目录

基于用户偏好分析的个性化引文推荐研究

李明发
西南大学计算机与信息科学学院 软件学院，重庆 400715

摘要：在互联网大兴起的背景下，导致信息的组织、存储、处理、传播等过程发生了巨大变化。海量数据下人们日益增长的高质量信息需求在不断的促进着信息检索的发展。为探讨如何解决论文作者在写文章时对参考文献信息高效准确检索需求，本文以自然语言处理领域的常用数据集 Association of Computational Linguistics Anthology Network 作为训练数据，以贝叶斯个性化排序和基于 Word2vec 与 Doc2vec 提出的三层图模型为研究主线，结合了机器学习与深度学习方法，提出基于用户偏好的个性化引文推荐算法，以作者信息和论文标题和摘要等信息作为输入，经系统的运算后为论文作者推荐参考文献。实验表明，该方法可以提高系统推荐的质量，得到较好的推荐结果。
关键词：引文推荐；用户偏好；词向量；贝叶斯个性化排序

Research on Personalized Citation Recommendation Based on User Preference Analysis

LI Mingfa
College of Computer & Information Science Software College，Southwest University, Chongqing 400715, PR China

Abstract: With the Development of Internet, great changes have taken place in the organization, storage, processing and dissemination of information. The ever-increasing demand for high-quality information under massive data is constantly promoting the development of information retrieval. In order to explore how to solve the problem of efficient and accurate information retrieval requirements when writing articles, this paper uses the commonly used data set Association of Computational Linguistics Anthology Network in the field of natural language processing as the training data, it will be handled with Bayesian personalized Rankin and a method named three layer graph model that based on Word2vec and Doc2vec. It combines with the method of machine learning and deep learning, which named Personalized Citation Recommendation Based on User Preference Analysis. The author information and the title and abstract of the paper are taken as input. After the system operation based on the paper's method, you will get a paper list as your paper reference. Experimental results show that the proposed method can effectively improve the quality of citation recommendation and get better recommendation results.
Key words: Citation Recommendation; User preference; Word2rec; Bayesian Personalized Ranking

第 1 章 引言

1.1 研究背景及意义

20 世纪 50 年代末，计算机的出现和逐步普及，把信息对整个社会的影响逐步提高到一种绝对重要的地位，信息量，信息传播的速度，信息处理的速度以及应用信息的程度等都以几何级数的方式在增长，人类进入了信息时代。计算机是传递信息的入口，是信息处理的强大工具。而承载信息在计算机之间传递的是互联网。互联的出现，突破了信息传递的时间和空间的限制，距离不再是问题，只要你有网络，无论你在世界的哪一个角落，而在世界另外一头的他依然能“重现”在你的面前。

根据中国互联网络信息中心（CNNIC）在北京发布的第 43 次《中国互联网络发展状况统计报告》指出[1]：截至 2018 年 12 月，我国域名总数为 3792.8 万个。用户量，网站数呈指数增长，如何在如此海量的信息当中找到用户真正需求和有价值的信息，这是我们这个时代面临的最大的问题。另一方面，各类搜索引擎，检索系统等如雨后春笋般出现，对一个检索系统的质量的评价也面临着巨大的挑战。

根据 ESI 数据库统计，在 2007-2017 年 10 年间我国研究人员发表的国际论文总共被引用了 1935 万次，位居全球第 2 位，发表在各个学科最具影响力国际期刊上的论文数目已连续七年排在世界第 2 位[2]。研究者在论文写作的是时候经常会大量的引用其他作者的论文来支持自己的观点，一方面是对原作者脑力劳动成果的尊重，另一方面也有利于作者梳理知识，了解知识的来龙去脉。论文写作是在借鉴前人研究成果的基础上的一种创新活动，而引文就是借鉴前人研究成果的一种方法。然而，随着科学和教育的发展，越来越多的学生投入到科研工作中，论文的产率不断提高，科研工作者们愈加迫切的希望能将其主要时间花费在创造性的活动中，尽量减少非创造性活动时间上的消费，如确定引文。如何在海量的文献资源中找到与作者写作和研究方向相关的成为了巨大的挑战。

搜索是目前最常用的解决方法，这是一种主动获取信息的方式，优点是用户可以自由的进行检索，检索的结果依赖于用户对需求的表达。推荐区别于搜索，是一种被动的方式，优点是可以发掘出用户的潜在需求，缺点是结果依赖于推荐系统模型的准确性，

对于用户的个性化需求难以准确把握。本文研究的目的是将这两种传统的方式结合，通过计算机技术构建用户个性化模型，改进排序算法，结合主动的搜索与被动的推荐，搭建引文推荐系统，力图解决用户从大量的文献信息中找到自己需要的信息的痛点，推动科学研究的进步与发展。

1.2 论文研究内容

本文研究的服务对象主要为学生、老师、研究者及其他有论文写作需求的相关人员。本文通过对引文推荐过程进行展开研究，主要分为推荐算法的理论简介和推荐系统实现两部分。为了使推荐结果更加准确，符号用户的个性化需求，本文结合前人的研究成果，使用现有的主流相关技术和方法，进行如下两方面内容的研究。

（1）引文推荐算法研究：根据系统文档数据进行相关性排序然后推荐。针对传统的推荐结果不纳入用户的个性化以及个性化表示不准确的问题，本文采用 BPR 算法进行用户兴趣建模。针对传统算法难以对文档的上下文以及文档的各类标注信息，充分利用文档之间的基于内容，基于参考文献和基于作者关系的相似信息，本文使用 Doc2vec 和 Word2vec 进行训练构建三层图型。最后融合以上算法，提出多因子融合的个性化推荐方法。

（2）引文推荐原型系统实现：从实际的生活应用场景出发，基于 BS 模式构建原型系统，根据用户在写的目标文档信息进行准确的返回推荐结果。同时为了提升系统的使用场景和后续优化可能，采用模块式的开发，适应更多的使用场景。

1.3 论文组织结构

本文共分为四个章节。具体文章结构安排如下：

第一章为绪论，首先对本文的研究的背景和意义进行大概了解，后对目前国内外的研究现状进行总结，梳理过去研究中的成果与不足，然后提出对前人研究的改进方法作为本文的研究内容。

第二章为相关的理论和技术简介，探讨过去研究中与本文相关的主要算法理论和本文的研究算法及其实现过程。

第三章为具体系统实现，在第二章提出了基于用户偏好分析的个性化引文推荐实现原理，本章则将其运用于实际的应用中，搭建引文推荐系统。主要介绍引文推荐系统构建的部分细节，包含系统总体框架设计、模块设计、数据库设计和系统的评价。

第四章为实验与结果，将会引用目前开源的数据集 AAN 进行实验，介绍实验设置，实验环境，实验结果与评价。

第五章为总结与展望，在对本文基于用户偏好分析的个性化引文推荐算法进行总结，分析其中的优点和不足之处，为后续的研究工作提出改进的思路和方向，推动引文推荐研究的发展。

第 2 章 国内外研究现状

2.1 引文推荐模型研究现状

引文推荐是研究如何向使用者提供引用备选项的问题[3] 。引文推荐涉及到较多的领域，包括预测理论，信息检索，计算机科学，管理科学等学科。和传统的文献推荐有一定的差别，传统的文献推荐基于文献元数据与用户偏好，为用户推荐符合的信息，与目标文档无关，而引文推荐则深入目标稳定的内容信息，推荐符合目标文档引文上下文的文献信息。引文推荐是一种更细致的推荐方式，在构建模型是参考的信息也会更多，如：文档基础标志信息、作者与作者之间的关系网络、文档引用关系网络、引文上下文关系网络等等。引文推荐起源于 20 世纪初，目前来说还是一个相对较新的研究方向。

19 世纪中期，第一篇基于协同过滤算法论文发表，推荐系统的研究步入正轨。主要应用于商品和电影的推荐。2002 年，Resnick P[4] 等将其应用于传统文献推荐。2007 年，Strohman T[5] 等首次为引文推荐作解释，认为区别于传统使用搜索引擎，用户更希望将整篇文档进行查询，而不是较短的查询语句，并基于图模型与文本相似方法进行了研究。2009 年之后，引文推荐得到快速的发展，主题模型，翻译模型，机器学习，深度学习等方法得到运用，在推荐的准确度和可用度有较大的改善。2009 年，Tang J[6] 等首次提出并使用了主题模型方法进行推荐。在之后才过去一年，He Q[7] 等对引文推荐的范围问题首次提出界定，根据上下文的相关性分为了局部引文推荐和全局引文推荐，区别在于前者是查询词是引文上下文组成的句子集合，而后者是目标文档的相关信息如标题，摘要等。并且，他们还试图通过机器学习来适应用户不标记引文的情景下的引文推荐。此后的发展，科研人员不断的在优化，提出翻译模型解决词汇异质性的问题，提出个性化引文推荐更加符合用户的偏好，提出跨语言引文推荐消除不同书写语言之间的推荐问题。

国内的研究相对来说比较的滞后，都是近几年才发展起来，相关的研究人员也比较少。石杰[8]等在研究时，基于多因素按照一定的规则生成引用关系网络图，并赋予一定的权值，后进行聚类推荐。李飞[9] 等提出了一种元路径时效衰减和引用模式划分的推荐方法，对新文献和非权威文献的推荐精度有较大的提高。此外，国内有部分的研究者另

辟蹊径，从用户的角度入手，如陈志涛[10] 等通过借助目前比较火热的深度学习，深度挖掘用户兴趣，以优化推荐结果更加符号用户的需求。下面将对常用的引文推荐算法进行介绍，包括文本相似度、主题模型、翻译模型、协同过滤算法、和混合推荐方法。

文本相似度：可类比检索问题，将目标文档的标题、摘要、引文上下文和用户兴趣等组成查询语句[10]。推荐的问题便转化为查询向量与文档向量的相似度问题。采用文本相似度方法存在的主要问题是查询语句的构建。实验表明，单一使用这一方法的效果并不理想，用户对需求的准确表达出现较大的问题，并且由于文化的差异性，不同作者对同一概念也有不同的表达。此外，文本相似度方法并没有充分考虑到不同文档的权重问题。

主题模型：一般通过使用迭代进行模型的训练，常常用与发掘大量文献资源中的潜在主题，可以对文档进行低纬度描述。在发展的过程中，研究者们解决了传统的主题模型对引用关系的忽略。在局部引文推荐的问题上，主题模型用于提取引文上下文的主题，然后对每一个主题推荐最适合的引文。在全局引文推荐的问题上，主题模型用于提取目标文档的潜在主题，然后推荐相关主题的文献。主题模型的缺点在于其训练过程耗时较长，不利于新文献的进入使得系统可以快速更新模型[10]。

翻译模型：主要的思想是把引文上下文和目标被引用文献看为不同的语言，通过最大似然估计来计算它们之间的翻译概率。其通过翻译的方式决定了它可以用于跨语言引文推荐的问题上，可以解决相同语言和不同语言在词汇表达上的鸿沟。研究表明，基于目标文档的摘要的翻译模型效果优于基于全文的翻译模型[10]。

协同过滤算法：其主要的思想是通过引用关系和合作者关系进行推荐[10]。。推荐的过程其实是建立在作者与合作者在兴趣上有共同点的基础上的。其算法对于较低引用频次的文献不够友好，有较大的局限性，并且由于协同并没有深入引文内容，因此很少被单独的拿出来使用。

混合推荐：不同的方法会有其不同的优缺点和使用场景，单独使用局限性较大，混合推荐的思想主要是结合不同方法的优点，弥补其中的不足，将优点最大化放大，达到更好的推荐效果。这是未来引文推荐研究的一个重要方向。

2.2 引文推荐工具使用现状

如何在大量的学术资源信息中找到与自己研究相关的信息是目前科研人员面临的巨大挑战。目前，科研人员确定引文的过程是：首先在搜索引擎或者是数据库中检索有关学术信息资源，在不断的检索过程中找到对自己有用的信息，然后整理存储，作为备用。常用的搜索引擎有：Google Scholar、SCIRUS 和 CiteSeer 等。常见的期刊论文数据口主要有：Web of Science 、CNKI、万方、维普等综合类数据库以及 Reaxys、BIOSIS Previews 等专业领域数据库。检索的过程对用户的信息素养要求比较高，最充分的体现在于对检索需求的正确表达。

检索之后，科研人员一般会通过一些学术资源管理工具来组织和整理参考文献。常见的工具主要有：LaTeX、ReadCube、NoteExpress、Papers、EndNote 等。以用户量较大的 LaTeX 为例，适合的场景为：科研人员手中已经检索到了合适的学术资源，并且对于筛选出要作为参考文献的部分与自己的研究反复斟酌，一一对应，然后录入到 LaTeX 中方便整理。也就是说这些工具本质上并没有大大减少科研人员在查找资源上的时间和精力，只是起到了一个方便组织管理的作用。而本文研究的引文推荐目的便是从前期的检索过程入手，自动的或半自动的完成引文推荐，返回符合要求的论文列表，缩小研究者的检索范围，使得这项研究非常有意义，引起了诸多学者的关注与讨论。

除了这一类文献资源管理工具外，也出现了一些引文推荐系统，但缺点主要是数据量小，使得推荐的效果比较差，没有被广泛的应用，如 Theadvisor 和 Scienstein。2014 年，CiteSigh 系统框架被 Livne A[12] 等提出来，它通过综合多种现有的推荐方法，借助引文耦合思想解决了低引文献和新文献被选中推荐概率较低的问题。基于结果，结合作者正在写作的目标与候选文档之间相似度的大小进行二次排序，最终选取前 k 篇相似度较大文档作为最终的推荐文档。

第 3 章 基于用户偏好分析的引文推荐理论及技术简介

3.1 Word2vec 模型

Word2vec 是谷歌开源的一个词向量计算工具，是深度学习在自然语言处理研究成果的一个产物，基于神经网络。首先，它的训练效率极高，可以在百万数量级的词典和上亿的数据集上进行高效地训练；其次，该工具得到的训练结果——词向量（Word Embedding），可以很好地度量词与词之间的相似性。词向量有非常好的语义特性，可以表示语言特征。

Word2vec 工具主要包含两个模型：跳字模型（Skip-gram）和连续词袋模型（Continuous Bag of Words，简称 CBOW），CBOW 又称连续词袋模型，包括了两种训练方法：负采样（Negative Nampling）和层序 SoftMax（Hierarchical SoftMax），是一个三层神经网络。CBOW 的目标是：给定一串文本，使用某个词的上下文来预测这个词。例如，对于句子“China is one of four ancient and civilizational countries in the world.”，预测 and 这个词时，可以使用 four、ancient、civilizational、countries 这四个词，它们构成了 and 的上下文。这样，从文本中就可以提取一系列的训练样本。如图 2.1 所示，该模型的特点是输入已知上下文，输出对当前单词的预测。

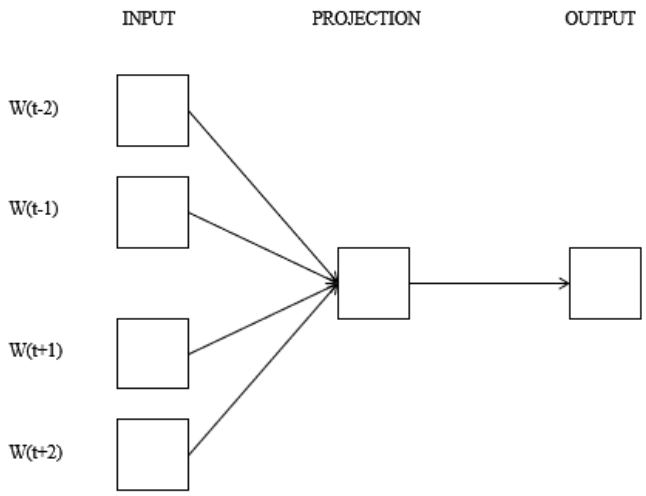

图 2.1 CBOW 模型

Figure 2.1 CBOW Model

Skip-gram 只是逆转了 CBOW 的因果关系而已，即已知当前词语 and，目标是预测 and 的上下文。其上下文词汇的处理与 CBOW 一样，也是将这些词的 One-hot 编码相加，作为训练使用的真值。其模型如图 2.2 所示。

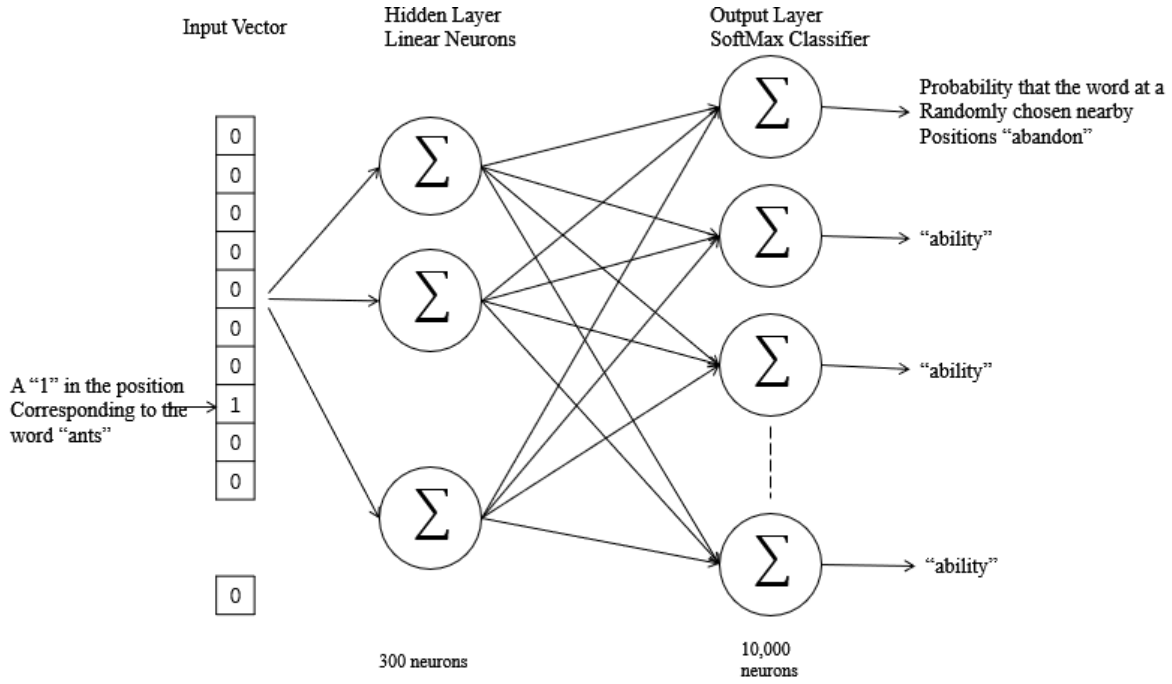

图 2.2 Skip-gram 模型

Figure 2.2 Skip-gram Model

3.2 Doc2vec 模型

Doc2vec 是 Mikolov[13] 在 Word2vec 基础上提出的另一个用于计算长文本向量的工具。它的工作原理与 Word2vec 极为相似，将长文本作为一个特殊的 token id 引入训练语料中。Doc2vec 也称做 Paragraph2vec 和 Sentence Embeddings，是一种非监督式算法，可以获得 Sentences/ Paragraphs/ Documents 的向量表达，是 Word2vec 的拓展，得到向量后可用于计算 Sentences/ Paragraphs/ Documents 之间的相似性，文本聚类，对于有标签的数据，还可以用监督学习的方法进行文本分类。

训练句向量的方法和词向量的方法非常类似。训练词向量的核心思想就是说可以根据每个单词的上下文预测，也就是说上下文的单词对是有影响的。同理，也可用同样的方式来训练 Doc2vec。例如对于一个句子 I want to drink water，如果要去预测句子中的单词 want，那么不仅可以根据其他单词生成 feature， 也可以根据其他单词和句子来生成 feature 进行预测。因此 Doc2vec 的框架如图 2.3 所示：

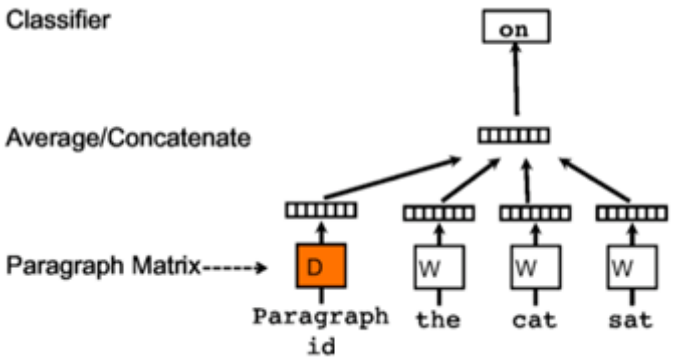

图 2.3 Doc2vec 框架

Figure 2.3 Doc2vec Framework

每个段落、单词或句子都被映射到向量空间中，可以用矩阵的一列来表示。然后将段落向量和词向量级联或者求平均得到其特征，预测句子中的下一个单词。这个段落向量或句向量也可以认为是一个单词，它的作用相当于是上下文的记忆单元或者是这个段落的主题，所以我们一般叫这种训练方法为 Distributed Memory Model of Paragraph Vectors(PV-DM)。在训练的时候我们固定上下文的长度，用滑动窗口的方法产生训练集。此时，段落向量和句向量在该上下文中共享。

Doc2vec 过程分为：训练模型和推断过程。训练模型，是指在已知的训练数据中得到词向量, SoftMax 的参数，以及段落向量或句向量。而推断过程（Inference Stage），是指对于新的段落，得到其向量表达。具体地，在矩阵中添加更多的列，利用上述方法进行训练，使用梯度下降的方法得到新的 D，从而得到新段落的向量表达。

Doc2 还有另外一种训练方法——Paragraph Vector without Word Ordering: Distributed Bag of Words（简称：PV-DBOW）。该方法忽略输入的上下文，让模型去预测段落中的随机一个单词。就是在每次迭代的时候，从文本中采样得到一个窗口，再从这个窗口中随机采样一个单词作为预测任务，让模型去预测，输入就是段落向量。如图 2.4 所示：

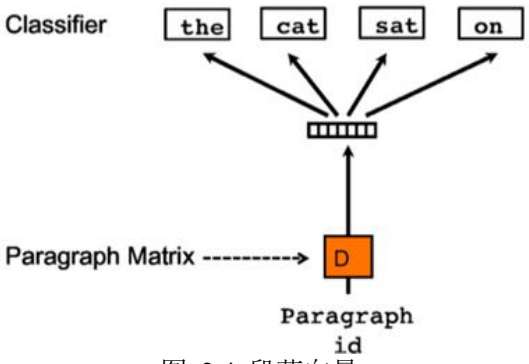

图 2.4 段落向量

Figure 2.4 Paragraph Vector

在上述两种方法中，我们可以使用 PV-DM 或者 PV-DBOW 得到段落向量或句向量。一般来说，PV-DM 的方法表现很好，但也可以将两种方法相结合，获得更好的效果。

3.3 用户偏好建模

在引文推荐的系统中，为了得到系统文档 V 的正确排序，本文采用贝叶斯个性化排序(Bayesian Personalized Ranking, 简称 BPR)算法。在购物的推荐场景中，我们都是基于现有的用户和商品之间的一些数据，得到用户对所有商品的评分，选择高分的商品推荐给用户，这是 Funk-SVD 之类算法的做法，使用起来也很有效。但是在有些推荐场景中，我们是为了在千万级别的商品中推荐个位数的商品给用户，此时，我们更关心的是用户来说，哪些极少数商品在用户心中有更高的优先级，也就是排序更靠前。也就是说，我们需要一个排序算法，这个算法可以把每个用户对应的所有商品按喜好排序。BPR 就是这样的一个我们需要的排序算法，由此我们可以通过 BPR 来构建用户偏好模型（User Preference Model，简称 UP）。

在 BPR 算法中，我们将任意用户 u 对应的文档进行标记，如果用户 u 在同时有文档 i 和 j 的时候点击了 i，那么我们就得到了一个三元组<u,i,j>，它表示对用户 u 来说，i 的排序要比 j 靠前。如果对于用户 u 来说我们有 m 组这样的反馈，那么我们就可以得到 m 组用户 u 对应的训练样本。BPR 算法属于贝叶斯概率，因此本文的 BPR 算法基于两个假设：一是每个用户 u 的偏好行为相互独立，即用在 i 和 j 的选择中与其他的用户偏好无关。二是同一用户 u 对不同文档的偏序相互独立，即用户选择 i 和 j 之间与其他文档无关。

在 BPR 中，这个排序关系符号>u 满足完全性，反对称性和传递性，即对于用户集 U 和物品集 I：

完整性：∀ i,j∈I:i≠j ⇒i>uj∪j>ui

反对称性：∀i,j∈I:i>uj∩j>ui ⇒i=j

传递性：∀i,j,k∈I:i>uj∩j>uk⇒i>uk

同时，BPR 也用了和 Funk-SVD 类似的矩阵分解模型，这里 BPR 对于用户集 U 和文档集 I 的对应的 U×I 的预测排序矩阵$\overline{X}$，我们期望得到两个分解后的用户矩阵 W(|U|×k)和文档矩阵 H(|I|×k)，k 和 Funk-SVD 类似，也是自己定义的，一般远远小于|U|,|I|。满足

$$\bar{X} = WH^T \tag{2-1}$$

由于 BPR 是基于用户维度的，所以对于任意一个用户 u，对应的任意一个文档 i 我们期望有：

$$\bar{x}_{ui} = w_u \cdot h_i = \sum_{f=1}^{k} w_u h_i f \tag{2-2}$$

最终我们的目标，是希望寻找合适的矩阵 W,H，让$\overline{X}$和 X 最相似。BPR 基于最大后验估计 P(W,H|>u)来求解模型参数 W,H,这里我们用θ来表示参数 W 和 H, >u 代表用户 u 对应的所有商品的全序关系,则优化目标是 P(θ|>u)。根据贝叶斯公式，我们有：

$$P(\theta| >_u) = \frac{P(>_u |\theta)P(\theta)}{P(>_u)} \tag{2-3}$$

基于假设一：用户偏好行为独立，可以得到 P(>u)对所有的物品一样，所以有：

$$P(\theta| >_u) \propto P(>_u |\theta)P(\theta) \tag{2-5}$$

式 2-5 中，文档之间的偏序关系>u 为用户 u 对全部文档的潜在偏好，θ矩阵分解模型的参数所构成的向量。$P(\theta)$为先验概率，$P(\theta| > u)$为似然函数。对 BPR 算法的优化分为两个部分，第一部分和样本数据集 D 有关，第二部分和样本数据集 D 无关。对于第一部分，由于我们假设每个用户之间的偏好行为相互独立，同一用户对不同物品的偏序相互独立，所以有：

$$\prod_{u\in U} P(\theta| >_u) = \prod_{(u,i,j)\in(U\times I\times I)} P(\theta| >_u)^{\delta(u,i,j)\in D} (1 - P(i >_u j|\theta))^{\delta(u,i,j)\notin D} \tag{2-6}$$

其中，

$$\delta(\mathrm{b}) = \begin{cases} 1, if b\ is\ true \\ 0, \quad else \end{cases} \tag{2-7}$$

根据上面的完整性和对称性，可以转化为：

$$\prod_{u \in U} P(\theta | >_u) = \prod_{(u,i,j) \in D} P(i >_u j | \theta) = \prod_{(u,i,j) \in D} \sigma(\bar{x}_{uij}(\theta)) \tag{2-8}$$

其中，$\sigma(x)$是 sigmoid 函数，$(\bar{x}_{uij}(\theta))$描述了论文 i、论文 j 和用户 u 之间的关系。

对于第二部分，$\mathrm{P}(\theta)$为先验概率，符合正态分布，且对应的平均值为 0，协方差矩阵为$\lambda_\theta \mathrm{I}$，即

$$\mathrm{P}(\theta) \sim \mathrm{N}(0, \lambda_\theta \mathrm{I}) \tag{2-9}$$

基于以上得到最大后验估计 BPR 模型：

$$\mathrm{BPR-Paper} = \sum_{(u,i,j) \in D_I} \ln \sigma\left(\widehat{x_{uij}}\right) - \lambda_\theta \|\theta\|^2 \tag{2-10}$$

3.4 三层图模型

为了充分的利用作者、文档的本身信息与相互关联信息。本身信息包括：作者姓名、所在组织、文档标题、关键字、摘要、参考文献等，相互关联信息指包括文档与文档在内容上的关联性，文档之间的引用关系、作者之间的合作关系。基于本文提出了三层图模型：由文档内容相关度（Content Relevancy，简称 CR）、文档 ID 相关度（ID Relevancy，简称 PR）、作者相关度（Author Relevancy，简称 AR）所构成的图模型。

3.4.1文档内容相关度表示

为将文档之间在内容上的相关关系程度表达出来，本文提出了文档相关度的概念。通过使用 Doc2vec 进行向量化表示，计算文档的标题与摘要组成的词向量之和求平均来表示一篇文档。再利用距离公式表达相关度信息，求得 CR。

3.4.2文档 ID 相关度表示

一篇文档中会有多个参考文献，文档与参考文献在内容上会存在较大的相关，同时，参考文献的先后顺序也会影响其相关性的大小，例如相对靠前的参考文献其引文较大可能的存在于文章比较前的位置，如摘要中引用的参考文献会表达出与目标文档的相关性较大，其参考文献的位置也会比较靠前。为了充分利用文档与文档的参考文献相关关系，同时为了降低计算量，本文提出文档 ID 相关度（ID Relevancy，简称 PR）概念。

每一篇文档可以分配一个独一无二的 ID，作为区分文档的标识。ID 可类比于一个词，将某篇文档 ID 以及其对应所有的参考文献组成一行，作为一个句子（Sentence），放入 Word2vec 进行训练，得到文档 ID 的向量化表示，进而通过欧式距离计算文档 ID 间的相关度 PR。

由于文档的参考文献可能比较少，一方面可能是因为所选的数据集中所标记的信息不够完整，另一方面是文档本身的参考文献就比较的少。由此会造成所构建的句子较短，预料库较小造成的结果就是训练出来的模型不够准确，因此有必要进行扩充。本文采用 Kimothi D[14] 提出的方法，将序列转化为重叠窗口为 3 的扩充序列。其思路如下：

假设某篇文档的 ID 为 1，其参考文献的 ID 为：2~9，共计 4 篇参考文献。未扩充前，由此文档所构建成的 sentence 为：123456789。设窗口大小为 3，此时可得如下 4 个序列，序列 D 为最后的扩充序列：

序列 A：1234 1567

序列 B：1345 1678

序列 C：1456 1789

序列 D：1234 1567 1345 1678 1456 1789

通过使用非重叠窗口来构建数据集，可以较大的提升数据规模，提升训练效果，得到的词向量也会更为准确，同时可以进一步挖掘文档引用网络中的信息。

3.4.3作相关度表示

每一篇文档一般都会由不少于一名作者书写，也会引用不少于一篇的参考文献，这些作者组成的集体在研究内容上存在一定交集，即研究内容相关。因此可以挖掘作者之间的相关关系推动引文推荐的准确率。

类比于文档 ID 相关度的表示，每一名作者分配一个 ID 值，将一篇文献的作者、协作者与参考文献作者组成一行，作为该文档的作者序列，借助上文提到的非重叠窗口方法来扩充数据集，放入 Word2vec 进行训练，得到作者序列的向量化表示，进而通过欧式距离计算作者相关度 AR。

3.5 多因子融合模型

本文提出的多因子融合模型（Multi-factor Fusion Model，简称 MF）是本文构建的基于用户偏好分析的个性化引文推荐模型的核心算法。多因子融合目的是为了将上文所提到的多个相关度因子进行有效结合，以发挥出多因子的最大化效用，提高引文推荐的准确率。融合的过程首先需要构建用户偏好模型与三层图模型，多因子包含了文档内容相关度因子 CR、文档 ID 相关度因子 PR、作者相关度因子 AR 与用户偏好因子。

本文提出的多因子融合模型（Multi-factor Fusion Model，简称 MF）是本文构建的基于用户偏好分析的个性化引文推荐模型的核心算法。多因子融合目的是为了将上文所提到的多个相关度因子进行有效结合，以发挥出多因子的最大化效用，提高引文推荐的准确率。融合的过程首先需要构建用户偏好模型与三层图模型。如图 2.5 所示，多因子包含了文档内容相关度因子 CR、文档 ID 相关度因子 PR、作者相关度因子 AR 与用户偏好因子。

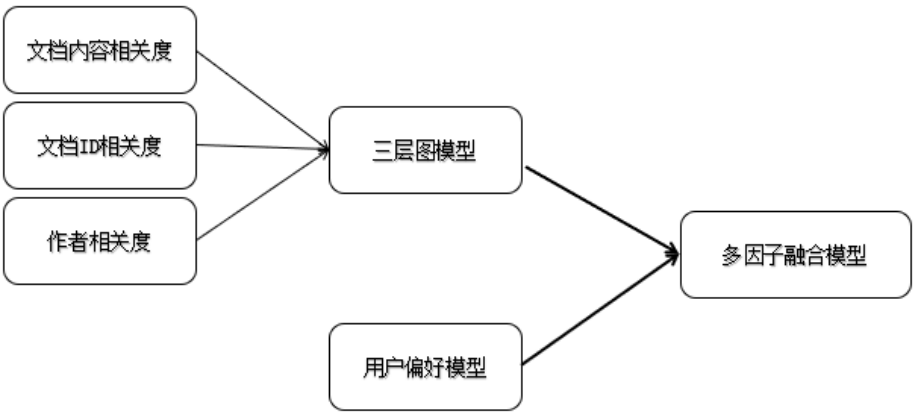

图 2.5 多因子融合模型

Figure 2.5 Multi-factor Fusion Model

在多因子融合的算法中，本文采用线性加权的方式，多因子融合模型计算公式如公式 2-11 所示：

$$\mathrm{MF} = \alpha \cdot \mathrm{CR} + \beta \cdot \mathrm{PR} + \gamma \cdot \mathrm{AR} + \delta \cdot \mathrm{UP} \tag{2-11}$$

第 4 章 实验与结果

4.1 实验设置

为了验证本文提出的算法的有效性，本章节将对算法在 P@N（Precision at N）和 MAP（Mean Average Precision）两个指标来对推荐结果进行评价。并介绍了常用自然语言处理领域的常用数据集 AAN（Association of Computational Linguistics (ACL) Anthology Network，以下简称 AAN）[15] 进行介绍。

4.1.1数据集介绍

本文所有进行的实验与测试均是基于 AAN 数据集所做。AAN 由耶鲁大学 Radev D R[15] 教授领导，为其学校下的 LILY 小组维护，根据 ACL Anthology 提供的原始 pdf 文件构建的，使用开源 OCR 技术，内部清理脚本以及通常繁琐的手工劳动所构建的一个开源语料库。包含了 1965 年到 2018 年共计 36272 篇文献资料。并对文档标注了作者、标题、摘要、年份、所属组织、引文上下文、参考文献等重要文献信息。

本文基于 AAN 网站给予的下载链接获取数据集，并结合该网站展示作者与文章的信息，使用网络爬虫抓取网站上较为规划化的信息，构建成为最终的数据集。数据库总集如表 4.1 所示。

表 4.1 数据统计
Table 4.1 Data Statistics

年份	论文数目	作者数目	被引用关系	引用关系
1965-2013	13030	4865	11541	68247
2013-2016	1313	968	1402	11578

4.1.2实验数据构建

考虑到 AAN 数据集较大，单机无法完成如大量的运算进行模型构建，本文旨在提供引文推荐算法一个可行的思路，故而对数据集进行筛选简化。实验数据分为训练数据（Train Data）和测试数据（Test Data）。构建过程如图 4.6 所示：

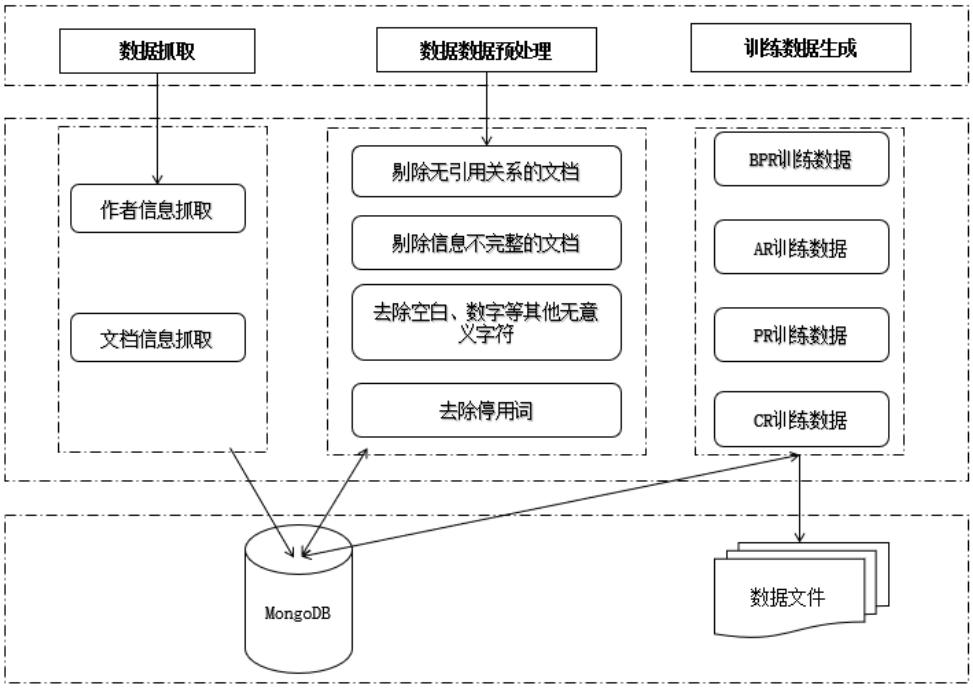

图 4.6 数据处理

Figure 4.6 Data Processing

Test Data :首先从 AAN 数据集中筛选出 2013 以后的发表的文档，剔除标记信息缺失与文档可用信息较少的部分（预处理），进而随机部分文档作为实验的测试文档集合（Testing Doc Corpora），借助爬虫可以抓取测试集对应的所有参考文献纳入训练集(Training Doc Corpora)，对训练集同样需要进行预处理。为获得作者信息，借助爬虫抓取训练集中的作者信息，如姓名、发表的文章和所在机构等信息。由此构建了文档和作者数据信息存储于远程的 MongoDB 数据库服务器中，在此过程之后，为方便将数据库中的信息转化为模型的本地训练数据文件，以文本文件方式进行存取。

经数据处理之后，实验数据集详细情况如表 4.2 所示：

表 4.2 实验数据集

Table 4.2 Experimental Data Corpora

类别	论文数目	引用关系	年份
训练集	10279	103457	1975-2015
测试集	153	410	2012-2014

4.1.3评价指标

为了对推荐结果进行评价，本文采用了信息检索系统中常用的指标准确率 Precision 和召回率 recall 来进行评价。Precision 可以评估推荐结果的准确率，其值越大，效果越

好，是检索结果中相关文档数与返回的总结果数的比值，反映的检索结果查准率。recall 则反映的是检索结果的查全率，是检索结果中相关文档与系统文档总数的比值。

定义：U 为用户集，对于某一个用户 u，X(u)为推荐的结果集,Y(u)为系统中该用户的所有感兴趣的文档集合，即为该 u 下的所有参考文献组成的文档集合。

准确率计算公式如式 4-1 所示。

$$Precision = \frac{\sum_{u \epsilon U} X(u) \cap Y(u)}{\sum_{u \epsilon U} X(u)} \tag{4-1}$$

recall 计算公式如式 4-2 所示：

$$\text{recall} = \frac{\sum_{u \epsilon U} X(u) \cap Y(u)}{\sum_{u \epsilon U} Y(u)} \tag{4-2}$$

4.2 实验结果

实验设置了不同返回长度的推荐文档集合，分析不同返回长度下以及单一模型与多因子融合模型对召回率和准确率的影响。

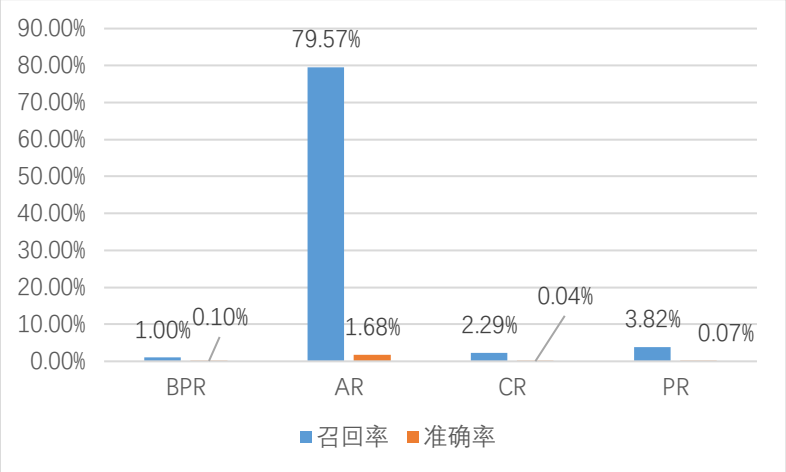

图 4.7 单一模型对比

Figure 4.7 Single Model Comparison

在单一模型下，为控制变量，以下实验的实验返回长度定为 100。由图 4.7 可以看到，AR 算法的召回率效果较好，大幅度高于其他算法，而其他算法在召回率以及准确度上较为相似，对整体的推荐效果都有一定的促进作用。AR 算法召回率结果较高可能原因为数据集本身的原因，由于数据集的限制，AR 算法的核心基于作者 ID 作为训练

数据，一方面可能原因为数据较少，对作者向量的表达存在较大误差，另一方面原因可能来源于测试数据集不具有普遍性，其相关度远高于普通数据集。

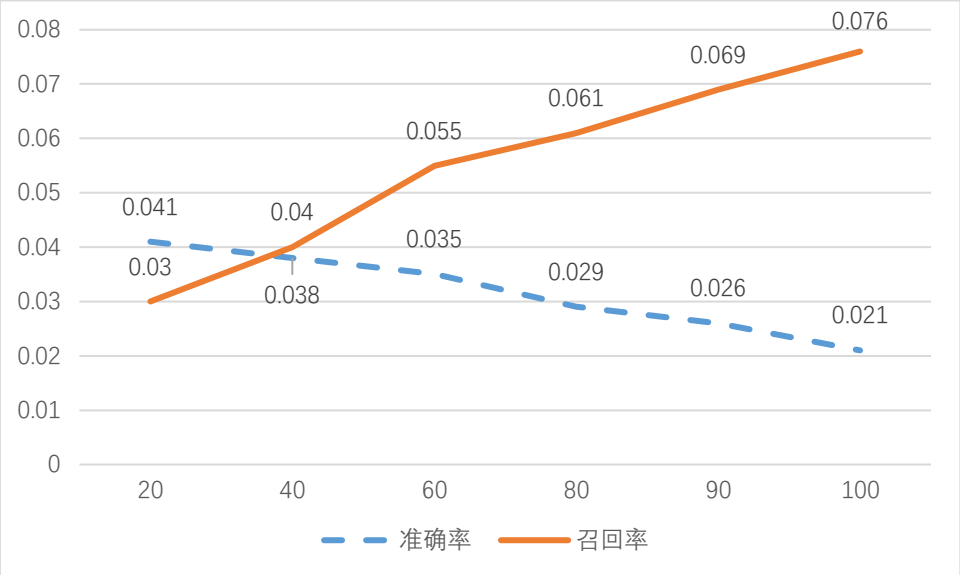

图 4.8 多因子融合实验结果

Figure 4.8 Multi-factory Fusion Experimental Result

在多因子融合模型下，由图 4.8 可以看到，召回率在 20-100 的长度下上升速度明显。准确率率则有一定降低，为了将算法达到最大优化的目的，在后续的研究中可以从适当调整返回的文档数来优化系统的推荐结果。由实验结果可知，本文所使用的基于用户偏好分析的个性化引文推荐算法的推荐结果较为满意，能在一定程度上满足现在大多数用户的要求。

第 5 章 基于用户偏好分析的个性化引文推荐原型系统实现

5.1 系统总体框架设计

系统基于 BS 模式进行实现，在设计的过程严格按照 MVC 理念对模型、视图、控制器三层设计模式进行实现，减少模块之间的耦合度，提升系统的运行效率，也便于后期的维护工作。本着平台无关和系统高效运行的理念，服务器端模型构建层与数据处理层基于 python 语言进行开发，由服务器端进行大量的运算，并将运算结果以 json 的形式由用户接口层传到浏览器。前端负责结果的展示，与用户直接进行接触，接收和记录用户的信息通过 http 的 post 方式回传到服务器，对相关的用户行为记录进入数据库中。同时为了提升系统中的语料库，会采用爬虫抓取网页与用户上传文档进行分析整理，借助文档处理程序将这些信息存入数据库中，进而可以构建更为准确的模型。系统框架设计如图 5.9 所示：

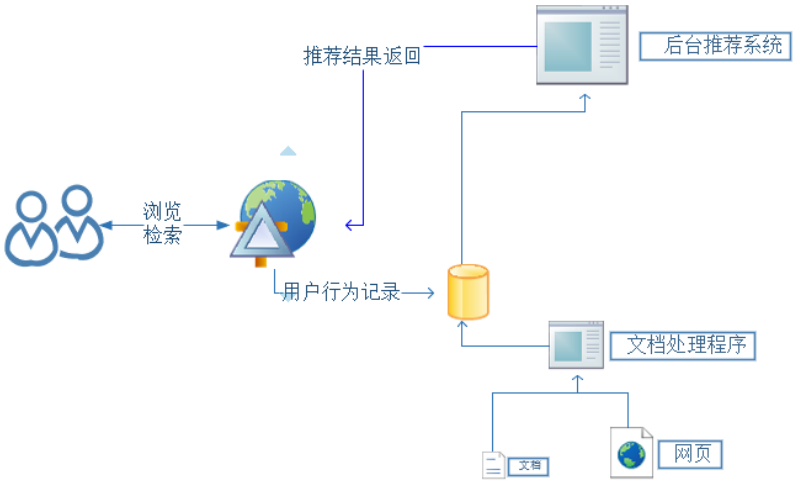

图 5.9 系统框架设计

Figure 5.9 System Framework

5.2 主要功能模块设计

基于上文对系统总体框架的分析，本段将对系统模块的设计进行展开。将基于用户偏好分析的个性化引文推荐系统划分为用户管理模块、论文推荐模块、信息展示模块和日志记录模块，共四大模块。用户管理模块下又划分为用户注册登录、权限管理和用户偏好模块，共三个小模块；在论文推荐模块划分为多因子融合推荐与推荐结果评价模块；信息展示模块划分为论文展示模块和作息信息模块；最后的日志记录模块划分为系统

错误日志、网站使用统计以及用户行为记录模块。各个模块分工合作，共同完成系统工作。模块设计如图 5.10 所示：

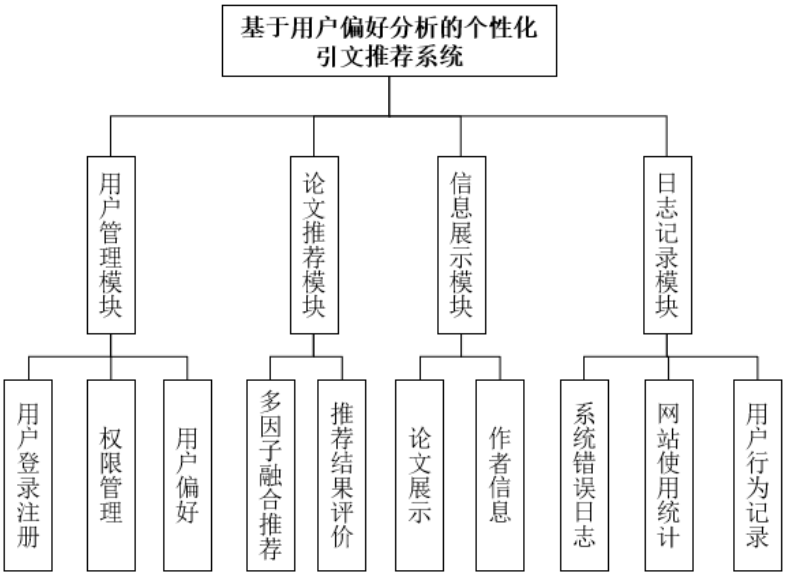

图 5.10 模块设计

Figure 5.10 Module Design

1. 用户管理模块：主要实现对用户的注册与登录、用户权限、以及用户的兴趣等信息的管理。登录的目的在于可以方便权限管理的同时也可以对一个用户在网站的行为进行最终，系统记录下的数据越多，对于用户的画像建模将会更加的准确。此外，每个人用户的个性不一样，用户管理模块可以针对用户的个性化需求做出一定的定制，实现 “千人千面”。

2. 论文推荐模块：本模块为系统的核心，包含了本文所提出的基于用户偏好分析的个性化引文推荐算法的实现，从数据的整理储存到预处理再到模型构建与计算，最后形成引文推荐的整个过程。同时，为了对推荐结果进行评价，系统需要设计评价模块，和用户直接进行交互，以此来优化系统中的模型参数，达到更准确的引文推荐结果。

3. 信息展示模块：本模块的主要功能是对信息的展示，数据库中存储的信息不便于用户浏览与理解，在本模块将会通过更加友好的方式将数据库中的文档、以及作者信息展示给用户，提升用户使用体验。

4. 日志记录模块：用户在使用系统的同时会产生许多的行为信息，如何对这些信息进行处理，本文暂未给出答案，但这些真实有效的行为信息可以在未来的研究中作为训练数据，以得到更好的用户画像。由此可见，对这些信息的存储显得非常有价值。由

于系统在长久的运行中会产生一些不可预测的问题，做好系统的日志的记录可以方便在出问题时及时依据日志进行回滚，使得系统尽快能够恢复使用。同时，系统日志信息有利于系统维护者查找问题的原因所在，便于及时找到问题然后解决问题。日志可以记录许多的信息，通过这些信息可以进行其他的研究，例如通过日志统计用户甚至网站的活跃时间度等等。基于日志记录的网站使用统计正是基于对对数据的充分利用与挖掘产生的。

5.3 数据库设计

在基于用户偏好分析的个性化引文推荐原型系统中，数据库对于数据的存储起到了至关重要的作用。利用数据库可以便于数据的存储、获取以及分析等。在基于本文的推荐算法所构建的原型系统，主要涉及的表包括：用户信息表、用户行为信息表、组织信息表、论文信息表、引文信息表、用户行为信息表、组织信息表等等。以下将对部门的表的字段设计进行说明：

用户信息表：包含了用户 ID、用户名、用户真实姓名、密码、邮箱、电话、组织、学历等。详细信息见表 5.3。

表 5.3 用户信息表

Table 5.3 Use Information

编号	名称	描述	类型	约束条件
1	Id	用户编号	Int (4)	Primary key, Not null
2	UserName	用户名	Varchar (30)	Not null
3	FullName	真实姓名	Varchar (30)	Not null
4	Password	密码	Varchar (30)	Not null
5	Email	邮箱	Varchar (30)	
6	Phone	电话号码	Varchar (15)	
7	Org	组织	Int (4)	Foreign key
8	Edu	学历	Int (4)	Foreign key

论文信息表：包含了文章 ID，网页浏览地址，论文标题、作者、发表年份、组织、文章摘要、参考文献等。详细信息见表 5.4。

表 5.4 论文信息表

Table 5.4 The Information about Sample Papers

编号	名称	描述	类型	约束条件
1	Id	文章编号	Int (4)	Primary key, Not null
2	url	网页链接	Varchar (100)	Not null
3	Title	标题	Varchar (100)	Not null
4	Author	作者	Int (4)	Foreign key
5	Year	发表年份	Varchar (30)	
6	Abstract	摘要	Text	
7	Org	组织	int (4)	Foreign key
8	references	参考文献	Text	

用户行为信息表：ID，开始时间、结束时间、浏览时长、网页链接、用户 ID 等。详细信息见表 5.5。

表 5.5 用户行为信息表

Table 5.5 Information about User Behavior

编号	名称	描述	类型	约束条件
1	Id	编号	Int (10)	Primary key, Not null
2	User	用户名	Int (4)	Foreign key
3	start	开始时间	Date	Not null
4	end	结束时间	Date	Not null
5	url	网页链接	Varchar (100)	
6	Time	浏览持续时间	Date	

5.4 原型实现效果

基于本文提出的算法，本文搭建了简单的系统原型实现，登录页、检索页、以及结果返回页效果图如图 5.11 与图 5.12 所示：

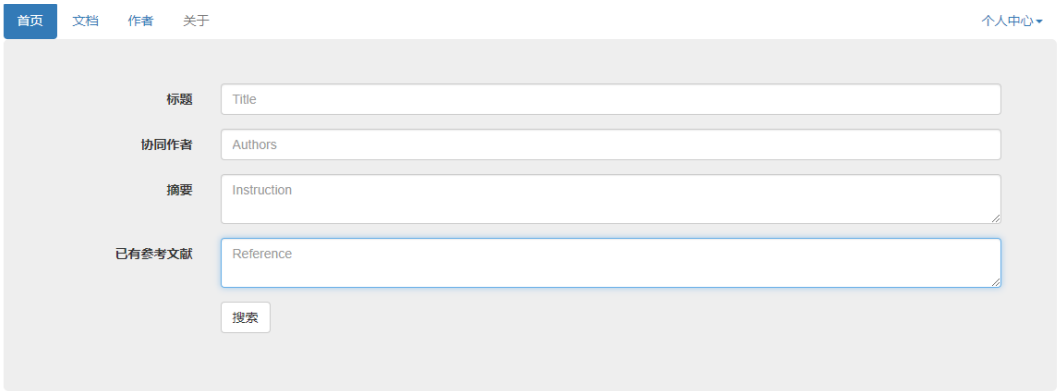

图 5.11 检索页面

Figure 5.11 Search Page

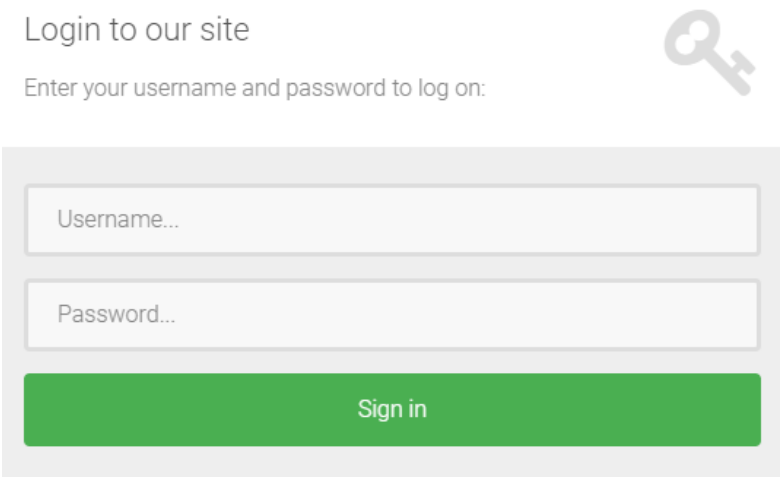

图 5.12 登录页面

Figure 5.12 Login Interface

在使用用户名为 Rebecca Dridan，密码为 0000 登入系统，得到的推荐结果如下图 5.13 所示，推荐结果基本符合用户需求。

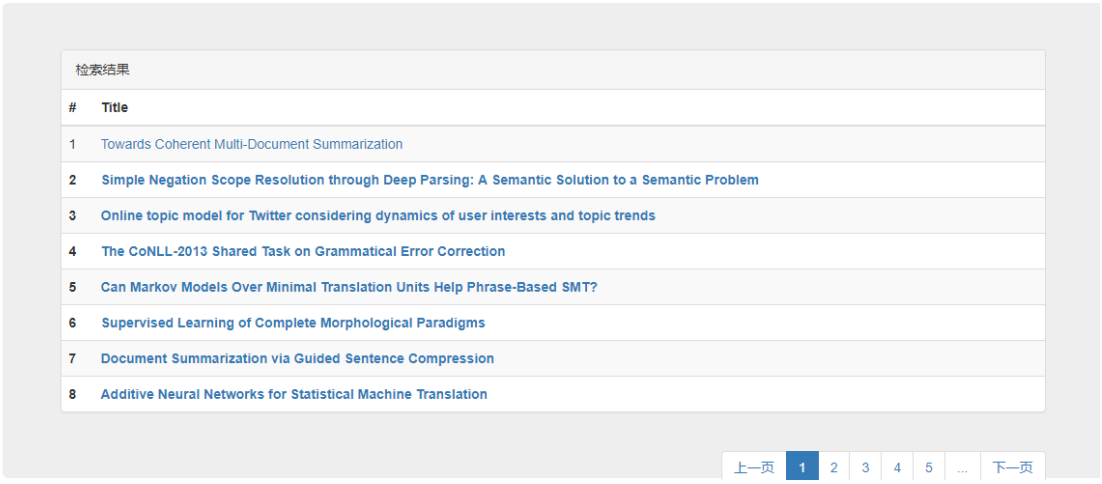

图 5.13 系统界面

Figure 5.13 System Interface

第 6 章 总结与展望

6.1 总结

计算机发展带领我们走进了数字化信息时代，相较之前的工业时代，数字化信息时代最大的特点在于互联网以及连接在互联网上各个节点对信息的存储、整理、组织、分析检索等信息过程的变化，这是一个革命式的变化。随着计算机技术的不断发展，会使用计算机已经是每个人工作生活的必备技能。计算机与各领域的结合得到了蓬勃的发展，计算机技术领域的研究也从未间断过。如何用好计算机，使其在我们的生活与工作中发挥更多更大的效用？计算机在实际应用的研究正是来源于生活，也最终回归到生活。

数字信息时代，信息呈指数增长，大数据给研究提供了更多可靠的有用信息，但对信息检索的用户来说确是一场灾难，尤其是科研工作者们。他们在检索与研究相关的文献是痛苦和费力的，引文推荐因此被提出来，是一项计算机技术在实际生活的应用，着眼于解决现在的论文作者在查找相关文献资源难、费时、资源质量参差不齐，相关度不高等问题。显而易见，引文推荐的相关研究变得非常有实际的应用价值。

在此背景下，众多的国内外科研工作者纷纷进入这个领域，展开了研究。本文在总结引文推荐的研究的工作进展的基础上，对目前常见的几种常用是算法模型进行分析比较，提出了一种基于用户偏好分析的个性化引文推荐算法，从理论推理、实验论证到最后的原型系统实现，实验表明，本文所提出的算法在一定的程度上能够提升引文推荐系统的质量。以下是本文的主要内容：

探讨和分析了本文课题的研究背景及意义，阐释了引文推荐问题提出的背景和在实际应用领域中的价值。

总结了当前国内外的研究进展以及对常用的模型算法和现有的工作使用情况进行分析和比较，把握引文推荐的理论基础。

提出了基于用户偏好分析的个性化引文推荐算法。利用用户是否参与一篇文章书写这一行为使用 BPR 算法进行构建用户兴趣模型，得到系统文档正确排序，以及用户对文档的兴趣值。

在基于用户偏好分析基础上，引入了作者与作者，作者与文档，文档与文档形成的三层图模型，有效的利用文档上下文和作者之间的关系特征，基于重启随机游走算法，对查询做出引文推荐结果的响应。最后通过实验证明其有效性。

基于本文提出的算法作为引文推荐系统的核心，搭建了一个引文推荐系统。同时对系统的框架设计、主要功能设计、数据库设计以及实现过程的相关技术进行介绍。

6.2 展望

本文所提出的基于用户偏好分析的个性化引文推荐算法是在前人的研究基础之上进行的，通过分析总结前人在引文推荐算法研究中的优点与不足，进一步的对算法进行优化提升。虽然实验数据证明，本文所提出的算法在一定的程度上对推荐的精准度有提升，但是依然存着一些需要改进的不足之处。以下是基于本文对未来研究的中引文推荐的思考：

1、本文所提出的算法虽然纳入了用户偏好进行考虑，但是在用户偏好建模上并不够准确。只考虑了用户与文章之间的书写关系，未来可以参考网页推荐研究进行用户偏好建模，纳入更多的因子，如：用户的点击行为、收藏行为、浏览时长等等。

2、本文所使用的三层图模型虽然尽可能的纳入作者、文档之间的关系特征，但在构图模型的时候所纳入的因子不够丰富，也没有考虑文档中词的关系特征信息，下一步研究可着眼于此。

3、在用户偏好与文档相似度进行结合的过程中，只是简单的使用了加权平均进行合并，权值也未优化和做深入的研究。

4、基于此算法搭建的系统原型，只是进行了简单粗糙的实现，仍然有诸多的细节未完善，功能模块的覆盖程度也不够。

参考文献

[1] 中国互联网络信息中心, 第 43 次《中国互联网络发展状况统计报告》[EB/OL]. [2019-05-10]. http://www.cac.gov.cn/2019-02/28/c_1124175686.htm.

[2] 中国日报, 中国国际论文被引用次数排名提升到世界第二[EB/OL] [2019-05-10]. http://cn.chinadaily.com.cn/2017-10/31/content_33928829.htm.

[3] 陆炀. 基于翻译模型的引文推荐[D]. 北京: 北京大学, 2013.

[4] Resnick P, Iacovou N, et al. An Open Architecture for Collaborative Filtering of Netnews[C]// Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work. New York: ACM Press, 1994: 175-186.

[5] Strohman T, Croft W. B, et al. Recommending Citations for Academic Papers[C]// Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. New York: ACM Press, 2007: 705-706

[6] Tang J, Zhang J, et al. A Discriminative Approach to Topic— Based Citation Recommendation[C]// Advances in Knowledge Discovery and Data Mining. Berlin: Springer Berlin Heidelberg, 2009.

[7] He Q, Pei J, et al. Context-aware Citation Recommendation[C]// Proceedings of the 19th International Conference on World Wide Web. Raleigh: ACM New York, 2010:421-430.

[8] 石杰. 基于多因素的引文推荐策略研究[D]. 沈阳: 东北大学, 2011.

[9] 李飞, 张宏鸣, 蔡晓妍. 一种改进的个性化查询引文推荐方法[J]. 计算机应用研究, 2019, 36(8): 1-8

[10] 陈志涛. 基于深度学习的个性化引文搜索推荐算法研究[D]. 杨凌: 西北农林科技大学, 2018.

[11] 陈海华, 孟睿, 陆伟. 学术文献引文推荐研究进展[J]. 图书情报工作, 2015, 59(15): 133-143, 147.

[12] Livne A, Gokuladas V. CiteSight: Supporting Contextual Citation Recommendation Using Differential Search[C]// Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval. New York: ACM Press, 2014:807-816.

[13] Mikolov T, Bengio Y, et al. On the Difficulty of Training Cecurrent Neural Networks[D]. New York: Cornell University, 2013.

[14] Kimothi D, Soni A, Biyani P. Distributed Representations for Biological Sequence Analysis[D]. New York: Cornell University, 2013.

[15] Radev D R, Muthukrishnan P, et al. The ACL Anthology Network Corpus [J]. Language Resources and Evaluation, 2013, 47(4): 919-944.