

分类号_____

学校代码 1 0 4 8 7

学号 M201272614

密级_____

华中科技大学

硕士学位论文

海量医疗信息系统推荐技术研究

学位申请人：刘 辉

学 科 专 业：计算机系统结构

指 导 教 师：李春花 副教授

答 辩 日 期：2015.5.20

**A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Engineering**

Research of Recommendation Technology for Massive Healthcare Data

Candidate : Liu Hui

Major : Computer Architecture

Supervisor : Assoc. Prof. Li Chunhua

Huazhong University of Science & Technology

Wuhan, Hubei 430074, P. R. China

April, 2015

独创性声明

本人声明所呈交的学位论文是我个人在导师指导下进行的研究工作及取得的成果。尽我所知，除文中已经标明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的研究成果。对本文的研究做出贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

学位论文作者签名：

日期： 年 月 日

学位论文版权使用授权书

本学位论文作者完全了解学校有关保留、使用学位论文的规定，即：学校有权保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权华中科技大学可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。

本论文属于 ☐ 保密， 在 _____ 年解密后适用本授权书。

☐ 不保密。

（请在以上方框内打“√”）

学位论文作者签名：

日期： 年 月 日

指导教师签名：

日期： 年 月 日

摘要

在信息爆炸式增长的背景下,医疗信息同样处于快速增长的阶段,为了获取信息,人们可以通过搜索引擎来满足自身的要求。但是,搜索引擎会返回符合要求的所有信息,通常来说信息量也很大,用户往往要翻阅大量的网页之后才能找到自己真正想要的信息,显然,这与用户快速获取信息的要求相悖。

设计并实现一个集存储、搜索和推荐于一体的海量医疗信息系统,本系统的医疗信息包括药品、医生、常识和处方信息。其中,存储和搜索部分利用开源搜索引擎 Elasticsearch 机制定义医疗信息的数据结构,以实现高性能和高可扩展性的要求;对于推荐部分,药品信息的推荐从其“适应症”文本的相似性、药品总使用量以及药品使用热度三方面综合考虑;医生信息的推荐从医生自身的流行程度、患者对医生评价相似性这两个策略来考虑;医疗常识信息则主要考虑常识的内容相似性和用户的历史兴趣点,为减少计算的时间开销,引入信息聚类技术;处方信息的推荐需要分析用户输入的诊断症状描述、检查结果和检验结果,并从处方数据库中找到跟用户输入最匹配的处方,匹配的原则首先是症状匹配,找到相同症状的案例,然后根据检查结果和检验结果匹配,检查结果是文字描述,可根据文本相似度来衡量匹配程度,检验结果是各项指标的定性定量表示,根据匹配项的数目来衡量匹配程度,然后根据用户的对药物的过敏情况进行过滤。

测试结果表明,本系统在存储、索引和搜索方面能够满足高性能、高可扩展性的要求,引入推荐功能和不引入推荐功能系统响应时间在同一个数量级上,不会显著增加响应时间。

关键词: 医疗常识, 处方信息, 数据聚类, Elasticsearch, 推荐

Abstract

Nowadays, with the explosion of information, Healthcare data is increasing rapidly. In order to get information, one can use search engine and do some searches, but search engine will return much data, which meet the requirement. He/she must scan a lot of documents until he/she find the right one, this contradicts with the demand of quickly find information.

In this paper, we propose a healthcare information platform which is used to store, index, search and recommend healthcare information. Healthcare information includes drug information, doctor information, common sense information and prescription. In order to store, index and search information and meet the requirement of high performance & high scalable, we use Elasticsearch, a powerful open source search and analytics engine, and design suitable data structure. As for recommendation, we propose special program for different kind of information. For drug information, we consider three aspects, indication text similarity, usage amount and the popularity of drug usage. For doctor information, we have two strategies, one considers the popularity of doctor, the other considers the similarity of patient praise. For common sense information, we recommend information with the standard of content similarity and user's interest, in order to reduce computation overhead, we should cluster the information. At last, for prescription, user should input symptom descriptions and inspection result, the system will find prescription most approach the inputs, then filter according to user's allergic drug.

The experimental results show that the system satisfies the demand of high performance & high scalable. With the introduction of recommendation, response delay is at the same order of magnitude, so real-time performance is perfect.

Keywords: common sense, prescription, data clustering, Elasticsearch, recommendation

目 录

摘 要	I
Abstract	II
1 绪论	
1.1 课题研究的背景	(1)
1.2 国内外研究现状的分析	(1)
1.3 本文研究的主要内容	(5)
2 医疗信息推荐方案设计	
2.1 系统需求分析	(6)
2.2 系统总体框架设计	(7)
2.3 医疗数据获取和解析模块设计	(9)
2.4 医疗信息存储和索引模块设计	(12)
2.5 医疗数据推荐模块设计	(14)
2.6 本章小结	(20)
3 医疗信息推荐实现	
3.1 信息存储和索引模块	(21)
3.2 药品信息推荐模块实现	(23)
3.3 医生信息推荐模块实现	(27)
3.4 常识信息推荐模块实现	(29)
3.5 处方推荐模块实现	(35)

3.6 本章小结	(37)
4 评估与测试	
4.1 测试环境	(38)
4.2 功能测试	(39)
4.3 性能测试	(40)
4.4 本章小结	(45)
5 总结与展望	
5.1 全文总结	(46)
5.2 展望	(47)
致 谢	(48)
参考文献	(49)

1 绪论

1.1 课题研究的背景

网络技术的发展引发了数据信息量的迅猛增长。统计结果表明,2009 年全球数据信息量累积达到了 80 万 PB,2010 年则增长至 1.2ZB。到 2020 年全世界数据信息总量可能达到 35ZB,这么庞大的数据规模对数据的获取和数据质量的甄别提出了巨大的挑战。医疗信息的情况也是如此。医疗信息,主要包括药品、处方、常识、医院、医生信息等。Web 信息过载问题出现的比较早,现在已经有一些解决方案,比如用于信息检索的搜索引擎及之后发展的个性化搜索技术、个性化推荐^[1,2]技术等。同时也出现了运用这些技术的实际应用,比如,Google 的个性化搜索 Google+和知名门户网站的新闻推荐。

我们设想一些情景,当患者购买药品时,他/她先是去药店,跟药店职员(具备一定的医学专业能力)说明自己的病情,然后药店职员就会根据用户的病情推荐相应的药品给他/她;当患者去医院看病时,一般情况下某个领域(比如妇科)的医生比较多,因为患者不清楚医生的各种信息,所以会感到迷茫,不知道应该选择哪个医生。他/她会根据医生的简介进行甄别。但是,患者最关心的信息,如其他病人对医生的评价等信息无从获取。总的来说,这些情况的解决方案都是比较片面的,主观因素影响患者的决定,不能给患者比较客观的推荐结果。

针对这些情景,我们提出一种医疗信息的推荐方案,对各种医疗信息(医生信息、常识信息、药品信息、处方信息)进行分析,对用户的反馈进行挖掘,然后根据特定用户给出特别的搜索结果和推荐结果,有效地减轻数据过载问题并减少医生误诊的情况。

1.2 国内外研究现状的分析

1.2.1 推荐算法的研究

个性化推荐^[1,2]就是根据信息属性或基于协同过滤对用户的历史行为记录进行深

度分析和挖掘，从而找到用户感兴趣的书籍、电影或文章等信息。这方面的研究已经非常丰富，并形成了具备各自特点的推荐系统。比如 Apache Lucene^[3]的子项目 Mahout^[4]。它包含了 3 个核心主题：推荐引擎（协同过滤）、聚类和分类，使用者可以以高效、可扩展的方式实现这些经典算法。根据所采用的推荐算法以及所处理的信息的不同，推荐系统可分为以下几种。

(1) 基于内容的推荐

基于内容的推荐（content-based recommendation），顾名思义，是根据用户选择过的物品或对象的自身属性，推荐给该用户其它具有相似属性的物品或对象。这类算法只跟物品或对象自身的属性有关，而与用户对物品或对象的评价信息无关。物品或对象需要某种方式来表示，这种表示方式即利用特征提取方法得到物品或对象的内容特征。再者，根据用户的历史关注对象的综合特征，并生成代表用户历史兴趣的向量，最后考察用户兴趣向量与可能被推荐的项目相匹配的程度。内容特征即物品或对象的文本描述信息的特征信息，特征信息的获取可使用应用比较广泛的词频-倒排文档频率（term frequency-inverse document frequency，即 TF-IDF）^[5]。用户兴趣模型的表示和生成有多种方法，常用的有贝叶斯分类算法^[6,7]、决策树、基于空间向量的方法、神经网络等。根据物品或对象的内容特征和用户兴趣向量，得到推荐度量公式如公式 1-1 所示^[2]。

$$u(c,s) = \text{score}(\text{ContentBasedProfile}(c), \text{Content}(s)) \quad (1-1)$$

score 值的计算有多种方式。其中夹角余弦如公式 1-2 所示。

$$u(c,s) = \cos(\vec{W}_c, \vec{W}_s) = \sum_{i=1}^K w_{i,c} / \left(\sqrt{\sum_{i=1}^K w_{i,c}^2} \sqrt{\sum_{i=1}^K w_{i,s}^2} \right) \quad (1-2)$$

其中 u 值是后续推荐结果排序的参考， u 值越大，则越有可能作为推荐结果。除了相似度计算之外，自适应过滤^[8,9]和阈值设定^[10,11]等也是此推荐类型的重要研究部分。自适应过滤^[8,9]研究怎样使用随时间不断增加的浏览对象，进而更新 ContentBasedProfile(c)，使其更加准确地反映用户的兴趣；阈值设定^[10,11]研究查询内容和物品或对象特征信息的匹配方法，从而得到 Content(c) 的精确值。

(2) 基于协同过滤的推荐

协同过滤推荐 (collaborative filtering recommendation) 技术从上世纪 90 年代开始研究, 随后出现了大量优秀的研究成果, 大大促进了推荐技术研究的发展, 并已经成为推荐领域使用最广泛的技术。

协同过滤的算法思想是: 根据用户对物品或对象的评分表, 找到某用户 c 的所有相似用户, 找到相似用户中具有且用户 c 不具有的物品或对象, 并根据相似用户和用户 c 的相似度以及用户相似对物品或对象的评分值来综合确定推荐值。对协同过滤最早的研究有 Grundy system^[12], 后来出现了 Tapestry system^[13], GroupLens^[14], Ringo^[15], PHOAKS system^[16]等。协同过滤算法分为: 启发式 (heuristic-based or memory-based) 方法^[23,24]和基于模型 (model-based) 的方法。

启发式方法^[23,24]的算法思想是: 根据用户 c 相似的用户 c' 对一个对象 s 的评价来预测 s 对用户 c 的推荐值。它包括两个步骤: (1) 计算用户之间的相似度; (2) 综合计算所有相似用户 c' 对物品或对象 s 的评分, 得到物品或对象 s 对用户 c 的推荐值。计算用户相似度的方法有: 基于关联的 (correlation-based) 和基于余弦距离的 (cosine-based) 方法^[2]。前者根据用户对所有物品或对象的评分相似度来计算关联值^[15]。而后者把评分作为向量并计算向量间余弦距离, 得到用户相似度的值^[17,18]。

基于模型的方法, 顾名思义, 利用用户 c 对所有对象的评分训练一个 c 的模型 (model)^[19-23], 并使用概率理论对每个对象的推荐值进行预测。描述公式如 1-3 所示。

$$\gamma_{c,s} = E(\gamma_{c,s}) = \sum_{i=0}^n i \times \Pr(\gamma_{c,s} = i \mid \gamma_{c,s}, S \in S_c) \quad (1-3)$$

其它协同过滤算法还包括贝叶斯模型^[24]、最大熵模型^[25]等。

(3) 基于关联规则的推荐

算法基本思想是: 挖掘物品或对象之间的关联关系, 并完成推荐。关联规则的内容是指和现在的物品具有某种关联的, 比如用户想买铅笔, 通常来说买铅笔的用户很容易的去买卷笔刀, 铅笔跟卷笔刀在属性上没有必然的关系, 但是它们具有经常被一起使用的关联关系, 关联关系使用关联规则发现算法得到。关联规则发现算法主要有 Apriori、Aprioritid、DHP、FP_troe 等。

(4) 混合推荐

根据各种推荐（比如基于内容推荐、基于协同过滤推荐、基于模型推荐等）的优缺点，组合起来完成推荐是混合推荐的基本思想。为了减少时间开销。有很多优秀的信息处理算法得到研究，其中信息分类和信息聚类是重要的两个部分。分类算法包括朴素贝叶斯算法、SGD、SVM、随机森林算法等；聚类算法分为划分法、层次法、基于密度、网格、模型的方法^[11]，具体来说，主要包括如 KMeans、Fuzzy-KMeans、Canopy、Latent Dirichlet、Minhash、Bayesian 等聚类算法^[6]。

1.2.2 医疗信息推荐研究

推荐技术有非常广泛的应用领域，比如电子商务，视频、电影类网站，音乐、社交、图书网站、甚至婚恋网站，但在医疗信息领域却比较少这方面的应用。门户网站，比如搜狐和新浪将医疗信息纳入健康频道，主要包含了药品、常识两个部分，其中药品部分实现了相似功能药品的推荐，而常识部分实现了首页推荐即大众化推荐，同时还具备“浏览了这篇文章的人同时还浏览了”的推荐，这种推荐使用的是关联规则发现算法，挖掘出跟本篇文章一起被其他用户阅读的其它文章。在科学研究领域，浙江大学的于宝福提出将医疗信息分成多个主题（医生、医院、常识、问答），根据用户的浏览记录，创建每个主题的兴趣模型，并作为个性化推荐^[1,2]的依据。他采用了混合推荐算法，即 BP 神经网络算法^[27,28]和 SOM 神经网络算法^[29]，首先将返回的信息根据主题分类，将特征类似的信息聚合在一起，这一步使用的是 BP 神经网络算法，然后将聚类后的信息分别与用户兴趣模型进行匹配，将大于实验设定的推荐阈值的信息推荐给用户，这一步使用的是 SOM 神经网络算法。概括来说，这篇论文本质上是实现对搜索结果的重新排序，根据返回的搜索结果来推荐的，并不能主动将信息推荐给用户，为了将用户感兴趣的信息推荐给用户，需要从整个数据集的角度来做推荐，而不仅仅只是将搜索引擎返回的结果作为推荐的数据源，而且这篇论文对各个领域的医疗信息的推荐所采用的推荐方法都是一样的，本质上都是针对数据本身的内容来做推荐，但是，类似于医生信息的推荐，患者对医生的评价是用户非常关心的，这一点这篇论文没有考虑。而且对于医疗常识信息，仅仅完成个性化推荐是不完善的，没有

考虑信息内容的因素。本文将对这些不足做出改变，并实现医疗处方的推荐。

1.3 本文研究的主要内容

本文主要利用 Elasticsearch^[30,31] 搜索引擎，然后对医疗信息推荐系统的进行研发。Elasticsearch^[30,31]是开源的分布式全文搜索引擎，基于 Apache Lucene^[3]类库创建的。本系统对海量的医疗信息，比如医学常识、处方、药品、医生等信息进行存储、索引、查询和推荐。针对不同种类的医疗信息提出不同的推荐方案。首先对获取的医疗信息数据（主要是 HTML 文档）进行解析，并且设计表示各种医疗信息的数据结构，将其转换成 JSON^[32]格式的传输文档并导入到 Elasticsearch^[30,31]服务器中存储，建立索引，最后针对不同种类的医疗信息给出不同的推荐方法。用户通过浏览器搜索相关信息时根据搜索信息的类别及信息本身得到相应的推荐信息，从而方便用户快速获取感兴趣的信息。本文的主要研究内容组织如下。

第一章，介绍了医疗信息推荐的研究背景、目的和意义，并介绍了相关的国内外研究现状。

第二章，介绍了海量医疗信息推荐的详细设计方案。包括数据获取和解析、数据存储和索引，还有数据推荐三部分。数据获取和解析主要包括各种医疗信息数据的获取、网页数据的抽取及特征数据的生成；数据存储和索引包括数据存储结构的设计、索引字段的选取和索引的建立；针对不同种类的医疗信息提出有针对性的推荐算法。

第三章，详细阐述了医疗信息推荐系统实现。包括：基于内容的药品信息推荐的实现，使用基于项目的协同过滤算法对医生信息进行推荐和基于大众化对医生信息进行推荐以及基于内容对医疗常识的推荐实现，用户兴趣模型构建及更新，设计处方推荐模型。

第四章，对实现的原型系统进行功能测试，展示了医疗数据的搜索和推荐功能，并给出了相关技术的性能测试，并对实验结果加以分析。

第五章，对本文内容进行分析和总结，同时给出了今后工作的研究方向和展望。

2 医疗信息推荐方案设计

医疗信息种类繁多，其中药品、医院、医生、常识、处方信息是用户最为关心的主题。本章针对药品、医生、常识和处方信息的推荐方案分别进行设计。获取医疗信息源数据的途径有：通过网络爬虫到指定网站获取；从项目中给予的数据库中得到。对不同种类的源数据，进行相应的解析，并导入到 Elasticsearch^[30,31]服务器中，从而方便用户进行信息检索。针对药品、医生、常识和处方信息，本章设计相应的推荐方法。

2.1 系统需求分析

功能上来说，本系统需要对药品、医生、常识、处方四类信息进行推荐，当用户搜索某类信息时，以网页的形式返回搜索和推荐结果，实现按需推荐。

系统性能而言，需要满足可扩展性高、实时性强，响应时间影响小，满意度高等要求，以下为具体的几个方面。

(1) 随着信息量的增长，单机显然难以满足信息的存储、检索等要求，因此，本系统需要满足可扩展性的要求；

(2) 随着信息的增长以及用户对信息的浏览记录的增长，推荐结果的精度会慢慢变得不精细，为了提高推荐结果的精度，满足一定的实时性要求，需要经常更新推荐结果，为此，在更新推荐结果时，要考虑减少计算量，从而降低时间开销；

(3) 对系统而言，推荐应用对响应时间的影响不应该太大，增加推荐部分后系统响应时间应该跟不增加推荐部分的系统响应时间在一个数量级上；

(4) 满意度即用户对推荐结果的接受程度，满意度的度量标准可通过经典的信息检索领域度量标准来表征，即查全率和查准率。查准率^[43]即 top 结果中相关结果的比例，查全率^[4]指所有相关结果包含在 top 结果中的比例，也可通过推荐信息的评分来得到。

2.2 系统总体框架设计

本系统工作流程如图 2-1 所示。图中包括两个步骤：第一、存储医疗数据并建立索引，如步骤(1)所示。这一步利用 Elasticsearch 提供的 command 或者 java API 来实现，通过调用 Elasticsearch 提供的 command 或者 java API，用户可批量的将数据导入到 Elasticsearch 服务器中，通过选择分词算法，可对数据进行关键词检索；第二、对医疗数据进行检索和推荐，当用户通过浏览器输入关键词时，如步骤(1)所示，通过调用 Elasticsearch 提供的 REST API，Elasticsearch 服务器端会将检索到的信息和推荐结果一并返回给用户，如步骤(2)所示。其中，药品推荐结果、医生推荐结果、内容相似的常识推荐结果通过离线计算并更新搜索引擎中的文档来实现，而常识的个性化推荐结果和处方推荐结果则需要在线计算得到，依据是预处理后存入到数据库中的特征数据。

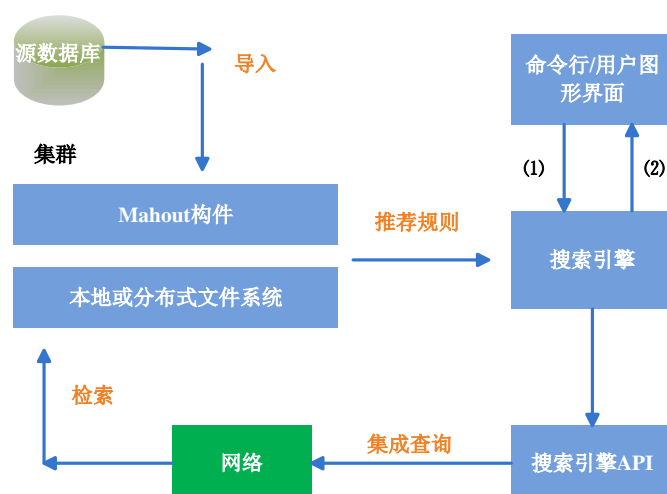


图 2-1 系统工作流程图

通过对系统各个功能模块相互关系的理解，我们提出如图 2-2 所示的系统整体设计框架，系统由上至下分为三个层次：数据预处理及输出层、中间层和信息存储层。其中各个层次解释如下。

(1) 数据预处理及输出层分为预处理模块和输入/输出模块，预处理模块中，源数据包括获取的 HTML 文件、数据库记录和用户浏览记录对应的文档。对于药品信息、常识信息和用户兴趣模型的构建来说，需要经过步骤(1)处理得到解析后的结构信息，然

后通过步骤(5)(6)分别得到药品“适应症”文本和常识内容文本的特征数据。药品信息和常识信息需要通过步骤(3)将信息导入 Elasticsearch 服务器中，特征数据通过步骤(7)存储到数据库中；因为医生信息和处方信息数据源为数据库，所以根据步骤(2)将二者直接导入到 Elasticsearch 服务器中，其中医生信息还需要生成两个数据库表，即医生能力表和患者对医生的评分表，这需要通过步骤(4)来生成，最后通过步骤(7)存储到数据库中；

(2) 中间层包括索引建立模块、数据检索模块、数据推荐模块、推荐更新模块。为减少系统响应时间，特征信息需存储在数据库中，方便快速获取。其中数据推荐模块分为两个部分：在线推荐和离线推荐。在线推荐部分根据数据检索的情况在线计算得到推荐结果，即步骤(11)，处方推荐以及常识信息的个性化推荐都属于这一部分；离线推荐部分主要考虑信息本身的因素，跟用户的行为无关，定时的计算并更新推荐结果；

(3) 信息存储层包括底层文件系统和数据库，底层文件系统用于保存信息本身，数据库保存信息的特征数据以及一些中间数据，比如偏好数据，评分数据等。

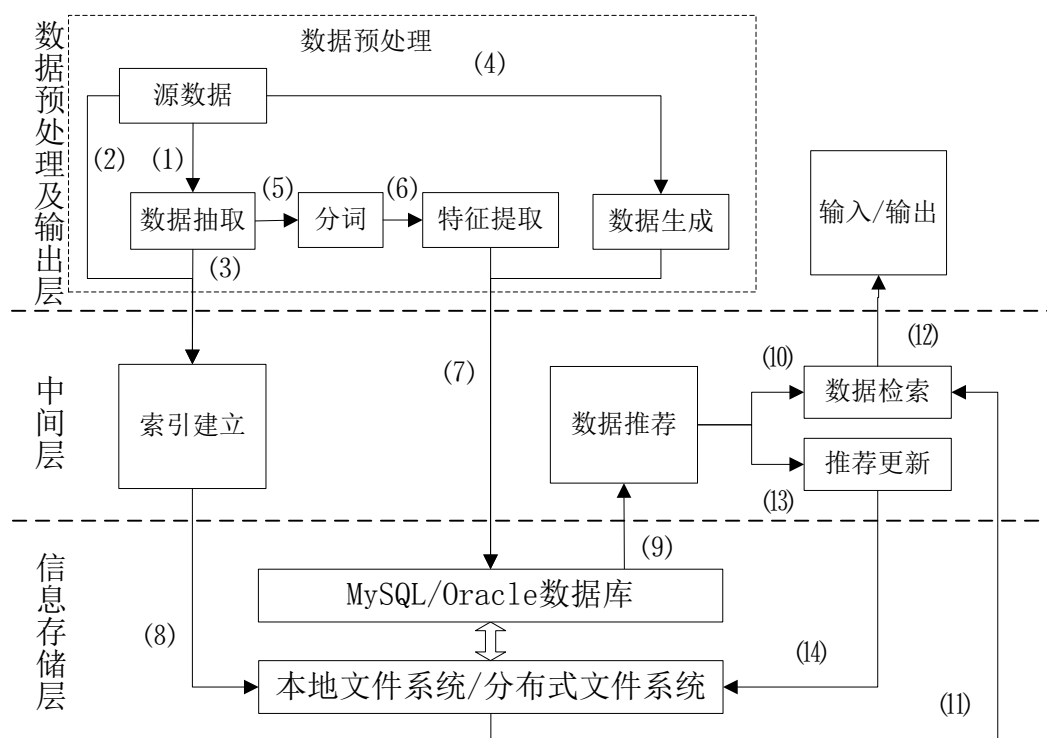


图 2-2 系统整体设计框架

2.3 医疗数据获取和解析模块设计

2.3.1 医疗数据获取模块

数据获取模块主要用于获取医疗数据，如药品、医生、处方、常识、检查结果、检验结果、诊断结果、病史等。系统从两个数据源获取相应数据：一是采取网络爬虫^[33]获取数据；二是从数据库中得到需要的数据。如图 2-3 所示。

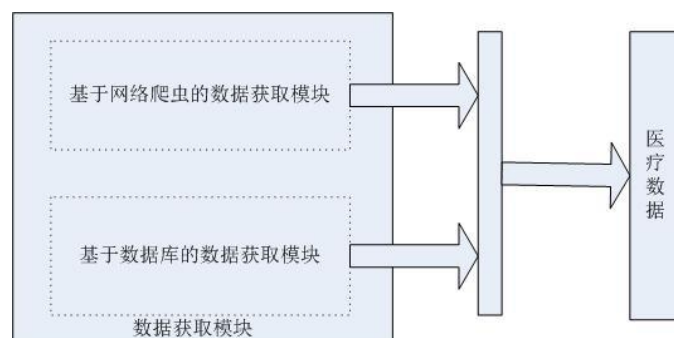


图 2-3 医疗数据获取

(1) 基于网络爬虫的数据获取

网络爬虫^[33]是搜索引擎的重要组成部分，搜索引擎定期使用爬虫从网络中的收集数据。在这里我们也将通过网络爬虫到相关的医疗网站上获取数据。药品数据来自于国家人口与健康科学数据共享平台-药学数据中心(<http://pharmdata.ncmi.cn>)，常识数据来自于新浪健康频道(<http://health.sina.com.cn>)。网络爬虫^[33]的基本工作原理如图 2-4 所示。本次实验中我们采用 WebCollector^[34]，它是一个无须配置，便于二次开发的 JAVA 爬虫框架。它提供精简的 API，只需少量代码就可以实现强大的爬虫，非常方便使用。

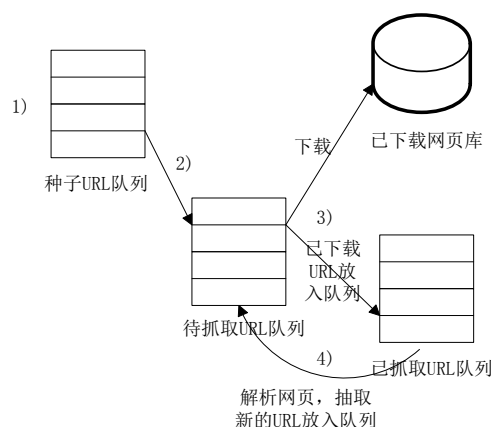


图 2-4 网络爬虫基本工作原理

基本工作原理如下。

- 1) 选取种子 URL;
 - 2) 将这些 URL 放入待抓取的队列;
 - 3) 从待抓取队列中读取并解析 URL, 下载并存储相应的网页, 标记 URL 为已抓取, 并将其放入已抓取队列;
 - 4) 解析网页, 得到网页中嵌入的 URL, 并放入待抓取队列。
- (2) 基于数据库的数据获取

由于实验过程中有很多数据是从数据库中获取, 因此, 基于数据库的数据获取模块用于获取数据库里面的信息, 这里我们选择关系型数据库 Oracle^[35,36], 因为 Oracle 相较于 MySQL^[37,38]来说更适合在大型项目中运用。创建数据库 DRUG, 并创建多个用户, 重要的用户名及其导入文件名如表 2-1 所示。

表 2-1 DRUG 数据库用户名及导入文件名

数据库用户名	导入文件名	备注
LABUSER	labrecord.dmp	检验记录
MEDRECUSER	medrecrecord.dmp	诊断记录
PHARMACYUSER	pharmacyrecord.dmp	处方记录
EXAMUSER	examrecord.dmp	检查记录
COMMUSER	commrecord.dmp	基本记录, 如病人, 医生, 各种字典等

2.3.2 医疗数据解析和特征提取模块

(1) 数据的描述

为更好的描述医疗信息数据, 我们设计了数据抽取模块, 主要是从网页和数据库记录中抽取表征某种药品、处方、医生和常识数据。根据数据来源不同可以分为下面两种类型的数据。

- 1) 非结构化数据: 主要通过网络爬虫获取, 药品、常识都属于这一类数据;
- 2) 结构化数据: 主要由数据库中获取的数据, 数据库中包括了多个医院的数据,

包括药品、病人、处方、医生、检查结果、检验结果、诊断结果、病史等数据。

其中，描述药品的数据字段有药品名、药理作用、适应症和不良反应等；描述医生的数据字段有医生的个人信息、专长、科室、级别等；而描述常识的数据域有类别，内容、题目等；处方数据字段有处方号、日期、相关的处方药品等。所有的信息将转换成 JSON 格式。

针对药品数据，我们使用开源的 Jsoup^[39]对网页的标签进行解析，Jsoup 是 java 实现的 HTML 解析器，可以从 URL、HTML 文件和字符串中解析 HTML，用户可以通过 Jsoup 选择器的定义来开发出非常强大的 HTML 解析功能。这里我们用到了 Jsoup 的 API 来解析描述药品的 HTML 文件，因为相关 HTML 文件是从专门的医疗网站中抓取，具备一定的结构，当标签解析完之后，比较容易选取标签的数据然后写入对应的字段。对于描述常识的 HTML 文档来说，需要提取其内容数据。因此，我们采用 Apache Tika，Apache Tika^[3]是通用的文本内容分析工具，能够解析多种格式（例如 HTML、PDF、Doc）的文件，侦测和提取文件的元数据和内容数据。在此我们利用其 API 来解析医疗信息的 HTML 文档中的文本内容数据信息。

(2) 特征数据提取

特征数据是基于数据内容推荐的重要组成部分。它是文档相似度计算的依据。因此，这部分是研究的重点之一。按照方案的设计，需要提取特征数据的信息有药品信息、常识信息。特征数据获取过程如图 2-5 所示。图中第四部分中，药品信息则采用分类算法^[43]对药品进行分类，而医疗常识信息采用聚类算法^[43]进行处理。二者的目的都是根据数据之间的相似度将数据集分成多个数据簇，在后续的文档相似度计算中只需计算同一簇内的对象之间的相似度，进而降低计算开销。这一处理在数据量非常大时具有很大的意义，分类算法和聚类算法的选取将在后文阐述。

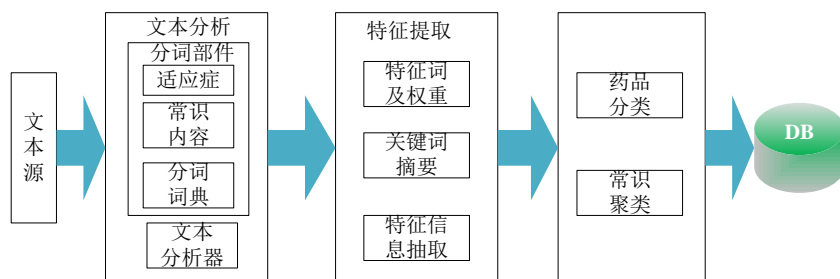


图 2-5 特征数据生成过程图

首先,对文本信息进行中文分词,这里我们采用 ik^[40]分词器。ik 具有非常快的分词速率,约为 60 万字/秒。此外,由于医疗信息里有很多专业术语,比如“逆转录酶”。为了更好的适用于医疗信息,需要针对医疗数据的分词进行特殊处理,主要有两套方案:方案一,收录专有的医疗术语,然后将其放入到自定义分词词典中;方案二,采用统计语言模型,其原理是根据句子切分的概率大小来确定怎样切分句子。例如: $p(\text{流行/性/感冒})=0.02$, $p(\text{流行性/感冒})=0.03$ 等等。对任意句子 s 给出不同分词方式的概率,概率最大者即为合适的分词。需要对大量的文档进行训练得到模型。这里我们选择方案一。

然后,对文档内容进行特征提取,将文本文档表示为向量,为后续的处理带来方便。我们采用空间向量模型 (Vector Space Model, VSM) 来表示文本文档的特征,向量的每一个域是文本中出现的一个词的权重。权重的计算方法有多种,包括对文档中出现的单词直接计数、TF、TF-IDF 等。TF 将词在文本中出现的频率作为词的权重,是对直接计数的一种归一化处理,目的是让不同长度的文本模型有统一的取值空间,从而方便文本相似度的比较。TF-IDF 是对 TF 的一种改进方法。这种方法的思想是词的重要性随着它在文件中出现的次数成正比增加,但同时随着它在所有文本中出现的频率成反比下降。这样,那些高频但无意义的词将得以过滤。本系统采用 TF-IDF 算法,它的使用最为广泛,也更加能够反映文档的内容。

2.4 医疗信息存储和索引模块设计

2.4.1 医疗信息存储

系统中出现了多种类型数据,有药品信息、医生信息、常识信息、处方信息以及对这些信息处理过后的特征数据。需要对它们进行持久化存储。对于药品、医生、常识、处方这些信息来说,它们是要呈现给用户的数据。因此,我们通过 Elasticsearch 的 API 将它们导入到 Elasticsearch^[30,31]服务器中。在将数据导入到 Elasticsearch 服务器之前,需要设计相应的数据结构将医疗信息转化为 JSON^[42]格式。Elasticsearch 有以下四种数据存储方式。

- (1) 在本地文件系统中存储;

(2) 在分布式文件系统中存储;

(3) 在 Hadoop 的 HDFS 中存储;

(4) 在亚马逊的 s3 云平台中存储。

本系统选择第一种存储方式。但是,为了满足可扩展性的要求,我们将构建由多个 Elasticsearch 服务器节点构成的集群。Elasticsearch 集群非常简洁,集群中有一个节点为主节点,其它为从节点。主节点可以通过选举产生,也可以通过配置来指定,集群的配置也很简单,在一个局域网内,只需将安装的 Elasticsearch 节点的集群名字 `cluster.name` 配置成相同的名字即可,因为同一局域网下集群名字相同的节点将自动构建成一个集群。

对于特征数据这类在预处理阶段得到的数据以及数据生成过程中产生的偏好数据,它是推荐的依据。我们将它放置到数据库中,方便利用数据库的索引机制快速获取到相关的数据。我们选取关系型数据库 MySQL。

2.4.2 医疗信息索引建立

搜索引擎的索引是“单词-文档”的映射,当用户检索信息时,搜索引擎首先将用户提供的关键词进行切分,然后查找索引返回相关的信息。索引的建立有多种方式,比如“倒排索引”、“签名文件”、“后缀树”等。实验表明,“倒排索引”是比较好的选择,通过倒排索引,根据切分的单词可以快速获取包含这个单词的文档列表。倒排索引的结构示意如图 2-6 所示。它主要有两部分组成:“词典”和“倒排文件”。其中,“词典”是导入到搜索引擎的文档集合中出现过的所有单词构成的单词集合;“倒排文件”由“倒排列表”组成,“倒排列表”记录了包含某个单词所有文档的文档列表以及该单词在该文档中出现的位置信息,每一条记录称为一个倒排项 (Posting),多个倒排项构成倒排列表。根据切分的关键词,查找词典,找出包含该关键词的所有文档,并根据某种规则,比如根据该关键词的在文档词频大小或者根据词频和文档长度的比值大小,对所有的文档进行排序,最终给用户返回检索结果。倒排索引具有结构简单,易于理解的特点,大多数商业搜索引擎采用这种结构来构建索引。

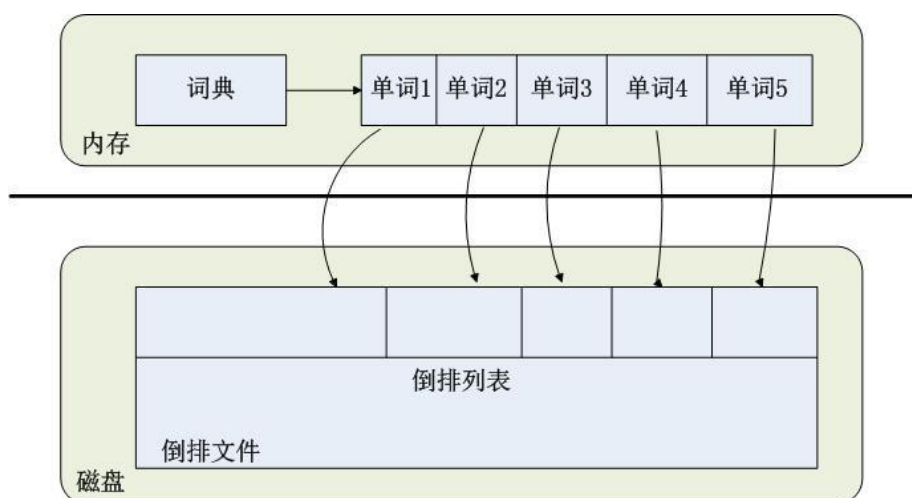


图 2-6 倒排索引示意图

Elasticsearch 的索引机制采用了 Lucene 的倒排索引机制,建立索引时,Elasticsearch 先要通过 Mapping 来对索引库中的索引的字段名及数据类型进行定义。索引建立后,Elasticsearch 会将一个完整的索引分成多个分片 (shard),并将多个分片均匀地分布到集群节点中,构成分布式搜索引擎。索引建立过程中可以通过配置产生索引的多个副本,从而提高系统的容错性,当系统中某个节点宕机或者某个分片丢失时,系统还能够通过相应的机制对外提供服务。

2.5 医疗数据推荐模块设计

2.5.1 药品信息推荐

为了更加客观的推荐主治功能相似的药品,我们的做法是根据药品信息中“适应症”字段计算得到药品之间的相似度。但是,为了降低计算开销,考虑将药品进行分类。在 Apache mahout^[4]项目中,有多种分类的方法,比如 SGD 算法、SVM 算法、朴素贝叶斯算法及补充朴素贝叶斯算法、随机森林算法^[4]等。但是,针对药品这类信息而言,可以用非常简便的方法来进行分类,我们考虑将药品“所属类别”这个字段作为分类的依据,比如“性传播合理用药”、“抗病毒合理用药”等,分类示意图如图 2-7 所示。这些类别让药品得到了很好地区分,不同类别的药品间的相似度极低,在计算药品间的相似度时,只需计算分类后数据簇内药品的相似度,簇间药品间相似度不需要计算。

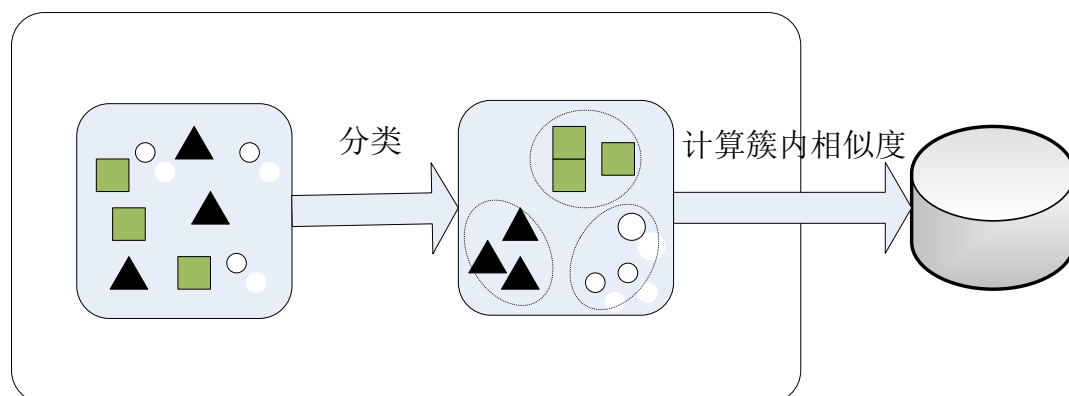


图 2-7

药品分类示意图

仅仅从药品“适应症”来考虑并作为药品推荐的依据有一些不足的地方，就像用户只通过观察药品简介一样，缺乏客观性。为此，我们考虑将药品的使用量来影响推荐结果，因为越好的产品其使用就越广泛、越流行。同时，设想这么一种情况，一个新出产的药品 A，另一个出产有一段时间的药品 B，药品 A 疗效好于药品 B，从历史累计使用量来看，药品 B 应该大于药品 A，推荐结果更偏向于将药品 B 推荐给用户，这样便不利于新品的推广。为此，我们考虑引入“数据热度”来影响推荐结果，“数据热度”即数据在近期被关注的程度，在某种程度上，它是更加“好”的信息的度量标准。我们的做法是统计 3 年内的产品使用量，以反映数据热度。推荐预处理流程图如图 2-8 所示。最后将综合考虑计算得到的药品相似度数据存入数据库中。

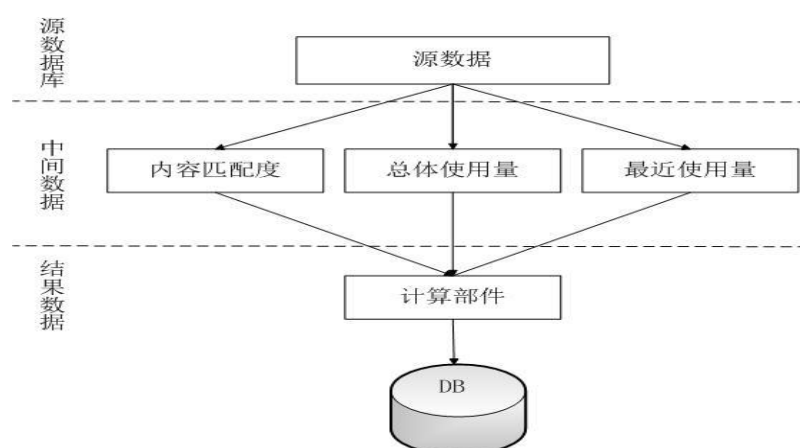


图 2-8 药品推荐预处理流程图

2.5.2 医生信息推荐

医生信息的搜索已经有一些应用，比如春雨医生、好大夫在线等等。它们的操作

模式是用户在线询问医生自己的症状以及获取相应的解决方法。它们收录了很多大型医院的医生信息，以目录的形式提供给用户，让用户自己去向医生询问。这种方式下用户选择医生的依据是医生的个人信息，比如工作单位、临床年限、年龄、专长等。用户可能还会关注其他用户对医生的评论。但是，总的来说，这些依据还是过于片面，缺乏客观性，不能很好的满足用户的要求。比如说其他用户对医生的评论有好有坏，而且评论的价值也是值得商榷的。

为了做到更好的将医生推荐给用户，我们考虑观察医生临床治疗的成果，每个用户来医院治疗都有备案，治疗结果也是其中的一项。治疗结果有“治愈”、“好转”、“恶化”、“死亡”等。通过处理这些治疗结果，可以得出用户对医生的评分表和医生总得分表。

医生治疗的病人越多、治愈或使其好转的病人越多，那么这个医生就越能得到用户的认可，用户就越可能选择这个医生为自己看病。因此，医生推荐的其中一个策略是根据某领域内医生的流行程度来推荐，即医生的总得分越高，他/她被接受的可能性就越大。另外，如果用户认可某个医生 A，但是因为医生 A 的预约已经排满，此时，他/她会选择其他医生，一般而言，他/她会选择跟医生 A 相似的其他医生，所以医生推荐的第二个策略是推荐和用户此前认可的医生相似的其他医生。这时，问题的关键是怎样定义两个医生之间是相似的。我们的决策是首先医生所在的科室应该一样，不同科室之间医生不具有可比性；然后是所有用户对二者的评价类似，相当于采用了基于项目的协同过滤算法。基于项目的协同过滤算法原理图如图 2-9 所示，通过图可以得到用户和医生间的评分表，如表 2-2 所示，星号代表用户对该医生的能力是认可的。对用户 3 来说，他/她认可医生 A，而从用户的评价相似度来看，医生 C 跟 A 比较类似，所以将医生 C 推荐给用户 3。评价相似度的计算可以根据所有用户对医生的评价构成评分向量，对每个医生而言，都有一个评分向量，评分向量的相似度可以用向量的夹角余弦来判定，也可以采用欧式距离的方法。在此，采用夹角余弦的方式，根据这种方式，能够找到评价比较一致的两个评分向量，因为根据评分向量生成方法可知，评分向量跟医生的从医时间有关，如果采用欧式距离，很容易受到从医时间的影响，不利于找到相似的医生。

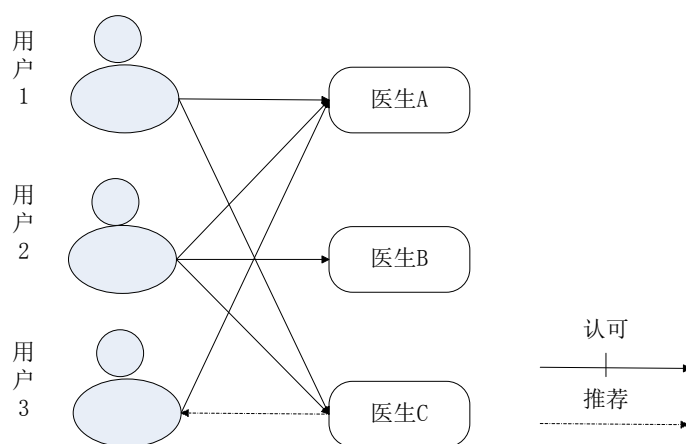


图 2-9 基于项目协同过滤算法原理图

表 2-2 用户对医生偏好表

用户 \ 医生	A	B	C
1	*		*
2	*	*	*
3	*		

2.5.3 常识信息推荐

常识信息具有极其广泛的主题。比如“养生常识”、“生活常识”、“饮食常识”，且更新速度也快。每个人每个时期关注的信息主题会有所不同。在某个时期内，他/她会持续关注同一类常识。为此，可采用两个策略完成推荐。

第一，推荐内容相似的其它信息。计算内容相似度与计算药品相似度类似，即分词，取得表示文档内容的特征向量，相似度计算三个步骤。但是，由于常识信息的信息量巨大且信息增长非常迅速，为保证推荐的准确性需要非常大的计算量，而计算量大就很难保证推荐的实时性。为此，需要一个降低计算量的优化方案。在本次实验中，我们的解决方法是先根据内容进行聚类，把大的数据分成一个个小的数据簇（Cluster），随后在计算常识信息相似度时就只需计算数据簇内信息之间的相似度。需要选择性能优异的聚类算法。聚类（Clustering）是数据挖掘领域的经典问题，根据不同的评判标准出现了比较多的算法，比如 K 均值聚类算法、华盖聚类算法、模糊 K 均值聚类算法、狄利克雷聚类算法^[43]等。其中，KMeans 算法^[4,41]将数据分成 k 个数

据簇，是一个简单的算法，具有算法复杂度低、簇的个数确定、簇之间的数据不重叠等特点；Canopy 算法^[4,41]是一种快速近似的聚类算法，聚类后数据簇之间有重叠，具有簇的数量不需要事先指定、计算速度快块的特点，只需遍历一次数据。但是，这也导致算法无法给出精确的簇结果；模糊 K 均值算法^[43]是在 K 均值算法基础上的扩展，它允许不同簇之间有重叠。在现实聚类问题中，给定一个数据集，事先很难知道应该分成多少簇，因此直接运用 K 均值聚类算法可能偏差较大，需要一个确定应该分成多少个簇的处理，所以本次实验首先使用 Canopy 算法对数据集进行预处理，得到需要分成多少个簇 k 的依据以及初始的聚类中心点，然后使用 k 均值聚类算法对数据进行聚类，聚类过程如图 2-10 所示。

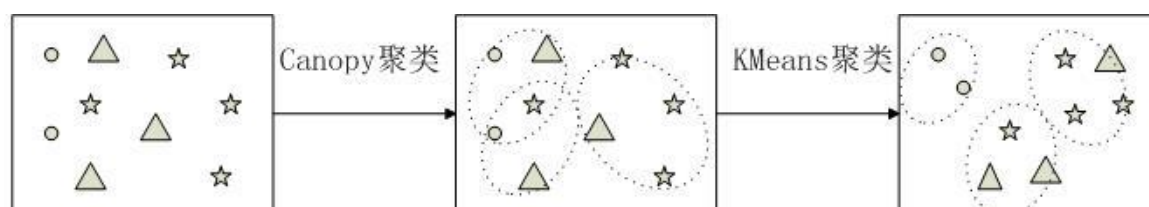


图 2-10 聚类过程

第二，根据最近关注话题的情况推荐信息。为此，需要挖掘用户的最近的浏览记录，构建用户的近期兴趣模型。兴趣模型的表示方式有多种，比如关键词表示法、向量空间表示法、主题表示法等。这里我们采用向量空间表示法（VSM）。向量空间表示法的表示方式为 $\{(t_1, w_1), (t_2, w_2), \dots, (t_n, w_n)\}$ ， t 代表关键词， w 代表权重。根据 TF-IDF 算法来实现。用户模型的更新方面，因为用户感兴趣的信息总是跟自己最近浏览过的信息有关。为了简化设计，我们总是让系统在用户每次登陆的时候重新生成模型，比如选取用户最近浏览的 20 篇文档，选取它们的特征，更新该用户的兴趣模型。推荐信息时，我们先计算得到兴趣模型大致在哪个数据簇中，然后比较数据簇信息和兴趣模型中信息的相似度，为了减少计算复杂度，当相似度达到一个阈值且跟用户之前浏览过的信息不一致即可作为推荐结果，推荐的信息规定为 5 个，达到 5 个以后将停止相似度的计算。

2.5.4 处方推荐

处方推荐旨在给医务人员、药店职员、甚至是普通用户一些诊断、处方的建议，

减少现实社会中医生误诊的问题，它能够根据用户的症状描述、检查结果、检验结果给出最为合理的一些处方方案。在本次实验中，我们用到诊断记录、检查记录、检验记录、处方记录等数据，首先对检验记录进行处理，得到男性女性在各方面的正常标准，计算公式可以采用计算数学期望的公式，即对所有正常的指标进行相加取平均值；然后找到每项指标中被判为高（H）和低（L）的案例，找出其中被判为高的最低值 H_{\min} 和被判为低的最高值 L_{\max} ，并以此作为判断诊断结果的标准。系统使用过程中，用户输入症状（这点需要符合疾病字典的描述）、检查结果、检验结果，系统根据这些信息，进行处理和判断，得到症状描述文本和检查结果文本的特征向量以及所提供的检验结果的定性判断结果，并利用数据库的信息，通过推荐模型得到最合理的一些处方方案，再根据用户的病历、对药物的过敏情况调整推荐结果。具体来说，考虑到症状描述和检查结果是使用文本进行描述的且都较为简短，因此获取它们的特征向量过程中，我们采用关键词表示法表示特征向量，包含的关键词的权重为 1，否则为 0，分别计算症状描述和检查结果的相似度 $\text{similarity}(\text{Description})$ 和 $\text{similarity}(\text{Test})$ ，再计算检验结果的相似度 $\text{similarity}(\text{Lab})$ ，检验结果的计算方式为定性结果（比如高 H，低 L）相同的项数 ItemSame 除以总项数 ItemSum ；最后加权各相似度得到相似度度量 similarity 。方案中主要包括数据预处理过程和处方推荐过程。处方推荐过程如图 2-11 所示。其中计算相似度是通过查询类似症状的诊断结果数据库，找到此前的记录中跟本次诊断结果相似的案例，具体的计算方式可以是统计二者在各项指标方面一致的项目数，项数越多则代表越相似，然后根据用户的病历、对药物的过敏情况进行过滤，得到推荐结果。

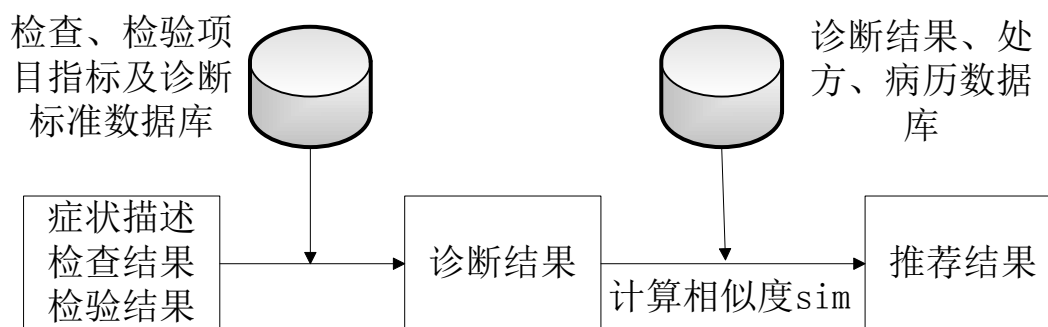


图 2-11 处方推荐过程示意图

2.6 本章小结

本章从需求的角度出发，指出医疗信息搜索和推荐具有重要意义。给出了系统总体设计框架，并分别解析了框架中的各组成部分，包括数据获取模块、数据存储和索引模块、最后详细介绍了药品、医生、常识、处方推荐的方案设计。

3 医疗信息推荐实现

本系统基于 Elasticsearch 搜索引擎对信息进行存储、索引和搜索。对每一类医疗方面的信息，设计对应的数据结构，然后将信息按照给定的数据结构转化为 JSON 格式，并调用 Elasticsearch 提供的 JAVA API，导入到 Elasticsearch 服务器集群中，建立索引。本章主要是在上一章的基础上详细介绍推荐方案各部分的实现。

3.1 信息存储和索引模块

本系统主要针对 4 类医疗信息：药品信息、医生信息、常识信息和处方信息。按照第二章的数据获取和解析方法，结合项目设计需求，我们提取了上述四种类型数据的以下信息（如表 3-1~表 3-4 所示）。

表 3-1 药品信息关键字段

关键字段	字段名称	说明
DATA_TYPE	数据类型	药品、医生、常识、处方
DRUG_CODE	药品代码	唯一识别某种药品
DRUG_TYPE	药品类别	比如抗病毒，妇产科用药等
DRUG_NAME	药品名称	
ACTION	药理作用	
INDICATION	适用症	药品功能，能治疗的疾病
ADVERSE_REACTION	不良反应	

表 3-2 医生信息关键字段

关键字段	字段名称	说明
DATA_TYPE	数据类型	药品、医生、常识、处方
DEPT_CODE	科室代码	医生所在科室
NAME	姓名	医生姓名
EMP_NO	人员编号	唯一识别某个医生

表 3-3 常识信息关键字段

关键字段	字段名称	说明
DATA_TYPE	数据类型	药品、医生、常识、处方
DOC_CODE	文档代码	唯一标识文档
TITLE	标题	字符串
CONTENT	内容	字符串

表 3-4 处方信息关键字段

关键字段	字段名称	说明
DATA_TYPE	数据类型	药品、医生、常识、处方
DIAGNOSIS	诊断结果	
EXAM	检查结果	可为空
LAB	检验结果	可为空
PRESC_DATE	处方日期	
PRESC_NO	项目序号	
DRUG_CODE	药品代码	所用药品的代码
DRUG_NAME	药品名称	

在信息处理过程中，会产生一些中间数据，比如药品适应症的特征数据和医疗常识的特征数据，它们主要是以 Key-Value 的形式存储在 MySQL 中。

确定上述数据结构后，通过 Elasticsearch^[30,31]的 JAVA API 定义 mapping 函数对索引库中索引的字段名及其数据类型进行定义，并建立索引。索引的分片规则、副本管理及集群定义都要通过配置文件来定义。重要的参数有 cluster.name、node.name、index.number_of_shards、index.number_of_replicas、index.analysis.analyzer，分别代表集群名、节点名、索引分片数、索引副本数、分词器。根据可靠性、高性能的要求。本系统采用两台主机构成一个小集群，集群的构建简单方便，只需将 cluster.name 均设置为 my_clusterName 即可。分片数按默认设置为 5，副本数设置为 2，分词器采用 ik。表 3-5 是药品信息关键字段的 mapping 定义。医生、常识、处方信息的 mapping 定义类似。

表 3-5 药品关键字段的 mapping 定义

	Type	indexAnalyzer	searchAnalyzer	store
dataType	Integer	not_analyzed	not_analyzed	yes
id	String	ik	ik	yes
drugType	String	ik	ik	yes
name	String	ik	ik	yes
indication	String	ik	ik	yes
interaction	String	ik	ik	yes

注：type 表示字段类型，indexAnalyzer、searchAnalyzer 分别表示建立索引和检索过程中使用的分词器，store 决定是否在索引中存储该字段；dataType 表示数据的类型，如药品、医生等；id 表示文档的唯一 id；drugType 表示药品类别；name 表示药品中文名；indicate 表示适应症；interaction 表示不良反应。

3.2 药品信息推荐模块实现

药品的推荐主要是基于药品适应症的相似度来做推荐，并引入药品的总体使用量和药品使用热度来综合影响推荐结果。相似度是通过计算表示“适应症”的特征向量间相似度来得到，而药品的总体使用量和药品使用热度主要是通过统计每种药品的使用量和最近 3 年该药品的使用量，二者都是根据药品出库记录来统计。药品出库记录由多个医院提供，药品出库记录的关键字段如表 3-6 和表 3-7 所示。然后归一化处理得到药品使用量和药品使用热度。最后综合三个因素进行推荐。

表 3-6 药品出库主记录的关键字段

字段中文名	字段英文名	备注
出库单号	DOCUMENT_NO	唯一标识一次出库操作
出库日期	EXPORT_DATE	

表 3-7 药品出库明细记录的关键字段

字段中文名	字段英文名	备注
出库单号	DOCUMENT_NO	唯一标识一次出库操作
药品代码	DRUG_CODE	药品字典定义
数量	QUANTITY	

推荐标准计算公式。

$$score(i) = s(i) * sim_weight + w(i) * con_weight + h(i) * pop_weight \quad (3-1)$$

其中 $score(i)$ 表示药品 i 的推荐评分, $s(i)$ 代表相似度, sim_weight 代表相似度权重, $w(i)$ 代表归一化后得到的药品总消耗量; con_weight 代表消耗量的权重, $h(i)$ 代表药品的热度, pop_weight 代表热度权重。其中 $sim_weight + con_weight + pop_weight = 1$ 。

3.2.1 药品信息相似度计算

(1) 预处理

通过网络爬虫获取到表示药品信息的 Html 文件并经过 Jsoup 对文件标签解析获取各个字段之后, 为了便于计算信息之间的相似度, 需要经过预处理得到文本的特征向量。具体步骤如下。

- 1) 遍历源数据, 即表示药品信息的 Html 文档;
- 2) 调用 Jsoup 的 API 进行标签解析, 并将各字段写入到药品信息结构中;
- 3) 对解析获取的“适应症”文本进行分词、运用 TF-IDF 算法得到特征向量, 将文件标识(这里使用文件名)和对应的 TF-IDF 特征向量作为键值对写入到 MySQL 中。

最终得到文档相似度计算的依据, 分词采用 ik 并自定义词典的方式。词典中收录医学专有名词, 比如“逆转录酶”, “抗病毒”等。表 3-8 列出一些专有名词。

表 3-8 专有名词

逆转录酶	抗病毒	聚羧乙烯	口含片
粘着性	松片	交联聚维酮	糊精
二甲基乙酰胺	泡腾片	酞剂	灌肠剂
脆碎度	环磷酰胺	单纯疱疹病毒	胸苷激酶磷酸

文档的特征向量代表单词的标识 T_i 和 TF-IDF 算法计算得出的权重 W_i 组成, 即 $\{T_1, W_1), \dots, (T_n, W_n)\}$ 。

(2) 相似度计算

文档间相似度的计算方法有多种。如欧式距离、夹角余弦、斯皮尔曼相关系数、基于对数似然比等等。这里我们使用夹角余弦方式，因为在文本相似度计算中夹角余弦的使用最为广泛。夹角余弦判断相似度的依据是原点到两点的两条射线间夹角的余弦值，计算公式如 3-2，它是从方向上考虑，作为相似度的标准度量。归一化得到 3-3。

$$sim_{\cos}(d_i, d_j) = \frac{\sum_k d_{i,k} \cdot d_{j,k}}{\sqrt{\sum_k d_{i,k}^2} \cdot \sqrt{\sum_k d_{j,k}^2}} \quad (3-2)$$

$$sim_{\cos}(d_i, d_j) = \frac{sim'_{\cos}(d_i, d_j) + 1}{2} \quad (3-3)$$

需要进行相似度计算的药品信息为药品类别一致的药品，药品类别不一致我们就认为二者不具备可比性。相似度计算的流程如图 3-1。

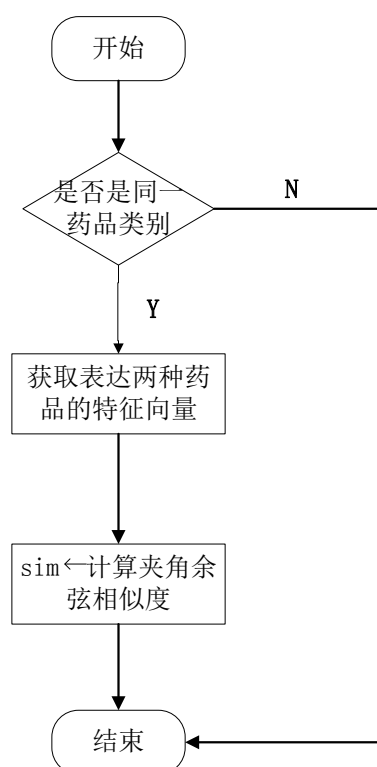


图 3-1 相似度计算

3.2.2 药品总体使用量和使用热度的归一化

每种药品的总体使用量和近 3 年使用量即使用热度可以通过统计获取。归一化处理依赖于根据相似度计算得出的结果，获取了这些结果后，需要进行归一化处理，从而方便计算推荐评分。直观上来讲，应该获取所有同类药品的使用量并计算占有率，如公式 3-4 和 3-5 所示。但是，获取同一“适应症”的所有药物并不容易，为了简化处理，在得到基于相似性的 m 个推荐结果后，我们再统计推荐结果中各药品的使用量 $u(i)$ ，药品最近 3 年的使用量 $h(i)$ ，归一化后得到 $w(i)$ ， $q(i)$ 。归一化公式为 3-6 和 3-7 所示。

$$w(i) = u(i) / \sum_{i=1} u(i) \quad (3-4)$$

$$q(i) = h(i) / \sum_{i=1} h(i) \quad (3-5)$$

$$w(i) = u(i) / \sum_{i=1}^m u(i) \quad (3-6)$$

$$q(i) = h(i) / \sum_{i=1}^m h(i) \quad (3-7)$$

3.2.3 权重决策及推荐

至此，我们已经计算得到了药品“适应症”相似度、药品总使用量、药品使用热度三个维度各自的评分，接下来我们需要给各个维度分配权重。我们的原则是以相似度作为主导因素。这里我们简单的分配权重，其中 $sim_weight=0.6$ ， $con_weight=0.3$ ， $pop_weight=0.1$ 。

正如第二章讲述的那样，当用户浏览某种药品的信息时，系统从数据库中获取与该药品的相似的 m 种药品、各药品使用量及近 3 年使用量。相似度、使用量通过离线计算得到，并根据上一节讲述的方法获取到归一化数据，并最终得到评分值，排序，过滤，再得到推荐结果。

3.3 医生信息推荐模块实现

3.3.1 基于医生流程度推荐实现

医生流程度的定义是患者找他/她看病的人数以及治愈或者病情好转的人数，因为越知名的医生看的病人就越多，且治愈的病人也越多，所以从结果可以反推医生的能力是否足够优秀。

根据这个思想，我们统计诊断记录，对不同的治疗情况我们打分，打分规则如表 3-9 所示。创建医生得分表，得分表各字段如表 3-10 所示。

表 3-9 打分规则

治疗结果	分值
治愈	2
好转	1
未治	0
无效	-1
死亡	-2

表 3-10 医生得分表

字段	说明
DEPT	科室
NAME	姓名
POINT	总得分

用户浏览医生信息时，从数据库中找到跟浏览的医生同一个科室的所有医生的得分情况，简单排序，过滤即得到推荐结果。

3.3.2 基于项目协同过滤推荐实现

为了获得相应的推荐，首先要处理数据库中的表得到患者对医生的评分表。这一步还是通过处理诊断记录表。评分表的字段如表 3-11 所示。

表 3-11 患者对医生评分表

字段	说明
PATIENT_ID	患者 ID
DOCTOR_NAME	医生姓名
POINT	评分

推荐过程如图 3-2 所示。

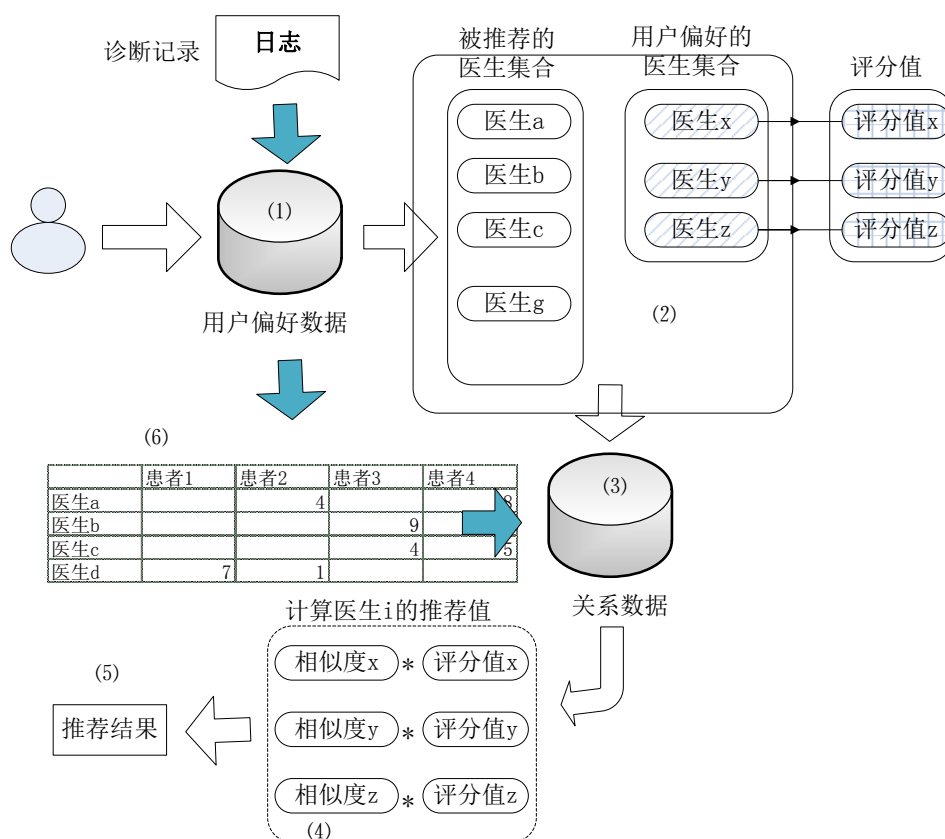


图 3-2 基于协同过滤的医生推荐过程

其中，用户偏好数据即患者对医生的评价表，关系数据表示医生间相似度，当用户浏览医生信息时，推荐步骤解析如下。

- (1) 从数据库中找到患者评价过的医生以及还未评价过的医生；
- (2) 过滤掉跟正在浏览的医生不在同一科室的医生信息，保留同一科室的医生信息，从而保证更加精准的推荐结果并降低计算开销；
- (3) 查找过滤后的已评价医生和未评价医生这两个数据集之间的相似度。相似度计算通过步骤(6)；

- (4) 计算推荐值。相似度 s_{i*} 表示医生 i 和医生 $*$ 的相似度, 评分值 r_{ui} 表示用户对医生 $*$ 的评分值;
- (5) 排序, 推荐;
- (6) 表示医生之间相似度的关系数据, 由偏好数据计算得来, 采用夹角余弦的方式。

3.4 常识信息推荐模块实现

常识推荐需从信息内容相似性和用户个性化推荐两个方面考虑。为了降低计算开销, 减小响应时间, 在计算信息间内容相似度之前先进行聚类处理, 得到多个信息簇, 这样在计算信息间相似度时只需计算信息簇内部信息间的相似度。为了能让用户迅速找到自己感兴趣的信息, 我们设计让系统挖掘用户浏览过的信息, 得到用户的兴趣模型, 并依据此模型为用户推荐信息。

3.4.1 常识信息聚类实现

信息聚类^[42,43]的目的是在计算信息间相似度时能够降低开销。如第二章所述, 我们先用 Canopy 算法^[4,41-44]获取到大致的聚类结果, 再通过 KMeans 算法^[4,41-44]得到最终的结果。下面是 KMeans 算法和 Canopy 算法的简单描述。

(1) KMeans 算法

- 步骤一: 任意选取 k 个中心点;
- 步骤二: 计算其它点分别到 k 个中心点的距离, 到哪个中心点的距离短则该点就归并为那个簇;
- 步骤三: 根据步骤二生成 k 个点簇, 分别移动 k 个点簇的中心点到各自点簇的中心;
- 步骤四: 重复步骤二、步骤三, 直到 k 个中心点的位置不再变化。

KMeans 算法最终能收敛, 但研究表明, KMeans 算法最终得到的中心点位置依赖于它们的初始位置, 并且在实际应用中, k 的值是不确定的。为此, 需要快速确定 k 值及中心点位置, 这里我们采用 Canopy 算法来实现, Canopy 是一种近似聚类算法技术, 虽无法得到精准的聚类结果但是运算速度非常快, 只需遍历一次数据集, 而

KMeans 算法需要不断的迭代，遍历多次数据集。

(2) Canopy 算法

步骤一：选取距离阈值 $T1$ 、 $T2$ ，且 $T1 > T2$ ；

步骤二：遍历点集 $list$ ，如果华盖集 $canopy_list$ 为空，则新建一个华盖并加入到 $canopy_list$ 中，如果华盖集不为空，则计算点到各华盖的距离，如果跟某华盖距离小于 $T1$ ，则将该点加入该华盖，如果都大于 $T1$ ，则创建新的华盖并加入到华盖集 $canopy_list$ 中；

步骤三：如果点跟某个华盖的距离小于 $T2$ ，则该点加入该华盖，并从点集 $list$ 中删除该点；

步骤四：重复步骤二、步骤三直到点集 $list$ 为空。

两个算法中我们都采用余弦距离作为文本间相似度的距离测度。阈值 $T1$ ， $T2$ 按表 3-12 选取测试。经过 Canopy 算法的聚类处理后，能够有效的减少 KMeans 算法迭代的次数并找到更好地 k 值和中心点。聚类的流程如图 3-3 所示。

表 3-12 阈值选取

$T1$	$T2$
2000	1500
2500	2000
3000	2500

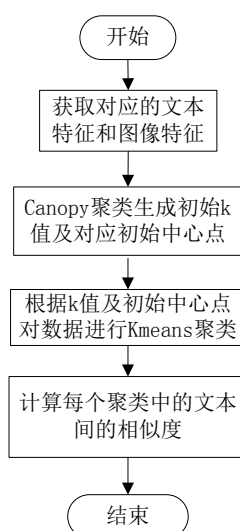


图 3-3 常识信息聚类流程图

3.4.2 用户兴趣模型的构建与更新

为了实现对常识信息的个性化推荐，需要有一种方式计算并表示用户最近关注的信息的特征。表示的方法有以下几种。

(1) 关键词表示法：用多个关键词表示。通常来说，关键词可以由用户提供，或者根据用户的浏览历史记录获取。这种方法的好处是简单且易于理解，但是，应该注意到，关键词表示法表示的粒度较粗，因为关键词的地位是平等的，比如有一个用户关注“养生”和“两性”，而二者之间又更关注“养生”，此时，作为关键字“养生”和“两性”都表示用户的兴趣，但是，反映不出这种差别；

(2) 空间向量表示法：它是关键词表示法的一种改进，在关键词的基础上增加了关键词权重，以显示用户关注信息的差别，同样地，比较（“养生”，3）和（“两性”，1），容易知道，用户更加关注“养生”，表示用户兴趣的粒度更细；

(3) 主题表示法：表示用户更加关注的话题，在三者之间粒度最粗。

下面阐述下模型构建和模型更新过程。

(1) 模型构建

为了更好地表达用户的兴趣，我们选取空间向量表示法来表示。其构建过程如图 3-4 所示。

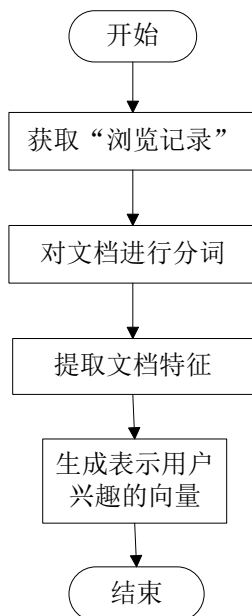


图 3-4 用户兴趣模型生成

其中“浏览记录”我们选取最近浏览的 20 篇文章，中文分词还是选取前文所采用的 ik，特征提取方式采用 TF-IDF 算法，并最终选取权值最高的前 20 个关键词代表用户兴趣。

(2) 模型更新

随着时间的推移以及用户关注话题的改变，需要不断地更新兴趣模型。为了简化设计和实现，我们将用户登录操作和时间结合起来考虑。具体流程如图 3-5 所示。

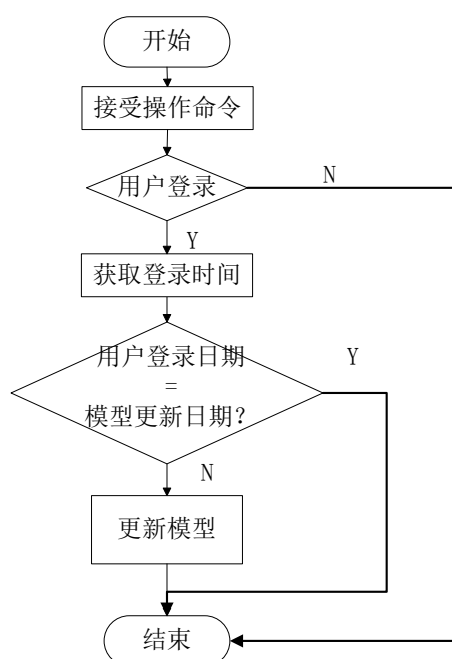


图 3-5 模型更新流程

简要来讲，只有当用户登录时才应该更新该用户的兴趣模型，且一天之内只能更新一次兴趣模型，这样的策略的好处是能减轻计算时间开销。

3.4.3 常识信息推荐

信息推荐从两个方面考虑：一是推荐内容相似的其它文档；二是推荐和用户最近浏览过的文档相似的其它文档。对于第一部分，只需离线计算好跟浏览的文档最相似的一些文档，并推荐给用户。而对于第二部分，因为用户模型是在用户登录后计算得到，所以不能离线计算好要推荐的文档，只能在线计算并推荐。我们采取的策略是：计算用户兴趣模型跟所有信息簇中心点的相似度，判断用户更倾向于喜欢哪个信息簇信息，然后依次遍历该信息簇中的文档，计算它跟用户兴趣模型的相似度，满足设定

的阈值后，就将作为推荐的一项，直到推荐的项数达到某个值 m 时，停止计算，返回推荐结果，这样能有效提高系统的实时性。

不管是文档和文档间的相似度还是文档与用户兴趣模型间的相似度的计算方法都是采用向量夹角余弦的方式，计算公式已经在前文有所阐述。图 3-6 是计算的示意图。将浏览的文档、要进行比较的文档和用户兴趣模型表示为向量，如公式 3-8~3-10 所示。

$$D1 = \{\text{权值 } 1, \text{权值 } 2, \dots, \text{权值 } n\} \quad (3-8)$$

$$D2 = \{\text{权值 } 1, \text{权值 } 2, \dots, \text{权值 } n\} \quad (3-9)$$

$$I = \{\text{权值 } 1, \text{权值 } 2, \dots, \text{权值 } n\} \quad (3-10)$$

然后根据公式(3-2)计算夹角 α 、 β 的余弦值，并作为相似度的度量。

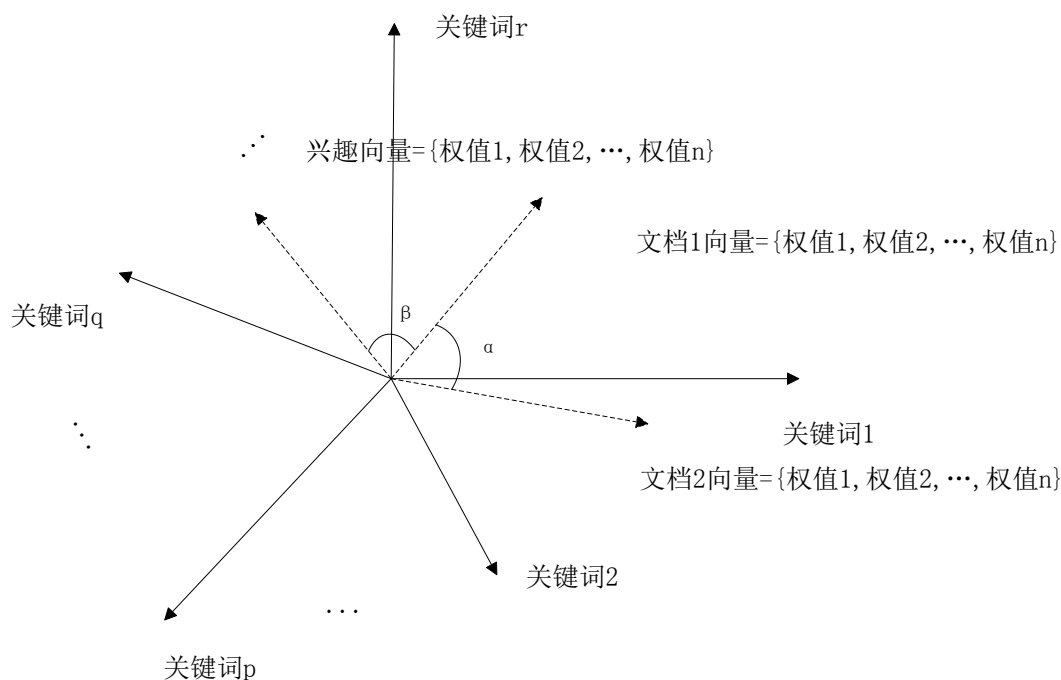


图 3-6 相似度计算的向量示意图

计算好信息簇内部文档间的相似度后，可将相似度存入到数据库中，以加快响应速度。图 3-7、图 3-8 分别是两种推荐策略的流程图。图中都涉及到过滤、排序步骤，目的是根据用户的浏览记录来过滤掉已浏览过的文档。

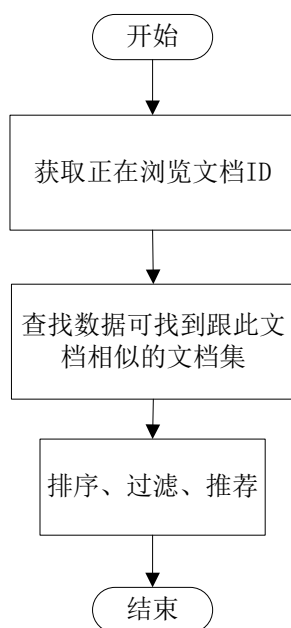


图 3-7 基于内容相似度推荐

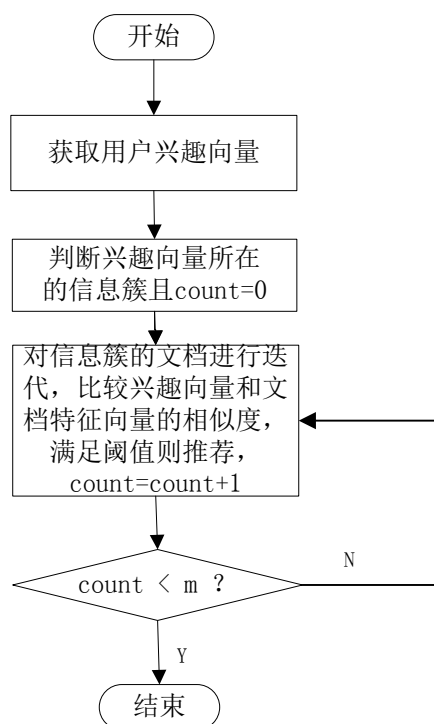


图 3-8 用户兴趣模型推荐示意图

3.5 处方推荐模块实现

处方推荐旨在给医疗工作人员、医疗学习人员、药店工作人员等一些合理的处方建议，处方推荐需要相关专业人士甄别之后，患者才能够采纳推荐结果。运用它能够减少误诊的情况，对于常见疾病，有比较好的效果。

对一些常见疾病来说，首先要在医院看门诊，医生根据患者的症状做出初步判断，为确定病情可能还需做一些检查/检验，然后根据检查/检验结果再做出诊断，开具处方。为了做到处方的推荐，有两种实现方案：第一、运用人工智能领域的知识做判断；第二、根据过往的检查、检验、诊断、处方等记录找到跟现在检查/检验结果最匹配的几个方案。在这里我们选择方案二，因为实施起来更加容易，而通过人工智能领域知识创造这个模型并代替医生的作用，还没有具备实用的研究成果。

整个实验过程中，用到的数据源有病人信息记录、诊断记录、门诊诊断记录、检查主记录、检查项目记录、检查报告、检验主记录、检验项目、检验结果、药品处方主记录、药品处方明细记录。下面是对这些记录的简要描述。

- (1) 病人信息记录描述所有在院注册的病人的基本信息；
- (2) 诊断记录描述医生为病人所下的各种诊断；
- (3) 门诊诊断记录记录的是诊断记录的门诊诊断部分；
- (4) 检查主记录记录病人各种检查的发生及执行情况，以及根据检查做出的临床诊断；
- (5) 检查项目记录描述检查的具体项目，是检查主记录的明细记录；
- (6) 检查报告描述检查报告内容；
- (7) 检验主记录记录所有检验申请；
- (8) 检验项目描述所有医院具备的检验项目；
- (9) 检验结果用于记录所有病人的检验结果；
- (10) 药品处方主记录和药品处方明细记录二者构成处方记录，每张处方可包含多条药品记录，根据经验，每张处方平均为 2~5 种药品。

处方推荐的流程如图 3-9 所示。

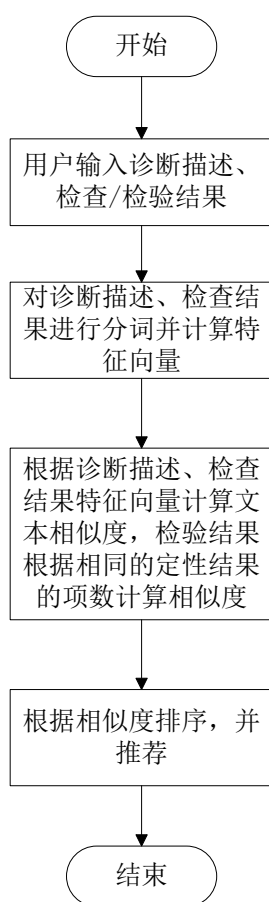


图 3-9 处方推荐流程图

整个流程中首先需要获取患者的诊断的症状描述，并取得检查结果和检验结果。当然，检查结果和检验结果可能都没有，仅凭症状即可开具处方，或者检查结果和检验结果只有一个；然后对检验结果进行定性判断，比如指标是高了还是低了，还是正常，为此需要统计数据库中各个指标的正常范围；最后，根据诊断症状描述，检查结果，检验结果跟数据库中的案例进行比对，并找到综合相似度最高的几项并推荐给用户，并且返回推荐结果时根据用户对药物的过敏情况进行过滤。

判断检验结果的相似性是通过统计定性的检查结果间相同的项数来决定的，比如：检验结果的定性分析为表 3-13 所示。

表 3-13 检查结果定性表示

总胆红素	直接胆红素	总蛋白	白蛋白	丙氨酸氨基转移酶	天冬氨酸氨基转移酶	碱性磷酸酶
H	H	H	L	L	L	L

数据库中相应的记录为表 3-14。

表 3-14 数据库中的检查结果记录

总胆红素	直接胆红素	总蛋白	白蛋白	丙氨酸氨基转移酶	天冬氨酸氨基转移酶	碱性磷酸酶
H	H	H	L	L	L	L
H	H	H	L	L	H	H
H	H	H	L	L	L	H

那么我们认为第一项跟检验结果的定性分析相似度高,因为它们有 7 项是一致的,而另两个只有 5 项和 6 项。

3.6 本章小结

本章主要讲述了实现部分。首先阐明各类信息的存储、索引和搜索的实现,为药品、医生、常识、处方设计了不同的数据结构,并解释了各字段的含义;然后分别讲述了药品推荐的实现、医生推荐的实现、常识推荐的实现和处方推荐的实现。

其中药品推荐主要基于适应症的相似度来推荐,辅助因素是药品的总使用量和近期使用量,医生推荐考虑医生的流行程度,而流行程度由治愈病人的人数来客观的反映,这样能够克服仅有医生简介判断造成主观性太强的问题,并且根据患者对医生的评价进行协同过滤的推荐,患者对医生的评价也是根据治疗结果客观得到的,对于常识而言,第一是根据内容相似度进行推荐,其中包含对信息的聚类处理,从而减少计算开销,第二根据用户浏览记录得到用户的兴趣点,从而主动推荐信息给用户,对处方来说,用户需要输入的信息是诊断症状描述、检查结果、检验结果,系统根据症状描述得到一些候选方案,再比较用户输入的检查结果、检验结果和数据库中记录的检查结果和检测结果的相似度,并从中选出相似度高的方案,最后根据用户对药物的过敏情况进行过滤,得到最终的推荐结果。

4 评估与测试

本文我们在开源的全文搜索引擎 Elasticsearch 上进行医疗信息推荐系统的开发，根据系统设计要求对 Elasticsearch 进行一些修改，增加推荐模块。本章从三个方面进行阐述：测试环境、功能测试和性能测试。考察了推荐对搜索的响应时间的影响，分析了聚类对计算开销减少的程度，并分析了推荐结果的有效性。

4.1 测试环境

本文测试平台有 2 台主机组成，系统配置如表 4-1 所示。两台主机构成 Elasticsearch 的服务器集群。

表 4-1 Hadoop 平台节点的硬件配置

处理器	Pentium(R) Dual-Core CPU E6700 @ 3.20GHz	AMD A4-3300M @3.8GHz
内存	2GB	4GB
硬盘	500GB	500GB
网卡	百兆网卡	百兆网卡
操作系统	Windows 7	Windows 7

本文实现的各数据处理模块的调度客户端与 Web 展示端均在主控制节点进行启动，原型系统搭建中使用的相关开源软件及其对应的版本号如表 4-2 所示，Mahout 算法需要 Hadoop^[45-47]平台的支持。

表 4-2 平台主要相关开源软件及版本号

软件	Hadoop	Jsoup	Mahout	WebCollector	Tika	MySQL	Elasticsearch	Oracle
版本号	0.20.2	1.8.1	0.9	1.3.1	0.8	5.1.6	1.3.4	11gR2

基于 Elasticsearch 搜索引擎的医疗信息推荐系统中 Elasticsearch 的主要配置参数如表 4-3 所示。

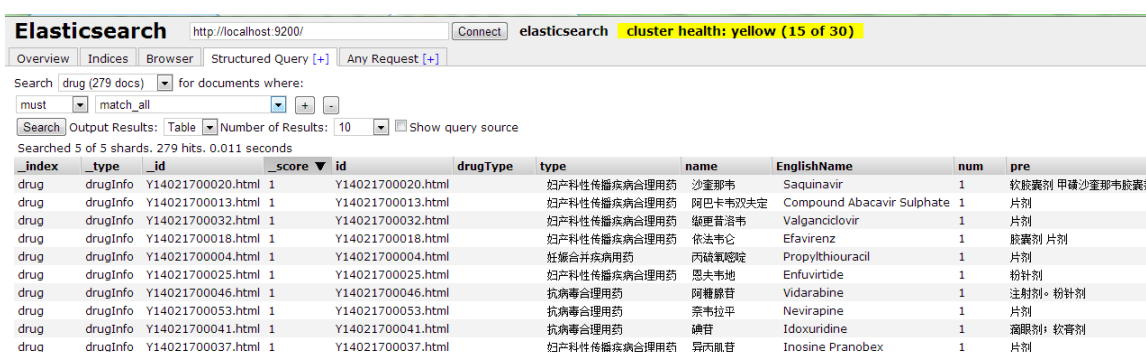
表 4-3 Elasticsearch 参数配置

配置项	节点 1	节点 2
cluster.name	Massive Healthcare Data	Massive Healthcare Data
node.name	es_node0	es_node1
node.data	true	true
index.number_of_shards	5	5
index.number_of_replicas	2	2
network.host	192.168.0.2	192.168.0.3
http.port	9200	9200
index.analysis.analyzer.ik.type	org.elasticsearch.ind ex.analysis.IkAnalyz erProvider	org.elasticsearch.inde x.analysis.IkAnalyzer Provider

4.2 功能测试

4.2.1 搜索测试

系统实现对药品、医生、医疗常识三种信息的搜索，图 4-1、图 4-2 分别展示了药品和常识信息的搜索过程，医生信息涉及到隐私，这里不公布。



_index	_type	_id	_score	id	drugType	type	name	EnglishName	num	pre
drug	drugInfo	Y14021700020.html	1	Y14021700020.html	妇产科性传播疾病合理用药	沙奎那韦	Saquinavir		1	胶囊剂 甲磺沙奎那韦胶囊剂
drug	drugInfo	Y14021700013.html	1	Y14021700013.html	妇产科性传播疾病合理用药	阿巴卡韦双夫定	Compound Abacavir Sulphate		1	片剂
drug	drugInfo	Y14021700032.html	1	Y14021700032.html	妇产科性传播疾病合理用药	缬更昔洛韦	Valganciclovir		1	片剂
drug	drugInfo	Y14021700018.html	1	Y14021700018.html	妇产科性传播疾病合理用药	依法韦仑	Efavirenz		1	胶囊剂 片剂
drug	drugInfo	Y14021700004.html	1	Y14021700004.html	妊娠合并疾病用药	丙硫氧嘧啶	Propylthiouracil		1	片剂
drug	drugInfo	Y14021700025.html	1	Y14021700025.html	妇产科性传播疾病合理用药	恩夫韦地	Enfuvirtide		1	粉针剂
drug	drugInfo	Y14021700046.html	1	Y14021700046.html	抗病毒合理用药	阿德福韦酯	Vidarabine		1	注射剂、粉针剂
drug	drugInfo	Y14021700053.html	1	Y14021700053.html	抗病毒合理用药	奈韦拉平	Nevirapine		1	片剂
drug	drugInfo	Y14021700041.html	1	Y14021700041.html	抗病毒合理用药	碘苷	Idoxuridine		1	滴眼剂、软膏剂
drug	drugInfo	Y14021700037.html	1	Y14021700037.html	妇产科性传播疾病合理用药	异丙肌苷	Inosine Pranobex		1	片剂

图 4-1 药品信息搜索

时的响应时间跟文档命中数的关系，实线部分为引入推荐功能后的响应时间跟文档命中数的关系。从图表上看，随着命中文档数的增多，响应时间呈线性增长，且无推荐情形下的响应时间总体上比有推荐情形下的响应时间略微少一些。这个结果符合实验的预期，因为有推荐的情况下系统返回的数据量更大，所以其响应时间总体上比无推荐的情况下更大，但结果不会相差太多，因为推荐结果是离线计算并更新搜索引擎中的文档得到，所以，多出来的部分是推荐结果所占字节的传输时间。两个图对比可知，医生检索相较于药品检索在有推荐和无推荐两种情况下的差值更大，即方差更大。这是因为增加的推荐结果字节在检索结果的所有字节中占比更大。

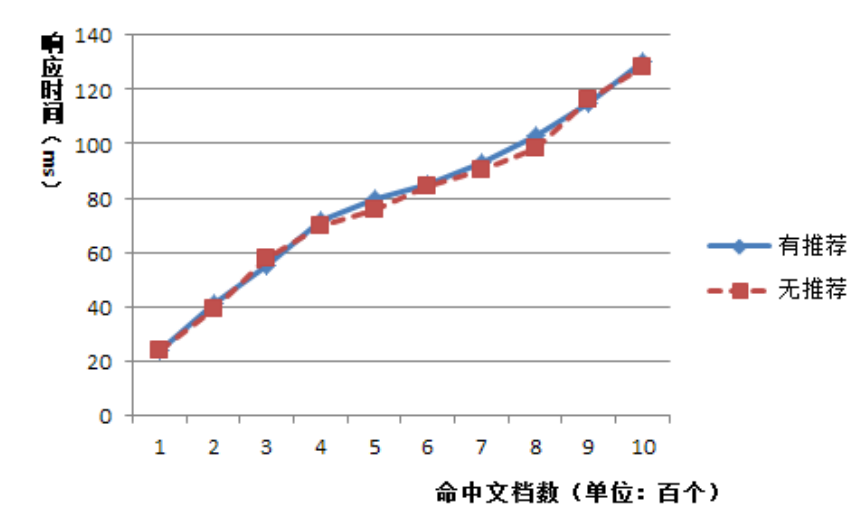


图 4-5 药品检索响应时间随命中文档数变化曲线

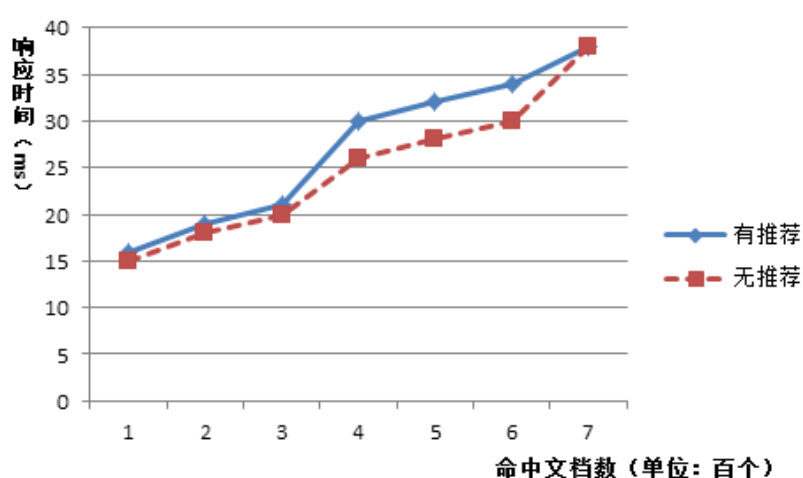


图 4-6 医生检索响应时间随命中文档数变化曲线

图 4-7 所示为常识信息检索响应时间随文档命中数的变化曲线，数据集中有 10 万个文档，且设置与用户兴趣向量间的相似度达到 0.5 即作为推荐结果的一项，得到 5 项即停止推荐。对于常识信息而言，采用文档相似性推荐和个性化推荐两种策略。个性化推荐跟命中文档数的多少无关，只与用户兴趣向量和数据簇中文档的相似性计算有关，故命中文档数不会影响个性化推荐的响应时间，因为常识信息的文本比较大，推荐结果占有的字节比重也不多，从而响应时间比较一致。

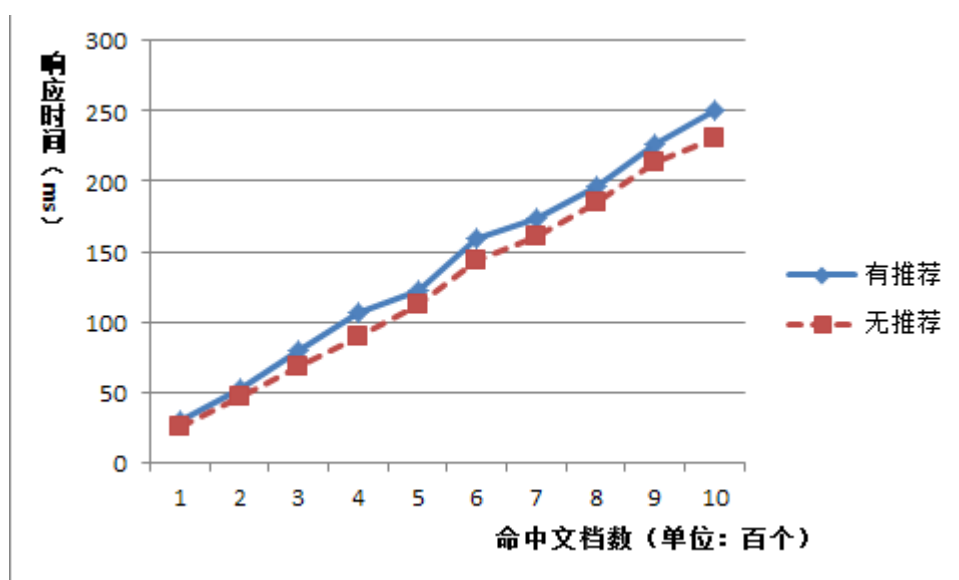


图 4-7 常识检索响应时间随命中文档数变化曲线

图 4-8 为处方推荐响应时间随命中文档数的变化曲线，处方推荐需要在线计算用户输入的诊断信息、检查信息和检验信息与记录的相似度，其中诊断信息和检查信息都是文本描述，使用信息相似度的一般计算方法，而检验结果相似度需根据检验结果的定性表示的相同项数决定。随着命中文档数的增多，响应时间增大，且越来越偏离横坐标轴，因为随着命中文档数的增多，在线计算的时间开销越来越大。

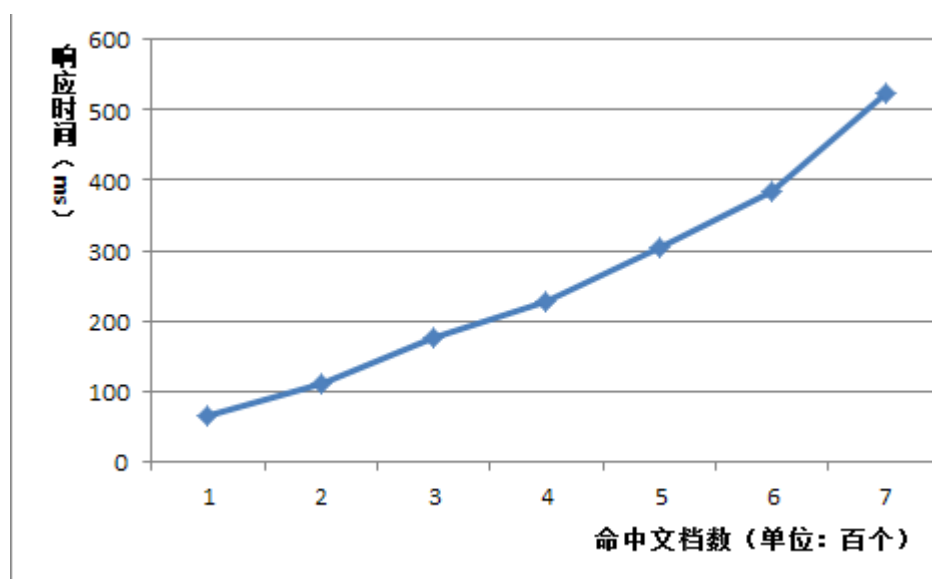


图 4-8 处方检索响应时间随命中文档数变化曲线

图 4-9 为响应时间(相较于不进行聚类处理时的比值)跟聚类簇数的关系,随着簇数的增多,响应时间呈对数级下降,因为相似度计算量也是随着簇数的增多呈指数级下降,相似度计算只需计算跟某个簇的文档的相似度,不过当数据簇数变得很多时,极端情况下每个文档一个簇,此时的响应时间也将非常大,因为在进行相似性计算时需要预先判断所在的簇。

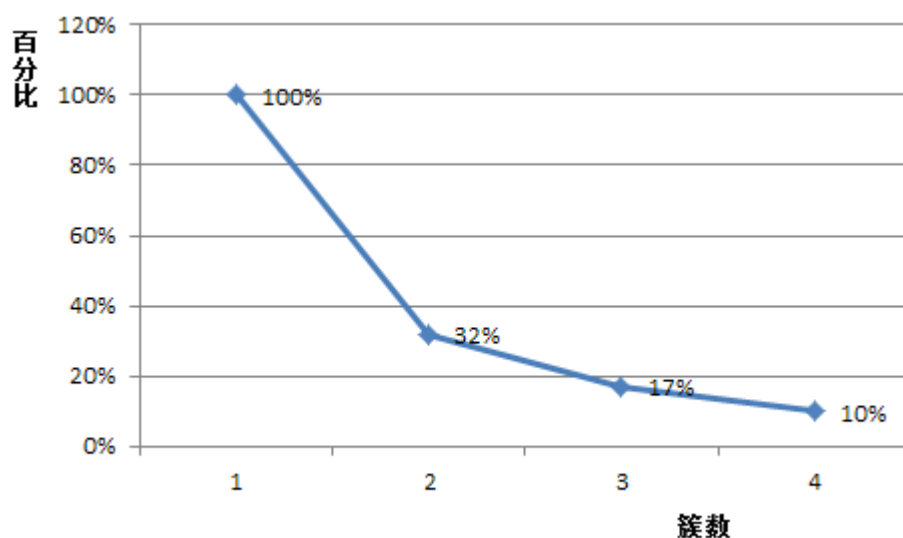


图 4-9 响应时间与簇数关系曲线

4.3.1 推荐性能

推荐的有效性即用户浏览推荐文档信息的可能性,而用户浏览推荐文档信息的可

能性需要在线运行系统来获取。但是，我们可以通过相似度、查准率和查全率衡量标准来表征推荐的有效性。表 4-4 分别表示 5 个推荐结果和 10 个推荐结果的相似度测试，对所有的推荐结果的推荐值取平均。处方信息的推荐相似度最高，因为其相似度的度量标准是诊断结果、检查结果和检验结果的定性表示，相关的记录中术语比较固定，比如“病毒性感冒”，所以相似度更高。表 4-5 分别表示 5 个推荐结果和 10 个推荐结果的内容相关性测试，查准率测试是判断推荐的结果是否跟检索的信息相关，比如药品信息来说，就是判断推荐的药品跟检索的药品适应症是否一致，对医生信息来说，就是判断推荐的医生跟检索的医生之间是否从事领域，擅长领域是否一致，而对常识信息而言，就是判断推荐的信息跟检索的信息在主题和内容上是否一致，对处方而言，就是判断推荐的处方能否对该疾病有效。查全率测试是判断推荐结果占有信息量的比例。查准率和查全率的计算如公式 4-1 和公式 4-2 所示。测试过程中对四类信息进行 1000 个检索，统计各自的推荐结果跟检索结果的相关性来表征查准率，并统计各自的推荐结果信息的覆盖程度表征查全率。结果如表 4-5 所示，返回推荐结果设置为 10。

表 4-4 推荐结果的相似度

	药品信息	医生信息	常识信息	处方信息
相似度(5)	0.78	0.63	0.66	0.83
相似度(10)	0.71	0.59	0.62	0.74

$$\text{查准率} = \frac{\text{返回相关信息量}}{\text{返回总信息量}} \times 100\% \quad (4-1)$$

$$\text{查全率} = \frac{\text{返回相关信息量}}{\text{系统总信息量}} \times 100\% \quad (4-2)$$

表 4-5 10 个推荐结果的查准率和查全率

	药品信息	医生信息	常识信息	处方信息
查准率 (%)	82.1	92.4	67.8	52.4
查全率 (%)	84.5	94.8	72.3	56.7

药品“适应症”是比较简短的文本描述，而常识信息是比较长的文本描述，对前

者进行预处理（分词、计算特征向量），能够得到更加能够反映文本内容的特征向量，所以查准率更高，对后者来说，因为文本长度的增加，很容易产生跟文本内容偏离的关键词，所以查准率相对较低，而医生信息推荐限制了只能推荐该领域的医生，所以查准率和查全率都更高，处方推荐考虑的因素比较多，有诊断结果、检查结果和检验结果，而且，只考虑这三个因素也并不能完全开出理想的处方，处方的获取应该要考虑更多的因素，而且对各个指标的定量结果也需要考虑，而本文只是考虑定性结果，综上所述，推荐性能最差。

4.4 本章小结

本章首先讲述测试环境以及工具软件；然后分别给出四类信息搜索和推荐的功能示意图；最后从响应时间和推荐性能方面评估本系统的性能。响应时间的测试表明推荐的引入不会显著影响响应时间，且信息聚类的运用能够有效降低响应时间，当聚类簇数达到 4 个以上时，相似度的计算开销降低 80% 以上，推荐性能测试采用相似度的度量标准。随着文档的增多，推荐效果将变得越来越好，但是时间开销也将随之增大。

5 总结与展望

5.1 全文总结

随着医疗信息的增长，造成了严重的信息过载问题，用户很难得到较为专业的指导、比较客观的信息以及自己关注的信息。为了促进医疗卫生事业的发展，本文对药品、医生、常识、处方四种数据进行研究，建立索引、提供搜索和推荐。所做的工作包括如下几个方面：

(1) 对信息过载问题的解决方案、信息推荐的研究现状、医疗信息推荐的研究现状进行分析和阐述，并分析它们的设计思想和相关技术。

(2) 对四类信息分别设计对应的数据结构，并设计对应的 mapping 定义，从而构建索引，方便信息的检索。

(3) 对药品信息来说，为保证推荐结果的客观性，提出结合药品“适应症”描述文本相似性、药品总体使用量以及药品使用热度三个方面，综合影响推荐结果，为减少计算开销，需要对药品进行分类，分类的依据是药品类别，比如抗病毒药物，抗癌药物等。

(4) 对医生信息而言，首先是收集某医院的所有医生信息以及他们的看病记录；然后处理成表示医生看病能力的数据表以及患者对医生的评价表，处理的依据是看病的治疗结果；最后采用两种策略进行推荐，第一是根据医生能力进行推荐，第二是根据患者评价的相似性对医生进行推荐，推荐结果还需要进行过滤，过滤的原则是用户关注的医生专业领域，比如儿科、妇产科等。

(5) 常识信息是利用网络爬虫从 web 上获取，具有 web 信息的特点，即数据量大、更新快等特点。为此，我们考虑每个用户在某个时期关注的信息具有局部性，个性化推荐处理用户近期浏览的信息，得到表示用户兴趣的向量。另一个方面，内容相似的信息往往会得到用户的关注，所以根据常识信息内容的相似性进行推荐也是研究的一部分，为了降低时间开销，我们需要对数据进行聚类处理。

(6) 对处方来说，我们利用到处方记录、检查记录、检验记录、诊断记录四个数据

源, 根据用户输入的检查结果、检验结果和诊断结果找到适合的处方。

(7) 实验结果表明: 用户对推荐结果有较高的满意度, 分类和聚类处理能够有效的减少时间开销。

5.2 展望

(1) 在以后的工作中考虑将数据迁移到分布式系统中; 同时数据库的使用利用分布式数据库 MongoDB, 从而增强系统的可扩展性。

(2) 为更好的减少计算开销, 需要进一步改进聚类算法, KMeans+Capony 需要选择好两个距离 T1 和 T2, 后续工作中需要对它们的各种选择进行深入研究, 从而获取更好的聚类结果, 提高推荐的精度。

(3) 本次实验中医学专有名词的收录并不是非常完整, 所以在系统运用过程中也要不断的收录, 这样才更加适合医疗信息这类信息的分词。

致 谢

3 年的研究生生活无论是在做人还是做事方面都给我带来了莫大的进步，通过这些年我更加懂得站在别人的角度去思考问题，我更加觉得多种观念、看法的存在有它的必然性，所以当自己跟别人的意见不一致时我所想到的是提高自己的认知，去发现问题的本质；在做事方面我变得更加严谨，也更加细致，性格更加沉稳，浮躁的气息变得越来越不明显。

在此期间我认识了一些对我的人生有极大帮助的人，首先是我的导师李春花老师，她是一个和蔼可亲的人，不论是学习上还是生活上都给了我极大的关注，在我觉得浮躁之时总是能够给我一些好的建议。还要感谢周可老师、王桦老师、郑胜老师、张胜老师和邹复好老师，更多的来说我跟你们探讨的是学习方面的事情，但是你们严谨的作风还是给了我极大的印象，在这个方面上，你们给了我很多有建设性的建议。让我在学习上不至于走上迷途。还有，实验室的小伙伴们，边泽明等，跟你们一起的三年让我在技术上和生活上都受益匪浅，我们一起学习，一起打球，一起进步，这些经历我都倍感珍惜，实验室学弟在态度上给了我极深的印象，帮我查找各种数据集，你们是值得一起工作的人。寝室的室友，边泽明，李哲，陈启蒙，跟你们一起生活在一个寝室是我的荣幸，你们的包容让我能更好的融入这个寝室，谢谢你们，感谢从事医学领域的同学张洁、刘海燕等，你们耐心的讲解对我的设计和测试工作帮助极大。最后，感谢默默支持我的父母，你们对我的支持是我最大的动力，你们的支持让我觉得生活更加充满意义，谢谢有你们一路相伴。

参考文献

- [1] 刘建国, 周涛, 汪秉宏. 个性化推荐系统的研究进展. 自然科学进展. 2009,19(1)
- [2] Adomavicius G, Tuzhilin A, Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE Trans. On Knowledge and Data Engineering, 2005,17(6):734~749
- [3] Gospodnetic O, Hatcher E, McCandless M, Lucene 实战, (第二版), 牛长流等译, 北京: 人民邮电出版社. 2011.228~255
- [4] Anil R, Dunning T, Friedman E 等著, Mahout in action, (第一版), 王斌等译, 北京: 人民邮电出版社, 2014
- [5] Resnick P, Varian HR. Recommender systems. Communications of the ACM, 1997,40(3):56~58
- [6] Mooney RJ, Bennett PN, Roy L. Book recommending using text categorization with extracted information. In: Proc. of the AAAI'98/ICML'98 Workshop on Learning for Text Categorization. Madison: AAAI Press, 1998. 49~54
- [7] Pazzani M, Billsus D. Learning and revising user profiles: The identification of interesting Web sites. Machine Learning, 1997, 27(3):313~331
- [8] Somlo G, Howe A. Adaptive lightweight text filtering. In: Proc. of the 4th Int'l Symp. on Intelligent Data Analysis. Berlin, Heidelberg: Springer-Verlag, 2001. 319~329
- [9] Zhang Y, Callan J, Minka T. Novelty and redundancy detection in adaptive filtering. In: Proc. of the 25th Annual Int'l ACM SIGIR Conf. New York: ACM Press, 2002. 81~88
- [10] Robertson S. Threshold setting and performance optimization in adaptive filtering. Information Retrieval, 2002,5(2-3):239~256
- [11] Zhang Y, Callan J. Maximum likelihood estimation for filtering thresholds. In: Proc. of the 24th Annual Int'l ACM SIGIR Conf. New York: ACM Press, 2001. 294~302
- [12] Rich E. User modeling via stereotypes. Cognitive Science, 1979,3(4); 329~354

- [13] Goldberg D, Nichols D, Oki BM, Terry D. Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 1992,35(12):61~70
- [14] Konstan JA, Miller BN, Maltz D, Herlocker JL, Gordon LR, Riedl J. GroupLens: Applying collaborative filtering to usenet news. *Communications of the ACM*, 1997.40(3):77~87
- [15] Shardanand U, Maes P. Social information filtering: Algorithms for automating “Word of Mouth”. In: *Proc. of the Conf. on Human Factors in Computing Systems*. New York: ACM Press, 1995. 210~217
- [16] Terveen L, Hill W, Amento B, McDonald D, Creter J. PHOAKS: A system for sharing recommendations. *Communications of the ACM*, 1997,40(3):59~62
- [17] Breese JS, Heckerman D, Kadie C. Empirical analysis of predictive algorithms for collaborative filtering. Technical Report, MSR-TR-98-12, Redmond: Microsoft Research, 1998
- [18] Sarwar B, Karypis G, Konstan J, Riedl J. Item-Based collaborative filtering recommendation algorithms. In: *Proc. of the 10th Int’l WWW Conf.* New York: ACM Press, 2001. 285~295
- [19] Billsus D, Pazzani M. Learning collaborative information filters. In: *Proc. of Int’l Conf. on Machine Learning*. San Francisco: Morgan Kaufmann Publishers Inc., 1998. 46~54
- [20] Getoor L, Sahami M. Using probabilistic relational models for collaborative filtering. In: *Proc. of the Workshop Web Usage Analysis and User Profiling*. 1999
- [21] Hofmann T. Collaborative filtering via Gaussian probabilistic latent semantic analysis. In: *Proc. of the 26th Int’l ACM SIGIR Conf.* New York: ACM Press, 2003. 259~266
- [22] Marlin B. Modeling user rating profiles for collaborative filtering. In: *Proc. of the 17th Annual Conf. on Neural Information Processing Systems*. Cambridge: MIT Press, 2003. 627~634

- [23] Ungar LH, Foster DP. Clustering methods for collaborative filtering. In: Proc. of the Workshop on Recommendation Systems. Menlo Park: AAAI Press, 1998. 112–125
- [24] Chien YH, George EI. A Bayesian model for collaborative filtering. In: Proc. of the 7th Int'l Workshop on Artificial Intelligence and Statistics. San Francisco: Morgan Kaufmann, 1999
- [25] Pavlov D, Pennock D. A maximum entropy approach to collaborative filtering in dynamic, sparse, high-dimensional domains. In: Proc. of the 16th Annual Conf. on Neural Information Processing Systems. 2002
- [26] 于宝福 个性化医疗信息推荐系统的研究与实现 [硕士学位论文] 浙江 浙江大学, 2012
- [27] Jianying Mai, Yongjian Fan, Yanguang Shen, “A Neural Networks-based Clustering Collaborative Filtering Algorithm in E-commerce Recommendation System”, International Conference on Web Information System and Mining[J], DOI 10.1109/WISM.2009.129
- [28] 方惠敏, 基于 BP 神经网络的个性化网站界面用户建模[硕士学位论文], 河南, 河南大学, 2008
- [29] Leandro A. Silva, Emilio Del-Moral-Hernandez, “A SOM combined with KNN for Classification Task”, Proceedings of International Joint Conference on Neural Networks[J], San Jose, California, USA, July 31-August 5, 2011
- [30] Elasticsearch: <http://www.elastic.co/guide/>. Accessed October 11, 2014
- [31] Radu Gheorghe 等著, Elasticsearch IN ACTION, (第二版), Manning Publications, 2013
- [32] JSON:<http://www.json.org/>. Accessed October 13, 2014
- [33] 罗刚著. 自己动手写网络爬虫. (第一版).北京: 清华大学出版社, 2010
- [34] WebCollector:<https://github.com/CrawlScript/WebCollector>. Accessed November 12, 2014
- [35] 佩里等著. Oracle 基础教程. (第三版). 钟鸣等译.北京: 人民邮电出版社, 2008

- [36] Karen Morto 等著, Oracle SQL 高级编程, (第一版), 朱浩波译, 北京: 人民邮电出版社, 2011
- [37] 帕奇维著, 深入理解 MySQL 核心技术, (第一版), 李芳等译, 北京: 中国电力出版社, 2009(5) Goldberg K, Roeder T, Gupta D, Perkins C. Eigentaste: A constant time collaborative filtering algorithm. Information Retrieval, 2001,4(2):133~151
- [38] 杜布瓦等著, MySQL 权威指南, (第二版), 林琪等译, 北京: 中国电力出版社, 2004
- [39] Jsoup:<http://jsoup.org/>. Accessed October 17, 2014
- [40] IKAnalyzer:<http://code.google.com/p/ik-analyzer/>. Accessed November 23, 2014
- [41] Delgado J, Ishii N. Memory-Based weighted-majority prediction for recommender systems. In: Proc. of the ACM SIGIR'99 Workshop Recommender Systems: Algorithms and Evaluation. New York: ACM Press, 1999
- [42] Larsen B, Aone C. Fast and effective text mining using linear-time document clustering. in: Proceedings of the 5th ACM SIGKDD international conference on Knowledge discovery and data mining. 1999. 16~22
- [43] 江小平, 李成华, 向文等. k-means 聚类算法的 MapReduce 并行化实现. 华中科技大学学报(自然科学版), 2011, 39: 120~124
- [44] 韩家炜, Kamber 著, 数据挖掘: 概念与技术, (第二版), 范明等译, 北京: 机械工业出版社, 2001,232~233
- [45] 姜文. 基于 Hadoop 平台的数据分析和应用: [硕士学位论文]. 北京: 北京邮电大学, 2011
- [46] Tom White, Hadoop 权威指南, (第二版), 曾大聃等译, 北京: 清华大学出版社, 2011
- [47] Chu C, Kim S K, Lin Y A. Map-reduce for machine learning on multicore Advance in neural information processing systems, 2007, 19: 281~291