密级：_____

浙江大学

硕 士 学 位 论 文

论文题目 科研文献开放获取系统中的推荐算法研究与应用

作者姓名 钱宇

指导教师 陈珂

学科(专业) 计算机科学与技术

所在学院 计算机科学与技术学院

提交日期 2017 年 1 月

A Dissertation Submitted to Zhejiang University for the Degree of Master of Engineering

TITLE: Research and Implementation of Recommendation Algorithm in Open Access System

Author: Qian Yu

Supervisor: Chen Ke

Subject: Computer Science and Technology

College: Computer Science and Technology

Submitted Date: 2017-01

摘要

随着开放获取在科研领域的发展，科研工作者们越来越习惯通过网络来进行文献获取和学术交流。如何在科研文献数量呈现爆炸式增长的情况下，为用户提供方便快捷的论文获取方式，成为了新的科研环境下开放获取系统面临的巨大挑战。论文推荐和专家推荐功能的引入能够为用户的论文发现和专家发现提供新的途径。传统的推荐方法多是以基于内容的算法为主，而随着开放获取系统的发展，加入对收集的用户行为数据的分析必将能增强系统的推荐效果。在这种情况下，传统单一的推荐算法将无法满足越来越多样化的用户需求，需要针对科研文献开放获取系统的独特应用场景进行创新和改进。

本文中我们在对传统推荐算法进行了充分研究的基础上，提出了一种将基于协同过滤的推荐算法与基于内容的推荐算法结合的混合论文推荐算法。算法首先使用词向量进行了论文内容的比较，然后使用最近邻模型进行用户行为的比较，最后将论文内容以及用户行为对推荐结果的影响进行了综合考虑，以得到最终的推荐结果。在论文推荐的基础上，我们又研究实现了专家推荐算法，通过对专家间论文内容相似度的比较，来进行专家相似度的计算以及相关专家的推荐。

为了验证本文算法的有效性，我们在公开数据集上将该混合算法与多种传统推荐算法进行了对比，结果证明混合推荐算法在多种情况下都能有稳定良好的表现，且与对比算法相比推荐效果有所提升。另外，我们在基金委开放获取系统中设计并实现了论文推荐和专家推荐的功能，其实现所采用的方法是基于本文算法的部分内容。同时我们还实现了系统中论文成果录入的功能，经测试系统运行效果良好。

关键词： 开放获取系统，论文推荐，专家推荐，混合推荐算法

Abstract

As the developing of open access in scientific and research area, researchers are getting more used to retrieving literature and communicating online. With explosive growth in the number of papers, how to provide researchers with convenient access to papers has become a great challenge for open access systems. The introduction of paper recommendation and expert recommendation provides new approach for researchers to access information. However, traditional recommendation methods mostly only concentrate on text content. In nowadays open access systems, as users' behavioral data are also abundant, taking this into consideration can surely improve recommendation system's performance.

In this paper, based on sufficient research of traditional recommendation algorithms, we propose a new hybrid method which combines collaborative filtering algorithm and content-based algorithm. We first use word embedding to compare similarity between papers' content, then use the nearest neighbor model to compare users' behavioral data, and finally recommend concerning papers based on overall consideration of paper content and users' behavior. Based on the results of previous step, we further study and implement an expert recommendation algorithm by comparing the similarity between experts' paper collection.

To verify the effectiveness of our algorithms, we compare our solution with many traditional recommendation algorithms on a public available dataset, and the result shows that our hybrid recommendation algorithm performs better than counterparts. We also implemented the function modules in our open access system using the content based part algorithm as users' behavior data not available yet. Finally, we add function modules in the system to provide convenience for researchers to record their papers into our system by either online or offline method.

Keywords： Open Access System, Paper Recommendation, Expert Recommendation, Hybrid Recommendation Algorithm

目录

图目录

表目录

第1章 绪论

1.1 课题背景

随着科学技术的飞速发展，新的技术和理论层出不穷，越来越多的学者投入到了科研事业中，近年来科研文献的数量呈现出爆炸式增长的趋势，专家学者们获取论文期刊的需求也日益增强。然而传统的期刊出版社将科技期刊的订阅价格大幅提高，大大超过了图书馆及教育科研机构的订购经费，使得其只能减少期刊的订购数量。电子期刊出现以后，出版社严格限制资源只能在机构的局域网范围内访问，这使得科技知识的传播和公共保存受到了严重的限制，科学技术知识俨然成为了出版社的垄断产品。为了打破这些日益严重的限制，促进科学知识的传播和普及，国际科技界在 2002 年 2 月发布的《布达佩斯开放获取计划》中明确提出了科研文献开放获取的概念[1]。随后又在 2003 年 10 月的《关于自然科学与人文科学知识开放获取的柏林宣言》中将开放获取扩展到了论文、数据、图表等资源，正式将开放获取的概念扩展到了科研领域。柏林宣言致力于推动科技文献的开放获取（Open Access，简称 OA），即用户可通过互联网免费阅读、下载、复制和传播作品。这一倡议得到了全球科技教育界的积极支持，包括中国科学院、中国国家自然科学基金委员会在内的数百家科研机构都签署了该宣言。开放获取的出现突破了传统出版的概念，其发展对于推动世界科技的进步、促进全球科技学者的合作与交流都有非常重要的意义。

开放获取系统的使用已成为全球学术界的潮流，且已经发展的比较成熟，但其在我国的使用情况还处于起步的阶段。截止 2014 年 5 月 11 日，在 DOAJ[2]（Directory of Open Access Journals）收录的 OA 期刊中，中国大陆出版的只有 75 种，香港出版的有 38 种，台湾出版的有 24 种。而在 OpenDOAR[3]（Directory of Open Access Repositories）注册的 OA 仓储中，中国大陆只有 39 个，中国香港有 7 个，

[1] http://www.oaj.cas.cn/aboutoa/index.jhtml
[2] http://www.doaj.org/
[3] http://www.opendoar.org/

中国台湾有 58 个。我国的科技期刊正在积极地适应开放获取这一全新的模式，根据中国科协的报告指出，截止 2013 年 8 月底，中国科协科技期刊中有开放获取期刊 364 种[1]，占中国科协全部期刊的 34.5%。开放获取在中国也逐渐成为了一种趋势，但由于起步较晚，在由传统出版到开放获取这一转型过程中，我国仍然需要向国外学习理念和经验。在研的国家自然科学基金委员会开放存取科研平台项目，旨在为国内的研究学者们提供海量论文数据的 OA 平台，推动开放获取系统在国内的广泛使用。本论文的工作即以该项目为基础，着力于其中的论文推荐模块和专家推荐模块，意图为科研工作者提供除关键字搜索以外的论文及专家发现方法。

推荐系统最早被运用于信息检索领域，经过二十多年的发展，已经产生了非常丰富的研究成果。在工业界，推荐系统被广泛应用于电子商务、社交网络等网络产品中，已经成为了信息爆炸时代人们获取信息的重要方式。而在学术界，对于推荐算法的研究也在不断发展，美国计算机协会自 2007 年起开始举办推荐系统年会（RecSys），成为了推荐系统领域的顶级会议。另外，一些知名 IT 公司也开始举办推荐算法相关比赛，比如国内阿里巴巴公司举办的天猫推荐算法挑战赛，这些比赛为全球的学者们提供了学术交流和研究成果展示的平台，大力促进了推荐算法的研究发展。通常来说，推荐系统有两个主要目的，一方面是激发用户潜在行为，比如购物网站中引导用户买新的商品。另一方面推荐系统也可以用来解决互联网时代的信息过载问题，从大量的项目集合中发掘用户最可能需要的信息，提高用户获取信息的效率。

OA 系统的基本功能是管理海量的科研数据，提供便利的获取科研文献的方法。在传统的 OA 系统中，关键字的搜索是科研专家查找论文的主要途径，然而在海量的数据中寻找感兴趣的论文，单单依靠人为的搜索是耗时耗力、效果有限的。于是出现了将论文推荐系统加入 OA 系统的趋势，实现了根据用户感兴趣的特定论文来推荐与此相关的其他论文，突破了原来单一的依靠关键字来搜索的局限。传统的推荐系统中通常有两种方法，即基于内容的推荐和基于协同过滤的推

[1] 《中国科协科技期刊发展报告(2014)》，2014 年 4 月。

荐。由于在论文推荐系统中的待推荐项目具有非常丰富的文本信息，比如论文的标题、简介、关键字等，所以通常是采用基于内容的推荐方式，即根据两篇论文的相似程度来推荐关联项目。但是随着科研论文数量的不断增加，以及在线论文网站，尤其是开放获取系统中用户个人定制仓库功能的发展，以往基于内容的推荐方式将受到很大限制，已经无法适应科研学者们越来越多样化的需要了。随着开放获取系统的进一步普及，用户的行为数据越来越多，从中将能得到大量反映用户兴趣的信息，如何合理利用这些信息，为每一位用户进行个性化的推荐，成为了论文推荐研究的重点。除论文推荐外，专家推荐也是开放获取系统中的重要组成部分。OA 系统中每个专家都有个人主页，其中包含了专家的个人信息，主要包括基本信息、教育信息等，另外还包括专家的研究领域，但这些信息通常是用户个人编辑的，与专家具体研究内容相关的信息并不丰富。由于系统中的每位学者都有自己的论文成果列表，通过对这些论文信息的分析，我们就能够概括出完整的学者学术特征，有了这些特征就可以对任意两个学者进行相似度的判断，从而进行相似专家的推荐。专家推荐在开放获取系统中能有效的帮助用户发现感兴趣领域的相关专家，从而为学者们的科研工作带来便利。

1.2 本文的工作与贡献

在本文中，我们主要研究开放获取系统中论文推荐和专家推荐问题，并以提升现有算法的推荐效果为目标。具体而言，本文所要解决的问题是，给定目标用户和论文列表，基于用户的历史行为信息，为其推荐可能感兴趣的论文，然后在此基础之上推荐相关专家。

论文推荐方面，文本内容的对比是非常重要的信息，但基于用户数据的相似度信息也是不可或缺的特征。因此为了得到更好的效果，我们在对常用推荐算法进行研究后，提出了一种将基于内容的推荐算法与基于协同过滤的推荐算法结合起来的混合推荐算法。我们首先将对论文与论文内容之间的相似度进行计算，主要依据是论文的标题及摘要信息。然后我们根据用户对论文的收藏关系建立用户-项目矩阵，并根据这个矩阵得到基于协同过滤的偏好关系。最后我们将这两方面

的相似关系相结合，通过综合计算为用户推荐最可能感兴趣的若干论文。在论文推荐的基础上，我们将进一步进行专家推荐，我们把专家的特征由其论文成果的组合来表示，并通过专家与专家论文列表的内容相似度来对专家间的相似度进行判断。与论文推荐不同的是，专家推荐并不是针对普通用户进行的，而是主要用于寻找专家的相关学者。我们在公开数据集 CiteULike 上对论文推荐的混合算法进行了实验，并通过与其他传统推荐算法的比较，对混合算法的有效性进行了验证分析。

在拥有较多用户数据的前提下，混合推荐方式将会带来比较好的结果，但在研的基金委 OA 系统中的用户数据比较缺乏，因此论文推荐和专家推荐将主要以基于内容的推荐为主。本文将通过计算论文相似度的方式来进行论文推荐，由于系统中论文数量比较庞大，这一过程将以离线进行为主，本文将对其实现进行详细介绍。专家推荐功能则是在论文推荐的基础上，通过对两个专家论文列表的内容对比来判断他们的相似程度。另外，在 OA 系统中，用户论文成果的添加也是重要的功能之一，本文也将对这一功能进行设计实现。

概括来说，本文的主要工作内容和贡献如下：

1. 研究分析了传统论文推荐方法的推荐效果与适用场景，重点对协同过滤推荐以及基于词向量的相似度计算方法进行了研究，提出了一种将协同过滤推荐与内容推荐结合的混合论文推荐方法。
2. 在论文推荐的基础上，将专家的特征概括为其所有论文成果的特征组合，研究实现了根据专家相似度进行的专家推荐算法。
3. 将本文算法与多种传统推荐算法进行比较，在公开数据集 CiteULike 上对实验结果进行验证，通过多种情况下的大量实验，分析对比了各个算法的推荐效果及各自特点。
4. 在基金委开放获取系统中，设计并实现论文推荐和专家推荐功能，并完成了系统中添加论文成果的功能。

1.3 本文的组织与结构

第一章说明了本文课题研究的背景与意义，本文的主要工作贡献，以及本文的基本架构和组织结构。

第二章介绍了与本文课题内容相关的技术与研究现状，包括常用算法的介绍及其优缺点的分析。

第三章分别介绍了论文推荐和专家推荐部分使用的算法，详细描述了算法的原理和特点，以及具体的计算过程。

第四章是对本文算法进行实验验证的部分，详细介绍了实验配置、对比算法、实验步骤和实验结果，并对实验结果进行了分析。

第五章是功能实现部分，详细介绍了在基金委开放获取系统中，论文推荐和专家推荐两个功能的设计与实现。本章还包含在系统实现方面其他部分的工作，并展示了这些功能的实现效果。

第六章对本文的全部工作进行了概括总结，反思了工作中的得失，同时对今后的研究工作进行了展望。

1.4 本章小结

本章是本文的绪论部分，首先阐述了在科研文献开放获取系统中论文推荐和专家推荐相关研究的背景与意义，然后从整体上概括了本文的主要工作内容，最后介绍了本文的总体组织结构。

第2章 相关工作

本章主要研究科研文献系统中的相关推荐算法，首先对开放获取系统进行了研究，然后详细介绍了传统推荐算法中基于内容的推荐和基于协同过滤的推荐，其中重点对本文工作涉及到的算法技术进行了阐述。

2.1 科研文献开放获取系统相关研究

2.1.1 开放获取系统综述

科研文献的开放获取已经成为全球学术界的发展趋势，出现了越来越多的开放获取系统，传统的期刊和图书出版商也在逐渐在从传统的基于订阅的文献扩展到了网络电子文献。对于广大科研工作者来说，对科研文献的获取和检索变得简便，研究工作变得更加便捷和全面。对于论文的作者来说，开放获取为他们提供了比从前更广泛的读者群体，提高了他们研究成果的影响力。而对期刊图书的出版者来说，开放获取同样为他们的文献提供了更高的可见性，能够吸引更多的读者，从而获取更多的潜在好处。总体来说，科研文献的开放获取对于科研学术界都有着非常重要的意义。

ScienceDirect[1]是世界上科学研究出版的最大在线收藏网站，它是由荷兰的学术期刊出版商 Elsevier 提供的网络服务。Elsevier 是一家历史悠久的跨国科学出版公司，其出版的期刊是世界公认的高质量学术期刊，被全球许多著名的二次文献数据库所收录。Elsevier 将其出版的 2500 多种期刊和 11000 种图书全部数字化，建立了 ScienceDirect 网站，其中包含了大约 1100 万篇文章[2]，涵盖了农业和生物科学、临床医学、计算机科学、材料科学等 12 个大类，目前为止其中有 25 万篇文章可供用户开放获取。ScienceDirect 为科研工作者们提供了便捷的文献获取方式，促进了开放获取在科技学术界的发展。网站庞大的论文数量会给用户的论文发现带来难题，而 ScienceDirect 也提供了论文推荐的功能以缓解这一问题。如图

[1] http://www.sciencedirect.com/
[2] https://zh.wikipedia.org/wiki/ScienceDirect

2.1 所示为 ScienceDirect 中某篇论文的详情页面，可以看到在页面主体部分的右侧为用户提供了多种发现论文的方式，包括论文推荐、引用本篇的论文、相关图书内容。其中论文推荐功能被放在了最优先的位置，推荐列表中默认显示了 3 篇论文，而点击查看更多论文按钮则可以看到多达 100 篇的推荐论文，可见论文推荐已经成为了用户发现论文的重要方式。

图 2.1 ScienceDirect 论文推荐功能示例

论文推荐在科研专家学者信息获取的过程中十分常用，我国的许多科研文献库或检索网站也提供了相应的推荐功能，如图 2.2 为国内论文检索库中国知网 CNKI[1]中的论文推荐功能。该功能所在页面是论文的详情页面，可以看到网站根据一篇论文的内容推荐出了相似文献，然后又根据网站其他读者的关注情况推荐出了同行关注较多的文献。除图中两部分之外，网站上还推荐了相关作者文献和相关机构文献，从多个角度为用户提供了论文推荐。在 CNKI 整个论文页面的内容中，推荐部分就占了一半以上的篇幅，比论文的基本信息还要多，这说明论文推荐是网站用户进行论文获取过程中非常重要的方式，能够为用户的科研信息获取带来便利。

[1] http://www.cnki.net/

图 2.2 CNKI 论文推荐功能示例

2.1.2 论文推荐算法研究

在论文推荐中有许多常用的算法和研究，比如 PageRank[44]算法就可以用在论文推荐系统[46,43,45]中，来给用户提供一个全局的优化过的推荐结果，但这种做法也存在一定的局限性，即无法针对每个用户的兴趣来推荐个性化的结果。为了解决无法做出个性化推荐的问题，像 Elsevier[1]、PubMed[2]、SpringerLink[3]这样的数字图书馆都在系统中实现了能够根据用户兴趣推送 RSS 订阅更新的功能。这些功能使得推荐系统更加积极主动，能够及时的向用户发送出匹配的论文。但是在这些系统中，推荐功能的实现通常需要用户预先指定自己感兴趣的分类，或者是在分析了用户的搜索记录已后才能推荐，这些都增加了用户的前期配置时间，对于新用户也不够友好。

[1] http://www.elsevier.com/wps/find/homepage.cws_home
[2] http://www.ncbi.nlm.nih.gov/pubmed/
[3] http://www.springerlink.com/home/main.mpx

另外，Sugiyama 等人认为专家们所发表的论文能够清楚的表明作者的研究领域，从而能够代表其感兴趣的方向，因此他们提出了一个通过分析专家过去发表过的论文来进行推荐的模型。在这个模型中，关键的一点是对专家过去发表论文的引用文献，以及引用了这篇文章的其他论文进行分析，从而构建更加精确的用户画像。在实验过程中，作者对只发表过一两篇论文的新学者以及成果比较丰硕的专家分别进行了处理，结果表明该模型在两种情况下都能够很好的提升论文推荐的准确度。

Wang 和 Blei 提出了一种给在线用户推荐科研论文的机器学习算法[10]，该算法使用了两种类型的数据，用户的论文收藏数据以及论文的内容数据。对每个用户来说，该算法不仅能够找到对于相似用户来说比较重要的早期论文，也能找到能够反映用户特定兴趣的新发表论文。算法结合了基于潜在因素模型（Latent Factor Models）的协同过滤算法，以及基于概率主题模型的内容分析思想。与潜在因素模型类似，该算法使用了其他用户的论文库信息。对于特定用户来说，这能够为他推荐来自其他相似用户喜爱的论文。潜在因素模型对于推荐已知的文章效果很好，但无法找到之前未出现过的文章。为了找到未见过的文章，算法使用了主题模型。主题模型依据从集合中发现的潜在主题，提供了一种文章表示方法。当在推荐系统中使用时，这部分能够推荐与用户喜欢的论文有相似内容的文章。论文的主题表示允许算法在有人给文章打分之前，就能做出一些合理的推荐。

Wang 和 Blei 把这些方法用概率模型结合了起来，使得给特定用户推荐论文的问题转换成了计算隐藏变量条件期望值的问题，对这些期望值的计算将能够很好的平衡论文内容的影响与其他用户论文库的影响。未被很多人看到过的文章将会更多的基于其内容进行推荐，而那些已经被很多人看过的文章将更多的基于其他用户的喜好来进行推荐。对于推荐科研论文来说，像上述算法一样将基于内容的和基于协同过滤的方法结合起来，能够达到较好的推荐效果。实验证明，这种方法比简单的矩阵分解方法有更好的表现，说明文章的内容信息能够提升推荐效果。另外，传统的协同过滤算法无法在没有用户评分的情况下推荐一篇论文，而这种混合方法则能够使用新论文的内容来预测哪些用户可能喜欢它们。

2.1.3 专家推荐算法研究

除了论文推荐以外，相关专家的推荐在科研工作中也非常有用，尤其是在知识密集的研究领域有着很重要的作用，因为专家们通常都需要依靠学习其他专家的相关工作来提供不同的思路和灵感。特别是在学术社区中，通过相关专家的推荐，甚至能够促成专家之间的合作研究。因此为每个专家推荐高质量的相关专家就显得尤为重要。然而在学术领域中，由于专家兴趣的动态变化，如何评价专家的专业性没有确定的标准，以及在网络学术社区中的海量数据资源，都给专家推荐的效果带来了挑战。

科研文献管理系统中，与专家相关的信息通常包括个人基本资料和发表论文信息，进行专家推荐时也主要是针对这些信息进行专家特征的概括抽取。Earl[48]在论文中通过分析用户提供的详尽的个人兴趣信息，构造出了用户画像，从而进行相关专家的推荐。而 Yimam-Seid 等人[49]则通过对专家的联系方式，以及论文成果的文本信息进行分析来做出推荐。近年来兴起的社交网络也为专家推荐任务带来了新的发展，Kirchhof 等人[50]就指出，通过对信息检索系统中社交网络的分析，能够找出在社交网络中哪个用户是更有影响力的。Fazel-Zarandi 等人[51]通过应用社会科学理论来计算用户的动机，然后利用得到的用户动机和专家间的关系来构建用户画像并进行个性化的专家推荐。

2.2 传统推荐算法综述

2.2.1 基于内容的推荐算法

在实际应用中，尤其是论文推荐中，待推荐项目通常包含了大量的文本信息，这些信息之间存在的关联性对于推荐系统是非常有用的。基于内容的推荐就是要解决这一问题，充分利用项目和用户的特征，判断用户与项目的相似度，获得更好的推荐效果。

基于内容的推荐过程通常包括三个步骤：

第一步，项目特征提取。项目原始信息可能包括文本或各种非结构化信息，这些信息往往没有明确的意义，也没有特定的取值，所以首先需要把这些信息以

结构化的形式进行表示。对于论文推荐来说，最常用的方法就是使用文章中出现单词的权重向量来代表一篇文章。

第二步，用户特征建立。在这一步中推荐系统将收集和概括用户偏好数据，通常采用机器学习的方法从用户历史行为数据中，根据用户喜欢和不喜欢的项目的特征，抽象出表示用户喜好特征的模型。这个问题可以看作为一个有监督分类问题，机器学习中的分类算法都可以在这一步中使用，例如最近邻算法、决策树算法、线性分类算法、朴素贝叶斯算法等。在实际应用中，用户的历史数据总是在不断更新的，如何实时响应用户新的行为，及时对用户的兴趣偏好改变做出反馈，是相关研究工作的重难点。

第三步，新项目匹配。前两步中的项目特征和用户特征是采用同样的表示方式，因此可直接计算用户与新项目的相似度来进行推荐。相似度的计算也有多种算法可以使用，如皮尔森相似度、余弦相似度等。

基于内容的推荐有以下一些优缺点：

- 文本内容的推荐效果较好。基于内容的推荐技术最早在信息检索领域使用，而在大多数应用场景中，如论文、新闻等，项目都包含丰富的文本信息，这使得基于内容的推荐能够达到很好的效果。
- 对新项目友好。基于内容的推荐不需要用到其他用户的行为数据，而仅仅考虑了项目本身的特征信息。对于新加入系统的项目来说，它们能够获得与已存在项目同等的推荐机会。而在协同过滤推荐中，新项目是无法进行推荐的。
- 多媒体特征难以提取。在实际应用中，项目包含的信息越来越丰富，尤其是多媒体信息越来越多。对于这种信息的提取，传统的文本特征提取方法将不再适用，如果采用人工标注的方法又将耗费大量资源，而协同过滤推荐则能够较好的解决这一问题。
- 过度专业化问题。基于内容的推荐结果往往会局限在用户接触过的几个领域之内，无法发现用户对新领域的潜在兴趣，因此其推荐结果没有协同过滤丰富。

在本文的论文推荐场景中，项目包含的信息以文本信息为主，主要是论文的标题、摘要等，这些信息足够提供比较完整的、有代表性的论文特征。并且在学术研究中，专家学者们通常对最新的研究技术非常关注，基于内容的推荐对于新项目的友好性将带来更好的推荐效果，更符合用户的需求。另外，虽然基于内容的推荐存在专业化的问题，但学术研究通常是比较专一的，用户对于论文的偏好大多都局限在某一个特定的领域，因此在论文推荐的场景中，这一点并不会对推荐的效果产生很大的影响。

2.2.2 基于协同过滤的推荐算法

基于协同过滤的推荐可能是推荐系统中最常用，也是最成熟的技术。协同过滤推荐系统将用户对项目的评分聚集在一起，根据不同用户对相同项目的评分来找出用户之间的共同点，最后基于这种用户间的比较来生成推荐结果。协同过滤推荐中一个典型的用户画像将包括项目和其对应评分的向量，在一些场景中的评分可能是二元的，即喜欢和不喜欢，或者是以特定数字的大小来表示用户对项目的偏好程度。

协同过滤有以下一些优缺点：

- 能够处理复杂的非结构化数据。由于协同过滤算法没有考虑项目的内容，只考虑了用户行为，因此对于比较复杂的非结构化数据，如视频、音乐等，无需进行困难的特征提取工作。
- 有利于发现用户潜在兴趣。若以项目的内容来进行推荐，结果可能会局限在同一个领域，而协同过滤算法则能够发现用户从未接触过的领域，挖掘出用户的潜在兴趣。
- 稀疏性问题。由于用户和项目的数量较大，用户-项目矩阵中有很多元素都是空的，这个矩阵将会非常稀疏，当用户的行为记录较少时，算法的推荐效果并不好，而随着时间推移用户数据增多，算法的推荐质量也会提高。
- 可扩展性较差。随着用户和项目数量的不断增大，推荐系统的时间和空

间复杂度将会越来越大，协同过滤算法中使用到的用户-项目矩阵和相似度矩阵也在增大，带来很大的计算开销。

协同过滤推荐可分为基于内存的推荐（Memory-based）和基于模型的推荐（Model-based）两种[4]。基于内存是指直接根据相似度或其他衡量方法，在用户之间相互比较。而基于模型是指首先根据历史评分数据训练出模型，然后再用这个模型进行预测。

2.2.2.1 基于内存的协同过滤推荐

基于内存的（Memory-based）协同过滤推荐使用用户和项目的关系数据来进行预测，是网络在线推荐系统中常用的方法，一般可分为以用户为基础的（User-based）和以项目为基础的（Item-based）。

以用户为基础的推荐，或称为基于邻居（Neighbor-based）的协同过滤推荐，是指根据相似用户的评分来对目标用户的项目兴趣偏好进行预测。每个用户都对应了用户-项目矩阵中的一个行向量，算法首先将根据皮尔森相关性或向量空间模型[7]，计算任意两个行向量之间的相似度，然后判断出与目标用户最相似的用户，最后会将这些相似用户对他们拥有项目的评分进行加权计算，并按照预测评分进行排序，来得出最终的推荐结果。

以项目为基础的推荐使用了项目间的相似度，而不是用户间的相似度。每一个项目对应了用户-项目矩阵中的一个列向量，计算出项目间的相似度后，未知的打分可以通过相似项目的评分来进行预测。与以用户为基础的推荐相比，以项目为基础的推荐最大的优势在于扩展性问题的改善。以项目为基础的协同过滤不需要对所有的用户进行搜索，来找到相似兴趣的用户，而是能够根据用户评分和属性来进行预评分，这样避免了大量的计算开销。

以用户为基础的推荐和以项目为基础的推荐，因其计算方式不同而适用于不同的场景。对于项目数量大、内容更新快，以及包含社交关系，用户数据丰富的网站来说，比如新闻、微博，使用基于用户的推荐能够达到更好的效果，因为用户的邻居越多越容易发现与他兴趣相似的用户。而对于项目数据相对固定，而社

交关系较弱的推荐，比如购物网站，则更适合使用以项目为基础的推荐，这样计算的开销比较小，且不用经常更新。

2.2.2.2 基于模型的协同过滤推荐

基于模型的协同过滤是指使用数据挖掘和机器学习的方法，通过离线计算的方式来进行推荐。首先将会基于历史数据训练出模型，并且对模型的效果进行评估和优化，当达到所需精度后再根据模型对真实数据进行预测。常用的算法包括贝叶斯网络（Bayesian Networks）、潜在语义分析（Latent Semantic Models）、隐含狄利克雷分布（Latent Dirichlet Allocation）等。这种方法的优点在于，比起基于内存的协同过滤算法，它能更好的处理稀疏性的问题，这增强了大数据集的可扩展性，并且提升了预测的效果。但是这种方法也存在一定的局限性，最主要的就是模型训练和评估的开销，开销的大小与数据集的规模、算法计算复杂度都有很大关系。

2.2.2.3 Top-N 推荐算法

推荐系统的作用一般有两种，评分预测和 Top-N 推荐[11]。评分预测是指在用户可打分的系统中，通过对用户未打分项目的评分预测来对用户进行推荐，而 Top-N 推荐是指找到用户最可能感兴趣的 N 个项目。在论文推荐场景中，通常并没有用户评分机制，用户与项目的关系是以收藏与未收藏来区分的，因此推荐系统的任务主要是进行 Top-N 推荐。在实际应用中，也大多是 Top-N 推荐的情况。Top-N 推荐非常适用于类似本文论文推荐的场景，即对项目没有评分，只有二元关系的情况下。它关注的重点在于用户会不会拥有项目，而不是拥有之后对项目更具体的喜好程度。但是评分预测和 Top-N 推荐并不是完全独立的，在算法设计中常常是相辅相成。在论文推荐系统中，Top-N 推荐实际上是以预测评分的排序来实现的，只是由于用户项目关系为 0-1 矩阵，预测的评分结果均为小数，但这个数值的大小关系可以反映用户对项目的偏好程度。

最近邻模型是推荐系统中常用的 Top-N 推荐算法，也是基于协同过滤的推荐中最常用的算法之一。其作为推荐系统中使用的最广泛的模型之一，在大多数问

题下都会有较好的表现。近邻模型最早是由 Goldberg 等人[32]提出并应用在了邮件过滤系统中，后来研究机构 GroupLens 又对其进行了更加深入的研究和推广，至今已发展的非常完善。选择近邻的方式有两种，给定某个相似度阈值，或给定某个近邻数阈值。K-最近邻（K-Nearest Neighbors，简称 KNN）是常用的给定近邻数的方法，它通常分为基于用户的模型（User-KNN）和基于项目的模型[5]（Item-KNN），这两种模型的基本思想类似，都是寻找最相似的 K 个邻居来进行加权计算，预测用户对未知项目的偏好情况。

相似度的计算是最近邻模型中的关键问题，计算方法也有很多种，常用的包括皮尔森相似度和余弦相似度，以用户相似度为例，其计算方法如以下两个所示，其中 u、v 为用户，i 为项目，r 表示用户对项目的评分。

$$w_{u,v} = \frac{\sum_{i \in I}(r_{u,i} - \bar{r}_u)(r_{v,i} - \bar{r}_v)}{\sqrt{\sum_{i \in I}(r_{u,i} - \bar{r}_u)^2}\sqrt{\sum_{i \in I}(r_{v,i} - \bar{r}_v)^2}} \tag{2.3}$$

$$w_{u,v} = \frac{r_u \cdot r_v}{\|r_u\|_2 \|r_v\|_2} \tag{2.4}$$

皮尔逊相似度是英国统计学家皮尔逊提出的一种计算距离的方法，也被称为积差相关系数。这种相似度算法主要是为了消除用户平均打分偏高或偏低的情况下出现的噪音，因为若只考虑用户评分的绝对值，无法正确反映不同用户之间的喜好情况，皮尔逊相似度主要关注用户对不同项目打分的相对情况是否相似，因而更适用于用户对项目有评分信息的情况下。

余弦相似度是对向量内积空间夹角余弦值的测量，主要考量了向量的方向而不是空间距离，由于高维空间中通常使用向量方向来对两个向量进行评价，因此余弦相似度适用于高维空间的推荐系统中。余弦相似度计算出来的值均介于[0,1]之间，由于它没有考虑用户评分的尺度问题，所以对于有用户评分的推荐系统可靠性并不高，相反在没有用户评分的推荐中比较适用，一般用来计算用户或项目之间的相似度。

2.2.3 混合推荐方法

推荐算法的发展非常成熟，也出现了各种各样的算法，尤其是协同过滤推荐算法十分丰富。每种方法都有自己的优点和缺点，面对实际应用中越来越复杂的需求，单独使用一种推荐方法是无法解决所有问题的，因此通常会采用多种方法混合的方式。几种常用的推荐机制组合方式如下[1]：

1. 加权混合（Weighted Hybridization）：将不同方法得到的推荐结果进行线性组合，在测试数据上对各方法的权重值进行实验优化，以达到最好的推荐效果。
2. 切换混合（Switching Hybridization）：在不同的数据量、系统运行情况、用户数量、项目数量的情况下，各个方法将带来不同的推荐效果，因此可以根据特定情况制定不同的推荐策略，选择最合适的算法。
3. 分区混合（Mixed Hybridization）：多种推荐方法会产生不同的结果，将这些结果有所区分的展示给用户。
4. 分层混合（Meta-Level Hybridization）：结合不同推荐方法的特点，可以采取分层的方式，把一种方法的输出作为另一种方法的输入，从而得到更加准确的推荐。

基本的推荐技术在长期发展的过程中已经比较成熟稳定，近年来推荐系统更多的是围绕混合推荐算法展开，混合方法的适用场景非常广泛，科研工作者们也在不断的探寻新的混合方法，以适应不断变化的现实需求。

2.3 文本特征相关研究

2.3.1 文本相似度计算

在基于内容的推荐算法中，对待推荐项目间相似情况的判断是非常重要的步骤，而在论文推荐中则主要是进行文本相似度的计算。文本相似度是表示两个或多个文本间匹配程度的度量参数，相似度的计算在推荐系统中被广泛用于基于内容的推荐中。多数基于内容的推荐系统都使用比较简单的检索模型[1]，比如关键

[1] https://www.ibm.com/developerworks/cn/web/1103_zhaoct_recommstudy1/

词匹配或向量空间模型（Vector Space Model，简称 VSM）。VSM 是文档的空间表示，在这个模型里，每一个文档都被表示为空间中的一个向量，其中空间的每个维度对应了给定文档集的单词表中的一个单词。在形式上，每个文档都被表示为单词权重的向量，其中每个权重表示文档和单词间的关联程度。单词在文档中的重要性可以使用词频-逆文档频率（Term Frequency-Inverse Document Frequency，简称 TF-IDF）来衡量，它以单词在文档中出现的频率来进行区分。TF-IDF 是最基础的文本相似度计算方法，其算法非常简单，并且对文章的所有元素都进行了综合考量。但由于它仅仅考虑了单词的词频信息，而没有利用文档的语义特征，所以对于一词多义和一义多词等情况，都无法进行很好的区分。另外，如果一个词条是一类文档中频繁出现的，说明这个词条能够很好的代表这类文档，应当获得较高的权重，但是用 TF-IDF 却会忽视这类词条。

近年来在推荐系统中经常出现的主题模型（Topic Model）可以很好的避免上述的问题，它是一种概率模型，引入了主题空间的概念，将文档从高维的词频空间降维到低维的主题空间，常用的方法包括 pLSI、LDA 等。主题模型的建模也是通过向量的形式，其核心在于将文档看做若干主题的概率分布组合，又将每个主题看做若干单词的概率分布。使用主题模型时首先要对文档的生成过程进行建模，然后再通过参数估计得到各个主题。在 VSM 模型中文本的表示维度通常是非常大的，与文本中单词的数目正相关，而使用了主题模型后，文本被映射到了主题空间上，表示维度仅与指定的主题个数有关，大大降低了文本相似度计算的开销。主题模型能够发现文档的语义信息，从而发现文档间的隐含关系，可以有效的解决一词多义，一义多词的问题，对分析文档内容，抽取文档特征具有重要意义。另外它还能对文档表示的维度进行压缩，给系统性能的提升也带来了改进。主题模型最初主要用于自然语言处理领域，现在已经被广泛延伸到了许多地方，如图像处理、生物信息学等。

LDA 是主题空间模型中比较有效的也是最常用的模型之一，它是一个三层贝叶斯模型，集合中的每一个文档都将用同一组主题的概率向量进行表示，同时每个主题由一组词的概率向量进行表示。我们认为一篇文档的每个词都是通过以一

定概率选择了某个主题，并从这个主题中以一定的概率选择某个词语。那么如果我们要生成一个文档，它里面每个词语出现的概率公式如下：

$$p(词语|文档)=\sum_{主题} p(词语|主题)\times p(主题|文档) \tag{2.1}$$

该概率公式可以用矩阵表示，如图 2.3 所示。

图 2.3 LDA 模型概率公式矩阵表示

其中的“文档-词语”矩阵表示每个文档中各个单词出现的词频，也就是出现的概率，“主题-词语”矩阵表示的则是在每个主题中各个单词出现的概率，“文档-主题”矩阵表示的是每个文档中每个主题出现的概率。在给定了一系列的文本以后，通过对这些文本进行分词、统计工作，就可以得到每个单词在文本中的词频，也就是“文档-词语”矩阵，主题模型就是通过对左边的这个矩阵进行训练，学习出右边的两个矩阵。

LDA 在对大规模文档集进行主题提取时能够得到比较好的效果和效率，因此在文本相似度计算方面有非常广泛的应用。在 LDA 建模的过程中最关键的问题是如何准确获取主题的关键词分布，以及目标文档的主题分布。在得到了这些结果以后，可以进一步融入在线系统中用户与文档的关系数据，从而对用户的主题特征分布进行计算，进而进行用户相似度计算、文档推荐等工作。LDA 对用户及项目特征高效的概括抽取，使得它在推荐系统中也成为了常用的方法。

2.3.2 词向量

在自然语言处理的过程中，词向量技术（Word embedding）是最常用的方法之一，词向量最早是由 Hinton[37]在 1986 年提出的，其基本思想是通过对文本语

料的训练，将单词从每一个词一个维度的高维空间映射到 N 维的实数向量空间，然后通过单词之间的向量距离来表达它们之间的语义相似度。当选择使用词向量作为底层的输入时，能够极大的对自然语言处理任务的性能进行改进，比如说语法解析[13]、情绪分析[14]等任务。词向量最大的优点在于它对语义的解析，它让有相同语义的词在数学意义上的距离更近了。词向量通常是使用特定方式对语言模型进行训练获得的，常用的训练方法包括神经网络语言模型、C&W[29]、M&H[30,31]等，其中神经网络是最成熟的一种方法，后文介绍的 Word2Vec 也是基于这种方法进行改进的。

词向量技术发展到现在已经有了非常多的研究成果[38,39]，例如非常流行的神经网络语言模型[37]（NNLM）结构，这个结构使用了由一个线性映射层和一个非线性隐含层构成的前馈神经网络，通过该网络来学习词的向量表示以及基于统计的语言模型，该工作成为后来的很多其他相关研究论文的基础。在另一篇论文[41,42]中提出的 NNLM 结构首先用只包含一个隐含层的神经网络来计算出词向量，得到的词向量之后会被用来训练 NNLM。通过这种方式词向量的获得就跳过了构造一个完整 NNLM 的过程。

Word2Vec 是 Mikolov 在上述工作的基础上，提出的一种用于计算词向量的新方法[3,16]，他在扩展了 NNLM 结构的基础上，将重点放在了第一步中如何用一个简单的模型来获得词向量的过程。Mikolov 提出了两个计算性能优异的计算词向量的模型结构[3]，他在研究之前的模型时发现，大量的计算时间都耗费在模型中的非线性隐含层中，于是他想在放宽一些数据表示的精确度的基础上，来大大的提高模型在大数据集上的训练效率。在论文中 Mikolov 一共提出了两种新的降低计算复杂度的用于学习单词分布表示的模型结构，连续词袋模型（Continuous Bag-of-Words Model，简称 CBOW）和连续 Skip-gram 模型（Continuous Skip-gram Model）。

和前馈 NNLM 类似，CBOW 模型移除了非线性隐含层并且让所有的单词共享同一个映射层，这样做可以将所有的单词映射到相同的位置上面。之所以叫做 Bag-of-Words 模型是因为在该模型中，单词出现的位置并不会对映射的结果造成

影响。但是不同于标准 Bag-of-Words 模型的是，CBOW 使用了语境的连续分布表示。CBOW 模型的结果如图 2.4 所示，可以看到和 NNLM 一样，所有的单词位置共享了输入层和映射层之间的权重矩阵。

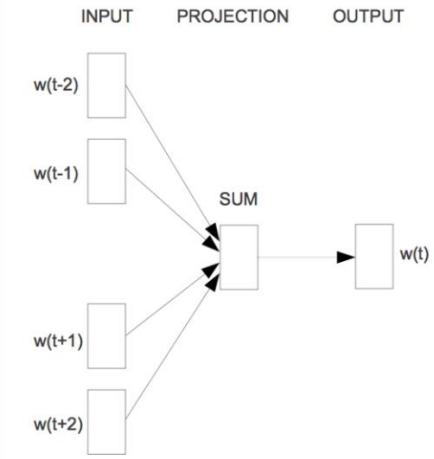

图 2.4 CBOW 模型结构图

第二个模型 Continuous Skip-gram 模型和 CBOW 相似，但是不同于 CBOW 的基于语境对当前单词进行预测，Skip-gram 是基于同一个句子中出现的另一个单词来最大化当前单词的分类可能性。更准确的说，Skip-gram 是通过把每个单词作为具有连续映射层的分类器的输入，来预测当前单词的前后一定范围内可能出现的单词。实验表明，扩大前后范围可以提高得到的词向量的质量，当然相应的计算量也会增大。Skip-gram 的模型结构如图 2.5 所示。

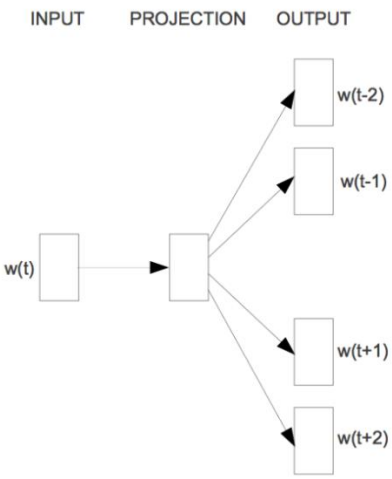

图 2.5 Skip-gram 模型结构图

Skip-gram 模型是用一个神经网络模型来学习每个单词的向量表示。Skip-gram 由一个输入层、一个映射层、一个输出层来预测附近可能出现的单词。其中，每个词向量都是通过最大化下面的这个概率公式来得到的：

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{j\in nb(t)}\log p(w_j|w_t) \tag{2.2}$$

其中 $nb(t)$ 是与单词 w_t 临近的单词集合，$p(w_j|w_t)$ 是和词向量 $V_{w(j)}$ 和 $V_{w(t)}$ 对应的分层 softmax。由于使用了简单的结构，Skip-gram 能够在一台普通的台式机上面进行每秒上百万的单词计算。这种高效的计算能力，使得模型在大数据集上进行训练成为可能，从而能够得到更加复杂的单词关系。另外，Word2Vec 一个最大的特点就是支持多角度线性运算，比如进行减法计算，学习到类似以下的关系：

$$vec(Japan) - vec(sushi) + vec(Germany) \sim vec(bratwurst)$$
$$vec(Einstein) - vec(scientist) + vec(Picasso) \sim vec(painter)$$

作者的实验结果表明[3]，和流行的神经网络模型（Feedforward NNLM 和 Recurrent NNLM 模型）相比，通过一个简单的模型结构来训练一个高质量的词向量是可行的。由于简单的模型结构可以大大的降低计算复杂度，使在大数据集上计算词向量成为可能，因此得到了更加精确的高维词向量表示。Word2Vec 能够在较短的时间内从大型文本中学习到高质量的词向量，在得到了词向量以后，就可以根据其结果计算出单词与单词之间的距离，而文档就是由一组单词表示的，因此我们便可进一步对文档与文档之间的距离进行计算。另外这种方法还具有很大的便捷性，其对词向量的学习过程不需要对语料库进行分类标记，只需要文档内容即可。

Mat J. Kusner[17]提出了一种基于 EMD[35,36]（Earth Mover's Distance）和 Word2Vec 的计算文档相似度的算法 WMD（Word Mover's Distance）。在用 Word2Vec 计算出词向量的基础上，WMD 通过计算将一篇文章所包含的词向量转

换成为另一篇文章所包含的词向量所需的最小距离，从而得到两篇文章之间的相似度。如图 2.6 所示，像 Obama speaks to the media in Illinois 和 The President greets the press in Chicago 这两个句子，虽然没有相同的单词，但语义却是接近的：（Obama，President）；（speaks，greets）；（media，press）；（Illinois，Chicago），用 WMD 算法就能得到这两个句子具有较高的相似度。

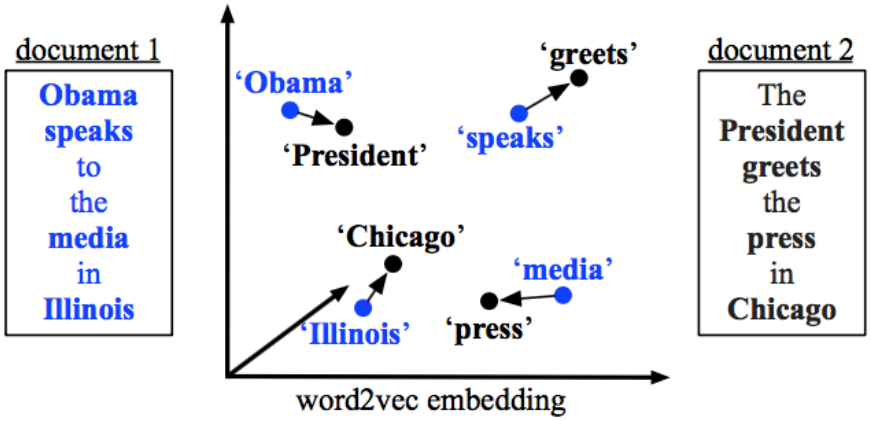

图 2.6 WMD 算法示例

2.4 本章小结

本章介绍了与本文工作相关的算法以及研究成果，首先对科研文献管理系统的相关工作进行了研究，并重点介绍了论文推荐和专家推荐在其中的应用。然后本章介绍了传统推荐算法中的基于内容的推荐算法和基于协同过滤的推荐算法，并重点介绍了部分算法技术，这些工作都是本文推荐算法设计以及系统功能实现的基础。

第3章 基于协同过滤与内容的科研文献推荐算法

本章将详细介绍在本文所研究实现的推荐算法，算法包括论文推荐和专家推荐两部分，本章对算法解决的问题、算法的详细步骤以及具体的计算过程进行完整的阐述，为推荐功能在 OA 系统中的实现打下理论基础。

3.1 问题描述

为了方便后续的介绍，我们将罗列出本文所涉及到的数学符号，如表 3.1 所示。所有符号的定义都是基于对比实验中所使用的数据集所得，关于数据集的详细描述见 4.1.2 节。

表 3.1 数据符号定义

符号	描述
U	所有用户的集合
P	所有论文的集合
u	某个特定用户
p	某篇特定论文
C_u	用户 u 收藏的论文集合
E_i	组成论文 i 的单词列表

在前述数据符号定义的基础上，本算法将解决论文的个性化推荐问题，具体来说，问题的定义为：给定用户 u 属于 U，以及用户的收藏数据 C_u，论文的内容 E_i，利用个性化推荐算法给用户推荐相关性高的论文集合，并按照相关性降序排列，最后将得到的有序列表呈献给用户 u，作为给用户的个性化推荐结果。

为了验证本文算法的有效性，我们需要将其与其他推荐算法进行比较，本文的算法参考了基本的基于内容和基于协同过滤的推荐算法，预期算法的推荐结果应该在整体表现上优于这些基本算法。

3.2 算法概述

在 OA 系统中有大量的内容信息，包括论文相关的文本信息、专家的个人资料信息，以及专家与论文之间的对应关系信息等。论文信息和专家信息主要以文本形式为主，对于文本内容的挖掘是进行推荐的重点。本文将在充分利用了这些信息的基础上，综合考虑文本内容和用户行为两方面因素，使用混合方法进行推荐。本文所研究的算法包括了论文推荐和专家推荐两个部分，两部分包含了重叠的计算过程，其关系如图 3.1 所示。

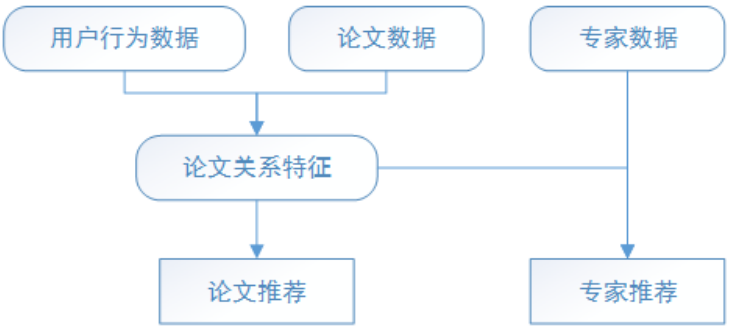

图 3.1 论文推荐与专家推荐关系示例

论文推荐部分，本文提出了一种将基于协同过滤的推荐算法与基于内容的推荐算法相结合的混合推荐算法，算法的计算过程主要包括了论文相似度计算和最近邻模型推荐两个步骤。首先是项目相似度的计算，这里的相似度包括了基于协同过滤的相似度和基于文本词向量距离的相似度两种，我们将根据这两种相似度的加权组合计算得到一个综合相似度值。接下来是对项目评分进行预测，针对用户未打过分的所有项目，我们将使用最近邻模型算法来对用户评分进行预测，即选取与该项目最相似的若干个项目，使用用户对这些相似项目的评分进行加权计算。最后我们将各个项目的预测评分排序得到最可能被用户喜欢的若干个项目，作为最后的推荐结果。

专家推荐部分，主要目标是对专家间相似情况进行比较，为每个专家推荐与其研究内容相似的专家学者。由于与专家相关的个人资料一般比较少，且其中与研究内容相关的多是概括性的描述，无法准确的对专家的研究特征进行概括。但

每个专家的论文成果信息非常完整，且论文的内容反映了专家的研究方向，因此我们将通过对任意两个专家论文列表的对比，来进行专家相似度的计算。我们将根据论文推荐过程中计算出来的论文与论文的相似度值加权得到两个专家的相似度，然后为每一个学者推荐与他最相似的专家。

3.3 论文推荐

3.3.1 基于用户评分的论文相似度计算

基于用户评分的相似度与项目内容无关，主要根据用户-项目关系矩阵计算所得。论文推荐的用户-项目评分矩阵中每一项的值都为 0 或 1，其中 1 表示用户收藏了这篇论文，而 0 表示用户未收藏。矩阵中的每一个行向量代表了一个用户，每个列向量都代表了一个项目，列向量的内容就是所有用户对它的拥有情况，而项目的特征是由这个向量表示的。两个列向量之间的距离越近也就是它们的相似度越高，进而能够说明从总体上来说用户对这两个项目的喜好情况越类似，这两个项目也就有较大的可能被同一个用户喜欢。

相似度的计算方法有皮尔森相似度、余弦相似度等，由于在论文推荐中没有详细的评分机制，仅有用户与论文的二元关系，因此我们将采用余弦相似度，即向量夹角的余弦值进行计算。论文 i 和论文 j 之间的评分相似度计算公式如下，其中 v_i 和 v_j 分别为两篇论文的列向量：

$$simi_{SCORE}(i,j) = \frac{v_j' v_i}{\sqrt{(v_i' v_i)(v_j' v_j)}} \tag{3.1}$$

根据以上公式计算出来的相似度结果为两篇论文列向量夹角的余弦值，取值范围在 0 到 1 之间，该数值越大表示两篇论文的相似度越高，两篇同样的论文相似度值将为 1。该算法的时间复杂度为$O(N_{user})$，其中N_{user}为用户的数量。

3.3.2 基于词向量的论文相似度计算

基于内容进行推荐的算法能够解决模型训练中的冷启动问题，即当系统中的

论文数还比较少的时候如何训练得到模型。由于单词语义关系是普遍存在于文本中的，论文文本与一般文本在语义上并没有差别，因此论文推荐中基于内容的模型并不一定要通过对论文集合的训练得到，也可以通过对诸如维基百科这样的普通文本来训练得到模型。而在已经有大量论文文本数据的情况下，完全通过论文数据集来进行的训练，将能够得到一个更加有效反映论文语境中语义表示的模型，这也是我们在本论文中所采用的做法。

基于内容的推荐算法上我们选择了基于词向量的方法，相对于 LDA 这类方法采用的 Bag-of-Words 假设，词向量算法的神经网络语言模型中加入了单词之间相对位置的考虑，这样便能够更加准确的表达单词之间的相关性。我们预期在基于 LDA 的推荐算法和基于词向量的推荐算法的对比中，词向量算法能够得到更好的推荐结果。

在多种计算词向量的算法的选择上，我们选择用 Mikolov 提出的 Word2Vec 模型来生成单词的向量表示。之所以选择用 Word2Vec 来计算词向量，是因为和其他的模型（如 Feedforward NNLM 和 Recurrent NNLM）相比，由于 Word2Vec 选择了一个相对简单的模型结构，使得模型在大数据集上的训练成为可能。由于大量训练文本带来的词向量精确度提升，从侧面超过了从前的复杂模型。在我们开发的基金委科研文献开放存取系统中，目前已经有原始的 200 多万篇论文，并且该数据量还在以每年 30 多万的速度增长，在如此庞大的数量级下，Word2Vec 带来的计算复杂度降低能够极大得缩短模型的训练时间。

在使用 Word2Vec 模型生成了单词的向量表示后，我们选择了基于词向量来计算文本相似度的 WMD 算法[17]来计算文档距离，WMD 算法的主要思想是，计算一个文档中的词最少移动多少距离能够变成另一个文档，将这个距离定义为两个文档之间的距离，距离越大则文档的相似度越低，其中一个文档跟自己的距离为 0。

在对所有的论文文本进行分词处理，并且去掉停用词、低频词等这些预处理操作后，将进行训练 Word2Vec 模型的步骤。这里我们根据经验选定了关键参数，特征向量维度选为 100，训练算法选择用 Skip-gram 算法，单词间最大距离设定

为 5。在得到训练好的 Word2Vec 模型以后，我们就得到了任意两个词之间的相似度，接着利用 WMD 算法可以计算得到两个文本之间的 WMD 距离，该值越大表明这两个文档之间的差异越大，反之则越相似。

$$dis(i,j) = wmd(i,j) \tag{3.2}$$

在对任意的两个文本计算 WMD 距离以后，就得到了所有文本两两之间的距离，但该距离与论文之间的相似情况是负相关的，即距离越大表示两篇论文越不相关。而我们在上一节的相似度计算中使用的是余弦距离，所得结果范围为 0 到 1，数值越大代表两篇文章的内容越相似。因此为了方便后续进行的加权操作，我们需要将基于词向量的文本距离转化为对应的相似度值。这里我们选择使用如下的公式来进行转换：

$$simi_{WMD}(i,j) = \frac{1}{1 + dis(i,j)} \tag{3.3}$$

经过上述公式的计算，当论文的距离为 0（即两篇一模一样的文章）时，相似度值为 1，而随着文本距离的增大相似度值会越来越小，且逐渐逼近于 0，符合我们的计算要求。

算法的时间复杂度方面，在训练模型的过程中，Word2Vec 方法使用 Skip-gram 模型的时间复杂度为$O(C\times(D + D\times \log_2 V))$，其中 C 是单词间的最大距离，D 是向量的维度，V 是集合中的单词数。在论文推荐的计算性能上，用 WMD 算法来计算两篇论文相似度的时间复杂度为 $O(p^2)$，其中 p 是组成两篇论文的所有不同的单词数量。

3.3.3 相似度加权

在得到了前两小节的相似度值后，我们需要将其综合考虑以进行最后的论文推荐，由于两种相似度值已经进行归一化，范围均在 0 到 1 之间，因此我们将直接对其进行加权，加权的公式如下：

$$simi(i,j) = (1-\lambda) \cdot simi_{SCORE} + \lambda \cdot simi_{WMD} \quad (3.4)$$

经过以上公式的计算我们将得到混合的论文相似度矩阵，公式中 λ 的取值范围在 0 到 1 之间，在后续的实验中，我们将对 λ 的不同取值进行对比，观察其对推荐效果的影响。我们通过调整系数 λ 来控制基于的两种算法的影响力，当 λ 为 1 时即退化为基于内容的推荐算法，当 λ 为 0 时即退化为基于协同过滤的推荐算法。

通过将两种方法结合，除了可以预期在最后的推荐效果上得到提升以外，在科研文献的推荐中还具有如下优势。一方面，相比基于协同过滤的算法来说，当有新的论文加入时，由于所有人都没有对此有过收藏评分行为，将导致新的论文不可能得到推荐。在加入了基于内容的推荐部分以后，即使协同过滤部分的值为 0，基于内容的推荐也能够做出评分，如果和目标论文的相似度很高的话，将会出现在列表的前面。另一方面，对比基于内容的推荐算法来说，虽然不会有冷启动的问题，但是由于推荐出来的内容高度相关的论文很有可能在质量上没有保证，所以单纯的基于内容进行推荐的算法往往在召回率的指标上会比协同过滤差很多。通过融入基于协同过滤的因子，我们能够找到一个平衡点来综合考虑两者在论文推荐的最终结果中的影响力。

混合算法的时间复杂度方面，由于基于协同过滤的相似度计算的时间复杂度为 $O(N_{user})$，而基于词向量的 WMD 算法的相似度计算时间复杂度为$O(p^2)$，所以最终混合算法的相似度计算时间复杂度将由 WMD 算法主导。

3.3.4 基于最近邻模型的推荐

利用上一节中的算法，我们已经得到混合了用户收藏关系和文本相似度的项目相似度矩阵，利用这个矩阵我们将给每个用户推荐感兴趣的项目。针对目标用户 u 与他未收藏过的项目 i，我们选取了与项目 i 相似度最高的 K 个项目，然后根据用户 u 对这些项目的评分来进行加权，从而得到用户对该项目的预测评分，计算公式如下，其中 $r(u,j)$ 表示用户 u 对物品 j 的评分。

$$predict(u,i)=\frac{\sum_{j=1}^{K} simi(i,j)\cdot r(u,j)}{\sum_{j=1}^{K} simi(i,j)} \tag{3.5}$$

由于在论文推荐中用户对项目的评分只有 0 或 1，因此得到的预测评分均为 0 到 1 范围内的小数，数值的大小代表了用户偏好程度的高低。在得到了用户 u 所有没有收藏的物品的评分之后，我们对该列表进行降序排列，排名越靠前的论文我们认为越有可能被用户喜欢，我们将评分最高的 N 个项目作为给用户的推荐结果。

3.4 专家推荐

在论文推荐之后，我们还将进行相似专家的推荐。与论文推荐不同，相似专家的推荐并不是针对普通用户进行的，其主要应用场景是为系统中的学者推荐与他相似的专家，以供普通用户在访问学者主页时能够发现相关专家。因此专家推荐主要的目标是基于专家的个人信息及成果信息，在专家与专家之间进行相似度的比较。网站中专家的个人信息往往是有限的，并且能够反映专家研究方向的信息更加稀少。但是专家的论文列表却代表了专家在相关领域上的主攻方向，能够有效的反映出专家的研究兴趣信息。通过比较不同专家论文集合之间的相似度，也就能够得到不同专家之间的相似度关系，进而可以做出相关专家的推荐。我们将利用论文推荐中得到的论文词向量距离，在此基础上进行专家推荐。

考虑到不同专家之间发表论文数量的差异，论文列表的绝对总和并没有实际意义，我们提出了如下的公式来计算专家之间的相似度：

$$simi(u,v)=\frac{1}{n_u}\sum_{i=1}^{n_u}\min\big(dis(i,j)\big),\forall j\in P_v \tag{3.6}$$

其中 $dis(i,j)$ 表示两篇文章 i 和 j 之间的 WMD 距离，P_v 表示专家 v 的论文集合，n_u 表示专家 u 的论文总数。上式中我们假设专家 u 的论文数不大于专家 v 的论文数，否则交换 u 和 v 的位置。

论文之间的距离和相似度是负相关的，这里我们对专家 u 集合中的每一篇论

文都到专家 v 的集合中找最相似的一篇，加总并做平滑处理。由于每次计算都保证了 u 的论文数小于 v 的论文数，可以减少一半的计算量，最终得到的用户相似度矩阵将是一个对称矩阵。

3.5 本章小结

本章对本文的混合推荐算法进行了详细介绍，论文推荐部分在有用户数据的前提下，提出了将协同过滤推荐与内容推荐结合的算法，专家推荐部分在论文推荐的基础上进行了专家论文列表相似度的比较，本章详细描述了算法的步骤以及计算过程。

第4章 实验结果与分析

本章是本论文的实验部分，将在公开数据集 CiteULike 上将本文算法与多个对比算法的推荐效果进行比较，验证算法的有效性。本章将详细介绍实验环境、实验数据、实验步骤，以及实验结果的展示和分析等。

4.1 实验配置

4.1.1 运行环境

本文实验环境包括实现算法和实现系统的环境，算法中的 Word2Vec 部分使用 Python 实现，版本为 Python 3.5，相关实验在作者实验室的服务器上进行。其余的算法均使用 Matlab 实现，版本为 Matlab 2016a，相关实验在作者个人电脑上运行。系统功能的实现是以网站的形式，主要使用到的语言为 HTML、CSS、JavaScript、Java，开发环境也为作者的个人电脑。实验的硬件配置如表 4.1 和表 4.2 所示。

表 4.1 实验服务器硬件配置

项目	内容
处理器	Xeon E5-2603
内存	200G
操作系统	CentOS 7

表 4.2 实验个人电脑配置

项目	内容
处理器	Intel(R) Core(TM) i5-4590
内存	16GB
操作系统	Windows 7 64bits

4.1.2 数据集描述

由于 OA 系统的真实数据只有论文与专家的信息，没有用户行为记录，为了验证本文算法的有效性，我们使用了科研社交网络 CiteULike 的公开数据集，主要对论文推荐部分的混合算法效果进行验证。CiteULike 是提供文献引用和分享的网络服务，用户可以在上面创建自己的论文库，收藏自己感兴趣的论文。CiteULike 的公开数据集包含的信息较为丰富，对于验证论文推荐相关算法的效果具有较强的说服力，本文主要用到的是其中提供的论文的标题，摘要信息以及用户收藏信息。另外，我们在实验中使用的是 Wang[10]已经处理过的数据，他将原始数据集中的空论文全部移除，并且仅保留了收藏论文数大于 10 篇的用户。对每篇论文来说，标题和摘要中的停用词全部被忽略，并且 Wang 还使用 TF-IDF 抽取了 8000 个不同的词作为单词表。数据集的一些具体统计信息如表 4.3 所示。

表 4.3 数据集基本信息统计

内容	数值
用户数	5551
项目数	16980
标签数	46391
引用数	44709
用户-项目对数	204987

表中的用户-项目对数指用户收藏了论文的关系对数，我们根据该数据生成了用户-项目矩阵。从表中可以看出，由于用户和项目数量都比较大，用户-项目矩阵将是非常稀疏的。在我们生成的矩阵中，1 表示用户收藏了对应项目，0 表示用户未收藏，而约有 99.8%的数据都为 0。根据统计，用户收藏的论文数量平均为 36 篇每人，其中最多的有 403 篇，最少的为 10 篇，约 93%的用户收藏数小于 100 篇。而选择每篇论文的用户数量平均为 12 人，最多的有 321 人，最少的为 1 人，其中 97%的文章被收藏量小于 40 人。

对于推荐算法来说，用户收藏论文数量的多少会对推荐结果造成影响，我们

在后续实验中也会对这一情况下各算法的表现进行考量，为了更全面的掌握数据集分布组成，我们对 5551 名用户收藏论文的数量进行了统计，结果如图 4.1 所示。

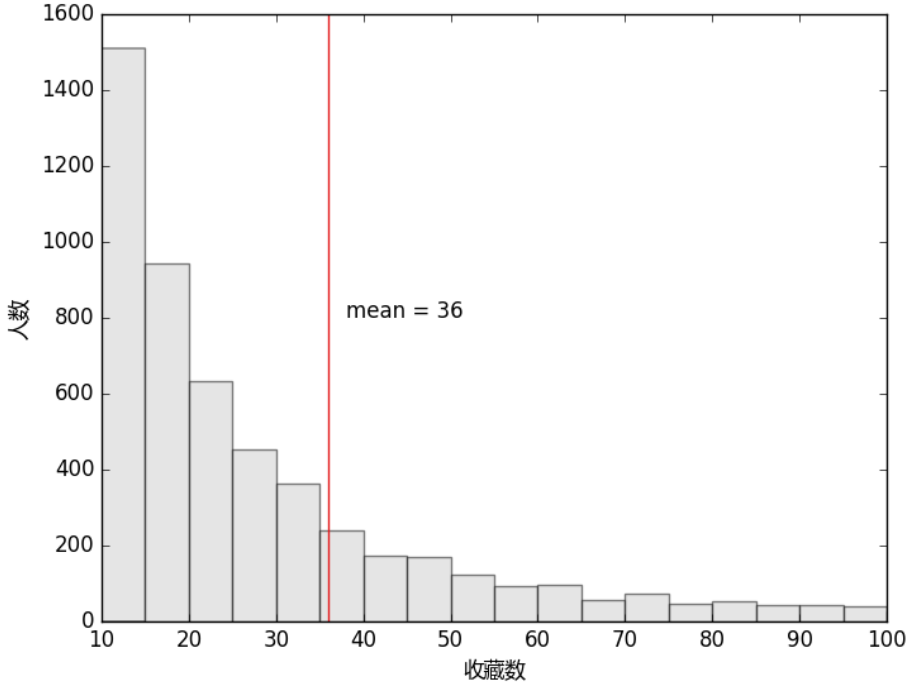

图 4.1 用户论文收藏数分布图

从图中可以看到，用户论文收藏数总体偏少，且数量越多对应的用户数也越少，大部分用户的收藏数在 10 到 30 篇。在论文推荐过程中，丰富的用户行为数据会使基于协同过滤的推荐效果变好，关于收藏数对不同算法推荐效果的影响将在 4.3.3 节进行具体实验分析。

另外，我们在进行文本相似度计算的过程中使用到了数据集中的论文内容信息，包括论文标题、摘要、关键字。而文本内容中单词数的多少也将对文本相似度计算，尤其是词向量 Word2Vec 算法的效果产生影响。因此我们对数据集中所有论文的单词数也进行了统计分析，结果如图 4.2 所示。从图中可以看到，论文单词数量分布基本比较集中，大多在 40 到 130 之间，仅个别论文的单词数量偏多或偏少，整体来说这种分布情况对于词向量计算是比较有优势的，能够比较准确的对论文特征进行概括。

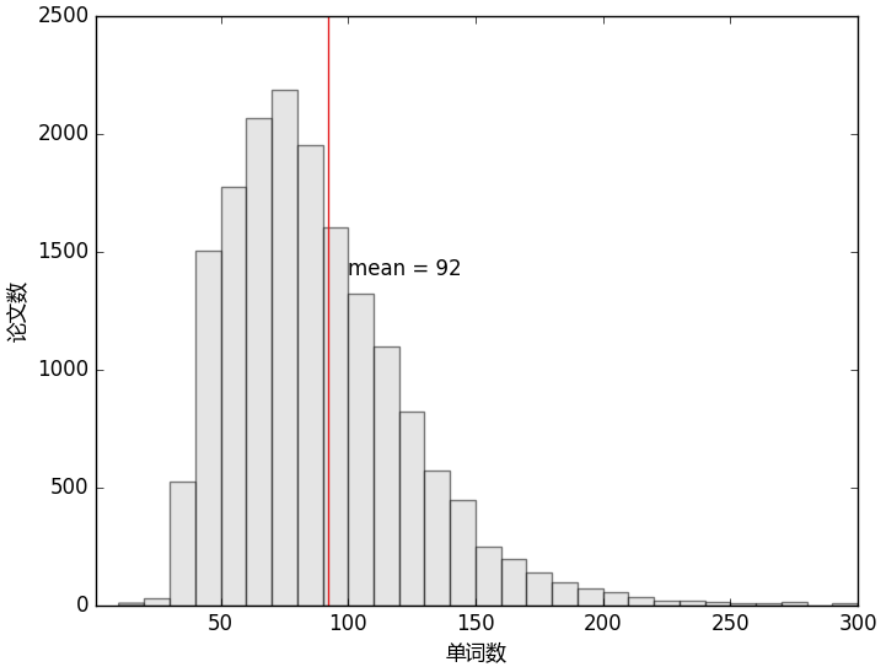

图 4.2 数据集论文单词数分布

4.1.3 对比算法

本文选择了若干算法进行对比实验，所选算法如表 4.4 所示。

表 4.4 实验对比算法介绍

算法名称	算法描述
UserKNN	基于用户的协同过滤 Top-N 推荐算法
ItemKNN	基于项目的协同过滤 Top-N 推荐算法
WMD	基于词向量的内容比较推荐算法
LDA	基于 LDA 主题模型的内容比较推荐算法
CTR	协同主题模型[10]
Hybrid	本文提出的混合推荐算法

上述算法中，ItemKNN 和 WMD 是本文混合算法所参考的主要依据，因此将重点进行对比。其中 WMD 算法和 LDA 算法主要是用于计算文档相似度的，我们将使用简单的推荐方式，在相似度的基础上将目标用户收藏的论文列表与目标论文的相似度进行加权，推荐最相似的论文。

4.1.4 衡量指标

推荐系统的衡量指标主要包括预测准确度的衡量和分类准确度的衡量等[15]。预测准确度的衡量通常使用平均绝对误差（Mean Absolute Error）和均方根误差（Root Mean Squared Error），然而本文的论文推荐场景中，仅有用户对项目的收藏与未收藏数据，没有用户评分信息，并不适合这种评价标准。在实际的实验过程中也发现，误差指标无法正确的反映算法的推荐效果。

在算法的实现中，我们针对测试集中的每个用户推荐了 N 个项目，将其与实际结果进行比较，通过使用分类准确度来衡量算法的效果，评估指标包括准确率（Precision）和召回率（Recall）。我们将改变向每个用户推荐项目的数量 N，来对结果进行详细的评判，将使用 Precision@N、Recall@N 来表示推荐 N 个项目的情况下各指标的大小。假设N_{select} 表示用户实际选择的项目，N_{return} 表示推荐的项目。各指标定义如下：

$$\text{Precision} = \frac{|N_{select} \cap N_{return}|}{N_{select}} \tag{4.1}$$

$$\text{Recall} = \frac{|N_{select} \cap N_{return}|}{N_{return}} \tag{4.2}$$

另外值得注意的是，由于本文实验中评分为 0 的项是不确定的，它可能是表示用户不喜欢这篇论文，也可能表示用户还没有看过它，因此准确率值的计算是非常困难的。与此同时评分为 1 的项代表用户喜欢这个项目，它的值就是正相关的值，所以对于召回率的比较更能反映结果的好坏，我们将主要使用该值来进行评估。

4.2 实验过程与步骤

4.2.1 文本距离计算

我们在实验中进行文本距离计算的第一个步骤是获取每篇论文的分词列表，以进行 Word2Vec 模型的训练，因此我们首先将对论文数据进行预处理，其步骤如图 4.3 所示，我们选取了数据集中的两篇论文，以它们的标题文本为例对预处理流程进行说明。

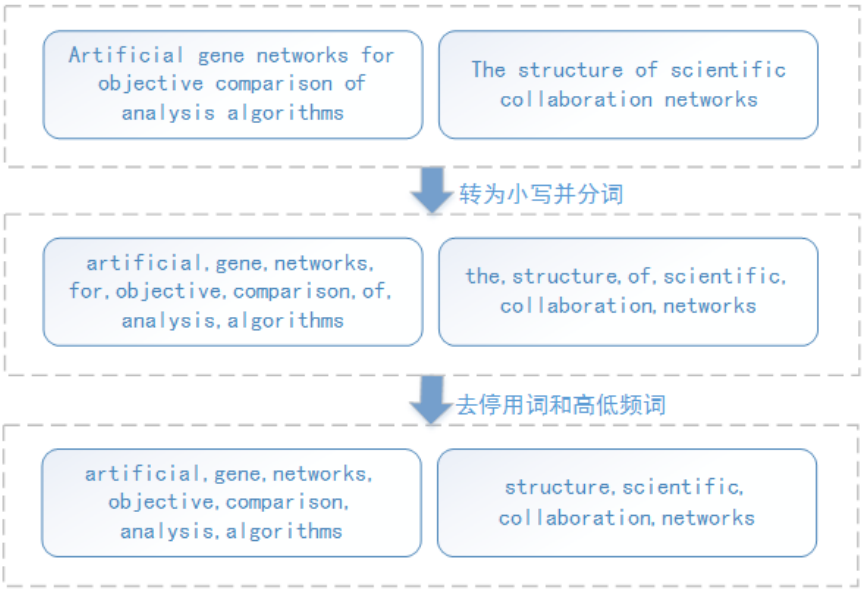

图 4.3 论文数据预处理流程

数据预处理的第一步是将文本转换为小写并进行分词操作，我们在实验中使用的 CiteULike 数据集里包含了每篇论文的标题和摘要文本信息，以及每篇论文对应的所有分词，但是该分词信息是以无序集合的形式给出的，而在 Word2Vec 算法实验中却需要使用包含位置关系的文本分词信息，因此我们首先需要对原始论文数据进行处理，得到有序的分词列表。接下来我们需要将停用词和高低频词移除，在原始论文数据中包含大量停用词，对论文特征的提取以及相似度的判断来说并没有意义，反而会增加计算的开销。由于 CiteULike 数据集中包含所有论文的合计单词表，已经将停用词剔除，因此我们将直接使用单词表对原始单词进行筛选，仅保留在单词表中的词语。

经过上述处理后我们得到了每篇文章对应的分词列表，列表中每行表示一篇文章，接下来使用该列表训练 Word2Vec 模型，我们使用了 Python 的主题模型算法库 gensim[1]中的 Word2Vec 方法来生成单词的向量表示模型。为了对模型的效果进行说明，我们随机选取了若干单词，使用 Word2Vec 模型分别计算出单词库中与其最相似的和最不相似的各 5 个单词，得到的结果如表 4.5 所示。从表中可以

[1] https://radimrehurek.com/gensim/

看出，Word2Vec 模型能够非常准确的体现单词之间的相似度关系。另外由于我们在对文本进行预处理的过程中没有进行词干化的操作，所以可以看到在搜索词的最相关词语列表中，出现了单词的单复数形式、名词形式等同词干的变形词，比如 engineering 的最相关词语出现了 engineers 和 engineer，这一点也从侧面体现出了 Word2Vec 在挖掘单词相似关系上的强大能力。

表 4.5 Word2Vec 模型单词相似度计算结果示例

搜索词	最相关词语	最不相关词语
machine	supervised, machines, svms, unsupervised, vector	abundance, observed, occurring, strain, differences
svm	classifiers, classifier, svms, multiclass, unlabeled	ecosystems, plants, responsible, viruses, widespread
algorithm	procedure, method, technique, algorithms, scheme	united, health, thought, labor, sex
mathematics	physics, econometrics, textbook, linguistics, foundations	regulated, upstream, putative, promoter, histone
engineering	engineers, experimentation, development, engineer, chemistry	region, position, locations, orientation, positions
physics	mathematics, mechanics, fields, optics, gravity	expressed, enriched, differentially, profiles, upstream
business	creative, adoption, corporate, managers, industry	pairs, score, experimentally, correctly, inferred
health	care, patient, national, institutes, staff	chain, matrix, linear, matrices, approximate
chemical	macromolecular, kinetics, coupled, macromolecules, molecules	reports, care, patients, trial, participants
psychology	anthropology, economics, sociology, contemporary, philosophy	generates, matches, containing, clusters, seed

在得到了所有单词的向量表示后，我们将使用 WMD 算法来进行文本距离的计算。我们从 CiteULike 数据集中随机挑选了 7 篇涵盖多个领域的论文，各个论文的标题如表 4.6 所示，我们对这些论文两两之间进行了 WMD 距离的计算，得

到的结果如表 4.7 所示。

表 4.6 WMD 算法论文标题示例

分类	ID	题目
SVM	1	the spectrum kernel a string kernel for svm protein classification
	2	recursive svm feature selection and sample classification for massspectrometry and microarray data
基因序列	3	comparative genome sequencing of drosophila pseudoobscura chromosomal gene and ciselement evolution
	4	rfam annotating noncoding rnas in complete genomes
经济	5	information rules a strategic guide to the network economy
	6	the cluetrain manifesto the end of business as usual
	7	small worlds the dynamics of networks between order and randomness princeton studies in complexity

表 4.7 WMD 算法论文距离计算结果示例

ID/ID	1	2	3	4	5	6	7
1	0	**5.84**	8.15	7.77	7.96	7.60	8.07
2		0	7.54	8.09	7.53	7.18	7.42
3			0	**6.27**	8.27	7.90	8.09
4				0	8.26	7.73	8.25
5					0	**4.99**	**5.95**
6						0	**6.07**
7							0

表 4.7 中计算的是两篇论文之间的距离，因此值越小表示论文内容越相似，表中加粗部分是值相对比较小的项，可以看到其对应的都是属于同一分类的两篇论文，说明 WMD 算法能够准确的对文本距离进行判断，将其作为论文相似度计算的方法是非常可靠的。在计算出数据集中所有论文间的相似度后，我们将其结

果整理为三列矩阵形式，每一列分别表示项目 1 的 ID、项目 2 的 ID、两个项目的距离，生成了文件 ii_distance.3，供后续算法处理使用。

4.2.2 推荐算法部分

CiteULike 的原始用户数据是以每个用户收藏的论文 ID 列表的形式给出的，我们首先根据该列表生成了用户关系矩阵，然后根据第 3 章中的公式进行项目相似度的计算，生成 n×n（n 为论文总数）的项目相似度矩阵。然后根据上一步中计算出的文本距离进一步计算出基于内容的相似度矩阵，具体计算方法见 3.3.2 小节。得到两个相似度矩阵后，将使用不同的权重比例将两个矩阵相加得到加权相似度，该相似度将作为最近邻算法的输入。

我们在进行实验时从全部数据集中抽取了 20%作为测试数据，共分了 5 组训练和测试数据，所有实验结果都是各组测试平均后的结果。另外，我们还将用户按照收藏论文数的不同进行了分割，分组包括 0 到 14、15 到 30、31 到 50、51 到 100、100 以上，分组依据主要考虑到各个组别的用户数量大致相等。针对每组用户同样分了 5 组训练集和测试集，以验证各个算法在不同条件下的表现，具体实验结果见后续 4.3.3 节。

4.3 实验结果与分析

4.3.1 混合模型参数对结果的影响

本节将对混合推荐模型中参数的不同取值进行实验对比，分析模型的表现情况，实验结果的好坏使用 recall 值的大小进行衡量，所用数据为全部数据集。混合模型的参数主要包括相似度加权系数 λ 和最近邻数量 K,λ 的取值范围为 0 到 1 之间，K 的取值范围根据经验取 20 到 200。我们在实验中发现推荐数量 N 也会对结果造成一定的影响，但它不属于模型参数，因此不在本节的讨论范围内，其具体的对比情况将在下一节进行分析，本节的实验中将取 N 为固定值 100。实验结果如图 4.4、图 4.5 所示。

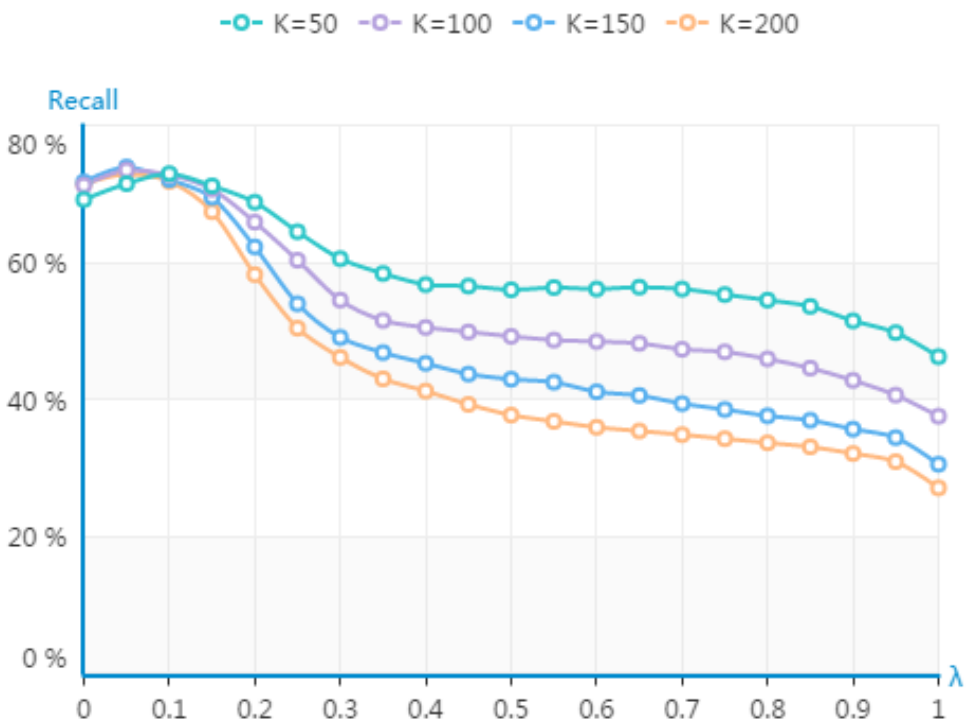

图 4.4 混合参数 λ 对结果的影响

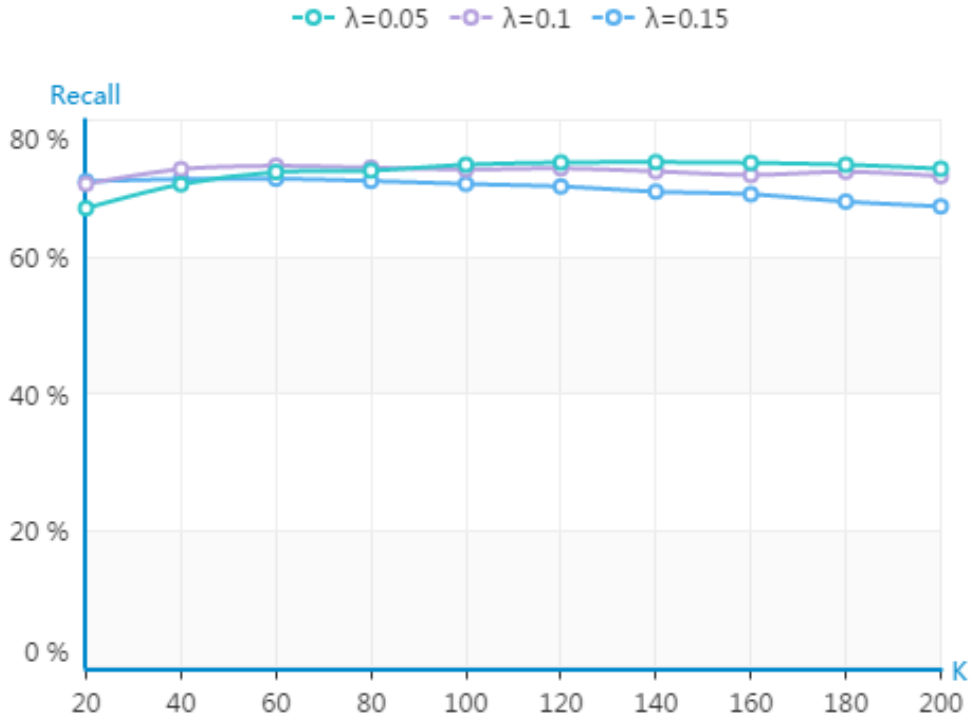

图 4.5 混合参数 K 对结果的影响

从图 4.4 中可以看出，在 K 的不同取值下，当 λ 的取值为 0.05 到 0.1 左右时 recall 值最高。这个结果说明基于词向量的相似度计算会对基于协同过滤的推荐结果带来一定的提升，但主要还是协同过滤相似度在起作用，其原因可能在于我们所使用的数据集中用户行为比较丰富，所以得到的相似度是非常权威的。但实际应用中的数据并非都是如此，往往会出现用户行为较少的情况，因此我们将在 4.3.3 节讨论用户收藏数对实验结果的影响。

从图 4.5 中可以看出，当 λ 在最优范围内取值时，K 的值对实验结果的影响总体差别不大，大概在 80 到 140 之间时表现较好。该实验结果说明选取更多的相似项目更能对目标项目的特征进行完整的表示，但当相似项目的数量足够多时，由于包含了较多弱相关的项目，其数量的提升将不会对结果带来太大的影响，因此 K 的取值不需要太大。

为了进一步精确地确定 λ 和 K 的具体取值，我们又对两个参数进行了更精确的混合调参，λ 的取值在 0.02 到 0.08 之间，步长为 0.01，K 的取值在 60 到 160 之间，步长为 20，部分结果如表 4.8 所示。结果显示在 λ 为 0.06，K 为 140 时达到最大值，我们在后续的实验中将使用这一参数结果。

表 4.8 混合模型参数对结果的影响

λ	K	Recall
0.05	120	0.7380
0.05	140	0.7385
0.05	160	0.7374
0.06	120	0.7380
0.06	140	0.7407
0.06	160	0.7358
0.07	120	0.7385
0.07	140	0.7369
0.07	160	0.7353

4.3.2 推荐数量对结果的影响

本节将对推荐结果数量 N 的不同取值进行实验，分析各个算法的准确率和召回率，对比结果如图 4.6 和图 4.7 所示。

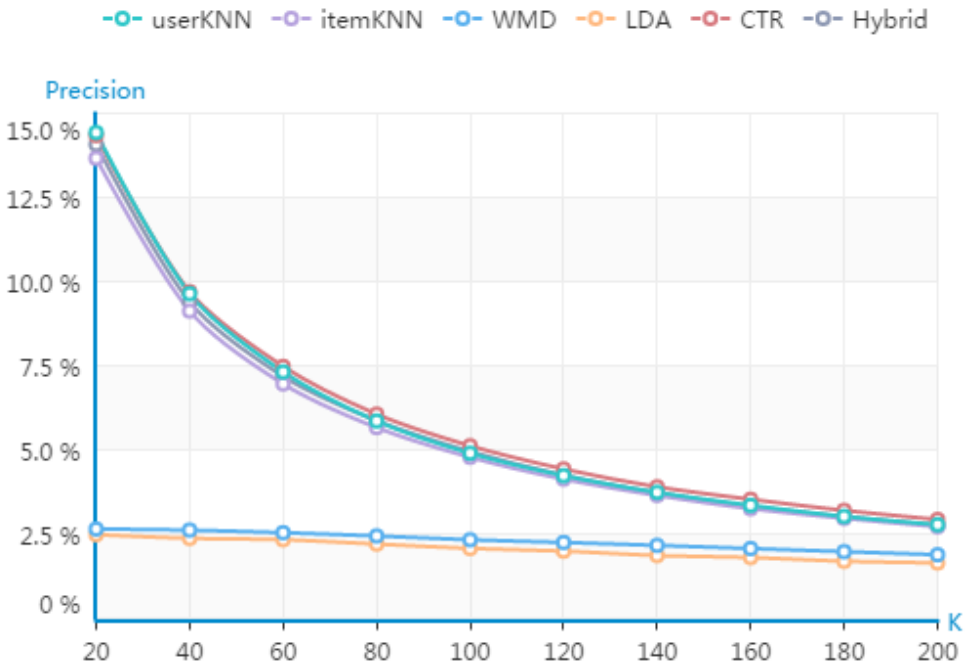

图 4.6 推荐数量不同时各算法的 precision 对比

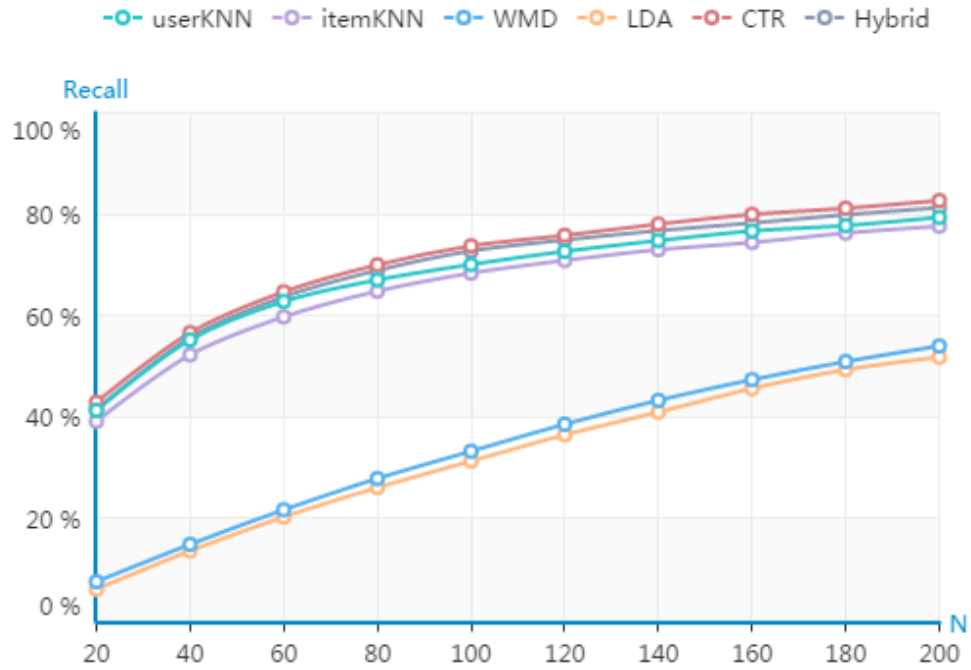

图 4.7 推荐数量不同时各算法的 recall 对比

由以上对比图可以看出，各个算法的 recall 值均随着 N 的增大而增大，这是因为随着推荐数量的增加，结果集中包含的真值数量也在增加。另外，precision 值是随着 N 的增大而减小的，并且绝对数值总体比较低。其主要原因在于数据集的稀疏性非常大，测试集中有大量值为 0（即用户未收藏论文）的项，每个用户平均收藏的论文数比较小，因此随着 K 的增大会有越来越多的无关项出现在结果集中。

在各个算法的总体表现上，几种基于协同过滤的算法明显比基于内容特征的算法表现更好，也证明了在推荐中基于内容算法的局限性。两种基于内容的算法中，WMD 的推荐效果比 LDA 更好，说明词向量比主题模型更能准确的对文本内容相似度进行判断。另外，两种基本协同过滤算法 userKNN 和 itemKNN 的结果差别不大，userKNN 略好一点，原因可能在于数据集过滤掉了收藏论文数低于 10 的用户，所以用户相似度的计算效果更好。本文提出的混合算法是以 itemKNN 为基础的，从对比中可以看出其对 itemKNN 的结果有了较大提升，并且也比 userKNN 结果更好，但相比 CTR 算法略差一点。CTR 算法也是一种混合推荐算法，结合了 LDA 与协同过滤算法，可以看到它比 LDA 和协同过滤算法效果都好。

4.3.3 用户收藏数对结果的影响

为了进一步验证各个算法在不同情况下的表现，我们根据每个用户拥有论文数量的不同，分割了多组训练集和测试集，分别测试了论文数较多的用户和论文数较少的用户，图 4.8 展示了用户收藏数不同时各个算法的表现。

从图中可以看出，由于协同过滤算法主要依赖用户历史数据来进行相似度的分析，所以当用户收藏的论文数比较多时，算法能够提取出更加准确的用户特征，所以得到了较好的推荐效果。而在论文数比较少时，单一的协同过滤算法的效果则会变差。对于基于内容的推荐算法来说，整体表现比较平均，都不是特别好。而本文提出的混合算法以及 CTR 算法由于综合了协同过滤与基于内容算法的优势，在各个情况下都能有相对较好的表现，更适合于应用在实际系统中。

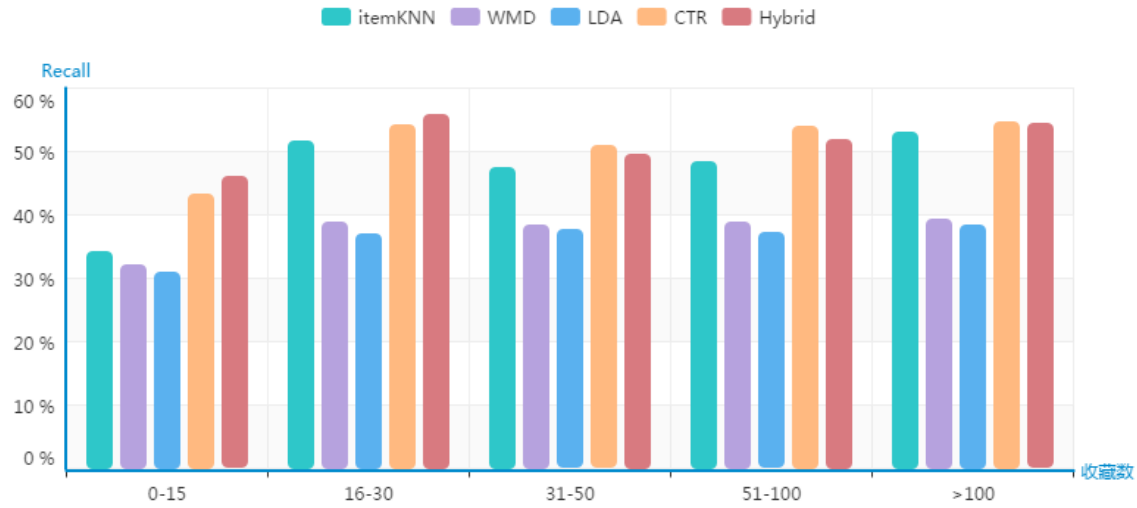

图 4.8 用户收藏数量对结果的影响

4.3.4 推荐算法性能分析

在实验过程中，我们记录了各个算法在进行推荐任务时花费的时间。我们使用的对比算法在计算复杂度上总体来说可以分成三类，第一类是基于协同过滤的两种算法 itemKNN 和 userKNN，第二类是使用主题模型的 LDA 算法和 CTR 算法，第三类是使用了词向量模型的 WMD 算法和混合推荐算法。

在模型训练阶段，我们使用相同的计算机配置，对 CiteULike 数据集包含的 16980 篇文章，共计 7826770 个单词训练模型。实验结果表明，训练 100 维的 Word2Vec 模型用了 10.8 秒，而训练同样维度的 LDA 模型则用了 88.2 秒，说明 Word2Vec 模型在大规模数据集上的模型训练效率比 LDA 更好。

在论文推荐的阶段，各类算法的用时差别较大。基于协同过滤的算法由于结构较简单，获得推荐结果的时间是最短的，整个过程用时约 30 分钟。而基于主题模型的算法在进行推荐时，主要的时间花费在了计算各个文本在主题空间的向量表示上，在得到了文本的向量表示以后，相似度的计算就变成计算两个向量之间的距离，整个计算过程耗时约 3 小时。最后用基于词向量的 WMD 算法来计算文档两两之间的相似度时，由于需要对任意两个文档在词向量模型中单独计算相似度，由此带来了较大的计算开销，整个过程持续了大约 30 小时。

上述结果表明，基于词向量的 WMD 算法和混合推荐算法由于使用了较复杂

的模型来计算文档间的距离，虽然在推荐结果的质量上比基于主题模型的方法好，但在计算量上远高于后者。针对混合算法在计算量上带来的挑战，在实际系统中，推荐算法模块的实现将采用后台离线并行计算的方式来进行。

4.4 本章小结

本章将多种推荐算法进行了对比实验，介绍了实验配置、实验步骤、实验结果，并且分析了在不同情况下各个算法的表现情况，对各个算法的优劣性进行了分析，验证了本文提出的混合算法的效果。

第5章 系统功能设计与实现

本章为系统功能的设计实现部分，我们在基金委开放获取系统中实现了论文推荐与专家推荐功能，另外本章还包括了论文成果录入的相关功能。本章对上述功能的架构设计与实现流程进行了详细介绍，并展示了最后的实现效果。

5.1 功能描述

OA 系统中包含了从 Scopus、CNKI、PDF 元数据等多个数据源中获取的大量论文信息和专家信息，系统的每个登录用户都是专家或学者，将拥有自己的论文成果列表，系统的主要目的是为他们提供科研成果的管理，以及科研论文的获取功能。其中，推荐功能主要包括论文推荐和专家推荐两部分，将为用户提供便捷的科研论文发现和专家发现方法，促进学术成果的交流。

由于 OA 系统中暂时没有用户收藏功能，没有足够的信息来反应用户对论文的喜好情况，因此论文推荐主要将使用在论文详情页面中，根据当前论文来推荐若干篇相关论文。论文详情页面上主要展示了论文的相关信息，包括中英文标题、作者、期刊、发表时间、摘要等信息。在论文信息的旁边，是文章作者列表，以及文章所属领域的热门学者。我们将在这个页面的基础上，增加论文推荐的相关内容。这部分的推荐与登陆的用户并没有关系，主要是根据论文内容的相关性进行计算，因此在论文库没有更新的情况下，同一篇论文的推荐结果是不变的。而当有新的论文加入到数据库中时，论文的相似度将重新进行计算排序，以及时对推荐结果进行更新。

专家推荐是在论文推荐的基础上进行的，系统中关于专家的个人信息，尤其是研究方向信息并不丰富，仅根据这些有限的信息无法对专家的个人特征进行很好的概括。但是每个专家都会有自己的论文列表，根据不同专家之间论文的相似情况，可以对专家的相似度进行估计，从而为每个人推荐相似的专家。这部分功能主要实现在专家个人主页中，主页上一般展示专家的个人信息、论文及项目成

果列表、合作学者等信息。我们将在该页面的基础上添加专家推荐，实现专家的相关学者功能。

除论文推荐和专家推荐功能外，本文工作还包括 OA 系统的成果录入模块，这部分功能帮助用户将自己的论文成果录入到系统中。成果录入的方式一般分为两种，手动成果录入和在线成果录入。手动成果录入指用户自己填写论文的相关信息，系统将这些信息添加到数据库中。在线成果录入指根据用户指定的条件在数据源中搜索相关论文，目前支持的数据源为 CNKI 中国知网，用户可直接在列表中选择属于自己的论文，系统将自动读取论文信息并将其添加到系统中。成果录入模块为用户提供了便捷的论文添加功能，使用户能够更方便的对自己的成果进行管理。

5.2 详细设计

5.2.1 总体架构

本文实现的功能是依托于基金委开放获取系统的，与本文工作相关的部分系统架构如图 5.1 所示，其中深色部分为本文工作内容，主要包括了推荐算法部分、成果录入部分，以及所对应的前端页面。

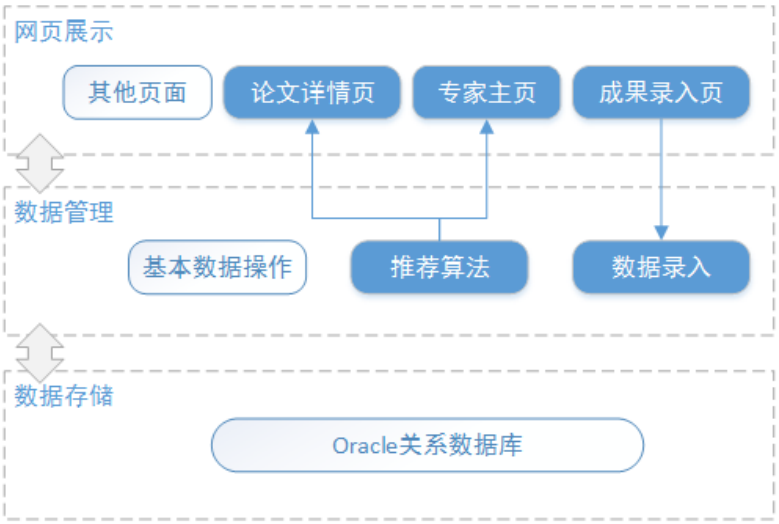

图 5.1 系统总体架构图

其中，推荐算法部分将直接从后台读取数据进行离线计算，在用户打开相关页面时系统将读取推荐结果并显示给用户。成果录入部分主要从前台读取用户输入的论文信息，然后将相关数据存入数据库，其中在线成果还包括了从数据源爬取元数据的步骤，具体功能将在后续章节进行介绍。

5.2.2 各模块介绍

5.2.2.1 论文推荐模块

本文中提出了一种依赖于论文内容及用户行为的混合推荐方法，但由于 OA 系统中用户数据的缺乏，无法完整的实现该方法，因此功能的实现主要使用了基于内容的推荐形式。对于每一篇论文来说，系统将使用 WMD 方法计算它与其他论文之间的相似度，并根据相似度排序得到相关论文的推荐结果。

从 4.3.4 小节“推荐算法性能分析”中我们知道，由于使用了较新的基于词向量的 WMD 算法，每对论文之间的相似度都要放到词向量模型中进行计算，因此带来了极大的计算量需求。为了应对这一挑战，我们将系统的推荐算法模块实现成离线并行计算结构。考虑到系统每天新增的论文数量是有限的，所以该结构能够很好的解决算法带来的计算力挑战。

推荐算法模块的实现使用了 Python 语言，itemKNN 算法使用了自己的实现，Word2Vec 和 WMD 算法使用了较成熟的主题模型库 gensim。由于 Python 语言的 GIL（Global Interpreter Lock）[1]限制，计算量密集的多线程任务在共享内存的时候会遇到瓶颈[2]，所以我们将新论文和已有论文计算相似度的任务通过新建进程来计算，以此来突破 GIL 锁的限制，提高算法模块的吞吐率。推荐模块的计算流程如图 5.2 所示。

[1] https://wiki.python.org/moin/GlobalInterpreterLock
[2] http://python-notes.curiousefficiency.org/en/latest/python3/multicore_python.html

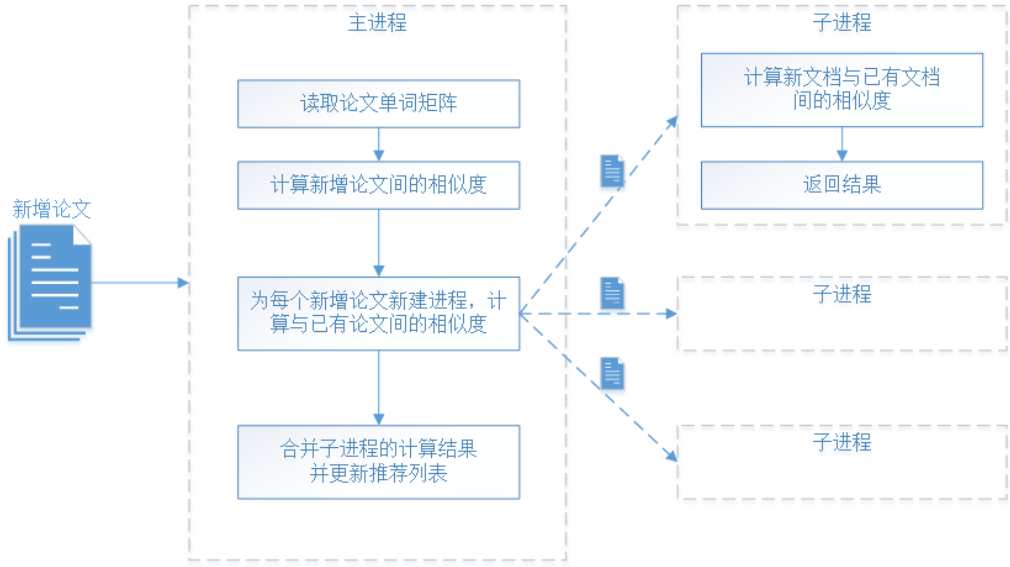

图 5.2 论文推荐模块计算流程

在有了论文之间的相似度值之后，我们将每篇论文选取相似度最高的 5 篇，显示在论文页面中。由于系统中论文数量庞大，对相似度的排序过程也将耗费大量资源。因此系统将在离线计算时保存每篇论文的相似度排序情况，在系统论文更新时对排序列表一并更新。

5.2.2.2 专家推荐模块

专家推荐模块以目标专家为基准，推荐与其相似的若干专家。具体实现中，该模块使用了第四章中提出的计算方法，在论文推荐模块得到的论文相似度基础上进行计算。由于系统中专家数量较多，所有专家两两进行相似度计算耗时较大，实现中将根据专家所属的分类信息，在相同的分类下进行相关专家的推荐。

考虑到专家发表的论文类型在一段时间内都会是相对固定的，所以没有必要每次一有新的论文加入就更新该专家和其他专家的相似度。如果每次有新的论文加入就重新计算专家之间的相似度的话，会使推荐系统的负担极大增加，而带来的收益却并不明显。在综合考虑了推荐效果和系统负载的基础上，我们在设计这部分实现的时候，使用了比上一节的更新论文相似度更低的频率来更新专家相似度。当更新专家相似度的任务周期到来时，推荐系统在完成上一节中提到的论文

相似度的计算后，会接着调用计算专家相似度的模块来重新计算该专家和同一分类下的其他专家的相似度，并更新涉及到的专家的推荐列表。最终的效果将在 5.3 节中展示。

5.2.2.3 成果录入模块

成果录入模块的功能流程如图 5.3 所示，该模块分为手动成果录入和在线成果录入两部分。其中手动成果录入指用户自行填写论文信息，系统直接将用户输入的数据存入数据库。在线成果录入指系统根据用户给定的搜索条件，从在线数据源爬取论文信息，进行解析后自动添加至系统中。

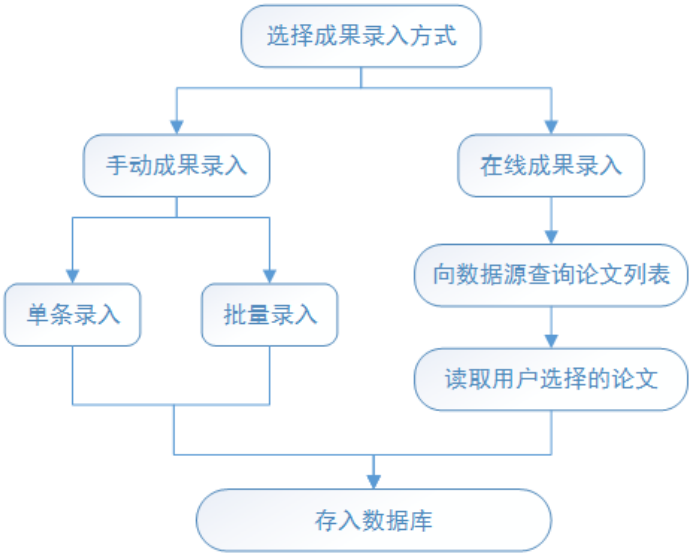

图 5.3 论文成果录入流程

手动成果录入部分又进一步分为单条录入和批量录入，单条录入中用户一次只能添加一篇论文，需要输入论文相关的详细信息，批量录入中用户可同时添加多篇论文，且只需输入论文最基本的信息。系统将对用户输入的内容格式进行判断，若用户输入了其他合作作者的邮箱，系统将查找该邮箱是否为已存在学者，若是则为这些合作作者添加论文关联。

在线成果录入部分主要使用的数据源为中国知网 CNKI[1]，由于 CNKI 没有提

[1] http://www.cnki.net/

供相应的 API 来进行检索操作，所以我们参考了爬虫的思想来请求搜索关键字的索引页面。索引页面是搜索结果的列表展示，包含了搜索结果论文的标题、作者等基本信息。检索请求相关的查询条件如表 5.1 所示。

表 5.1 在线成果录入检索查询条件

请求地址	
http://epub.cnki.net/kns/request/SearchHandler.ashx	
请求关键参数	对应含义
txt_1_value1	论文标题
au_1_value1	作者姓名
au_1_value2	作者所属机构/单位
publishdate_from / publishdate_to	发表日期的起始时间/终止时间
DisplayMode	是否显示论文的摘要信息

得到索引页面后，我们需要对其内容进行解析。我们发现页面中的元数据信息都位于 class 为 GridTableContent 的 table 标签中，后面针对论文信息相对于该标签的 tbody/tr/td/div/ul 位置罗列规则。我们主要提取其中的题目、作者列表、期刊、发表时间以及摘要信息，相关规则如表 5.2 所示。

表 5.2 在线成果录入索引页面解析规则

目标内容	对应的 XPath 规则
题目	li/div/div[2]/div[1]/h3/a
作者列表	li/div/div[2]/div[2]/div[1]/a[1].KnowledgeNetLink
期刊	li/div/div[2]/div[2]/div[1]/a[4]
发表时间	li/div/div[2]/div[2]/div[2]/label[3]
摘要信息	li/div/div[2]/div[2]/script

索引页面中仅包含了论文的基本信息，我们将这些信息解析后的结果显示给用户，供用户判断这些论文是否属于自己。但是上述的索引页面中并不包含论文

的关键词信息以及作者的机构信息，而这些都是对论文非常重要的属性，因此在用户选择了要导入的论文后，我们将进一步解析 CNKI 的论文详情页面来获取这两个信息，相关规则如所示。

表 5.3 在线成果录入详情页面解析规则

目标内容	对应的 XPath 规则
关键字	//*[@id="ChDivKeyWord"]
作者机构	//*[@id="content"]/div[1]/div[3]/div[2]/p[2]

在得到上述的所有信息以后，我们便得到了需要的论文元数据信息。由于可能存在录入的论文是系统中已有论文的情况，我们需要在正式将论文录入系统数据库之前进行查重的步骤，以避免重复录入情况的出现。通过比较待录入的论文的题目和作者列表信息与在库论文的对应字段之间的编辑距离（Edit Distance），我们规定当两篇论文的相似度超过 90%时，我们认为这是重复的论文，将不实际执行录入操作，只是将已有的这篇论文归入作者的论文集合中。

5.3 效果展示

5.3.1 论文推荐部分

如图 5.4 所示为论文“Mining association rules in incomplete information systems”的详情页面，左侧主体部分展示了论文的作者、关键字、发表时间、摘要、相关项目等基本信息，右侧分别是论文发表刊物、合作学者、同领域热门学者的链接。在左侧论文基本信息下面就是针对这篇论文做出推荐的相关论文列表，默认显示了其中的 5 篇，点击右下角的更多按钮可以显示最多 10 篇的论文标题，当鼠标移至论文标题上时可以看到该论文的一些基本信息，包括标题、期刊、发表时间和关键词，效果如图 5.5 所示，点击论文标题则可以进入对应的论文详情页面。

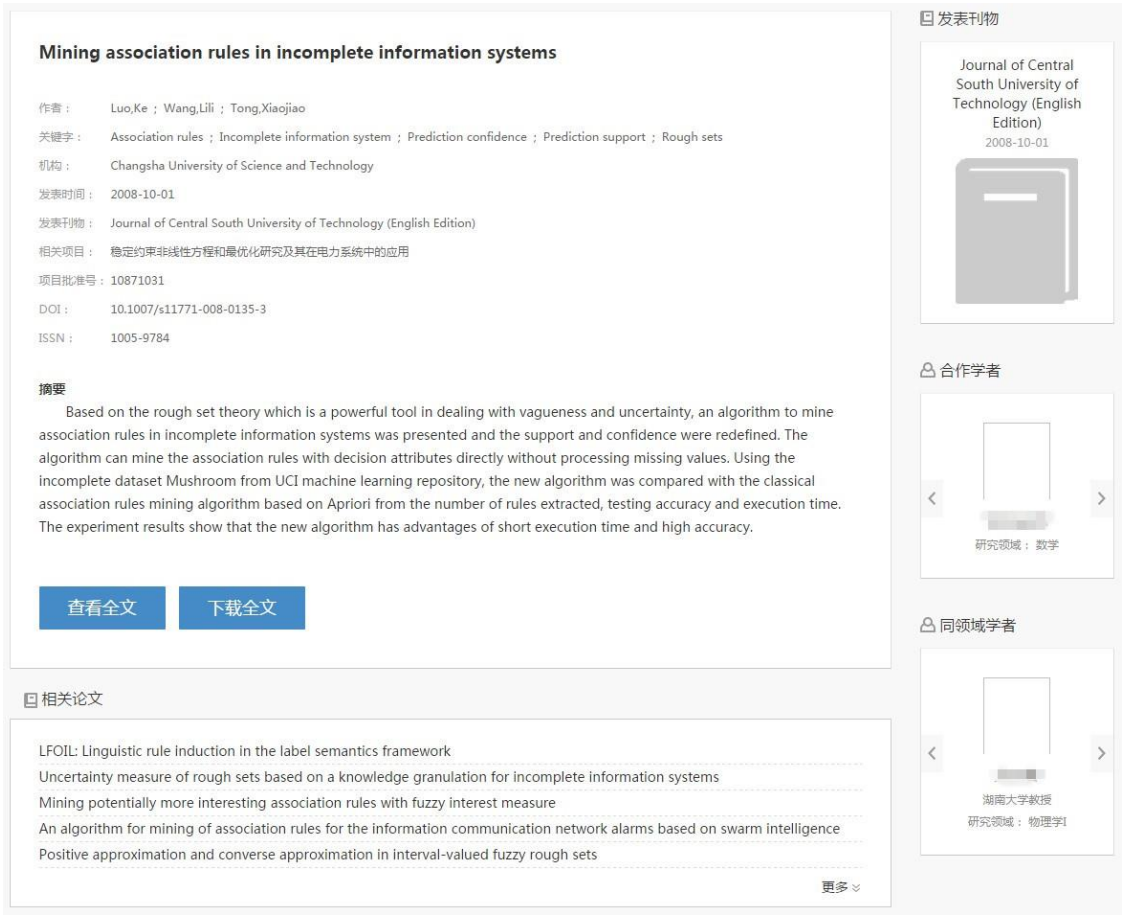

图 5.4 论文推荐整体效果展示

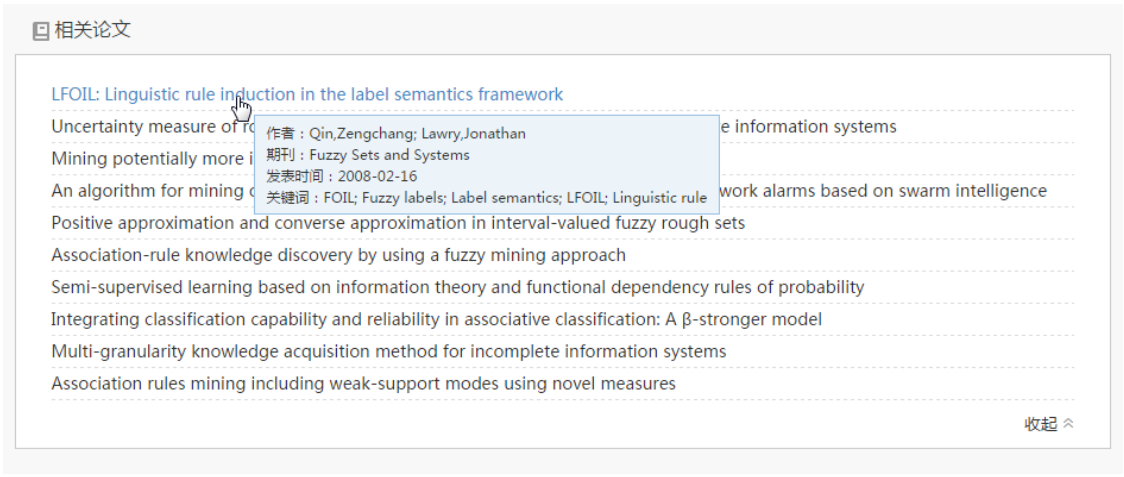

图 5.5 论文推荐具体效果展示

从上图示例论文的信息可以看到，其主要内容是与信息系统中的数据挖掘和

关联规则算法相关的，而推荐出来的结果也都是和机器学习相关的文章。在推荐的 10 篇结果中，有 4 篇文章明确的提到了关联规则算法，说明利用 WMD 算法能够推荐出和主旨相关的论文。另外的 6 篇论文虽然没有直接出现关联规则这样的词语，但是从标题可以看出，涉及的是包括了标签语义、机器学习算法、集群智能、信息通信网络这样的相关领域，说明我们提出的算法并不会局限于某几个权重大的词来进行推荐，这就为专家们发现更多的研究方向提供了可能。综上所述，我们提出的推荐算法达到了预期的推荐效果。

5.3.2 专家推荐部分

如图 5.6 所示为某个专家的个人主页，主要展示了专家的个人基本信息、学术信息、教育信息等。左侧导航栏中包括了三个标签，其中个人成果标签展示了专家的论文成果和项目成果页面，而相关专家标签就是我们实现的专家推荐功能页面，如图 5.7 所示。相关专家页面中以列表的形式展示了专家推荐结果，列表中展示出了专家的头像、姓名、研究领域信息，点击专家的头像可进入相应的个人主页。

图 5.6 专家推荐个人信息页面

图 5.7 专家推荐相关专家页面

我们的例子中所选取的专家研究领域为金属科学，可以看到根据他的研究成果推荐出来的相关专家，研究领域多为金属科学、物理学 I、无机非金属材料、有机化学等，推荐的结果并不局限于专家所在的金属科学领域，通过个人论文集合来进行内容相关性的匹配，能够发现专家目前研究的内容在其他领域的可能性，对扩展专家今后的研究方向有很重要的现实意义。

5.3.3 成果录入部分

手动成果录入分为单条录入和批量录入两种模式。单条录入部分如图 5.8 所示，用户可选择要添加的是期刊论文还是会议论文，两种论文的内容有所差别。然后用户可填写论文的详细信息，其中星号部分为所需必要信息，其他部分可选填，同时用户可选择上传指定格式的论文全文数据。另外用户还可以填写论文作者的详细信息，系统将根据作者邮箱寻找是否是系统中已存在的学者，如果是则为作者与该论文添加关联关系。

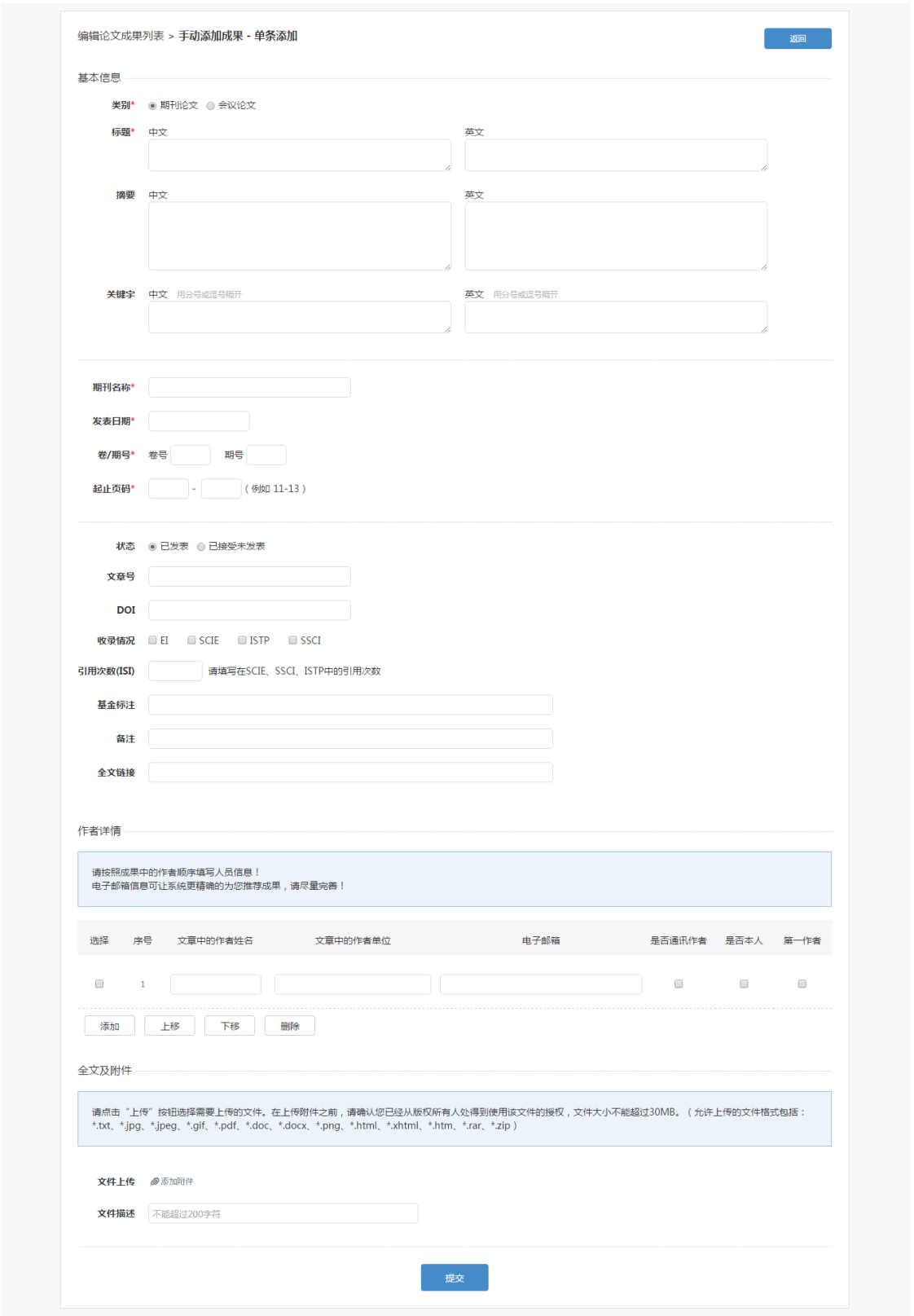

图 5.8 手动成果录入单条添加页面

批量论文成果录入功能如图 5.9 所示，用户只需要填写论文必要的最基本信息，如标题、作者名、期刊名、发表日期等，用户可同时添加多条数据，包括期刊论文和会议论文两种类型。由于作者列表仅有姓名而没有邮箱信息，无法与系统中的专家完全匹配，因此这里将仅为当前用户添加论文关联。

图 5.9 手动成果录入批量添加页面

图 5.10 为在线成果录入部分的检索页面，用户可选择填写若干搜索条件，系统将会把外部数据源中检索到的论文信息以列表的形式展示给用户，如图 5.11 所示，用户可以选择属于自己的论文导入系统。

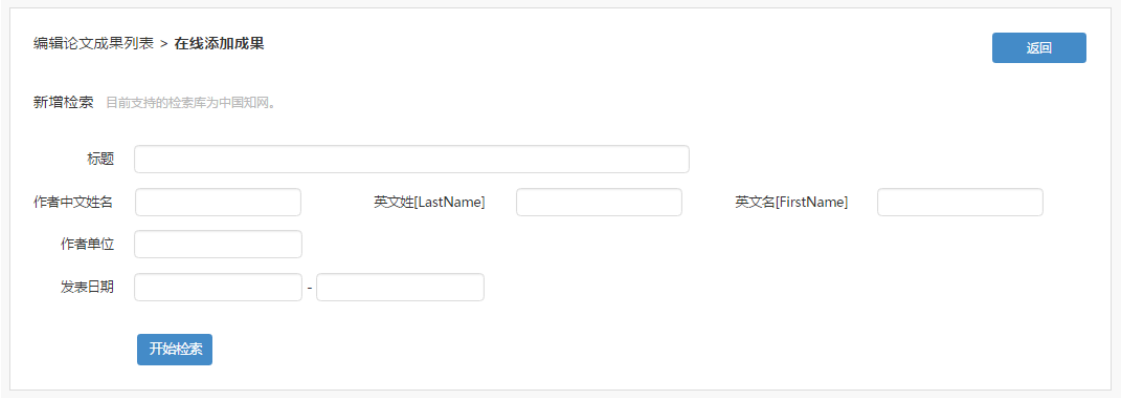

图 5.10 在线成果录入检索页面

图 5.11 在线成果搜索结果

5.4 本章小结

本章中我们将论文推荐和专家推荐的功能在 OA 系统中进行了实现，同时介绍了对论文成果录入功能的实现，详细描述了功能的架构设计以及各模块的具体内容，最后展示了各个模块的效果，对推荐算法的实用性进行了验证。

第6章 总结与展望

6.1 本文工作总结

科研文献的开放获取已经成为世界科技领域的新潮流，广大科研工作者们也在越来越习惯这种搜索学术信息的方式。如何在论文数量呈现爆炸式增长的情况下，为用户提供方便快捷的论文获取方式，成为了新的科研环境下开放获取系统面临的巨大挑战。论文推荐和专家推荐功能的引入能够为用户的信息获取提供新的途径，但如何在海量专业数据中为每一个用户提供个性化推荐，也是学术研究和工业界都亟待解决的问题。

由于论文包含丰富的文本信息，传统的推荐方法多是以基于内容的为主，而随着开放获取系统的发展，用户行为数据的加入必将大幅增强用户偏好的预测结果，也是未来个性化推荐的重点方向所在。将二者结合的混合推荐方法一方面保留了对于文本内容信息的考量，另一方面也能够适应开放获取系统未来的发展趋势，这方面的研究是非常有意义的。本文提出了一种混合推荐方法，综合了基于内容的推荐与基于协同过滤的推荐算法。为了验证本文算法的有效性，我们在真实的公开数据集上进行了大量对比实验，结果表明我们的混合算法与基础算法相比，推荐结果有比较明显的提升，并且在用户行为数据量不同的情况下能够有比较稳定的表现，更加适合于实际系统中的应用。

我们在基金委开放获取系统中设计实现了论文推荐和专家推荐功能，其实现是基于本文算法的部分内容。另外，我们还实现了系统中论文成果录入的功能，经测试系统运行效果良好。

概括来讲，本文的主要工作包括：

1. 通过对协同过滤推荐以及基于词向量的相似度计算方法进行了研究和分析，提出了一种将二者相结合的混合推荐方法，为开放存取系统中的用户提供论文推荐功能。并在论文相似度计算结果的基础上，对相关专家推荐算法进行了研究和实现。

2. 在公开数据集上进行了大量实验，将本文算法与参考的一些基本算法进行了对比，并分析了在多种情况下各算法的表现情况，验证了本文算法推荐结果的有效性。
3. 在基金委开放获取系统中设计实现了论文推荐和专家推荐功能、在线和手动添加成果的功能，其中推荐功能使用了本文所研究算法中的部分内容。

除了上述取得的成果之外，由于各种原因的限制，本文的研究工作也存在一些不足之处。首先是算法的有效性验证方面，由于无法在 OA 系统中实现完整的算法，我们仅在公开数据集上进行了实验分析，使得我们无法对混合算法的实用性以及执行效率等问题进行判断和比较，从而也就无法根据系统实际运行情况对算法进行改进。另外在系统实现方面，推荐结果目前仅仅是以列表的形式进行展示，虽然在大多数推荐系统中都是类似的形式，但对于用户来说，如果能以更加丰富直观的方式将推荐结果与目标项目的关系展示出来，将更好的帮助其进行选择，也从一定程度上增加了推荐结果对于用户的说服力。

6.2 未来工作展望

在未来的研究工作中，主要考虑从以下几个方面来对本文工作进行进一步的完善和扩充：

1. 本文的算法中，基于词向量的相似度计算是非常耗时的，虽然实际应用中这部分是离线处理，但也会消耗较多的系统资源，未来可继续对该部分的效率优化问题进行研究探讨。
2. 本文实验验证所使用的数据集中，剔除了论文收藏数量较少的用户，而在真实的系统中，冷启动问题通常是非常严重的，实验对于实际应用情况的模拟还不够全面，未来应该使用更加丰富全面的数据，对算法的效果进行进一步验证和改进。
3. 专家推荐部分目前主要使用了专家的论文列表信息来进行计算，未来若系统用户数量增多，专家拥有了更加完整的个人信息，可利用这些信息

进行更加多样化的推荐。

4. 系统实现方面，随着 OA 系统用户的不断增多，系统将收集到大量的用户行为信息，有了这些信息，便可以将本文中介绍的混合推荐算法应用到系统中，为用户提供个性化定制的论文推荐和专家推荐。

6.3 本章小结

本章是本论文的最后一章，对整篇论文的工作进行了概括总结。突出了本文所做的研究工作，主要包括算法的研究以及系统的实现。也指出了当前工作的不足之处，进行了反思。另外本章还提出了未来科研工作的改进方向，在算法效果以及系统实现方面都还有很大的进步空间。

参考文献

[1] Lops P, De Gemmis M, Semeraro G. Content-based recommender systems: State of the art and trends[M]//Recommender systems handbook. Springer US, 2011: 73-105.

[2] Ramos J. Using tf-idf to determine word relevance in document queries[C]//Proceedings of the first instructional conference on machine learning. 2003.

[3] Mikolov T, Chen K, Corrado G, et al. Efficient estimation of word representations in vector space[J]. arXiv preprint arXiv:1301.3781, 2013.

[4] Su X, Khoshgoftaar T M. A survey of collaborative filtering techniques[J]. Advances in artificial intelligence, 2009, 2009: 4.

[5] Linden G, Smith B, York J. Amazon. com recommendations: Item-to-item collaborative filtering[J]. IEEE Internet computing, 2003, 7(1): 76-80.

[6] Sarwar B, Karypis G, Konstan J, et al. Item-based collaborative filtering recommendation algorithms[C]//Proceedings of the 10th international conference on World Wide Web. ACM, 2001: 285-295.

[7] Breese J S, Heckerman D, Kadie C. Empirical analysis of predictive algorithms for collaborative filtering[C]//Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence. Morgan Kaufmann Publishers Inc., 1998: 43-52.

[8] Karypis G. Evaluation of item-based top-n recommendation algorithms[C]//Proceedings of the tenth international conference on Information and knowledge management. ACM, 2001: 247-254.

[9] Deshpande M, Karypis G. Item-based top-n recommendation algorithms[J]. ACM Transactions on Information Systems (TOIS), 2004, 22(1): 143-177.

[10] Wang C, Blei D M. Collaborative topic modeling for recommending scientific articles[C]//Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2011: 448-456.

[11] Cremonesi P, Koren Y, Turrin R. Performance of recommender algorithms on top-n recommendation tasks[C]//Proceedings of the fourth ACM conference on Recommender systems. ACM, 2010: 39-46.

[12] Gomaa W H, Fahmy A A. A survey of text similarity approaches[J]. International Journal of Computer Applications, 2013, 68(13).

[13] Socher R, Bauer J, Manning C D, et al. Parsing with Compositional Vector Grammars[C]//ACL (1). 2013: 455-465.

[14] Socher R, Perelygin A, Wu J Y, et al. Recursive deep models for semantic compositionality over a sentiment treebank[C]//Proceedings of the conference on empirical methods in natural language processing (EMNLP). 2013, 1631: 1642.

[15] Herlocker J L, Konstan J A, Terveen L G, et al. Evaluating collaborative filtering recommender systems[J]. ACM Transactions on Information Systems (TOIS), 2004, 22(1): 5-53.

[16] Mikolov T, Sutskever I, Chen K, et al. Distributed representations of words and phrases and their compositionality[C]//Advances in neural information processing systems. 2013: 3111-3119.

[17] Kusner M J, Sun Y, Kolkin N I, et al. From word embeddings to document distances[C]//Proceedings of the 32nd International Conference on Machine Learning (ICML 2015). 2015: 957-966.

[18] Zhang J, Tang J, Li J. Expert finding in a social network[C]//International Conference on Database Systems for Advanced Applications. Springer Berlin Heidelberg, 2007: 1066-1069.

[19] Osmanli O N. A singular value decomposition approach for recommendation systems[J]. The graduate school of natureal and applied sciences of middle east technical unversity, 2010: 1-67.

[20] Blei D M, Ng A Y, Jordan M I. Latent dirichlet allocation[J]. Journal of machine Learning research, 2003, 3(Jan): 993-1022.

[21] Chang J, Gerrish S, Wang C, et al. Reading tea leaves: How humans interpret topic models[C]//Advances in neural information processing systems. 2009: 288-296.

[22] Teh Y W, Jordan M I, Beal M J, et al. Sharing clusters among related groups:

Hierarchical Dirichlet processess[C]//Proc. Neural Information Processing Systems (NIPS). 2005.

[23] Salakhutdinov R, Mnih A. Bayesian probabilistic matrix factorization using Markov chain Monte Carlo[C]//Proceedings of the 25th international conference on Machine learning. ACM, 2008: 880-887.

[24] Salakhutdinov R, Mnih A. Probabilistic matrix factorization[C]//NIPS. 2011, 20: 1-8.

[25] Koren Y, Bell R, Volinsky C. Matrix factorization techniques for recommender systems[J]. Computer, 2009, 42(8): 30-37.

[26] Agarwal D, Chen B C. Regression-based latent factor models[C]//Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2009: 19-28.

[27] Yu K, Lafferty J, Zhu S, et al. Large-scale collaborative prediction using a nonparametric random effects model[C]//Proceedings of the 26th Annual International Conference on Machine Learning. ACM, 2009: 1185-1192.

[28] Sarwar B, Karypis G, Konstan J, et al. Analysis of recommendation algorithms for e-commerce[C]//Proceedings of the 2nd ACM conference on Electronic commerce. ACM, 2000: 158-167.

[29] Collobert R, Weston J. A unified architecture for natural language processing: Deep neural networks with multitask learning[C]//Proceedings of the 25th international conference on Machine learning. ACM, 2008: 160-167.

[30] Mnih A, Hinton G. Three new graphical models for statistical language modelling[C]//Proceedings of the 24th international conference on Machine learning. ACM, 2007: 641-648.

[31] Mnih A, Hinton G E. A scalable hierarchical distributed language model[C]//Advances in neural information processing systems. 2009: 1081-1088.

[32] Goldberg D, Nichols D, Oki B M, et al. Using collaborative filtering to weave an information tapestry[J]. Communications of the ACM, 1992, 35(12): 61-70.

[33] Haruechaiyasak C, Damrongrat C. Article recommendation based on a topic model for wikipedia selection for schools[C]//International Conference on Asian

Digital Libraries. Springer Berlin Heidelberg, 2008: 339-342.

[34] Pazzani M J, Billsus D. Content-based recommendation systems[M]//The adaptive web. Springer Berlin Heidelberg, 2007: 325-341.

[35] Pele O, Werman M. A linear time histogram metric for improved sift matching[C]//European conference on computer vision. Springer Berlin Heidelberg, 2008: 495-508.

[36] Pele O, Werman M. Fast and robust earth mover's distances[C]//2009 IEEE 12th International Conference on Computer Vision. IEEE, 2009: 460-467.

[37] Hinton G E, Mcclelland J L, Rumelhart D E. Distributed representations, Parallel distributed processing: explorations in the microstructure of cognition, vol. 1: foundations[J]. 1986.

[38] Hinton G E, Rumelhart D E, Williams R J. Learning internal representations by back-propagating errors[J]. Parallel Distributed Processing: Explorations in the Microstructure of Cognition, 1985, 1.

[39] Elman J L. Finding structure in time[J]. Cognitive science, 1990, 14(2): 179-211.

[40] Bengio Y, Ducharme R, Vincent P, et al. A neural probabilistic language model[J]. journal of machine learning research, 2003, 3(Feb): 1137-1155.

[41] Mikolov T. Language Modeling for Speech Recognition in Czech[D]. Masters thesis, Brno University of Technology, 2007.

[42] Mikolov T, Kopecky J, Burget L, et al. Neural network based language models for highly inflective languages[C]//2009 IEEE International Conference on Acoustics, Speech and Signal Processing. IEEE, 2009: 4725-4728.

[43] Krapivin M, Marchese M. Focused page rank in scientific papers ranking[C]//International Conference on Asian Digital Libraries. Springer Berlin Heidelberg, 2008: 144-153.

[44] Page L, Brin S, Motwani R, et al. The PageRank citation ranking: Bringing order to the web[R]. Stanford InfoLab, 1999.

[45] Sayyadi H, Getoor L. Futurerank: Ranking scientific articles by predicting their future pagerank[C]//Proceedings of the 2009 SIAM International Conference on Data Mining. Society for Industrial and Applied Mathematics, 2009: 533-544.

[46] Sun Y, Giles C L. Popularity weighted ranking for academic digital libraries[C]//European Conference on Information Retrieval. Springer Berlin Heidelberg, 2007: 605-612.

[47] Sugiyama K, Kan M Y. Scholarly paper recommendation via user's recent research interests[C]//Proceedings of the 10th annual joint conference on Digital libraries. ACM, 2010: 29-38.

[48] Earl M. Knowledge management strategies: Toward a taxonomy[J]. Journal of management information systems, 2001, 18(1): 215-233.

[49] Yimam-Seid D, Kobsa A. Expert-finding systems for organizations: Problem and domain analysis and the DEMOIR approach[J]. Journal of Organizational Computing and Electronic Commerce, 2003, 13(1): 1-24.

[50] Kirchhoff L, Stanoevska-Slabeva K, Nicolai T, et al. Using social network analysis to enhance information retrieval systems[J]. 2008.

[51] Fazel-Zarandi M, Devlin H J, Huang Y, et al. Expert recommendation based on social drivers, social network analysis, and semantic data representation[C]//Proceedings of the 2nd International Workshop on Information Heterogeneity and Fusion in Recommender Systems. ACM, 2011: 41-48.

攻读硕士学位期间主要的研究成果

致谢

时光飞逝，两年多的研究生生涯即将结束。回想三年前第一次踏入浙大的校园，对未来的生活充满了期待，希望自己能够把握这短暂的时光，锻炼自己的综合能力。读研以来，我在实验室以的老师及同学们的帮助下，在工程能力以及个人素质方面有了极大提升。

首先要感谢的是导师组的各位老师，陈刚教授、寿黎但教授、陈珂副教授、胡天磊副教授、伍赛副教授。作为学术研究者，各位老师的专业素养以及广博的知识都使我受益匪浅。而作为导师，各位老师在学习和生活上对我的关心和帮助也使我倍感温暖。衷心希望各位老师未来能够工作顺利、身体健康。在这里尤其要感谢陈珂老师，不仅给了我大量工作上的指导，还对我个人未来的发展提出了许多建议和帮助。另外要特别感谢胡天磊老师，在我的工程能力以及项目协作能力方面给予了很多指导。

感谢实验室全体同学，尤其是王铖微师兄，师兄在论文的主题和方法上给了悉心指导，带我走入了学术的殿堂。这里衷心感谢师兄并预祝师兄学业顺利、事业腾飞。实验室的同学们给我的工作、学习和生活上面都增添了很多色彩，预祝大家在未来的人生道路上面都能够过上理想的生活。另外感谢父母这些年来的支持与理解，家人的关爱和支持是自己在前进道路上不竭的动力源泉。在未来的人生道路上，我会更加努力的工作、生活。

再次感谢所有身边的人，祝你们身体健康，前程似锦。

署名

当前日期