西華大學

毕业设计说明书

题　　目：基于协同过滤算法的个性化新闻推荐系统

学　　院：计算机与软件工程学院

年级专业：2013 级软件工程

姓　　名：

学　　号：3120130905415

指导教师：

完成时间：　　年　　月　　日

摘要

基于协同过滤算法的个性化新闻推荐系统，使用基于模型的协同过滤算法，通过对用户在网站内的历史操作行为的分析，对用户的兴趣偏好进行预测，为用户推荐可能喜欢的内容。该系统推荐流程为先训练一个表示用户偏好预测的用户模型，根据对用户模型里的用户操作历史的分析，得到关键字和分类权重排序的集合结果进行相同关键字或者相同分类的新闻推荐。

基于协同过滤算法的个性化新闻推荐系统是使用 php 语言、mysql 数据库完成的。测试得到的结果表明，基于协同过滤算法的个性化新闻推荐系统实现了比较准确的个性化新闻推荐，基本符合新闻推荐系统的实际要求。

【关键词】 协同过滤　　新闻推荐　　模型　　个性化

Abstract

Personalized news recommendation system based on collaborative filtering algorithm, using collaborative filtering algorithm based on the model, based on user actions in the history of the site analysis, forecast the user interest preferences, the recommended might like content for the user.The system recommended process first training a user model for preference prediction, based on the user model in the analysis of the history of users, and get the key words and categorization weight sorting collection of results for the same keyword or category news is recommended.

Personalized news recommendation system based on collaborative filtering algorithm is done by using PHP language and mysql database.Test results show that this system has realized the relatively accurate news recommendation, in line with the actual needs of news recommendation systems.

【Key Words】Collaborative Filtering News Recommend Model Individuation

目录

1 绪论

1.1 前言

当今社会已经步入了互联网的大数据时代，人们从以前的获取数据信息匮乏状态逐渐演变成了现在的多方式获取数据信息，而阅读方式也从以前的有什么看什么演变成了喜欢什么看什么。当前获取信息的途径主要为互联网，人们面对互联网的大量的数据，无法快速的从这么多的数据中快速找到喜欢的内容，并且，人们也往往没有明确的阅读兴趣指向，这时我们就需要一个个性化的内容推荐引擎来根据人们的隐性的兴趣指向来为用户推荐其可能喜欢的信息内容，而用户的隐性的兴趣指向，便可以根据用户的一些在网上的行为进行预测。为提高用户的获取想要的信息的速度，用户行为分析和个性化推荐成为了其研究者们重点的研究目标之一。

1.2 主要研究内容

协同过滤分三类，基于用户的协同过滤（user-based）、基于项目的协同过滤（item-based）和基于模型的协同过滤（model-based）。本系统主要目标为实现基于模型的协同过滤的新闻推荐。

系统主要整体流程：

首先，进行新闻内容采集，利用新闻爬虫，抓取新闻之后进行自动提取新闻的关键字，供新闻推荐使用。

其次，用户画像模型的训练，根据用户的操作历史分析出一个可以预测用户偏好的兴趣模型，即形成系统自定的表示该用户近期的兴趣指标的数据集。

最后，进行新闻推荐，根据用户画像模型分析得到一个关联内容的权重排序的集合结果，根据该集合给用户推荐相同关联内容相同的新闻。

2 需求分析

2.1 需求概述

基于协同过滤算法的个性化新闻推荐系统能够根据对用户在网站内的操作记录的分析，为用户推荐可能喜欢的新闻内容。另外，该系统还实现了新闻的新增、改、查、删操作，以及新闻的评论和回复、新闻评论管理等。

2.2 需求功能点概述

根据用户的需求概述，可得到以下功能点：

1）新闻浏览：用户可以在系统前台首页对新闻进行浏览，包括游客（游客不能进行新闻推荐）。

2）评论回复：除游客以外，所有用户可以对新闻进行评论、回复操作。

3）评论点赞：除游客以外，所有用户可以对新闻评论进行店在哪操作。

4）发布者申请：用户可以进入发布者中心，申请成为发布者。

5）发布者内容管理：发布者可以进入发布者中心，发布内容，也可以删除自己发布过的新闻，另外还可以删除自己发布过的评论。

6）关注：用户可以进入个人中心对特定的用户进行关注操作。

7）管理员授权/取消授权：超级管理员可以对普通用户进行普通管理员授权和管理员用户进行管理员取消授权。

8）后台管理员新闻管理：管理员登录后台管理系统后可发布新闻、查看新闻、发布新闻、修改新闻。

9）后台管理员评论管理：管理员可登录后台管理系统查看评论、删除评论。

10）新闻推荐：系统可根据对用户喜好的猜测进行推荐新闻。

2.3 总体用例图

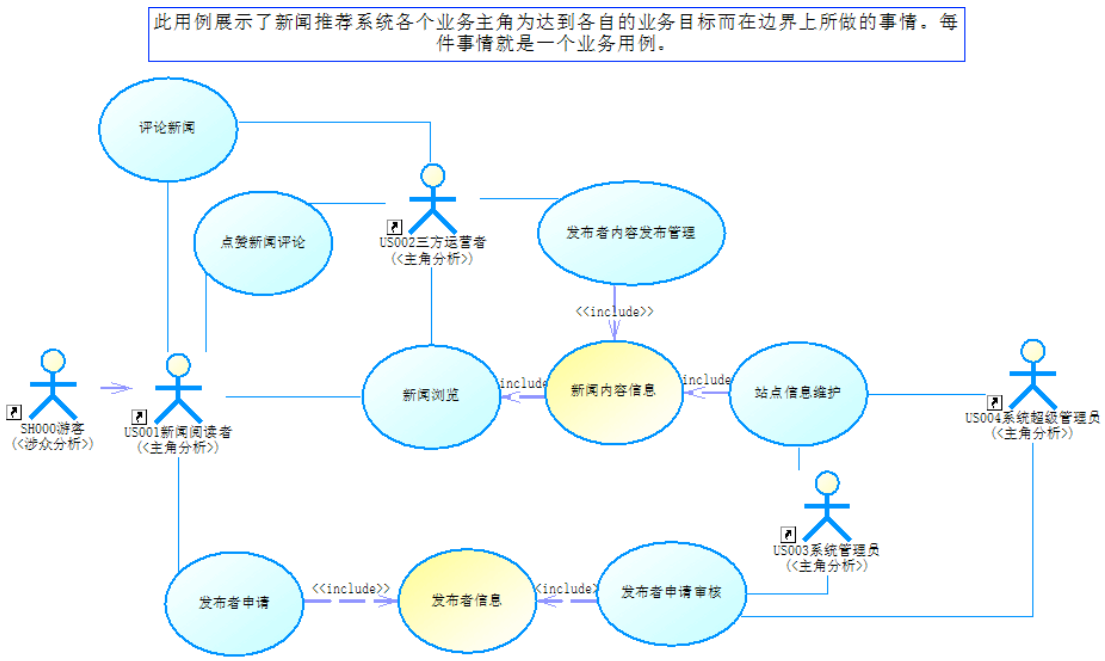

图 2-1 总体用例图

2.4 用例与参与者关系列表

表 2-1 用例与参与者关系表

主要参与者	用例
普通用户	发布者申请
普通用户、发布者、系统管理员、系统超级管理员	查看新闻、评论新闻、点赞新闻评论、评论回复
发布者	删除发布过的新闻、删除发布的新闻的评论
发布者、普通管理员、系统超级管理员	发布新闻
普通管理员、系统超级管理员	新闻管理、评论管理、新闻抓取、发布者申请审核
系统超级管理员	普通管理员授权

2.5 数据库需求概述

根据需求分析概述，应该保存的数据又新闻、用户、评论信息、回复信息、新闻浏览记录、点赞记录、收藏记录、用户关注记录。详细数据如下：

1）新闻：内容标题、内容、所属分类、时间、发布者、封面。

2）用户：手机号码、密码、邮箱、昵称、权限、注册时间、生日、学校、地区、自我介绍。

3）评论时间：内容、评论时间、评论的新闻。

4）新闻浏览记录：浏览新闻、浏览用户、浏览时间、浏览位置。

5）评论点赞：所属评论、点赞用户。

6）用户关注：被关注用户、关注用户。

7）收藏记录：收藏新闻、收藏用户。

3 软件概要设计

3.1 软件模块结构

软件的具体模块如图 3-1。

图 3-1 软件结构图

3.2 软件模块介绍

3.2.1 系统前台模块

3.2.1.1 首页

前台主要模块，新闻列表展示（含个性化新闻推荐结果）、新闻详情展示、相似好友推荐，拥有新闻评论、新闻评论回复、新闻评论/回复点赞操作。

图 3-2 前台-首页模块

3.2.1.2 个人中心

登录状态下显示模块，展示内容包括动态、粉丝、关注、资料、消息、收藏，操作有关注、个人信息编辑、账号密码修改。

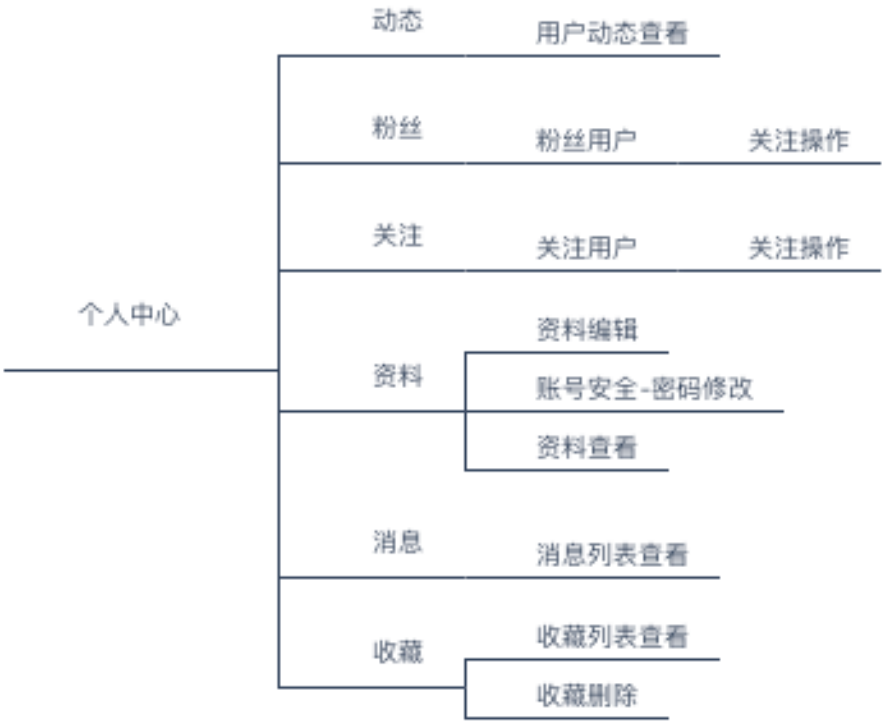

图 3-3 前台-个人中心模块

3.2.1.3 发布者中心

发布者操作模块，包括发布的新闻展示、粉丝指数查看、评论查看、发布者资料查看四个内容展示，发布新闻、删除新闻、评论删除三个操作。

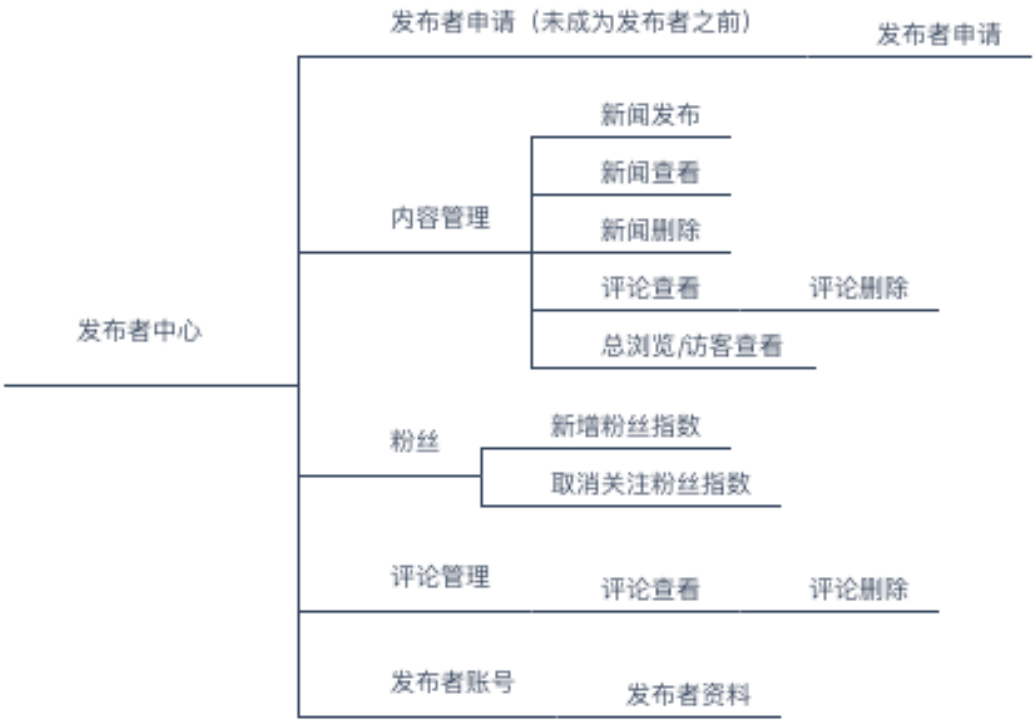

图 3-4 前台-发布者中心模块

3.2.2 系统后台模块

3.2.2.1 首页

后台欢迎页，展示 24 小时内的热门新闻、PV（page view 缩写 pv 或者 PV，下同）和 UV（user view 缩写 uv 或者 UV，下同）可视化数据视图、活跃用户柱状图、平均 PV/UV 柱状图、地区 UV 比例饼图、地区 PV 比例饼图、类型 PV 饼图。

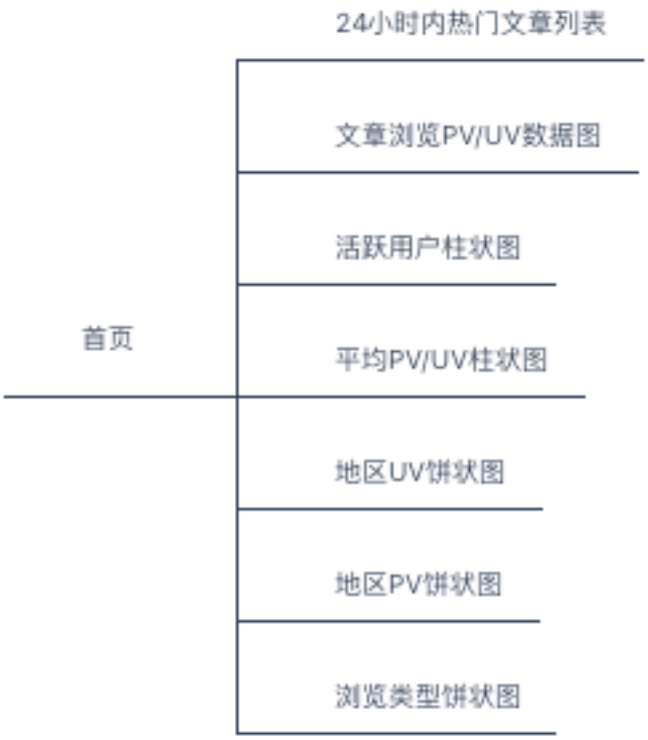

图 3-5 后台-首页模块

3.2.2.2 新闻管理

后台的新闻内容管理中心，管理员在此模块对新闻进行发布、修改、删除。也可在此模块对新闻的分类进行管理，包括分类的增、删、改。另外也可在该模块进行新闻的抓取和前台的离线的新闻相似度计算与保存。发布者发布的新闻审核也在此模块进行。

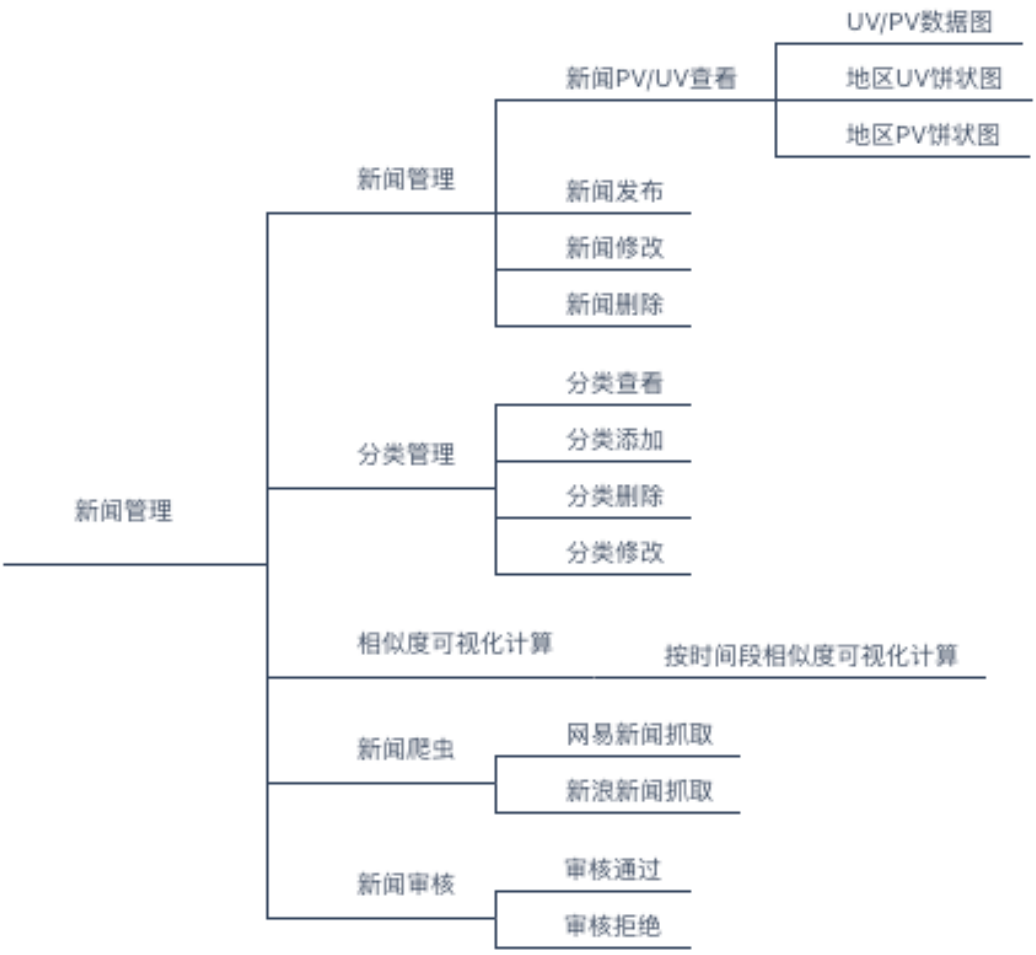

图 3-6 后台-新闻管理模块

3.2.2.3 发布者审核

管理员在该模块中查看普通用户的发布者申请，对其进行通过或拒绝操作。管理员也可在该模块查看已通过或者已拒绝的申请。

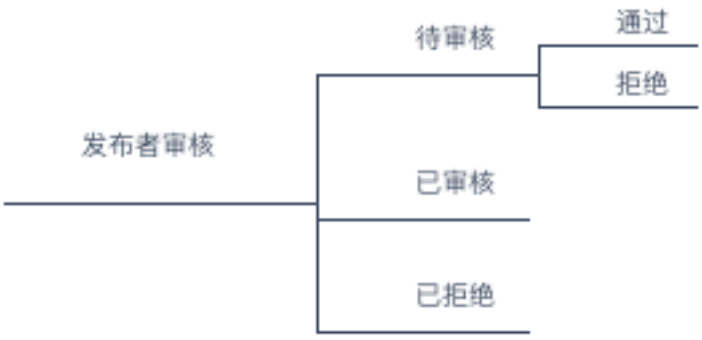

图 3-7 后台-发布者审核模块

3.2.2.4 用户管理

普通管理员在此模块查看用户的列表，超级管理员在该模块对普通用户进行普通管理员授权或者普通管理员的取消授权，另外可在该模块查看用户的相似度力导向图和用户注册的时间统计图。

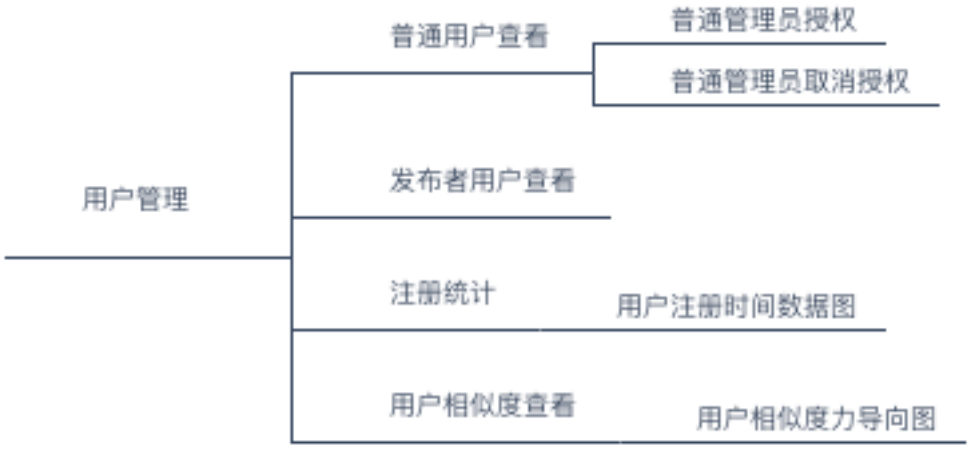

图 3-8 后台-用户管理模块

3.2.2.5 评论管理

管理员在此模块对新闻评论内容进行查看、删除。

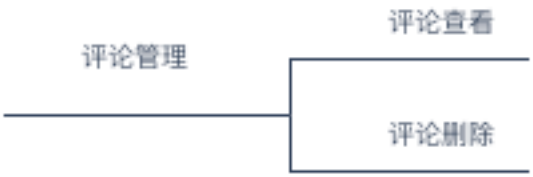

图 3-9 后台-评论管理模块

3.2.2.6 推荐配置管理

系统推荐机制的配置内容管理，管理员在此模块管理推荐权重系数，包括配置增、改、删和启用操作。

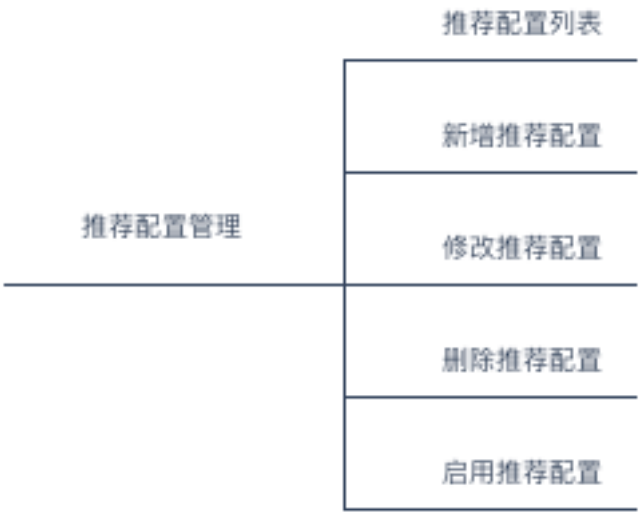

图 3-10 后台-推荐配置管理模块

3.3 数据结构

3.3.1 数据字典

本次设计总过包括 36 个数据表，下面列出主要表设计。

发布者申请表 re_apply，保存发布者申请的内容。

表 3-1 发布者申请表

列名	类型	描述
id	int(11) unsigned	编号
user_id	int(11)	用户编号
name	varchar(50)	申请人姓名
id_number	varchar(30)	身份证编号
file	varchar(255)	图片文件路径
state	int(11)	状态
time	Datetime	时间
email	varchar(50)	邮箱

浏览记录表 re_browse：保存新闻的浏览记录

表 3-2 浏览记录表

列名	类型	描述
id	bigint(20)	编号
news_id	int(11)	新闻编号
ip	varchar(20)	IP 地址
time	Datetime	时间
user_id	varchar(13)	用户编号
area	varchar(100)	地区
type	int(11)	地域判断 1 中国 2 中国不知省份 3 国外

收藏表 re_collection：保存用户的收藏记录。

表 3-3 收藏表

列名	类型	描述
id	bigint(11)	编号
collection_id	int(11)	收藏 ID
user_id	int(11)	用户 ID

表 3-4 评论表

列名	类型	描述
id	int(11) unsigned	编号
content	Text	内容
time	Datetime	时间
delete_tag	bit(1)	删除标识 0 为未删除 1 为删除
news_id	int(11)	新闻 id
reply	int(11)	自关联：评论回复 ID
user_id	int(11)	用户 ID
zan_count	int(11)	点赞数

关注表 re_follow：保存用户关注信息

表 3-5 关注表

列名	类型	描述
id	int(11)	编号
user_id	int(11)	关注的人的 id
follow_id	int(11)	被关注用户 ID
delete_tag	bit(1)	取消关注标识 0 为未取消 1 为取消关注
time	Datetime	时间

用户表 re_login：保存用户的登录信息

表 3-6 用户表

列名	类型	描述
id	bigint(21)	编号
tel	varchar(11)	电话
password	varchar(255)	密码
icon	varchar(255)	头像
nickname	varchar(10)	昵称
email	varchar(40)	邮箱
power	int(2)	权限
reg_time	Datetime	注册时间
userId	bigint(20)	用户信息编号
last_fans_read_time	Datetime	最后粉丝阅读时间
last_message_read_time	Datetime	最后消息阅读时间
last_dynamics_read_time	Datetime	最后动态阅读时间

关键字表 re_news_keyword：保存关键字信息

表 3-7 关键字表

列名	类型	描述
id	int(11) unsigned	编号
keyword	varchar(10)	关键字

新闻关键字表 re_news_keyword_belong：保存关键字和新闻的关系

表 3-8 新闻关键字表

列名	类型	描述
id	int(11) unsigned	编号
news_id	int(11)	新闻标号
keyword_id	int(11)	关键字编号

新闻表 re_news：保存新闻信息

表 3-9 新闻表

列名	类型	描述
id	bigint(1)	编号
title	varchar(255)	标题
content	Text	内容
publish_time	Datetime	发布时间
browse	int(11)	浏览数
state	Int(11)	状态 0 为审核通过 1 为待审核 2 为拒绝
contributor	varchar(13)	发布者
type	int(2)	类型
image	varchar(255)	图片
image_thumb	varchar(255)	压缩图
comment_count	int(11)	评论数
delete_tag	bit(1)	删除标识 0 为未删除 1 为删除

新闻相似度 re_news_similarity：新闻之间的相似度

表 3-10 新闻相似度表

列名	类型	描述
id	int(11) unsigned	编号
news_id1	int(11)	新闻 1
news_id2	int(11)	新闻 2
similarity	Float	相似度
last_modify_time	Datetime	最后计算时间

用户画像表 re_portrayal：保存用户的画像信息，即用户模型

表 3-11 用户画像表

列名	类型	描述
id	int(11) unsigned	编号
user_id	int(11)	用户编号
portrayal	Text	用户画像
last_modify_time	Datetime	最后修改时间

用户相似度表 re_similarity：保存用户相似度信息

表 3-12 用户相似度表

列名	类型	描述
id	int(11) unsigned	编号
user_id1	int(11)	用户 1
user_id2	int(11)	用户 2
similarity	Float	相似度
last_modify_time	Datetime	最后修改时间

点赞表 re_zan：保存评论点赞信息

表 3-13 点赞表

列名	类型	描述
id	int(11) unsigned	编号
comment_id	int(11)	评论编号
user_id	int(11)	用户编号
time	Datetime	时间

推荐配置表 re_recommend_config：保存推荐配置信息

表 3-14 推荐配置表

列名	类型	描述
id	int(11) unsigned	编号
browse_type	int(11)	浏览分类权重
browse_keyword	int(11)	浏览关键字权重
comment_type	int(11)	评论分类权重
comment_keyword	int(11)	浏览关键字权重
zan_type	int(11)	点赞分类权重
zan_keyword	int(11)	点赞关键字权重
follow_keyword	int(11)	关注的人喜欢权重
allow_recommend_time	int(11)	可推荐时间跨度 单位/天
calculate_time_span	int(11)	计算时间跨度，单位/天
display_name	varchar(50)	配置显示名
description	Text	描述
state	bit(11)	1 为启用 0 为不启用

新闻访客表 re_visitor_news：新闻的访客信息

表 3-15 新闻访客表

列名	类型	描述
id	bigint(11)	编号
ip	varchar(20)	IP 地址
area	varchar(100)	地区
news_id	bigint(255)	新闻编号
date	Datetime	时间
user_id	int(11)	用户编号

用户信息表 re_user：保存用户的账号信息

表 3-16 用户信息表

列名	类型	描述
id	bigint(20) unsigned	编号
name	varchar(20)	姓名
birthDate	Date	生日
sex	int(1)	0 表示女，1 表示男，相当于 false 和 true 3 未知
schoolName	varchar(255)	学校
profession	varchar(20)	专业
province	varchar(255)	省份
city	varchar(255)	城市
area	varchar(255)	所在地区
shelfIntroduction	varchar(255)	自我简介
createTime	Datetime	创建时间
modifyTime	Datetime	最后修改时间
deleteTime	Datetime	删除时间
deleteTag	bit(1)	删除标志：0 表示未删除，1 表示已删除

3.3.2 数据模型

3.3.2.1 概念模型

主要概念模型如图 3-11：

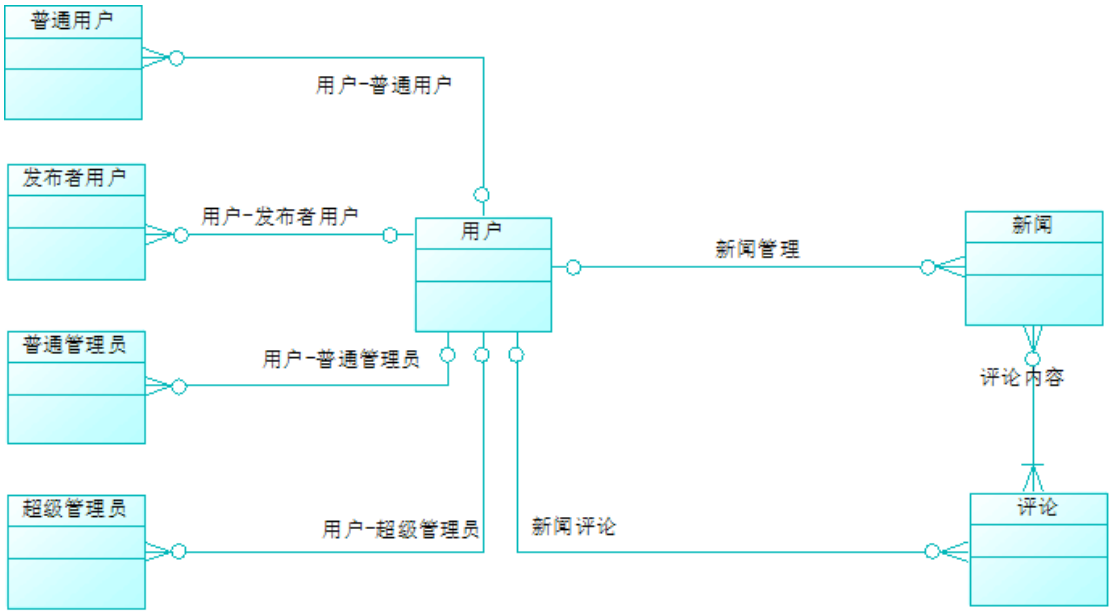

图 3-11 概念模型图

3.3.2.2 逻辑模型

主要逻辑模型如下图 3-12：

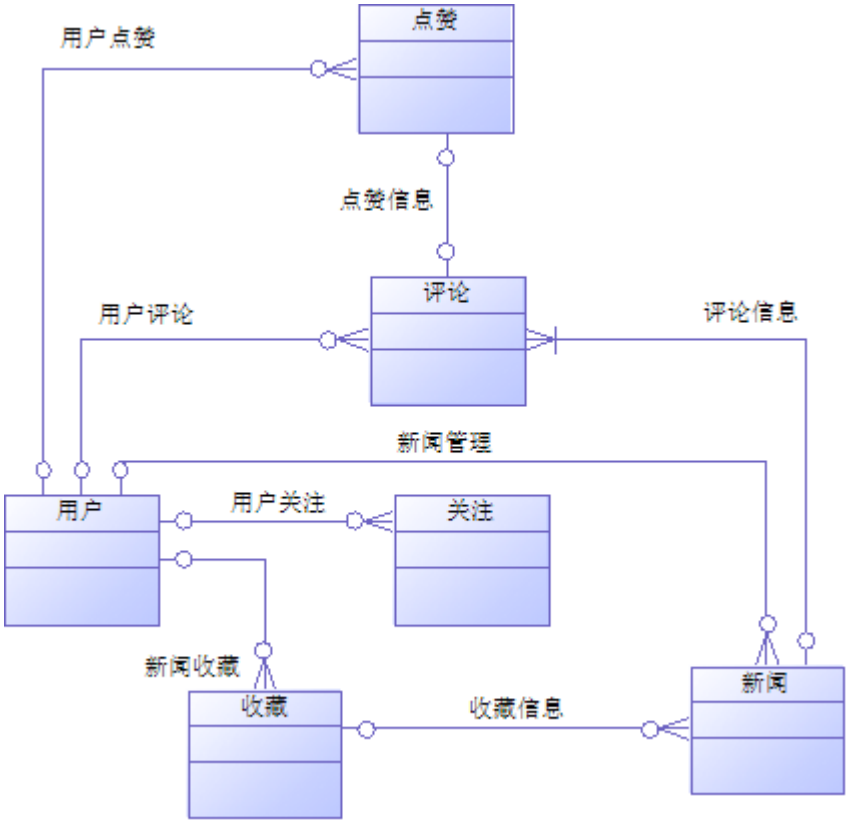

图 3-12 逻辑模型图

3.3.2.3 物理模型图

主要物理模型如下图 3-13：

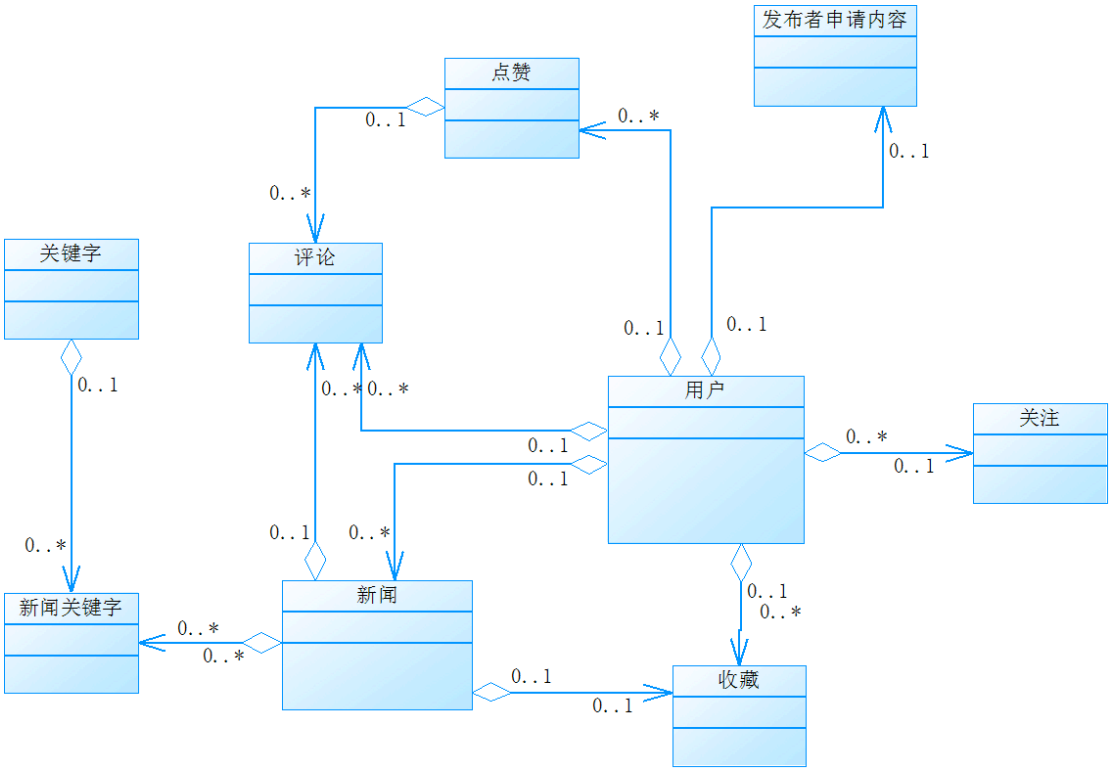

图 3-13 物理模型图

4 系统详细设计

4.1 新闻数据采集功能详细设计

4.1.1 新闻采集理论基础

4.1.1.1 网络爬虫

网络爬虫，一种之前预设好的规则结构，自动的爬取网页内的信息的程序。用比较通俗易懂的话来说，就是利用程序或者脚本打开一个网页，把页面内可以看到的内容保存到自己的数据库中。

4.1.1.2 中文分词

一篇文档的特征，我们可以用若干个词组来描述，这是有道理的。一片文章组成单位从大到小依次是章、篇、段、句、词、字，如果用单个的字作为描述信息的单位，表述的能力会很差；如果用句子做描述单位，包含信息则太精确，泛化的性能会很差。于是词就成了理想的描述单位的选择。文献摘要下面一般请看下都回附有关键词，既方便后续快速查找，也利于搜索引擎的分类与抓取。

中文分词，即将特定的汉字字符串以词作为单位切分。以句子“我刚才吃了午饭”为例，那么比较理想的分词结果是“我/刚才/吃了/午饭”。

4.1.1.3 TF-IDF 自动提取关键字

现假设有一篇长文《中国蜜蜂养殖》，对其进行自动提取关键字[1]。

一个比较容易想到的方法，找到文章中出现次数最多的词。因为如果一个词在一篇文章中文重要，那么它的出现次数不会少，于是可以对其进行"词频"（Term Frequency，缩写为 TF）统计。

在这篇文章中出现次数最多的可能会是"的"、"是"、"在"等等这一类词。他们被叫做"停用词"（stop words），表示会影响到结果、必须过滤的词。

如果已经把他们过滤掉了，剩下一些有实际意义的词，可能会遇到另外一个问题，那就是"中国"、"蜜蜂"、"养殖"这些词的出现次数一样多。那就代表着，作为关键字，他们的重要性一样？

很明显不是这样，因为相对于"蜜蜂"、"养殖"这两个不常见的词而言，"中国"是非常容易看到的词。如果这三个词在文章中出现的次数一样多，可以认为，"蜜蜂"、"养殖"比"中国"的重要程度大，所以，在关键字的排序里，"蜜蜂"、"养殖"的位置要比"中国"靠前。

因此，需要能表示一个词是否常见的系数来调整要计算的关键字的排序关系。一个词，如果很不常见，而且正好在这片文章中的出现次数也不少，可以认为这片文章的一个重要特征就是这个词，即关键字。

在词频的基础上，对出现的词都分配一个系数权重。非常常见的词给最小系数权重，比较常见的词分配较小系数权重，很少见的分配较大系数权重。这个分配的系数权重叫做“逆文档频率”（Inverse Document Frequency，缩写 IDF），大小与一个词的常见度成反比。

了解了"词频"（TF）和"逆文档频率"（IDF）以后，这两个值乘积，就得到一个词的 TF-IDF 值。当 TF-IDF 的值越大的时候，这个词对这片文章越重要，越重要就越可能为这片文章的关键字，所以 TF-IDF 最高的词，就是文章的关键字。

下面就是 TF-IDF 算法的细节。

（一）计算词频。

$$\text{词频(TF)} = \text{某个词在文章中的出现次数}$$

图 4-1 词频表达式

因为文章有长有短，为了更方便的进行对比，进行"词频"标准化。

$$\text{词频(TF)} = \frac{\text{某个词在文章中的出现次数}}{\text{文章的总词数}}$$

图 4-2 词频表达式

或

$$\text{词频(TF)} = \frac{\text{某个词在文章中的出现次数}}{\text{该文出现次数最多的词的出现次数}}$$

图 4-3 词频表达式

（二）计算逆文档频率。

在这个时候，需要一个比较真实的语言的使用环境的库，即语料库（corpus）。此设计中为用来储存词在本系统中的出现情况的库。

$$\text{逆文档频率(IDF)} = \log\left(\frac{\text{语料库的文档总数}}{\text{包含该词的文档数} + 1}\right)$$

图 4-4 逆文档频率

如果一个词很常见，分母就越大，逆文档频率越小。分母加 1 是为了避免分母为 0（即所有文档都不包含该词）。

（三）计算 TF-IDF。

$$\text{TF} - \text{IDF} = \text{词频(TF)} \times \text{逆文档频率（IDF）}$$

图 4-5 TF-IDF 表达式

公式表示，TF-IDF 值与该词在整个语言中出现次数成反比，与该词在文档中出现次数成正比。

4.1.2 新闻采集设计思路

从网易、新浪的滚动新闻页里找到滚动新闻刷新接口，通过该接口获取新闻列表，然后通过列表用爬虫逐一获取目标网址内容，之后将获取到的内容的数据结构统一到自己站点内的数据结构，其中业务逻辑包括文章中文分词、自动关键字获取、站点语料库更新，之后便于推荐引擎使用。

4.1.3 新闻采集实现方法

使用 php 的 curl 调用网易、新浪滚动新闻接口，得到新闻的 javascript 格式数据，用 artTemplate 模板引擎渲染页面，之后使用前端 ES7 的新关键字 async/wait 异步循环新闻队列，后端接收到请求后使用 phpQuery 打开目标地址，抓取其新闻标题、发布时间、内容等信息，利用 phpanalysis 进行中文分词，得到的中文

分词，将所有中文分词写入站点的语料库，并根据中文分词结果和语料库计算出分词的 TF-IDF 值，得到其关键字，之后将所有结果写入库中。

流程图如下：

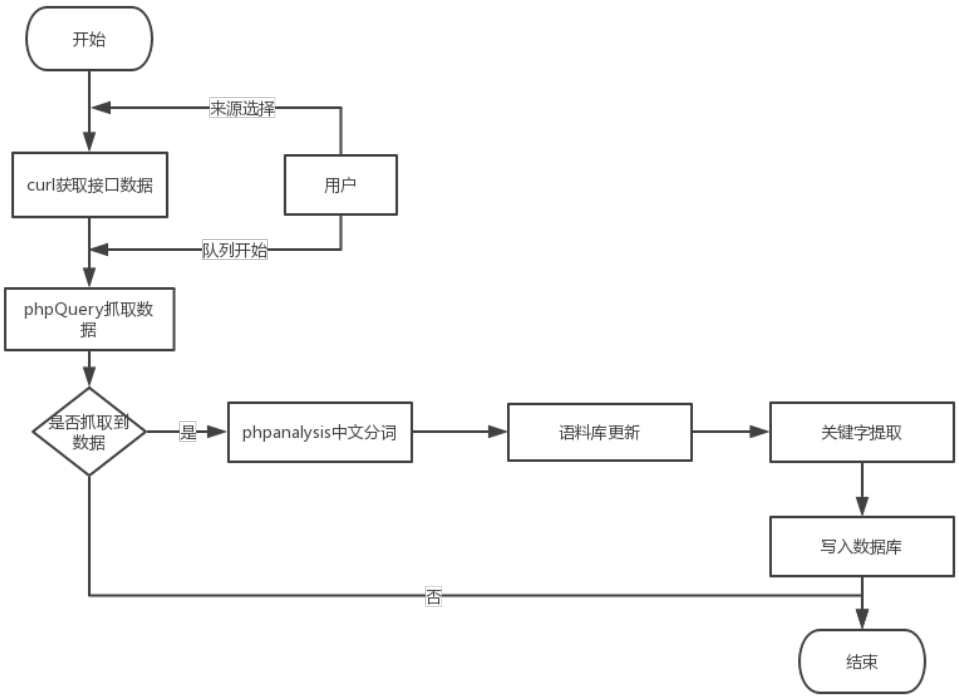

图 4-6 新闻采集流程图

4.1.4 新闻采集核心代码

4.1.4.1 中文分词核心代码

```
//中文分词及获取关键字
public function getWords($content){
    vendor('phpanalysis.phpanalysis');
    // 严格开发模式
    ini_set('display_errors', 'On');
```

```
        ini_set('memory_limit', '64M');
        error_reporting(E_ALL);
        //岐义处理
        $do_fork =    true ;
        //新词识别
        $do_unit = true ;
        //多元切分
        $do_multi = true ;
        //初始化类
        \PhpAnalysis::$loadInit = false;
        $pa = new \PhpAnalysis('utf-8', 'utf-8', $pri_dict);
        //载入词典
        $pa->LoadDict();
        //执行分词
        $pa->SetSource($content);
        $pa->differMax = $do_multi;
        $pa->unitWord = $do_unit;
        $pa->StartAnalysis( $do_fork );
        $result = $pa->GetFinallyIndex();
        $keywords = $pa->GetFinallyKeywords();
        return array(
            'words' => $result,
            'keywords' => $keywords
        );
    }
```

4.1.4.2 TD-IDF 计算核心代码

```
//语料库更新与计算 TD-IDF 获取关键字
    public function keywordCalculate($content){
        //去除 html 中的特殊字符
        $content = str_replace(' ', '', $content);
        $content = str_replace('&rdquo;', '', $content);
        $content = str_replace('&ldquo;', '', $content);
        $content = str_replace('&hellip;', '', $content);
        $content = str_replace('&mdash;', '', $content);
        $content = str_replace('&middot;', '', $content);
        $wordsAndKeyword = $this->getWords(trim(strip_tags($content)));
        $wordLibraryModel = M('WordLibrary');
        $newsCount = M('News')->count(); //文档总数
        $words = $wordsAndKeyword['words'];
        $keywordsStr = $wordsAndKeyword['keywords'];
        $keywords = explode(',',$keywordsStr);
        $keywordsObject = array(); //用于保存出现该词的文档数
        foreach($words as $word => $num){ //写入词料库
            $wordInfo = $wordLibraryModel -> where(array('word'=>$word)) -> find();
            if( $wordInfo ) {
                $wordInfo['num'] = (int)$wordInfo['num']+1;
                if( in_array( $word , $keywords)) { //保存出现该关键字的文档次数
                    $keywordsObject[$word] = $wordInfo['num'];
                }
                $wordResult=$wordLibraryModel->where(array('id'=>  $wordInfo['id']))  ->
save($wordInfo);
```

```
            }else{
                if( in_array( $word , $keywords ) ) {
                    $keywordsObject[$word] = 1;
                }
                $wordResult=$wordLibraryModel-> add(array('word'=>$word,'num'=> 1));
            }
        }
        $wordsCount = count($words); //文章词个数
        $finallyKeywordsInfo = array();
        $i = 0;
        foreach( $keywords  as  $keyword ){
            $wordFrequency = $words[$keyword]/$wordsCount; //词频
            $reverseDocumentFrequency                                                    =
log((int)$newsCount)/($keywordsObject[$keyword]+1); //逆文档频率
            $TF_IDF = $wordFrequency * $reverseDocumentFrequency; //tf-idf
            $finallyKeywordsInfo[$i]['TF_IDF'] = $TF_IDF;
            $finallyKeywordsInfo[$i]['keyword'] = $keyword;
            $i++;
        }
        for( $i = 0 ; $i < count($finallyKeywordsInfo); $i++ ) {
            for ($j = $i+1 ; $j < count($finallyKeywordsInfo) ; $j++ ) {
                if($finallyKeywordsInfo[$j]['TF_IDF']<=$finallyKeywordsInfo[$j+1]['TF_IDF']) {
                    $a = $finallyKeywordsInfo[$j];
                    $finallyKeywordsInfo[$j] = $finallyKeywordsInfo[$j+1];
                    $finallyKeywordsInfo[$j+1] = $a;
                }
```

```
            }
        }
        return $finallyKeywordsInfo;
    }
```

4.1.4.3 新闻爬虫关键代码

```
$document = \phpQuery::newDocumentFile($url);//打开目标地址
            //站点结构   储存标题与文章内容 dom
            $structures = array(
                    'news.sina.com.cn' => '#artibodyTitle,#artibody',
                    'sports.sina.com.cn' => '#j_title,#artibody',
                    'finance.sina.com.cn' => '#artibodyTitle,#artibody',
                    'tech.sina.com.cn' => '#main_title,#artibody',
                    'ent.sina.com.cn' => '#main_title,#artibody',
                    'news.163.com' => '#epContentLeft>h1:first,#endText',
                    'money.163.com' => '#epContentLeft>h1:first,#endText',
                    'mil.news.sina.com.cn' => '#main_title,#artibody',
            );
            $remove = array(   //需要过滤的 dom 节点
                    '.caijing_bq',
                    'style',
                    '.special_tag_wrap',
                    'pre',
                    '.ep-source',
                    '.article-editor',
                    '.show_author',
```

```
                'script',
                '.ct_hqimg',
                '.fin_reference',
                '.finance_app_zqtg',
                'link',
                '#j_album_1',
                '#blk_weiboBox_01'
        );
        $structure = $urlInfo['host'];
        $structureArr = explode(',',$structures[$structure]);
        $document->find($structureArr[1])->find(join(',',$remove))->remove();  //过滤一些广告信息
        $title = $document->find($structureArr[0])->text();//标题
        $content = $document->find($structureArr[1])->html();//内容
        if ($structure == 'news.163.com' || $structure == 'money.163.com') { //解决网易抓取的中文乱码问题
            $title = mb_convert_encoding($title, 'ISO-8859-1', 'utf-8');
            $title = mb_convert_encoding($title, 'utf-8', 'GBK');
            $content = mb_convert_encoding($content, 'ISO-8859-1', 'utf-8');
            $content = mb_convert_encoding($content, 'utf-8', 'GBK');
        }
        if ($structure == 'tech.sina.com.cn' && !$title) {
            $title = $document->find('#artibodyTitle')->text();
        }
```

4.2 相似用户推荐机制详细设计

4.2.1 相似用户推荐机制理论基础

4.2.1.1 余弦相似性

判断两个内容的是否相似的时候，可以把这两个内容看做是空间中的两条线段，这两条线段之间存在着一个夹角，当夹角越小时，即接近 0 度，表示这两条线段的方向相同而且线段重合在一起，带夹角越大时，即接近 180 度，表示方向相反，两天线段完全不一样。因此，可以通过内容表示的线段的夹角来判断这两个内容是否相似，夹角越小时，内容越相似。

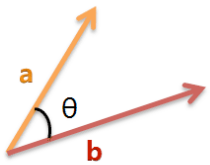

图 4-7 线段夹角图

以二维空间为例，图 4-7 的 a 和 b 两个向量，要计算 a 向量和 b 向量的夹角 θ。余弦定理可以用下面的公式求得：

假如 x[a1, b1]，y[a2, b2]，那么可以将余弦定理改写成下面的形式：

$$\cos\theta = \frac{x_1 x_2 + y_1 y_2}{\sqrt{x_1^2 + y_1^2} \times \sqrt{x_2^2 + y_2^2}}$$

图 4-8 余弦表达式

根据数学家们已经做好的证明，余弦定理在多维向量的情况下也成立。假设 X、Y 为两个 n 维向量，X [A1, A2, A3..., An] ，Y [B1, B2,B3..., Bn] ，那么 X 和 Y 的夹角 θ 的余弦等于：

$$\cos\theta = \frac{\sum_{i=1}^{n}(A_i \times B_i)}{\sqrt{\sum_{i=1}^{n}(A_i)^2} \times \sqrt{\sum_{i=1}^{n}(B_i)^2}} = \frac{A \cdot B}{|A| \times |B|}$$

图 4-9 n 维余弦定理表达式

最后计算得的结果，即余弦值越接近 1 时，就表示夹角越接近 0 度，也就表示两个向量越相似，这就叫"余弦相似性[2]"。

4.2.2 相似用户推荐机制设计思路

通过用户的浏览、评论、点赞操作，形成一个用户近期的操作历史，通过这些数据形成一个用户画像，通过这个用户画像建立用户模型，即向量模型，利用余弦定理计算这些向量模型的相似性。

4.2.3 相似用户推荐机制实现方法

数据库建立用户画像表，该表有着一个 similarity 字段，该字段保存一个拥有着三个键值的 json 数据，这三个键值分别为 browseInfo、commentInfo、zanInfo，分别表示近期的浏览信息、评论信息、点赞信息，利用这三个信息的保存内容，计算出一个与浏览分类、浏览关键字、评论分类、评论关键字、点赞分类、点赞关键字六个模型，利用余弦相似度分别计算这六个模型的相似性取平均值得到最后的用户相似度结果，最后通过相似度排序给用户推荐相似度最高的用户。

用户画像计算流程图如下：

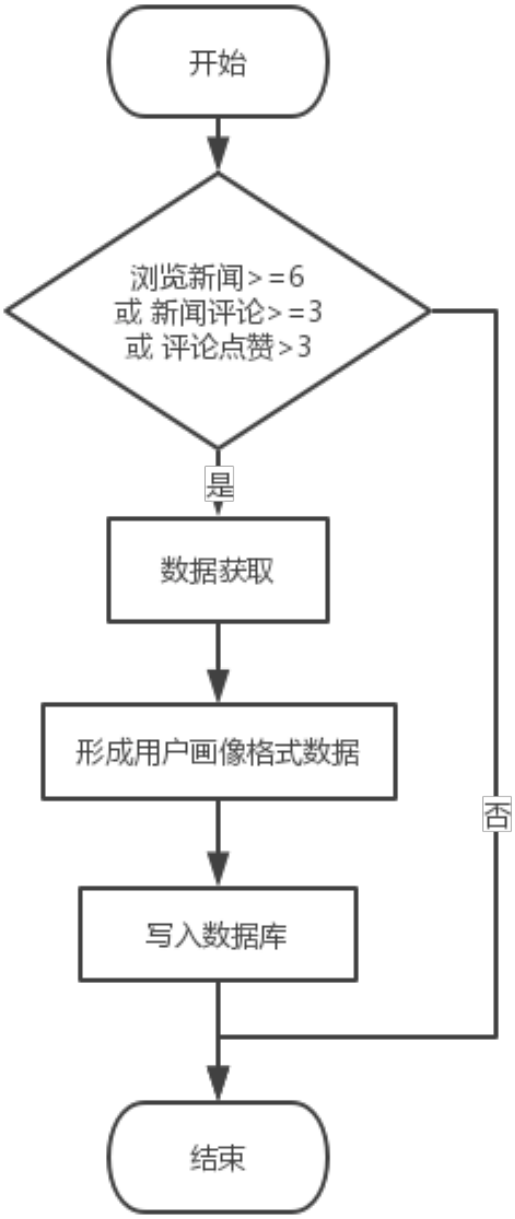

图 4-10 用户画像计算流程图

用户相似度结果表示如下图，相似度为 66%的两个用户，蓝色表色分类，黄色表示关键字。

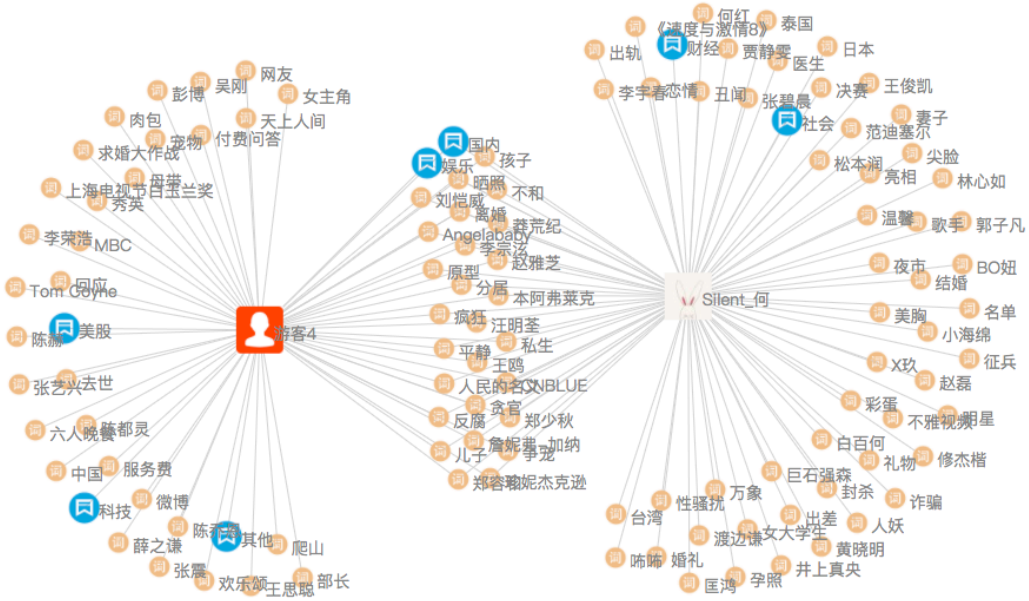

图 4-11 用户相似度力导向图

4.2.4 相似用户推荐机制核心代码

4.2.4.1 相似度计算算法

```
//计算两个向量的相似度
    public function similarity($vector1,$vector2){
        //初始化单词个数
        foreach ($vector1 as $key => $num) {
            if( !$vector2[$key] ){
                $vector2[$key] = 0;
            }
        }
        foreach ($vector2 as $key => $num) {
            if( !$vector1[$key] ){
```

```
                $vector1[$key] = 0;
            }
        }
        $fz = 0;
        foreach ($vector1 as $key => $num) {
            $fz += $vector1[$key] * $vector2[$key];
        }
        $fm1 = 0.0 ;
        foreach ($vector1 as $key => $num) {
            $fm1 += $num * $num;
        }
        $fm2 = 0.0;
        foreach ($vector2 as $key => $num) {
            $fm2 += $num*$num;
        }
        $fm = sqrt($fm1) * sqrt($fm2) ;
        return $fz/$fm;
    }
```

4.3 协同过滤推荐新闻机制详细设计

4.3.1 协同过滤推荐新闻机制理论基础

4.3.1.1 协同过滤算法

协同过滤算法(Collaborative Filtering)作为最经典的推荐算法之一，主要包括在线协同、离线过滤两大部分。在线协同，通过在线的数据找出用户可能感兴趣

的内容，而所谓离线过滤，则是过滤掉用户不喜欢的、不感兴趣的数据。

协同过滤分为三种类型。

第一种，基于用户的协同过滤（user-based）。该协同过滤考虑的以用户之间的相似度为主体，找到用户感兴趣的内容的相似内容，分析用户对感兴趣的内容的评分，找到相似的并且评分最高的内容推荐给用户。

第二种，基于项目的协同过滤（item-based）。与第一种协同过滤相比，有点类似，该类协同过滤主要以内容的相似度为主体，先分析用户对部分内容进行评分，然后对这部分内容的相似度的最高的内容进行预测，把预测得到的最高评分的内容推荐给用户。比如，我在网上买了一本程序学习相关的书，系统会推荐一堆程序学习，编程相关的书给我，这种推荐方式就用到了基于项目的协同过滤。

第三种，基于模型的协同过滤（model-based）。该类协同过滤是现在最流行的协同过滤，简单描述就是根据用户的一些操作历史，生成一个可以对用户的兴趣爱好进行预测的模型，根据该模型进行新闻推荐。

4.3.2 协同过滤推荐新闻机制设计思路

该系统使用基于模型的协同过滤进行新闻推荐，主要流程为先计算新闻的相似度，接下来根据对训练好的可以预测用户兴趣爱好的模型的分析，得到分析的结果集之后为用户推荐可能喜欢的内容。

要进行协同过滤推荐，首先要进行推荐内容相似度的计算与保存，而能表示新闻是否相似的表现有两种，一为计算新闻文本的相似矩阵，二为新闻的具有相同的关键字。而当文本内容比较大的时候，文本的相似度相对于关键字来说，不太能表示新闻相似，即核心内容相似，故我们采用关键字推荐的方式。

其次，根据基于模型的协同过滤算法的原理，我们需要训练一个用户模型，用该模型来对用户的兴趣进行预测。正好我们在进行相似用户推荐的时候已经记录了一个用户画像，该画像正好反应了用户的兴趣偏好。

最后，根据这个用户画像和推荐配置，分析计算出一个以新闻分类、新闻关

键字为指标的比例，根据该比例降序排序查找其未推荐过的相同的分类新闻、类似新闻（即关键字相同）。

4.3.3 协同过滤推荐新闻机制的实现方法

首先，新闻添加的时候，自动计算获取新闻的关键字，该计算方法在前面的章节有介绍过，方式为对新闻文本进行中文分词后，计算每个分词的 TF-IDF 值，根据其 TF-IDF 值决定其关键字，公式如下：

$$\mathrm{TF-IDF} = \text{词频}(\mathrm{TF}) \times \text{逆文档频率}(\mathrm{IDF})$$

图 4-12 TF-IDF 计算公式

其次，根据之前将的用户画像，即我们之前保存的拥有三个分别表示，用户的浏览信息 browseInfo、评论信息 commentInfo、点赞信息 zanInfo 的 json 字符串，通过分析计算出一个与浏览分类 browse_type、浏览关键字 browse_keyword、评论分类 comment_type、评论关键字 comment_keyword、点赞分类 zan_type、点赞关键字 zan_keyword 六个模型，在根据这六个模型，集合分析得到一个关键字的相关总和和一个新闻分类相关的总和，最后根据这两个总和数据库中启用的推荐配置，计算出一个按权重所占比降序的关键字或者新闻分类的数据集，该数据集的每个个体信息包括是否为关键字推荐、关键字或者分类编号、权重所占百分比、需要推荐的个数。

最后，根据这个数据集到逐一到数据库中查找相同关键字或者分类的新闻形成推荐结果列表。

其推荐流程如下图：

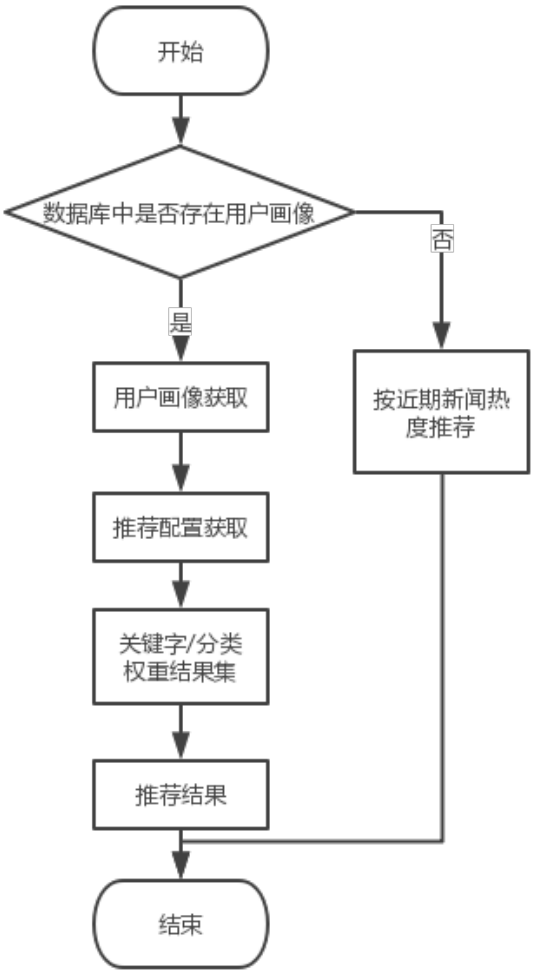

图 4-13 新闻推荐流程图

4.3.4 协同过滤推荐新闻机制核心代码

4.3.4.1 协同过滤新闻推荐核心代码

```
$recommendConfigModel = M('RecommendConfig');
//获取推荐配置信息
$recommendConfig = $recommendConfigModel->where(array('state' =>
```

```
array('neq', '0')))->find();
        $recommendModel = M('Recommend');
        $timestamp = time();
        $portrayalData = $this->getPortrayal($user_id);
        $portrayalInfo = $portrayalData['data'];    //用户画像数据
        $dataSaveResult = $portrayalData['dataSaveResult']; //用户画像信息是否保存成
功
        $browseList = $portrayalInfo['browseInfo'];
        $recommendNum = count($browseList) > 10 ? 10 : count($browseList);
        $portrayal = $this->portrayal($portrayalInfo);
        $similarityUserKeyword=$this->getSimilarityUserKeyword($user_id);// 相 似 用 户
的关联度
        $data=$this->getRecommendWeight($portrayal,$recommendConfig,$similarityUs
erKeyword);
        $data = $this->multi_array_sort($data,'weightScore');
        $read=$recommendModel->where(array('user_id'=>
$user_id))->field('news_id')->select(false);
        $recommendArray=$this->getRecommendNum($data, $recommendNum);
        $recommendList=$this->getRecommendData($read,          $recommendArray,
$recommendConfig, $recommendNum);
        $recommendLength = count($recommendList);
        if ( $recommendLength < 10 ) {
        //推荐个数不够,补充推荐个数(以热度为准  或者兴趣爱好为准)
            $notInArray = array();
            foreach ($recommendList as $item) {
                array_push($notInArray,$item['id']);
```

```
                }
                $count = 10 - $recommendLength;
                $newsModel = D('News');
                $allow_recommend_time = $timestamp - 60 * 60 * 24 *
(int)$recommendConfig['allow_recommend_time'];
                $supplement = $newsModel->getByBeginTimeAndNum(date('Y-m-d
H:i:s',$allow_recommend_time),$count,array($read,join(',',$notInArray)));
                $recommendList = array_merge($recommendList,$supplement);
            }
```

5 软件测试

5.1 测试方法及工具

测试方法：黑盒测试的边界值测试、等价类划分等。

工具：手工测试。

5.2 测试类型

5.2.1 功能性测试

5.2.1.1 测试范围

前台：新闻浏览、新闻评论、新闻发布、新闻评论点赞、新闻评论回复、评论删除。

后台：推荐配置管理、新闻管理、用户管理、发布者审核、评论管理。

5.2.1.2 测试目的

确保所有功能已正常实现并正常运行 。

业务流程检验：用户需求各个业务流程得到满足，用户使用的时候不会有疑问。

5.2.2 易用性测试

5.2.2.1 测试范围

界面结构包括菜单栏、侧边栏、网站的背景、主体颜色和站点字体、站点按钮、弹框提示信息的一致性。

5.2.2.2 测试目的

检查网站的操作设计是否符合当前人们的操作习惯，网站的页面风格是否在可接受的范围内。

5.3 测试用例

测试用例总共 92 个，由于测试用例过多，下面列举部分测试用例，详情请见”基于协同过滤算法的个性化新闻推荐系统测试用例.xlsx”。

表 5-1 新闻发布测试用例

测试类型	功能性测试	模块名称	后台
功能	发布新闻		
用例描述	测试后台新闻发布功能是否正常		
用例编号	操作步骤	输入	期望
admin_001	1. 管理员登录后台系统。 2. 进入新闻管理。 3. 点击发布新闻。 4. 编辑新闻内容。 5. 点击发布新闻。	标题：测试标题 关键字选择：测试 分类选择：创业 输入内容： 封面选择：无	1. 提示请输入内容。
admin_003	1. 管理员登录后台系统。 2. 进入新闻管理。 3. 点击发布新闻。 4. 编辑新闻内容。 5. 点击发布新闻。	标题：测试标题 关键字选择：测试、白百何 分类选择：娱乐 输入内容：合法化的飞洒好的撒 封面选择：无	1. 显示发布成功 2. 列表中可看到刚刚发布的内容。
admin_005	1. 管理员登录后台系统。 2. 进入新闻管理。 3. 点击发布新闻。 4. 编辑新闻内容。 5. 点击发布新闻。	标题：测试标题 关键字选择： 分类选择：娱乐 输入内容：合法化的飞洒好的撒 封面选择：无	1. 提示关键字至少选择一个

表 5-2 新闻推荐测试用例

测试类型	功能性测试	模块名称	前台
功能	新闻推荐		
用例描述	测试新闻推荐，推荐内容进行个性化推荐		
用例编号	操作步骤	输入	期望
home_002	1. 新账号，进入首页。 2. 一直浏览娱乐分类新闻。 3. 在推荐栏点击查看更多内容。	N/A	1. 新闻列表显示大部分为娱乐的 10 条新闻。
home_003	1. 用经常浏览娱乐新闻的账户进入系统。 2. 一直浏览体育类新闻。 3. 在推荐栏点击查看更多内容。	N/A	1. 新闻列表显示大部分为娱乐或者体育的十条新闻。

5.4 测试执行

本次执行的测试用例总共 92 个，分别为前台模块 35 个，后台模块 57 个，测试用例的执行详情请参考”基于协同过滤算法的个性化新闻推荐系统测试用例执行.xlsx”。

5.4.1 前台模块

执行用例数量：35 个。

bug 数量：4 个。

bug 描述：新闻浏览加载更多按钮没有防止重复点击、未登陆状态下可点赞、未登陆状态下可查看并点击回复按钮、新闻评论回复没有放置重复提交。

表 5-3 新闻推荐测试用例执行

模块名称	后台				
功能	新闻推荐				
用例描述	测试新闻推荐，推荐内容进行个性化推荐				
用例编号	操作步骤	输入	期望	实际结果	是否通过
home_002	1. 新账号，进入首页。 2. 一直浏览娱乐分类新闻。 3. 在推荐栏点击查看更多内容。	N/A	1. 新闻列表显示大部分为娱乐的 10 条新闻。	1. 新闻列表显示大部分为娱乐的 10 条新闻。	通过
home_003	1. 用经常浏览娱乐新闻的账户进入系统。 2. 一直浏览体育类新闻。 3. 在推荐栏点击查看更多内容。	N/A	1. 新闻列表显示大部分为娱乐或者体育的十条新闻。	1. 新闻列表显示大部分为娱乐或者体育的十条新闻。	通过

5.4.2 后台模块

执行用例数量：57 个。

bug 数量：2 个。

bug 描述：新闻发布修改未输入内容也可发布、新建推荐配置未输入名称也可添加。

表 5-4 新闻发布测试用例执行

模块名称	前台				
功能	新闻推荐				
用例描述	测试后台新闻发布功能是否正常				
用例编号	操作步骤	输入	期望	实际结果	是否通过
admin_001	1.管理员登录后台系统。 2.进入新闻管理。 3.点击发布新闻。 4.编辑新闻内容。 5.点击发布新闻。	标题：测试标题 关键字选择：测试 分类选择：创业 输入内容： 封面选择：无	1.提示请输入内容。	1.显示发布成功 2.列表中可看到刚刚发布的内容。	不通过
admin_003	1.管理员登录后台系统。 2.进入新闻管理。 3.点击发布新闻。 4.编辑新闻内容。 5.点击发布新闻。	标题：测试标题 关键字选择：测试、白百何 分类选择：娱乐 输入内容：合法化的飞洒好的撒 封面选择：无	1.显示发布成功。 2.列表中可看到刚刚发布的内容。	1.显示发布成功 2.列表中可看到刚刚发布的内容。	通过
admin_005	1.管理员登录后台系统。 2.进入新闻管理。 3.点击发布新闻。 4.编辑新闻内容。 5.点击发布新闻。	标题：测试标题 关键字选择： 分类选择：娱乐 输入内容：合法化的飞洒好的撒 封面选择：无	1.提示关键字至少选择一个。	1.提示关键字至少选择一个。	通过

5.5 测试结果统计

5.5.1 BUG 类型统计

表 5-5 BUG 类型统计

BUG 类型	数量
功能性问题	6
界面问题	0

5.5.2 BUG 严重程度统计

表 5-6 BUG 严重程度统计

严重程度	数量
严重	0
较严重	0
一般	0
微小	6

5.5.3 缺陷倾向及主要原因

缺陷主要倾向于功能性问题上，全部属于微小问题，对系统并没有其大的影响，主要集中在新闻发布这一块，其中主要形成原因是开发者对需求的业务规则了解不透，业务规则规定新闻发布内容不能为空，开发者并没有对其进行限定。残留对系统的功能性并没有影响。

5.6 测试结论

5.6.1 功能性

系统基本实现了按个性化推荐新闻的功能，其中的发布者中心模块、后台管理模块、前台模块的所有所有功能性正常，暂无验证缺陷，系统功能可以正常运行，包括新闻管理、评论管理、点赞、新闻推荐等所有功能。

5.6.2 易用性

界面内容直观、简洁、不唐突、不拥挤，布局合理。按钮内容提示易理解。页面风格可以在可接受的范围，用户界面的具有易操作性、友好型。页面操作习惯也符合用户习惯。

6 开发环境和软件运行结果

6.1 软件环境

操作系统：win7/win8/win10/macOS/Linux。

编程语言：php+javascript。

开发工具：phpStrom+atom。

6.2 运行环境

软件要求：支持 IE11 以上特性的浏览器。

硬件要求：内存：1GB（或更高）。

6.3 软件部分运行结果

用户推荐如下图 6-1。

图 6-1 用户推荐

新闻列表如图 6-2，新闻显示方式有三种，分别为无图、一张图、三张图，显示方式取决于是否有封面和文章内的本地图片个数，为解决图片拉伸问题，图片的地址为用 php 程序处理后输出固定长宽比的 url 地址。

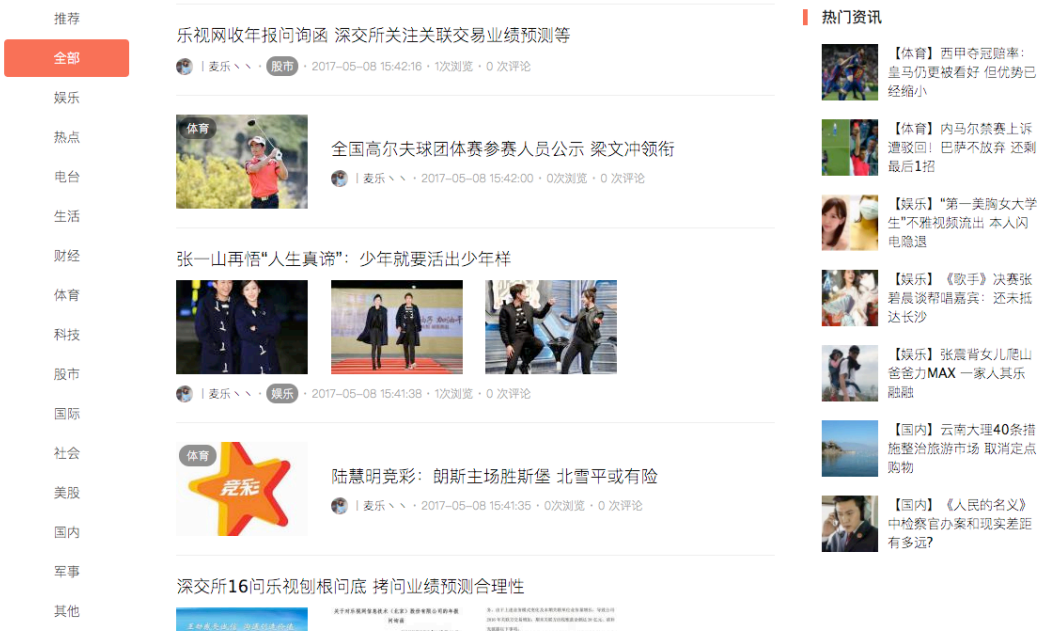

图 6-2 新闻列表

个人中心动态如图 6-3，该页面显示用户的动态信息。

图 6-3 用户动态

粉丝指数曲线图如图 6-4，为发布者提供粉丝的变化指数。

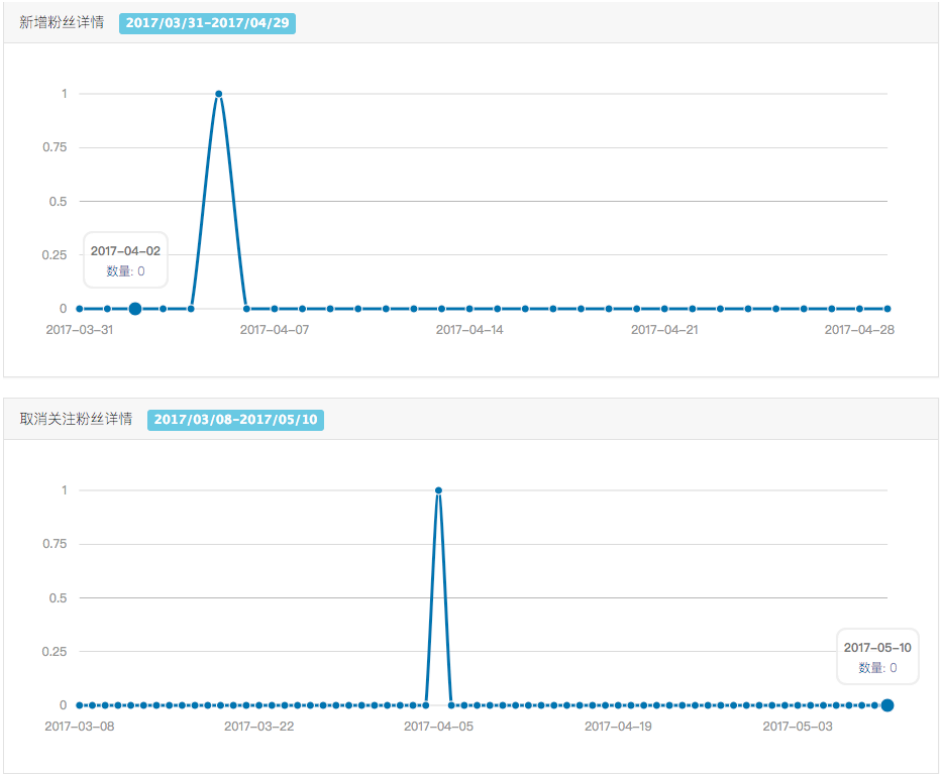

图 6-4 粉丝指数

文章浏览 PV、UV 数据图如图 6-5，整个站点的文章被浏览的数据的可视化。

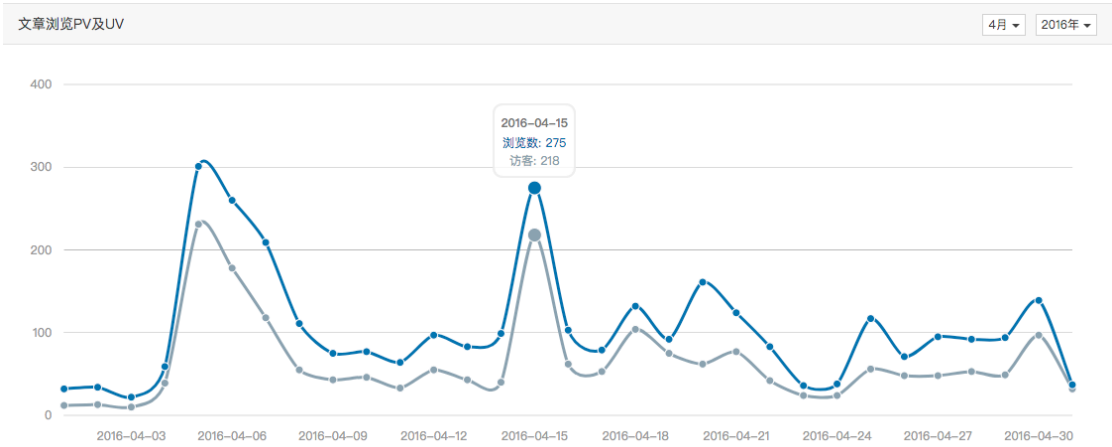

图 6-5 文章浏览 pv/uv

站点活跃用户柱状图如图 6-6，活跃用户表示每天在站点浏览文章超过三过三篇的用户。

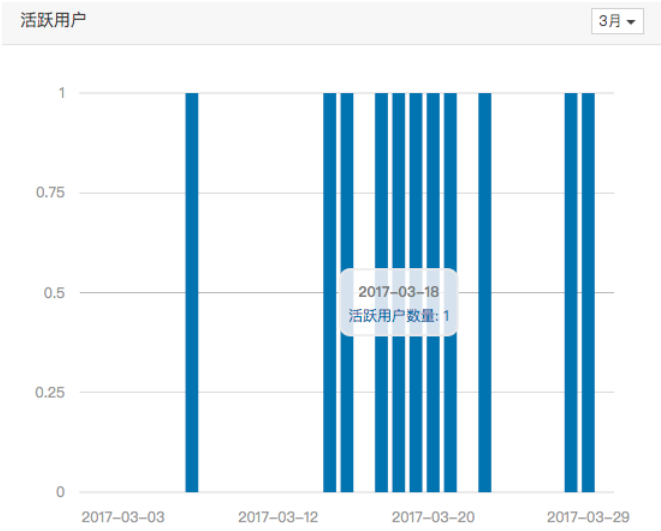

图 6-6 活跃用户柱状图

地区 uv/pv 饼状图如图 6-7，表示整个站点被 pv/uv 的分布地区数据的可视化。

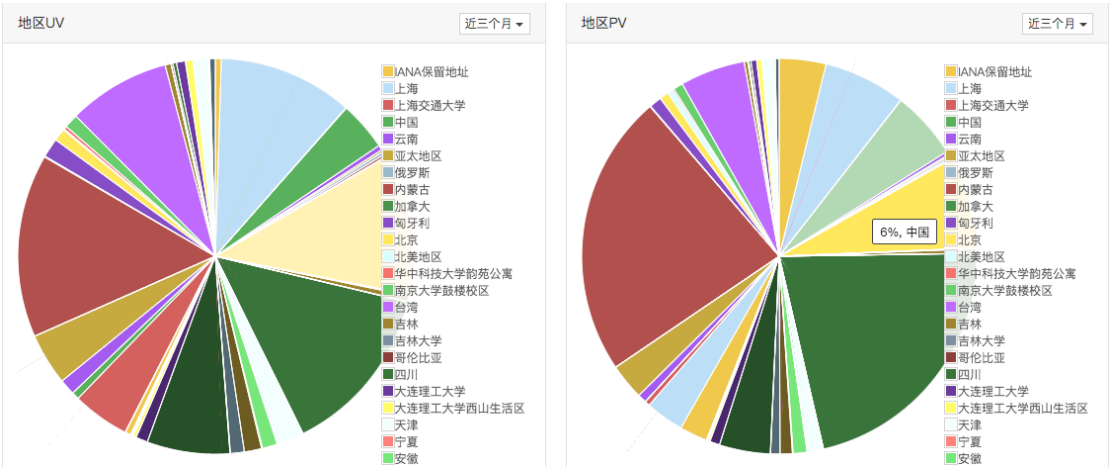

图 6-7 地区 uv/pv 饼状图

站点浏览的分类统计饼状图如 6-8，表示站点新闻分类被浏览的数据的统计的可视化。

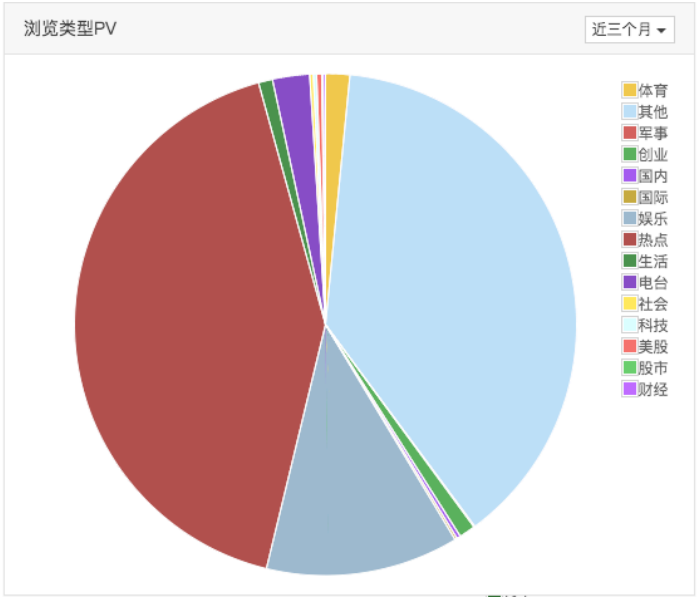

图 6-8 新闻分类饼状图

新闻抓取如图 6-9，抓取新浪和网易的新闻。

图 6-9 新闻抓取

新闻文本相似度计算结果如图 6-10，后台在线计算新闻文本的相似度，只计算计算近期关键字相同的新闻的文本的相似度。

图 6-10 新闻文本相似度计算结果

用户相似度力导向图如图 6-11，用来查看用户之间的相似的　　关联。

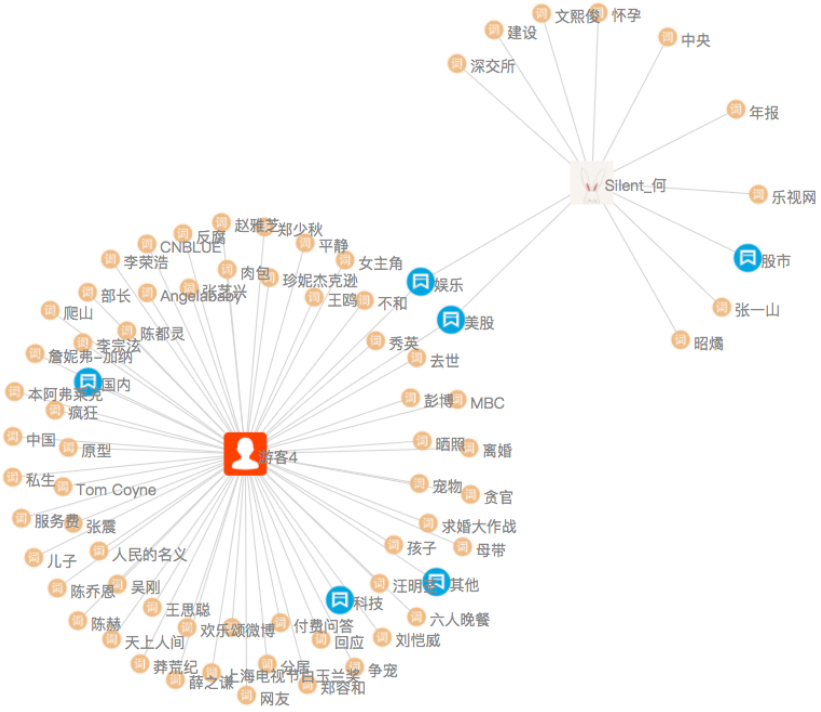

图 6-11 用户相似度力导向图

6.4 存在的问题和不足

由于编程时间紧促，在新闻推荐业务逻辑中没有将标记的用户兴趣爱好和用户的地区个人信息加入到推荐的权重中，所以项目中存在一些冗余的内容。另外，推荐结果也很难严重其正确性，故还需一个长期的验证过程。

总结

在此次设计选题之前从来没有对新闻推荐有过了解，只是对推荐比较感兴趣，不断的查阅相关的资料以及以周老师交流，了解到了协同过滤算法的一些基本理论知识，判定了其技术上的可行性，所以选择了该题目。

选题之后，将这个新闻推荐系统的一些重要问题列了出来，逐一去解决，比如推荐的数据从哪里来、推荐是以什么作为标准、数据的来源的数据结构怎么统一化等一系列问题，也正因为这一系列问题，在查资料和工具的工程中接触到了网页爬虫、中文分词、关键字自动提取这一些技术，并通过写 demo 来学习与实战它们，其中包括了 phpQuery、phpanalysis 这两个插件的使用，TF-IDF 算法的实现以及余弦相似性算法的的实现，在这个不断的实战的过程中，不断的得到了成长，慢慢的我对我毕业设计的完成更加有信心了。

总的来说，这次毕设完成之后，自己的前端编程、后端编程、数据库设计等各方面的能力都得到了前所未有的进步，这些能力会在我日后的生活和工作中起到很大的作用。

致谢

在周立章老师的细心指导下顺利了完成了本次毕业设计的任务。周老师教导我们，平时要多进行独立思考，有什么不懂的地方亲自动手查资料，真的解决不了的问题，再问周老师，周老师会耐心为我们解答。周老师在这段时间给了我非常大的帮助，我自己通过这次毕业设计也得到了很大的提升，这些将对我以后的工作和学习有着非常大的帮助，非常感谢周立章老师。

也感谢教过我数据结构、php、网页设计这些课程的老师，让我在毕业设计之前有相关的知识基础，以至于在开始的时候没有一头雾水。

参考文献

[1] TF-IDF 与余弦相似性的应用（一）：自动提取关键词. http://www.ruanyifeng.com/blog/2013/03/tf-idf.html

[2] TF-IDF 与余弦相似性的应用（二）：找出相似文章. http://www.ruanyifeng.com/blog/2013/03/cosine_similarity.html

[3]冷亚军,陆青,梁昌勇. 协同过滤推荐技术综述[J]. 模式识别与人工智能,2014,(08):720-734.

[4]刘青文. 基于协同过滤的推荐算法研究[D].中国科学技术大学,2013.

[5] 王鹏,王晶晶,俞能海. 基于核方法的 User-Based 协同过滤推荐算法[J]. 计算机研究与发展,2013,07:1444-1451.

[6]陈浩. 基于 Javascript 的异步编程分析[J]. 电脑知识与技术,2015,(13):80-81.

[7] 孙辉,马跃,杨海波,张红松. 一种相似度改进的用户聚类协同过滤推荐算法[J]. 小型微型计算机系统,2014,09:1967-1970.

[8]王芳. 当前流行 Web 开发语言——PHP[J]. 信息系统工程,2014,(05):30.

[9]徐立艳. 浅议 PHP 与 MySQL 之间的操作[J]. 电脑知识与技术,2014,15:3478-3480.

[10] 吕智强. 基于 MVC 模式的 PHP 框架设计[J]. 科技视界,2013,24:65-66.