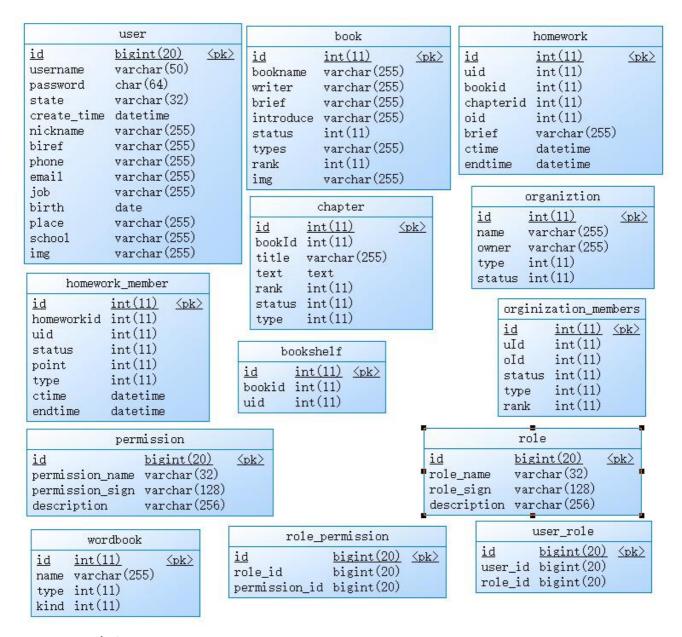
3.3 项目实现技术架构概要



3.4详细实现方案

3.4.1 数据库构建

数据库使用 Mysql 建立,本项目数据库模型如下



3.4.2 Web 前端

● 开发目的:为了方便管理员能对整个系统进行数据管理而设计的基于网页的用户交互界面

● 开发工具: idea

● 开发语言: JavaScript html css

● Web 应用服务器: tomcat

● 技术亮点:

(1) 本项目使用了 bootstrap 前端框架, Bootstrap, 来自 Twitter, 是目前很受欢迎的前端框架。 Bootstrap 是基于 HTML、CSS、JavaScript 的,它简洁灵活,使得 Web 开发更加快捷。它由 Twitter 的设计师 Mark Otto 和 Jacob Thornton 合作开发,是一个 CSS/HTML 框架。Bootstrap

提供了优雅的 HTML 和 CSS 规范,它即是由动态 CSS 语言 Less 写成。Bootstrap 一经推出后颇受欢迎,一直是 GitHub 上的热门开源项目,包括 NASA 的 MSNBC(微软全国广播公司)的 Breaking News 都使用了该项目。国内一些移动开发者较为熟悉的框架,如 WeX5 前端开源框架等,也是基于 Bootstrap 源码进行性能优化而来。

- (2) 整体界面简洁大方,操作简便易懂,用户体验极佳,由于采用了响应式布局设计,使得此网站在移动端设备上也有很好的表现。
- (3) 使用了Ajax 技术即 "Asynchronous Javascript And XML" (异步 JavaScript 和 XML),是指一种创建交互式网页应用的网页开发技术。Ajax 是一种在无需重新加载整个网页的情况下,能够更新部分网页的技术。通过在后台与服务器进行少量数据交换, Ajax 可以使网页实现异步更新。这意味着可以在不重新加载整个网页的情况下, 对网页的某部分进行更新。传统的网页(不使用 Ajax)如果需要更新内容,必须重载整个网页页面。能够很明显的提升系统的运行效率和用户体验。

3.4.3 WebServer

● 开发目的:为了给 APP 端和 WEB 前端提供网络服务已经和数据库交互数据而开发

● 开发语言: java

● Web 应用服务器: tomcat

● 技术亮点:本项目基于 SSM (spring MVC +spring+mybatis) 框架开发而成。

• Spring:

spring 是一个开源框架,Spring 是于 2003 年兴起的一个轻量级的 Java 开发框架,由Rod Johnson 在其著作 Expert One-On-One J2EE Development and Design 中阐述的部分理念和原型衍生而来。它是为了解决企业应用开发的复杂性而创建的。Spring 使用基本的 JavaBean 来完成以前只可能由 EJB 完成的事情。然而,Spring 的用途不仅限于服务器端的开发。从简单性、可测试性和松耦合的角度而言,任何 Java 应用都可以从 Spring 中受益。 简单来说,Spring 是一个轻量级的控制反转(IoC)和面向切面(AOP)的容器框架。

Spring MVC:

Spring MVC 属于 SpringFrameWork 的后续产品,已经融合在 Spring Web Flow 里面。Spring MVC 分离了控制器、模型对象、分派器以及处理程序对象的角色,这种分离让它们更容易进行定制。

MyBatis:

MyBatis 本 是 apache 的 一 个 开 源 项 目 iBatis, 2010 年 这 个 项 目 由 apache software foundation 迁移到了google code,并且改名为MyBatis。MyBatis是一个基于Java的持久层框架。iBATIS 提供的持久层框架包括 SQL Maps 和 Data Access Objects (DAO) MyBatis 消除了几乎所有的 JDBC 代码和参数的手工设置以及结果集的检索。MyBatis 使用简单的 XML 或注解用于配置和原始映射,将接口和 Java 的 POJOs (Plain Old Java Objects, 普通的 Java 对象)映射成数据库中的记录。

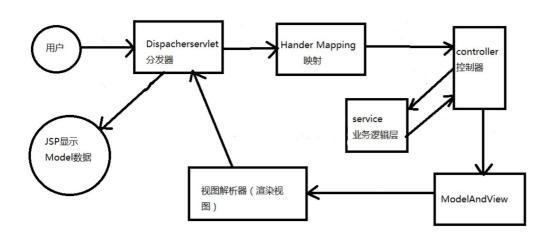
3.4.4作用

SSM 框架是 spring MVC, spring 和 mybatis 框架的整合,是标准的 MVC 模式,将整个系统划分为表现层,controller 层,service 层,DAO 层四层.使用 spring MVC 负责请求的转发和视图管理。spring 实现业务对象管理,mybatis 作为数据对象的持久化引擎

3.4.5 原理

> Spring MVC:

- (1) 客户端发送请求到 DispacherServlet (分发器)
- (2) 由 DispacherServlet 控制器查询 HanderMapping, 找到处理请求的 Controller
- (3) Controller 调用业务逻辑处理后, 返回 Model And View
- (4) DispacherSerclet 查询视图解析器,找到 ModelAndView 指定的视图
- (5) 视图负责将结果显示到客户端



Spring

它可以装载 bean (也就是我们 Java 中的类, 当然也包括 service dao 里面的), 有了这个机制, 我们就不用在每次使用这个类的时候为它初始化, 很少看到关键字 new。另外 spring 的 aop, 事务管理等等都是我们经常用到的。

Mybatis

mybatis 是对 jdbc 的封装,它让数据库底层操作变的透明。mybatis 的操作都是围绕一个sqlSessionFactory实例展开的。mybatis 通过配置文件关联到各实体类的 Mapper 文件, Mapper 文件中配置了每个类对数据库所需进行的 sql 语句映射。在每次与数据库交互时,通过sqlSessionFactory拿到一个sqlSession,再执行sql 命令。

3.4.6 工程结构目录

controller

- ▼ controller
 - ▼ 🛅 system
 - © b UserController
 - © La AuthController
 - © BookController
 - © b DataChart
 - © h HomeController
 - © > HomeworkController
 - C LabaelController
 - © b orginizationController
 - © PageController
 - © ShareddataController
 - c testController

> dao

- ▼ 🛅 dao
 - BookMapper
 - BookshelfMapper
 - ChapterMapper
 - HomeworkMapper
 - HomeworkMemberMapper
 - OrganiztionMapper
 - OrginizationMembersMapper
 - PermissionMapper
 - RoleMapper
 - RolePermissionMapper
 - UserMapper
 - UserRoleMapper
 - WordbookMapper

entity

▼ I model

- C & Book
- © b BookExample
- © Bookshelf
- © bookshelfExample
- Chapter
- ChapterExample
- C Homework
- © homeworkExample
- HomeworkMember
- © 🖫 HomeworkMemberExample
- © 🔓 Organiztion
- © 🔓 OrganiztionExample
- OrginizationMembers
- OrginizationMembersExample
- © Permission
- © PermissionExample
- C % Role
- © la RoleExample
- © B RolePermission
- © % RolePermissionExample
- C b User
- © b UserExample
- C & UserRole
- © b UserRoleExample

mapper

- ▼ 🛅 mapper
 - BookMapper.xml
 - BookshelfMapper.xml
 - ChapterMapper.xml
 - HomeworkMapper.xml
 - HomeworkMemberMapper.xml
 - OrganiztionMapper.xml
 - OrginizationMembersMapper.xml
 - PermissionMapper.xml
 - RoleMapper.xml
 - RolePermissionMapper.xml
 - UserMapper.xml
 - UserRoleMapper.xml
 - ₩ordbookMapper.xml