

2017 年夏季 Java 小学期大作业

秦岳 2015011333

陈宇 2015011343

贾越凯 2015011335

1 功能实现

	功能	子功能	百分比 (%)	是否实现
基础功能	系统支持	要保证程序在安卓机上正常运行，测试过程中程序不崩溃。	5	是
	页面布局	布局合理，点击处理正确	10	是
	分类列表	删除和添加操作	10	是
	新闻列表	正确显示新闻列表的消息，布局和展示，点击进入新闻详情页面正确。	10	是
		实现新闻的本地存储，看过的新闻列表在离线的环境下也可以浏览	10	是
		上拉获取更多新闻	5	是
		新闻是否看过的页面灰色标记	5	是
		新闻搜索	5	是
	分享收藏	使用微信、微博等 SDK 分享，新闻详情页面点击分享可以分享到常用的 app，分享内容带有新闻摘要、URL 和图片	10	是
		新闻详情页面点击收藏的添加和删除，实现收藏新闻的本地存储。收藏也的正确展示，点击可以进入新闻详情等	10	是
加分功能	新闻推荐	根据用户看过的新闻推荐相关的新闻，参考今日头条等	10	是
	语音播报	可以语音读出新闻等	20	是
	新闻人物地点链接	用户可以跳转到相应的百科词条等	5	是
	用户体验	新闻屏蔽功能，通过进一步询问用户想要屏蔽掉关于什么内容的新闻而实现基于关键词的屏蔽等	5	是
		夜间模式，用户可以调整背景色等	5	是
		文字模式和图片模式转换，文字模式不显示图片，帮助用户节省流量。	5	是

		流畅性强	视效果而定	是
		界面颜值高	视效果而定	是
		基于机器学习的图像填补与广告屏蔽与画质增强（对于服务器中无配图或配图质量过差，自动用优秀的图片替换并增强）	视效果而定	是
	代码和项目管理	使用了较好的框架（RxJava）	酌情加分	是
		有较完整的单元测试 (Junit4)	5	是
		使用了 github 等好用的代码版本管理工具	2	是
		使用了 gradle 项目管理工具	2	是
		使用了出色的设计模式（MVP）	酌情加分	是
		使用了优秀的动态 UI 设计语言（Material Design）	酌情加分	是
	网络检测	网络环境过差无法加载新闻时会进行提示与重新加载选项，提高用户交互体验	希望获得 2 分	是
	稀有词检测	通过 Wiki 百科词条进行数据挖掘，额外检测专有名词等，提供百度百科链接	希望获得 2 分	是
	清空缓存	帮助用户节约存储空间，及时清除新闻缓存	希望获得 2 分	是
	检查更新	帮助用户及时检查客户端是否是最新版	希望获得 2 分	是
	下拉刷新自动筛选	帮助用户在同一栏目浏览到不同的新闻，提供下拉方式的交互体验	希望获得 2 分	是
	关于我们	帮助用户了解我们，提高界面颜值与交互体验	希望获得 2 分	是

2 小组分工

功能	实现人
服务端通信接口实现与封装	秦岳
分享功能注册、集成及封装	秦岳
语音功能注册、集成与封装	秦岳
智能推荐系统实现与封装	秦岳
关键词挖掘与匹配算法实现与封装	秦岳
图像挖掘与识别后处理	秦岳
配色与图形艺术设计	秦岳
单元测试	秦岳
配置储存	陈宇
API 管理及整合	陈宇
新闻收藏、已读记录	陈宇
离线数据管理	陈宇
图片搜索	陈宇
百度词条可用性检查	陈宇
屏蔽词过滤	陈宇
新闻列表展示	贾越凯
收藏列表展示	贾越凯
新闻搜索展示	贾越凯
新闻详情展示	贾越凯
上拉获取更多新闻	贾越凯
设置页面	贾越凯
分类列表以及添加删除操作	贾越凯

3 具体实现

注：全部功能均在 5 款不同配置的 Android 手机上进行了测试，全部运行良好。检查时若发生功能无法正常使用 的情况请及时联系组长，以排除外部影响(如微博授权问题等)，过于简单的功能在本文中有所省略

项目地址：https://git.oschina.net/_yuguo_tianqing/Knows

秦岳：

分享功能：

集成 Github 第三方开源库 ShareSDK，在 QQ/微信/微博开放平台进行应用注册，并在本地配置相应的 APPKEY 等, 简单的实现窗体展示、回调接口和异常处理等，并修复了第三方库中关于版本问题的若干 BUG。

语音功能：

集成科大讯飞在线语音合成 SDK，并进行注册，简单封装调用接口。

基于内容的推荐系统：

由于用户使用次数少，缺少必要的大数据，故协同过滤等推荐算法并不适用，最终采用基于内容的推荐算法，具体实现如下：使用服务端已经提供了新闻的文本向量与类别等标签，基于 Bag-of-words 模型，以余弦相似度作为距离度量，实现了 Rocchio 相关性反馈算法。考虑到新闻数量较少，故不采用高效的数据结构与算法即可实时维护用户的浏览记录、收藏记录与屏蔽词的影响，进行排序推荐。为了避免重复推荐，加入了一些简单的筛选策略。

效果截图（在收藏了一篇经济相关的新闻之后）：

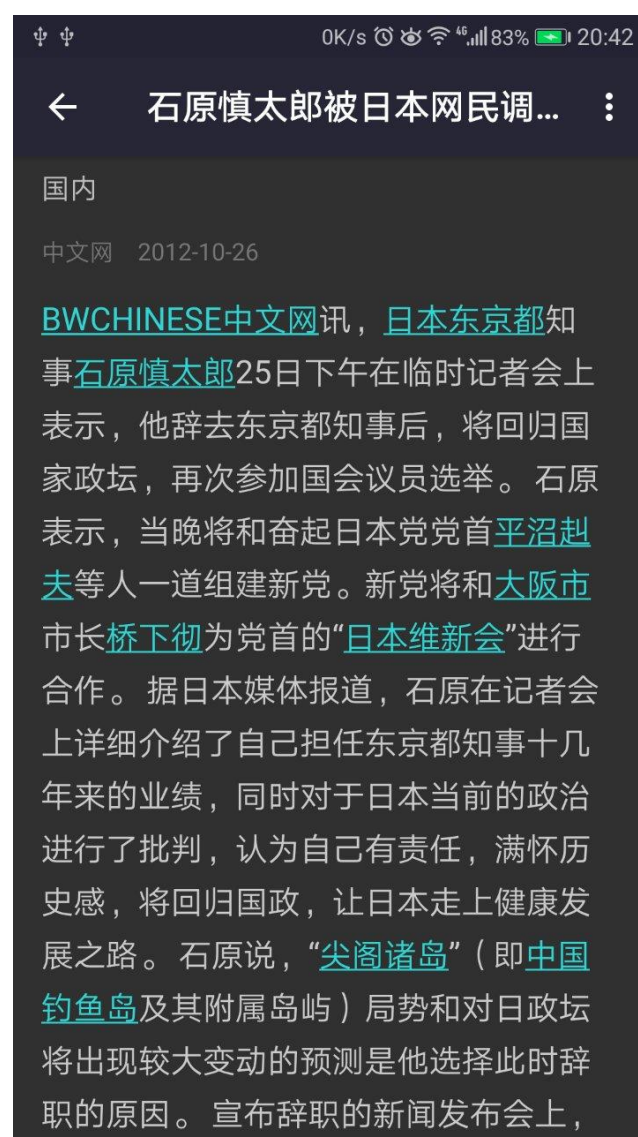


关键词挖掘与匹配算法：

由于服务端提供的地名人名过于简陋且有大量错误，故独立实现了文本挖掘算法并将数据库集成与 APP 中。以 Wiki 百科开源数据库为基础，通过爬虫爬取百度百科词条介绍，综合开源汉语词典数据库，进行评判新闻中有价值的词表。首先根据百度百科词条的存在性进行一步筛查，再根据汉语词典和百科词条进行词性与常用性筛查，可去除例如”尊敬”这种常用词，在根据新闻数据库进行比对筛查保留合适的词表。最终用词表建立数据结构，在展示新闻详情时动态标注高亮。（为了避免高亮词过多，在展示时只保留了少数词汇）

由于频繁涉及大规模多串匹配，正则表达式的时间复杂度无法达到要求，故实现了 Aho-Corasick automaton 数据结构的线性时间匹配算法。

效果截图：



图像挖掘与识别后处理：

智能补图算法中，对于若干张候选图片，如何有效鉴别哪些是错误图片，并选择一张最好的缩略图是一个十分困难的问题。通过观察分析陈宇爬取的图片库中的图片与服务端图片库，设计了一个简单而有效的决策树分类方法，结合人类认知的先验与数据样本的分布，通过人为设置长宽/色彩饱和度/纯色聚合度等直观特征的基础上通过简单的学习方法得到综合搜索结果与服务端结果的分类模型，进行离线与在线结合的图片标识与缓存，在尽可能确保有效性的基础上提高画面质量。

对比效果截图：



陈宇：

配置储存

APP 将所有配置（包括夜间模式、文字模式等）序列化为一个 JSON 格式的文件进行读取和访问。

数据储存

为了实现离线使用的功能，并且提高 APP 的流畅度，APP 的后台管理在每次从网络上获取到数据之后均会放入数据库中，使用 **sqlite** 进行存储；用户使用的时候，优先使用网络获取的数据，如果获取失败则自动从数据库中载入数据，从而实现离线访问。

在程序运行启动时，会启动线程从数据库中加载，并向用户展示 **logo**，等到载入完成再切换到主界面；但是，查找百度链接的模块（基于自制的词典，使用 **AC** 自动机进行查找）初始化速度太慢，并不能将全部的任务放置到载入界面，最终的解决办法是在程序启动时启动一个线程做初始化，等到打开详情页面时再等待线程的结束，由于用户打开详情页面的速度很慢，所以此时已经做好了初始化，并不会影响用户的使用。

后台管理

后台管理包括提供各种新闻查询的接口，以及已读和收藏等接口。

后台管理使用 **RxJava** 框架，基于该框架，可以非常方便的将获取数据的过程放在 **IO** 线程，而将界面更新放在主线程。并且，**RxJava** 采用类似流水线的工作模式，比如获取新闻数据：首先是要从网络上获取数据，如果获取失败则从数据库中载入数据，然后，每个新闻对象都需要判断是否已读和是否已收藏，而第一个新闻和第二个新闻的这些属性是完全分开的，所以这些任务可以同时进行，而 **RxJava** 的调度器则能够实现这个目的。

此外，使用 **RxJava** 进行后端和前端的通信，可以极大的降低耦合性，提高开发效率。

图片搜索

考虑到有些新闻没有图片，所以 APP 对于没有图片的新闻进行了智能补图：先获取新闻内容的关键字，然后用该关键字去图片搜索引擎上搜索，最后选择较好的一张作为新闻的图片；一开始时打算使用百度的图片搜索引擎，但是百度的图片链接有防盗链设置，虽然可以获取到搜索结果，但是并不能在 APP 中正常展示，最终确定使用搜狗的图片搜索引擎。

百度百科词条可用性判断

为了实现百度词条跳转功能，需要对分析出的名词进行判断，过滤掉不能访问的名词；APP 的判断方式是尝试访问百度百科的连接，根据返回结果判断是否可用；同时，为了加快这个步骤的速度，使用了 **RxJava** 对不同词条同时进行处理，同时会将判断结果记录到数据库中，方便下次使用。

贾越凯：

页面结构

本 App 共有 3 个 Activity，分别为 SplashActivity(载入页面)、MainActivity(主页面)和 NewsDetailActivity(新闻详情页面)。

在 MainActivity 使用 4 个 Fragment 实现导航的切换，分别为 NewsFragment(新闻页面)、FavoritesFragment(收藏页面)、SettingsFragment(设置页面)和 AboutFragment(关于页面)。

新闻页面

新闻页面主要由一个 TabLayout 和 ViewPager 构成，用于实现各个新闻类别之间的切换。ViewPager 的适配器返回 NewsListFragment，用于显示每一类的新闻列表。

新闻列表使用的是 RecyclerView 而不是 ListView。RecyclerView 是一种新的视图组，目标是为任何基于适配器的视图提供相似的渲染方式。RecyclerView 内部的 item 可以复用，在列表中项数较多时能大大提高性能。而且，RecyclerView 内置了一些动画效果，当 item 被插入或删除时，能展现出华丽的动画效果。

对于下拉刷新，使用的是 android 内置的 SwipeRefreshLayout，监听 onRefresh 事件即可。

对于上拉获取更多新闻，监听 RecyclerView 的 onScrollListener 事件，然后通过判断最后一个可见 item 是否是列表的最后一个 item，而向服务器请求更多新闻。

对于列表中图片的加载，使用第三方库 universalimageloader 实现，能够在不阻塞主线程的情况下，异步加载出大量图片。

新闻搜索

在新闻页面的标题栏右边有搜索按钮，点击可以让标题栏变成一个搜索框。输入关键字后能过进行新闻的搜索。在搜索过程中，可以随时切换到其他类别的页面，搜索的结果也会相应改变。

新闻详情

新闻详情页面由顶部新闻图片、中部新闻内容和底部操作栏构成。

顶部新闻图片嵌套在一个 CollapsingToolbarLayout 里，这个控件能够让图片随页面滑动而收缩，最终能够收缩到顶部的标题栏。

中部的新闻内容使用一个嵌套在 NestedScrollView 的 TextView 实现。对于新闻内容中的百科链接，使用 SpannableString 加以高亮，并使用 LinkMovementMethod 实现在浏览器中跳转到该链接。

底部的操作栏由收藏按钮、朗读按钮和分享按钮组成，可随页面的滑动自动隐藏或出现。点击收藏可以收藏/取消收藏该条新闻，点击朗读可以调用后台 API 朗读/停止朗读新闻，点击分享可以调用后台 API 分享该条新闻到微博、微信等其他 app。

从 MainActivity 到 NewsDetailActivity 的切换，加入了 ActivityOptionsCompat.makeSceneTransitionAnimation 选项，能够让新闻列表中的缩略图平滑过渡到新闻详情页面的大图。

收藏页面

收藏页面和首页列表一样，使用了 RecyclerView。

设置页面

在设置页面，可以进行夜间模式、文字模式的切换，首页类别的添加删除，过滤关键词的设置以及清空缓存。

对于夜间模式，使用了 AppCompatActivity.setLocalNightMode() 方法实现，同时在切换的瞬间加入了动画效果，提升用户体验。

对于无图模式，会在新闻列表缩略图那里显示一张默认图片，而在新闻详情页面会直接隐藏顶部的图片，从而帮助用户节省流量。

对于首页类别的和过滤关键词的设置，还是用 RecyclerView 实现。同时使用了第三方库 ChipsLayoutManager 和 MaterialChipsInput，实现标签的显示。

关于页面

关于页面提供了本项目的简单介绍和版本信息，能够帮助用户了解我们，提高界面颜值与交互体验。

4 总结与心得

秦岳：在本次大作业实验过程中，我主要负责后端接口的提供与各种复杂功能的实现，极大的锻炼了 Java 编程能力与数据处理能力，积累了 SDK 集成经验与设计经验，也对 Java 有了更加深入的理解，对 android 开发相关知识学习到了很多，感到非常高兴。

贾越凯：这次大作业我主要负责前端页面显示的部分，前端页面需要实现高颜值的界面和良好的用户体验，为此，我学习了 Material Design 的相关知识。在各种操作之间，需要有平滑的过渡，所以要实现很多动画效果，好在 Android 的很多组件都自带华丽的动画效果。此外，前端页面还需要对用户的各种非法操作和后台返回的错误信息进行处理。

通过这次大作业，我对 Java 语言有了更深入的理解，同时也了解了 android 开发的相关知识。一个星期做出这么一个精美的 App 的成就感很大。

关于这个 project，以后能不能换个 app 的类型，每年都是新闻太没意思了。