# 详细设计说明书

给用户推荐用户感兴趣的国外新闻，科研成果。

为了实现这个功能，我们需要用到用户推荐系统。我们先设计一个基于标签的推荐系统。

各种推荐系统系统之间都有一定的联系，有的是以某一个中间变量为媒介，计算用户和商品之间的关系

基于标签的推荐系统 和 之前所学的 隐语义模型我觉得有一定的相似性。

在隐语义模型中，我们用隐语义向量表示，用户的兴趣。我们把用户对每个隐语义的兴趣向量表示某个特定的用户

用一个特定的隐语义向量，表示某一个确定的物品。

通过对这两向量之间的距离我们计算得到用户和商品之间的关系

在标签系统中基本的思想和这个相似，我们只是把晦涩的隐语义，用一个明确的标签表示。

两者的基本思想便在这和方向上具有相似性

1.我们先得到一个矩阵 ui[用户][标签] 表示 用户在使用过程中 对每个标签使用的次数 。

2.我们继续得到一个矩阵 si[标签][物品] 表示 每个标签在每个物品上的使用次数

3.我们假设 物品和用户之间的关系 可以 用 两个矩阵的乘法得到 us[用户][物品] = ui[用户][标签] \* si[标签][物品]

4.我们考虑到 ， 热门标签被用户使用频率高 和 热门商品被热门标签标记的频率也比较高， 这会影响我们的推荐，这会使冷门的商品被推荐的概率减少。所以我们要去热门话

5.为了去热门话 ，我们对us矩阵进行处理。我们让us矩阵中的每个数据 除以 各自标签的被不同人，被不同商品使用次数

6.新的用户商品矩阵我们可以这样表示 new\_ui[用户][标签] = ui[用户][标签]/log\_(1+len(ui[标签])) ， new\_si[标签][物品] = si[标签][商品]/log\_(1+len(si[标签]))

7.最后我们得到去热门话的用户和商品之间的关系矩阵 new\_us[用户][物品] = new\_ui[用户][标签] \* new\_si[标签][物品]

提供专门的搜索引擎，定向搜索用户感兴趣的数据

我们需要设计一款搜索引擎，或者直接使用已有的。

搜索引擎工作流程主要有[数据采集](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E9%87%87%E9%9B%86/219239" \t "_blank)、[数据预处理](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E9%A2%84%E5%A4%84%E7%90%86/2711288)、数据处理、结果展示等阶段。在各工作阶段分别使用了网络爬虫、中文分词、大数据处理、数据挖掘等技术。 [2]

网络爬虫也被称为蜘蛛或者网络机器人，它是搜索引擎抓取系统的重要组成部分。网络爬虫根据相应的规则，以某些站点作为起始站点通过各页面上的超链接遍历整个互联网，利用URL弓I用根据广度优先遍历策略从一个html文档爬行到另一个html文档来抓取信息。 [2]

中文分词是中文搜索引擎中一个相当关键的技术，在创建索引之前需要将中文内容合理的进行分词。中文分词是文本挖掘的基础，对于输入的一段中文，成功的进行中文分词，可以达到电脑自动识别语句含义的效果。 [2]

大数据处理技术是通过运用大数据处理计算框架，对数据进行分布式计算。由于互联网数据量相当庞大，需要利用大数据处理技术来提高数据处理的效率。在搜索引擎中，大数据处理技术主要用来执行对网页重要度进行打分等数据计算。 [2]

数据挖掘就是从海量的数据中采用自动或半自动的建模算法，寻找隐藏在数据中的信息，是从数据库中发现知识的过程。数据挖掘一般和计算机科学相关，并通过机器学习、模式识别、统计学等方法来实现知识挖掘。在搜索引擎中主要是进行文本挖掘，搜索文本信息需要理解人类的自然语言，文本挖掘指从大量文本数据中抽取隐含的、未知的、可能有用的信息。

提供用户给收藏功能

在用户浏览的过程中，用户会发现一些令他感兴趣的，用户可以把这篇文章，这个数据集收藏下来。

一、前台页面

1.收藏按钮，默认显示“收藏”

2.页面加载，调用pdUserCollection()方法进行判断当前信息是否已收藏，判读的依据是数据库中收藏表中是否存在该信息（也可以判断数据表中该信息的状态）

1>.如果已收藏，更新sctxt的显示文字为：取消收藏

2>.如果未收藏，更新sctxt的显示文字为：收藏

3.点击按钮，触发点击事件，调用addUserCollection()方法，后台方法进行判断数据表中是否收藏，

1>.如果已收藏，执行删除操作（或者改变信息状态）

2>.如果未收藏，执行添加操作（或者改变信息状态）

二、后台

后台进行逻辑判断执行相近的数据库操作

# 注册功能

1. 用户注册
   1. 用户注册页面
   2. 注册成功，可选择进入信息编辑界面
2. 用户编写自己的基本信息
   1. 登记自己的基本信息
   2. 登记成功，可选择进入添加推荐好友界面

好友功能

在这个软件中，用户可以添加现实中相识的，网络中兴趣相似的人为好友

1. **添加关注**

添加关注分为两步：

1、将对方id添加到自己的关注列表中；

Redis::ZADD("me:follow", time(), other)

2、将自己的id添加到对方的粉丝列表中：

Redis::ZADD("other:fans", time(), me)

2. **取消关注**

取消关注同样分为两步：

1、将对方id从自己的关注列表中移除；

Redis::ZREM("me:follow", other)

2、将自己的id从对方的粉丝列表中移除：

Redis::ZREM("other:fans", me)

3. **关注列表**

1、查看我的关注列表：

Redis::ZRANGE("me:follow", 0 , -1)

2、查看别人的把id换掉就可以

Redis::ZRANGE("other:follow", 0 , -1)

4. **粉丝列表**

1、查看我的粉丝列表：

Redis::ZRANGE("me:fans", 0 , -1)

2、查看别人的把id换掉就可以

Redis::ZRANGE("other:fans", 0 , -1)

5. **人物关系**

**1、 我单向关注他**

我单向关注他，要同时满足两个条件：1、我的关注列表中有他(或他的粉丝列表中有我)；2、我的粉丝列表中没有他(或他的关注列表中没有我)。

Redis::ZSCORE("me:fans", other)   #未返回分数

Redis::ZSCORE("me:follow", other) #返回分数

### ****2、 他单向关注我****

他单向关注我，要同时满足两个条件：1、我的关注列表中没有他(或他的粉丝列表中没有我)；2、我的粉丝列表中有他(或他的关注列表中有我)。

Redis::ZSCORE("other:fans", me)   #未返回分数

Redis::ZSCORE("other:follow", me) #返回分数

### ****3、 是否互粉****

我和某人是否互粉，要同时满足两个条件：1、我的关注列表中有他(或他的粉丝列表中有我)；2、我的粉丝列表中有他(或他的关注列表中有我)。同时成立才为互粉。

Redis::ZSCORE("me:fans", other)   #返回分数

Redis::ZSCORE("me:follow", other) #返回分数

6. **数量相关**

### ****1、 我的关注数****

Redis::ZCARD("me:follow");    #返回数量

### ****2、 我的粉丝数****

Redis::ZCARD("me:fans");     #返回数量

## 7. ****排序取出所有的人****

**1、根据关注的时间倒叙取出用户的id**

Redis::ZREVRANGE("me:fans", 0, -1, TRUE); #倒序取值

**2、根据关注时间顺序取出用户的id**

Redis::ZRANGE("me:fans", 0, -1, TRUE); #顺序取值

# 数据库设计

## 逻辑模型



## 物理模型

1. 为每个表初始设置300M的存储空间，以10%的大小扩展。
2. 为上传的图片预留空间。