新闻搜索系统　项目报告

计 82　郑凯文　2018011314
计 82　孙昭言　2018011311
计 82　张舒丹　2018080122

2021 年 1 月 10 日

目录

1 综述

本项目为从零构建的新闻搜索系统。我们实现了腾讯、新浪新闻的全量与增量爬取，使用 Lucene 作为全文搜索引擎，并在前端完成用户交互、结果展现等。此外，支持一定程度的用户服务，跟踪用户信息并进行推荐。

在系统的部署上，采取分布式的架构，不同服务运行在不同 Docker 容器或服务器上，通过设计 RESTful API 完成交互。

项目的亮点有：

- 新闻爬取全面，覆盖了近一年大部分腾讯新闻与近两年大部分新浪新闻；截止目前，库内新闻总数约 1230W。
- 新闻更新迅速，实现秒级入库与索引；最新新闻覆盖大多数腾讯、新浪首页及各个子频道下新闻列表。
- 系统鲁棒性强，持续部署后维持良好的新闻更新速度与质量。
- 通过维护带有过期时间的 Token 库实现激活码与邮箱注册。
- 前端界面简洁美观，功能丰富，有适当动画；全部页面均为响应式，可通过移动终端访问。
- 支持与百度搜索类似的高级搜索选项。
- 根据用户标签加权推荐首页新闻，准确性更强。

2 需求分析

2.1 概况

项目需求可以分为爬虫、索引/检索、展现三个部分。

2.1.1 爬虫服务

- 全量抓取
- 增量更新

- （扩展）爬虫抓取多新闻站点、多线程
- （扩展）具有一定新闻去重能力

2.1.2 索引/检索服务

- 基于 Lucene 全量、增量索引
- 基于 Lucene 召回、排序
- （扩展）按照时间/相关度对新闻排序，可筛选来源和时间

2.1.3 展现服务

- 搜索主页：搜索框、新闻、用户选项，（扩展）分频道、首页在用户登录时根据点击行为展现其感兴趣的新闻、上拉加载更多
- 搜索结果页：搜索框、搜索结果、分页、标识命中关键词
- 用户注册、登录、退出
- 用户主页：显示、修改注册信息和用户标签
- （扩展）新闻图片展示、适当动画、响应式页面

2.2 重要需求建模

2.2.1 用户注册

用户注册时，前端会对用户名、邮箱地址、密码进行格式检查，并在正确输入图片验证码后才请求后端发送激活邮件，后端根据激活码判断用户邮箱地址的真实性从而完成注册。

2.2.2 搜索获取新闻详情页

搜索时，用户可以选择使用历史记录，也可以在搜索框中输入关键词，搜索行为会记录在历史记录中；搜索结果可以通过自定义搜索方式进行调整；用户找到合适的分页点击进入新闻详情页，点击行为会被记录在用户标签中。

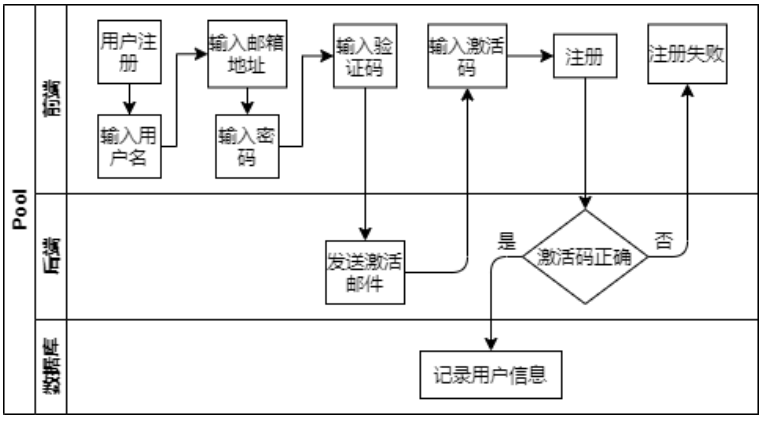

图 1: 用户注册泳道图

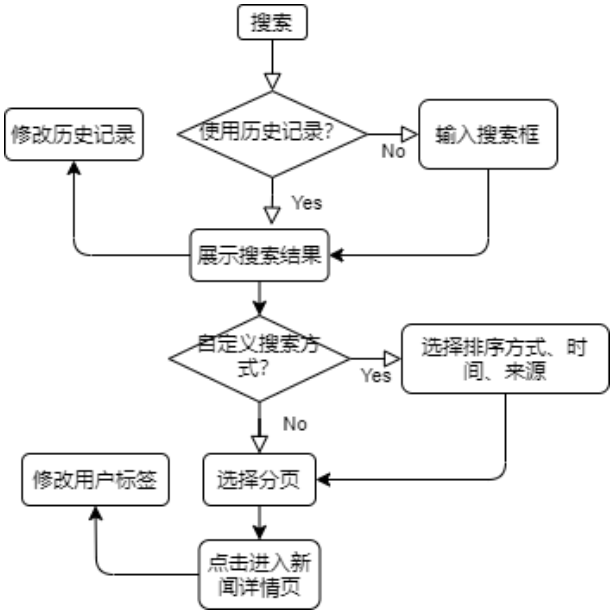

图 2: 搜索获取新闻详情页流程图

3 模块与接口设计

如图3所示，系统由前端、后端、Lucene、数据库、爬虫等几个部分组成，SECoder 平台上将各部分部署为一个或多个独立的仓库。前端采用 Vue 实现；后端采用 Django 实现；索引与检索由 Lucene 提供，并利用 Java SpringBoot 框架进行封装；数据库采用 PostgreSQL，通

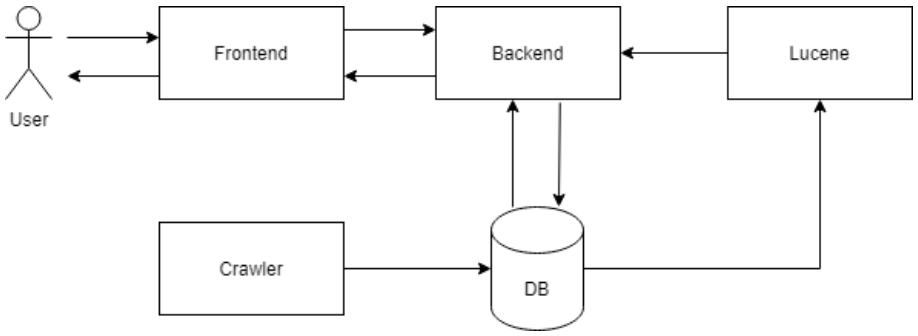

图 3: 系统模块设计

过管理员账户与特定端口进行远程访问；爬虫使用 Scrapy 实现，爬虫的不同部分部署在不同容器中，不对外暴露接口，而直接将爬取的数据存入数据库。

3.1 接口设计

对于有错误信息返回的接口，错误响应均是 HTTP 状态码为 400 的 Json 对象，格式为

```
data: {
    'data': 'error message'
}
```

若未发生错误，则响应的 HTTP 状态码为 200。之后不再赘述。

3.1.1 后端

最新新闻获取 请求：

```
url: '/news'
method: 'get'
data: {
    'start': 0,         //起始位置，不小于0
    'size': 10,         //新闻数量，位于区间(0,100]
    'channel': "history" //频道，没有此字段时默认为所有频道
}
```

响应：

```
data: {
    'data':[            //返回的新闻列表
       {
          'title': "", //标题
          'source': "", //来源
```

```
            'time': "",  //时间
            'content': "", //带有飘红的正文
            'href': "",  //链接
            'image': "", //头图链接，为空时代表没有头图
            'channel': "", //频道
            'tags': "",  //标签
        },
    ]
}
```

新闻搜索 请求：

```
url: '/search'
method: 'get'
data: {
    'start': 0,                  //起始位置
    'size': 10,                  //新闻数量
    'keyword': "xx",             //搜索的关键词
    'approach': "relativity"或"time" //按相关度或时间排序
    //以下两个字段对新闻的起止时间进行过滤
    'timestart': 1605500000      //若有此字段，则代表起始时间的时间戳，即去除小于此时间的新闻
    'timeend': 1605600000        //若有此字段，则代表结束时间的时间戳，即去除大于此时间的新闻
    'hrefcontains': 'new.qq.com' //若有此字段，去除链接中不包含此字段的新闻
}
```

响应：

```
data: {
    'newsnum': 10000,   //检索出的新闻总数
    'data':[            //返回的新闻列表
        {
            'title': "", //标题
            'source': "", //来源
            'time': "",  //时间
            'content': "", //带有飘红的正文
            'href': "",  //链接
            'image': "", //头图链接，为空时代表没有头图
            'channel': "", //频道
            'tags': "",  //标签
        },
    ]
}
```

用户登录 请求：

```
url: '/login'
method: 'post'
data: {
    'username': "xx", //用户名
    'password': "xx", //密码
}
```

响应:

```
data: {}
```

发送验证邮件 请求:

```
url: '/email'
method: 'post'
data: {
    'email': "xx", //邮箱
}
```

响应:

```
data: {}
```

用户注册 请求:

```
url: '/register'
method: 'post'
data: {
    'username': "xx", //用户名
    'password': "xx", //密码
    'email': "xx",  //邮箱
    'authcode': "xx" //邮箱验证码
}
```

响应:

```
data: {}
```

用户修改密码 请求:

```
url: '/update'
method: 'post'
data: {
    'password': "xx", //新密码
}
```

响应：

```
data: {}
```

获取用户信息 请求：

```
url: '/getinfo'
method: 'get'
data: {}
```

响应：

```
data: {
    'tags': "足球,1,1605500000;篮球,1,1605500000;", //标签,访问次数,最后一次访问的时间戳;
    'email':"xxx"                                    //邮箱
}
```

设置用户标签 请求：

```
url: '/settags'
method: 'post'
data: {
    'data': [
        {
            'tag': "篮球",          //标签
            'num': 1,               //访问次数
            'lasttime': 1605500000, //最后一次访问的时间戳
        },
    ]
}
```

响应：

```
data: {}
```

3.1.2 Lucene

新闻检索 请求：

```
url: '/get'
...
```

请求的其余部分及响应与后端的/search 接口完全相同。后端进行更多鲁棒性处理后，将信息转发给 Lucene 引擎的这个接口进行新闻检索。

按标签进行新闻检索 请求：

```
url: '/tags'
...
}
```

请求的其余部分及响应与后端的/search 接口完全相同。这个接口调用 Lucene 对新闻的标签域进行搜索，用于新闻推荐。

3.2 模块设计

3.2.1 前端

前端从页面上可以划分为搜索主页、搜索结果页、用户注册页、用户主页，分别对应 views 文件夹下 Home、Search、Register、User 四个文件。此外，在 components 文件夹下有若干组件供页面调用。前端还使用了 Element 组件库辅助设计。

组件

- AuthCode/SIdentify：生成随机验证码、图片，点击验证码图片或输入错误时更新验证码。
- HomeNewsList/SearchNewsList：显示新闻信息，对新闻无图片、标题/来源过长等情况做了鲁棒性处理；一天内的新闻时间转换为几分钟前、几小时前显示；HomeNewsList 通过监听滚动和距离计算实现上拉加载更多。
- SearchBar：封装了搜索框和历史记录操作；只有在搜索框为空时显示历史记录，边框动画模仿百度；历史记录最上方总是最近搜索项。
- SearchHeader/UserHeader：分别为搜索页、用户页的头部。
- UserInformation/UserPassword/UserTags：对应用户主页的三个分页面，UserTags 封装了对用户标签的操作。

页面

- User：将 UserHeader、UserInformation、UserPassword、UserTags 组装成用户主页。
- Home：将 SearchHeader、HomeNewsList 等组装成搜索主页；通过访问腾讯、新浪热榜实现今日话题模块。

- Search：将 SearchHeader、SearchNewsList 等组装成搜索主页；通过对 Element 组件的调整实现分页和搜索工具。
- Register：实现注册页，通过对 Element 表单组件的调整实现了较复杂的输入框内容检查。

3.2.2 后端

后端主要作用是与前端直接沟通，并将数据库、Lucene 服务等内部模块进行包装。

- 对于检索请求，后端进行鲁棒性处理后转发给 Lucene，并将 Lucene 返回的结果返回给前端。
- 对于用户相关操作，后端直接对信息进行处理后直接查询或修改 User 表。
- 进行新闻推荐时，后端将查询 User 表中用户对应的标签信息，处理后调用 Lucene 按标签检索的接口，进行一定随机化后返回推荐的新闻。
- 用户注册时，后端在发送验证邮件后，会生成并在 Token 表中记录 token-code-expiretime 项，其中 token 是一个长度为 20、由字母和数字组成的随机字符串。token 会被存入用户的 cookie 中，当用户输入邮箱验证码时，将检查 code 是否匹配以及是否过期。

3.2.3 Lucene

Lucene 完成的功能包括增量索引及各种过滤条件下的检索。SpringBoot 框架提供了对后端开放的 RESTful 接口，在收到检索请求时，将调用 Lucene 进行检索、飘红，并将诸如起止时间的过滤条件进行应用。由于 Lucene 收到的只可能是后端的检索请求，而后端已经进行了鲁棒性处理，因此 Lucene 并没有过多考虑边界情况。此外，Lucene 并行创建了一个定时任务，每间隔数秒时间，便会去数据库的 Article 表中获取最新的未索引新闻并进行索引，即增量索引功能。

3.2.4 爬虫

为了爬虫的高效与并行化，我们实现并部署了 7 种类型的爬虫。

- 腾讯新闻链接静态爬虫
- 腾讯新闻链接动态爬虫 1
- 腾讯新闻链接动态爬虫 2

- 腾讯新闻详情页面爬虫
- 新浪新闻链接静态爬虫
- 新浪新闻链接动态爬虫
- 新浪新闻详情页面爬虫

其中，静态爬虫进行非最新新闻的全量爬取，动态爬虫进行最新新闻的快速爬取。腾讯新闻的两个动态爬虫爬取的是两个不同种类页面的新闻。新闻的爬取与解析分为两步：

- 链接爬虫作为生产者，不断爬取新闻列表中的新闻链接与头图，将其存入 Links 表中。
- 详情页面爬虫作为消费者，爬取 Links 表中未解析的链接对应的新闻详情页面，验证页面有效性并进行去重，将有效新闻的正文等详细信息存入 Article 表中。

4 数据库设计

数据库采用 PostgreSQL，部署在单独的服务器上，其它模块对数据库进行远程访问与操作；如 Django 使用 Model 层，Python 爬虫使用 psycopg2。系统共使用了 Links,Article,User,Token 共 4 种类型的表。

表 1: Article 表

列	类型	描述
id	integer	自增序号
title	text	标题
source	text	来源
time	timestamp	发布时间，为加速查询最新新闻，已建立索引
content	text	正文
channel	text	频道，为加速分类查看新闻，已建立索引
tags	text	标签
href	text	链接
image	text	头图链接，为空表示无头图

表 2: Links 表

列	类型	描述
id	integer	自增序号
href	text	新闻链接
image	text	新闻头图
processed	boolean	是否已解析，若解析则不会再被详情页面爬虫处理
valid	boolean	是否有效

表 3: User 表

列	类型	描述
id	integer	自增序号
name	text	用户名
password	text	密码
tags	text	标签，在前端展示并用于新闻推荐
email	text	邮箱

表 4: Token 表

列	类型	描述
id	integer	自增序号
token	text	token
code	text	邮箱验证码
expiretime	integer	过期时间