

2020 年夏季 Java 小学期大作业

2018011359 乐阳

2018011388 王一诺

1 代码结构

本程序实现了大作业的所有基础功能和附加功能。针对新闻查阅、历史记录查看、知识图谱、知疫学者、事件聚类、数据可视化、设定的功能，我们定义了 MainActivity 之下的六个 Fragment，分别为 NewsFragment、HistoryFragment、InfoFragment、ScholarFragment、ClusterFragment 以及 SettingFragment 用以解决最终数据的呈现。

在各个 Fragment 中，本程序采用 MVP 模型，即将每个活动的数据展示 (View) 和数据获取、操作 (Presenter) 分离，Contract 中协商了 View 与 Presenter 之间可以相互调用的接口。Fragment 负责最终的展示，而 Presenter 负责连接前端的视图与后端的模型，将前端的点击等事件转化为数据请求，接受数据并通知前端进行显示的更新，实现了数据与显示与控制的解耦合。

后端向前端提供数据的过程采用事件流 (rxjava 库)，相当于开启新的线程来获取、处理数据。后端发射一个 Single 事件从网络下载数据，期间界面展示“正在加载”图标；下载完毕后，前端作为 Consumer 解析并呈现数据。

具体而言，前端与后端的代码结构如下：

1.1 后端结构

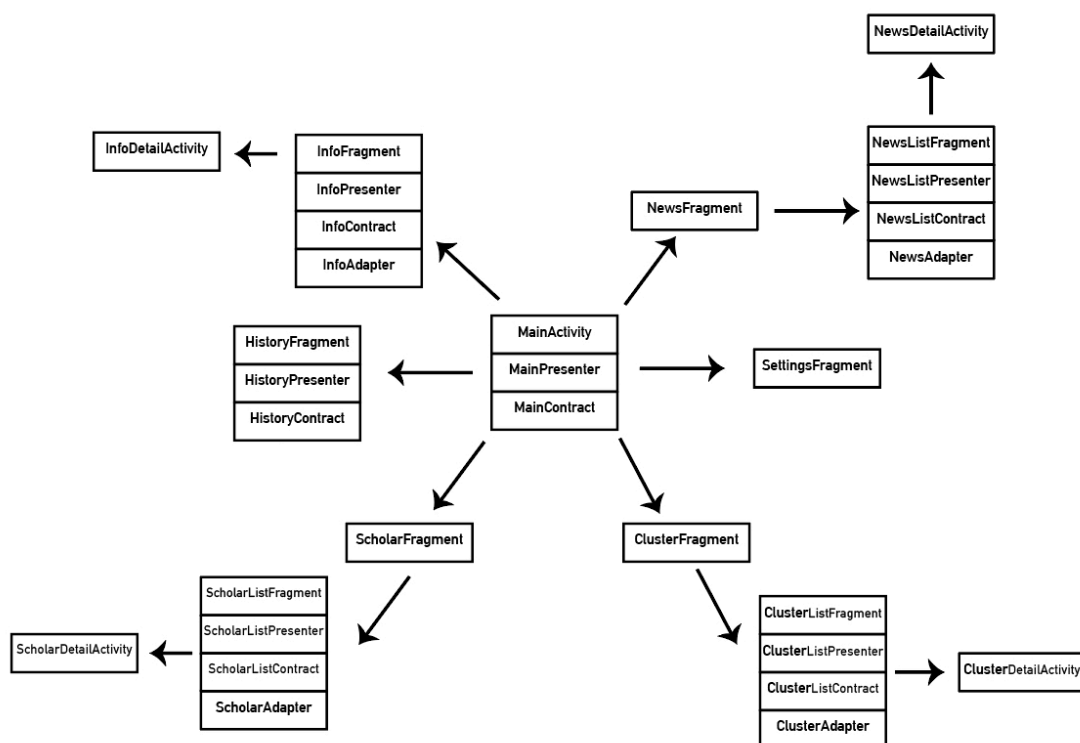
后端功能在 com.java.yueyang.data 包中实现，具体而言有如下 java 文件：

- (1) API：负责通过指定接口从网站申请数据，获得新闻、图谱、学者、数据资料
- (2) Config：负责存放配置信息，例如在新闻界面中可选种类
- (3) COVIDInfo、Epidemic、NewsItem、Scholar：四个类将四个模块所需要的数据类型组织成有序形式
- (4) HistoryDB：数据库管理，能够支持将历史记录的查询、插入、删除功能
- (5) ImageLoader：包装第三方库，实现知疫学者与知识图谱的图像显示
- (6) Manager：后端功能的管理，通过 Manager 以事件流的形式向前端提供所需要的数据任务

1.2 前端结构

前端分为 main、news、history、info、scholar、settings、cluster，各模块

之间的关系如下图。



下面分别进行详细介绍：

1.2.1 com.java.yueyang.main

包中以 MVP 形式进行组织，分为 MainContract、MainActivity、MainPresenter。MainActivity 的布局中包含界面上方的工具栏以及一个左侧呼出的导航栏。程序的六大主要功能都是属于 MainActivity 的 Fragment，通过导航栏进行切换，填充到主界面部分。

1.2.2 com.java.yueyang.news

新闻界面根据新闻类别分栏，每栏内为一个可以滑动刷新的新闻列表，点击新闻列表的一个条目可以查看详情。NewsFragment 为顶层活动，管理搜索栏与不同的新闻分区；下层的 NewsListFragment 以 MVP 形式进行组织，并且建立了 NewsAdapter 对于 NewsListFragment 中的 RecyclerView 进行适配，填充新闻条目。NewsPresenter 负责向后端请求数据（包含搜索功能）；NewsListFragment 负责监听刷新事件以及触发 NewsDetailActivity。NewsDetailActivity 负责显示新闻的具体内容，启动该活动时会将新闻信息作为 Intent 传入，不需要与后端交互。在用户点击一个新闻条目，启动 NewsDetailActivity 的同时也会将该条新闻传入 HistoryDB。新闻详情页有分享功能，将调用安卓原生的分享功能将新闻文本分享到各平台。

1.2.3 com.java.yueyang.history

历史记录的布局复用了 NewsAdapter，以 MVP 的形式进行组织。HistoryPresenter 负责通过 Manager 向 HistoryDB 请求储存的历史记录并且递交给 HistoryFragment；HistoryFragment 负责列表的呈现并且触发打开 NewsDetailActivity。

1.2.4 com.java.yueyang.info

知识图谱的展示仿照新闻界面。用户搜索一个关键词时返回包含该关键词的所有实体（用类似新闻列表的布局来展示）。点击一个实体时会进入实体详细信息（InfoDetailActivity），内部会展示该实体的信息，同时会展示与该实体相关的五个实体，支持用户点击相关实体并跳转到另一实体详情页面。

1.2.5 com.java.yueyang.scholar

知疫学者界面分为“知疫学者”和“追忆学者”两栏，前者展示所有学者信息，后者展示已经故去的学者信息。点击一个学者条目可以查看详情信息，展示了学者的各种指数、单位、教育经历、工作经历、研究领域等。

1.2.6 com.java.yueyang.chart

由于数据可视化功能不涉及复杂的布局调整，仅用一个 Fragment 完成。内含三个 Spinner 让用户选择国家、省份和数据类型，点击“查询”则展示一张折线图，按照自统计以来的日期顺序。

1.2.7 com.java.yueyang.settings

设置界面主要实现了新闻类别增添与删除和清空历史记录功能。

1.2.8 com.java.yueyang.cluster

用一个列表来展示各聚类的几个主要关键词，点击聚类条目可以查看属于该聚类的所有新闻。聚类算法的实现实际上是在 PC 上完成的，将结果存在 assets/cluster.json、assets/events.json 两个文件中。

1.3 聚类算法介绍

首先根据分词后的文本利用 TF-IDF 算法求出每一个词在文本中的权重，设在第 j 个文本中的第 i 个词的权重为 w_{ij} ，则 w_{ij} 表达式为：

$$w_{ij} = (1 + \log(tf_{ij}))idf_i$$

其中 tf_{ij} 为第 i 个词在文档 j 中出现的次数， idf_i 为 $\log \frac{N}{N_i}$ 。

可以看出，在一个文本中独特的词将会被赋予较大的权重。由于文本的词汇过多，在计算 Tfidf 向量时只选择权重较高的 M 个词作为特征，以降低维度。在计算 Tfidf 向量之后，采用 K-Means 算法进行聚类，通过随机选择 K 个初始点，将文档的特征向量按照欧氏距离划分为聚类，并重新调整聚类的中心点，在此之后不断迭代获得 K 个聚类。通过统计在每一个聚类中各个文本标签的出现次数，选择最高频率的 top-k 标签作为聚类的标签，即实现了聚类算法。

2 具体实现

2.1 分工

乐阳同学负责整体大作业的框架设计以及项目进度的控制。在大作业中完成了数据库的设计、传输数据的接口约定，并编写了 Manager 类中的流式处理。在功能方面完成了历史记录、知疫学者、数据可视化的绘图功能实现，以及新闻细

节展示、新闻聚类与知识图谱的细节展示。

王一诺同学主要进行辅助与前端工作，负责前期的开源项目调研、android 版本适配、布局 xml 文件的设计与 API 接口，整理数据可视化的疫情数据信息；并完成了设置界面、分享功能与聚类算法的实现。

DEBUG 工作由全组同学共同完成，乐阳组长修复了 API 接口中 JSON 文件读取的错误；王一诺同学完成了新闻列表中 RecyclerView 无法刷新与加载闪退的 BUG 修复。

2.2 任务的难点

本次大作业我们采用了较为新颖的 MVP 框架与任务流式处理，一方面这些方法使得我们的大作业的框架较为清晰简洁，并且在前后端连接时较为顺畅；但另外一方面这些同样是较为新鲜的事务，需要花费较长的时间进行学习与掌握。

此外在 RecyclerView 的适配器与 TabLayout 中的 PageViewer 同样是课本外的内容，学习的任务量较大，但这些新的布局与视图使得最终的呈现效果良好。

我们的设计主要通过 Fragment 实现功能的切换，减少了 Activity 的数目，降低了结构的复杂度并且提升了运行的流畅度，但在 Fragment 之间切换的布局转换、数据传递同样是之前不了解的领域，最终我们通过查阅 Android 文档与参考开源项目的设计完成了我们自身的大作业设计。

界面的美化虽然不体现在代码行数上，但也占用了一定的时间。如何添加合适的动画、选择合适的配色和图片都需要尝试和探索。

3 总结与心得

面对不十分熟悉的编程语言和完全陌生的框架，本次大作业的起步是很艰难的。新闻界面的编写几乎占据了一半的工作时间。而当新闻列表、详情完成后，其他功能更像是以新闻界面为基础的改动，可以较为快速的完成。

在实现聚类时，两个需要考虑的点是使用 label 还是 seg_text 以及聚类算法的选择。经过实验，我们发现使用 label 得出的结果更符合语义，使用 DBScan 的效果不如 K-means 理想。

两人合作的工作模式能充分的交流，任务分配也较为均匀。整体上来讲项目的完成比较顺利。