申请上海交通大学硕士学位论文

基于内容的自适应推荐系统研究

学　　校： 上海交通大学

院　　系： 电子信息与电气工程学院

班　　级： B1203691

学　　号： 1120369003

硕 士 生： 段淮

工程领域： 信息与通信工程

导　　师： 刘功申

上海交通大学电子信息与电气工程学院

2014 年 12 月

A Dissertation Submitted to Shanghai Jiao Tong University for the Degree of Master

THE RESEARCH OF ADAPTIVE RECOMMENDATION SYSTEM BASED ON CONTENT

Author: Duan Zhun

Specialty: Information and Communication Enginieering

Advisor: Liu Gongshen

School of Electronic Information
and Electrical Engineering
Shanghai Jiao Tong University
Shanghai, P. R. China
December, 2014

基于内容的自适应推荐系统研究

摘 要

随着互联网技术的迅猛发展，人们逐渐地从曾经的信息匮乏时代步入了信息过载的时代。如何从海量信息里获取自己所需要的信息迅速成为研究的热点。由于在信息过滤中的良好表现，推荐系统成为解决信息过载问题有效方法，并产生了巨大的商业利润。由此推荐系统在商业应用以及学术研究方面都有极大的研究价值。

在众多的推荐系统中，基于内容的推荐系统在文本推荐领域有着广泛的应用。本文主要对内容推荐系统中初始模板的构建以及用户模板的更新进行分析和研究，提出了一种应用于文本推荐的基于内容的自适应推荐系统，以提高推荐的准确性及效率。

在建立初始用户模板方面，本文提出了一种基于 TextRank 算法建立初始模板的方法，利用了用户提供信息中的集簇性。通过确定词义项，聚类，建立图模型，引入各种影响力因子修正 TextRank 概率转移矩阵等一系列操作，在只有少量数据的情况下，建立起一个精确的用户模板，有效地提升了推荐精度。

在用户模板更新方面，本文在使用用户提供的新数据更新模板的同时将信息检索中的伪相关反馈概念引入系统，更新用户模板。通过一系列操作挑选最优反馈文档，筛选关键词，结合改进的 Rocchio 算法更新模板，减少噪声引入，扩大了推荐范围，达到更好的推荐效果。

实验表明本文提出的基于内容自适应推荐系统有较好的推荐效果。

关键词：内容推荐算法，义项确定，TextRank，伪相关反馈，Rocchio 算法

THE RESEARCH OF ADAPTIVE RECOMMENDATION SYSTEM BASED ON CONTENT

ABSTRACT

With the rapid development of internet technology，people step into the era of information overload from an era of information scarcity gradually. How to get our required information from huge amounts of information in this era quickly became a research hotspot. Due to good performance in information filtering, recommendation system become an effective way to solve information overload problem and bring huge commercial profits. Therefore recommendation system has great research value not only in business application but also in academic research.

Among many kinds of recommendation system，recommendation system based on content has been widely used in the field of text recommendation. This paper majors in the construction of the initial user profile and user profile updating in the process of recommendation in content recommendation system. The author puts forward an adaptive recommendation system based on content applied in text recommendation, in order to improve the accuracy and efficiency of recommendation.

In the respect of building initial user profile, this paper presents a method of building initial user profile based on TextRank which make full use of the character of cluster in users' information. By taking a series of measures like determining the meaning of each word, clustering, establishing graph models, introducing various influence factors to make the TextRank transition probability matrix better, an accurate initial user profile is built when just having little information, which can improve the

accuracy of recommendation effectively.

In the respect of updating user template, this paper introduces the concept of pseudo relevance feedback in information retrieval to this recommendation system to update user profile at the same time of using the new data user providing updating the template. A series of operations are taken to get optimal feedback documents, select keywords and update user profile by using improved Rocchio algorithm so that the system can avoid the introduction of the noise term, expand the scope of recommendation and achieve better recommendation results.

Experiments show that the accuracy of the recommendation is high in this system.

Keywords: content recommendation algorithm, determining meanings, TextRank, pseudo relevance feedback, Rocchio algorithm

目 录

图目录

表目录

1 绪论

1.1 研究背景和意义

近年来,随着互联网技术的迅猛发展,人们逐渐进入了海量信息同时呈现的信息爆炸时代。如何应对随之而来的信息过载问题[1]成为研究的热点。推荐系统作为解决信息过载问题的一个重要手段有效地解决了互联网中的信息过滤问题[2]。特别是在电子商务快速发展的今天，商家所提供的商品种类和数量都在快速的变化之中，用户需求也通常是不确定或者是模糊的。于是，各种能够通过分析用户特征有针对性的推荐商品的个性化推荐系统[3]对于各个网站来说有着重大意义。一方面，为用户过滤掉大量的无用信息可以大大提升用户体验和用户忠诚度。另一方面，将用户的潜在需求转换为实际需求给商家带来了巨大利润，同时也使得一些重要的“暗信息”被用户及时获取。

包括 Amazon、YouTube、eBay 等诸多网站都部署了不同形式的推荐系统。据统计，其中 Amazon 的推荐系统将其商品销售额提高了 35%。以 Amazon 的一个最简单的应用场景为例，当用户浏览《Thinking In Java》这本书的网页时，Amazon 会快速地将其他的一些 java 经典书籍推荐给用户。这样，给用户带来了极大的便利。目前，推荐系统的研究吸引了来自计算机，数学，市场营销，物理，管理等众多领域的研究者。在不同领域研究人员的努力下，推荐系统正在进一步发展和完善之中。2006 年，Netflix 主办的推荐算法竞赛吸引了多达 4 万多个团队的参加。由此可见，推荐系统不仅在商业上被极大重视，也成为学术领域的一个研究热点。

现阶段，常用的推荐系统有基于内容的推荐系统，协同过滤推荐系统，基于图结构的推荐系统以及混合推荐系统等[4]。其中协同过滤推荐系统指的是通过对用户相关信息的分析找到与该用户偏好相似的其他用户,这些用户的喜好将成为一个重要参考，由此为用户推荐对应的商品。基于内容的推荐指的是由用户所选对象的特性推断用户偏好，为用户推荐具有其他类似属性的对象，主要广泛应用于文本等产品特征易于提取的领域。混合推荐是上述几种推荐算法的结合，包含有两种或多种算法的特性。其可以通过单独使用协同过滤算法，基于内容推荐算法，基于图结构推荐算法,最后用投票等操作综合推荐结果，或者将多者在推荐过程中互相融合，得到推荐结果，以此达到取长补短的作用。此外还有基于关联规则[5]等等其他推荐算法。

综上所述，在商业应用上，一个优秀的推荐系统有助于提高用户的粘连度，培养用户忠诚度，产生大量利润。在学术领域，推荐系统的研究涉及多种学科。可见，推荐系统的研究有很大的实际意义，需要我们持续的深入研究和推广。

1.2 国内外研究现状

目前，国内外已有不少学者对推荐系统中的各种算法进行了研究。自20世纪90年代中期出现了一些关于协同过滤的文章后，推荐系统持续发展，各种推荐算法层出不穷，并且涵盖多种学科，包括人工智能，管理，市场营销等等。

协同过滤推荐系统是最早被提出并得到成功应用的推荐系统，其首先通过用户的评分数据计算用户之间的相似度，找到相似度较高的邻居，然后通过邻居的评分数据估计该用户对待推荐产品的评分，以此确定是否为用户推荐该产品。基于这个思想，协同过滤的研究包括基于用户的记忆推荐算法，基于项目的记忆推荐算法，基于用户的top-N记忆推荐算法，基于项目的top-N记忆推荐算法，基于朴素贝叶斯分类的推荐算法，基于马尔科夫决策过程MDP的推荐算法等。Grundy[6]是第一个采用协同过滤并且投入应用的推荐系统。其通过建立用户兴趣模板,为每个用户推荐相关的书籍。此外，Ringo的音乐推荐，Amazon书籍推荐系统,Jester的笑话推荐系统,也同样使用了协同过滤。

显然，协同过滤推荐系统的优点是没有特征提取的困难。视频，音乐等不易结构化表示的项目均可以通过协同过滤进行推荐。但是，协同过滤推荐系统也存在一些显而易见的缺点。这主要体现为“冷启动”问题[7]：由于协同过滤主要依赖用户对商品的评分数据，当系统加入新用户时，新用户并没有对商品的评分，无法寻找喜好相同的邻居；当系统加入新商品时，由于所有用户都没有对该商品进行评分，也无法对该商品进行处理。上面的情况都是协同过滤推荐系统中棘手的问题。

基于内容的推荐算法是推荐系统中的另一个主要算法。基于内容是指通过用户选择对象的特征推断用户偏好，为用户推荐具有其他类似属性的对象。由于自然语言处理和机器学习等技术日趋成熟，该推荐算法在文本推荐领域有着广泛的应用。同时，采用基于内容的推荐算法有助于改善推荐系统中的冷启动问题。其典型流程为：收集用户爱好信息；建立用户模板；对待推荐文本集内的文本生成文本向量；计算用户向量与文本向量的相关系数，将相关系数高的文本推荐给用户；根据用户的反馈信息进行更新模版以提高模板精度。

在收集用户爱好信息时，首先要提取特征表征一个项目。Resnick 等人将自然语言处理领域中经典的 TF-IDF(词频-倒排文档频率)的特征提取方法引入推荐算法[8]，项目特征由关键词的出现次数确定。用户偏好文档和推荐项目文档通常均使用VSM模型表示，使用由关键词组成的一个向量表征，向量的每一维为一个关键词，权重为关键词重要程度的数值体现。例如Syskill和Webert系统[9]的一个文件由128个信息量最多的词表示。在Fab系统[10]中一个网页由100个最重要的关键词表征。在收集用户信息之后，可以使用这些信息得到一个用户模板，这样就可以通过计算待推荐项目于模板的相关程度决定是否推荐该项目。向量空间中计算相关程度常用余弦相似度的方法。在这个过程中，一些学者将机器学习中的相关技术引入内容推荐系统。比如通过引入贝叶斯分类技术[11]，将推荐问题转换为一个分类问题，通过用户数据训练分类器，当网页通过分类器时，可以通过分类器输出概率确定该网页是否为用户感兴趣文档。类似的，人工神经网络[12]，决策树[13]，支持向量机[14]等等技术都被引入推荐系统以

改善系统性能。还有一些学者使用了其他形式计算相关程度。例如Cui Chun-sheng等人运用Vague集计算商品相似度[15]，使用 Vague值表示商品特征，计算商品间的相关程度。类似的，也可以使用Fuzzy集分别表征用户模板和待推荐文档，计算两个Fuzzy集的相关程度决定是否推荐项目。

用户模板的更新是内容推荐系统的另一个重点和难点。随着时间的变化，用户的兴趣可能会动态变化，此时需要更新偏好文档，确保为用户推荐内容的准确性。同时，为了避免小范围内推荐造成的推荐效果不佳的问题，有必要引入反馈。有学者提出自适应更新用户模板方法。 其将与用户模板相关程度高的文档推荐给用户,于此同时，使用相似度高的文档更新用户模板，以此动态调整用户偏好模板。

此外，还有许多研究有助于提高内容推荐系统的推荐精度。比如：基于潜在语义分析的推荐；利用WordNet等外部资源建立一个包含语义信息的用户模板，以提高模板精度以及计算相似度时的精度；结合心理学，社会学等相关知识建立合理的交互系统，以尽可能精确地获取用户偏好信息；在系统初始时，通过概率知识建立一个较为精确地用户模板，避免偏移等[16]。

1.3 论文的主要工作

本文提出了一种基于内容的自适应推荐系统。

首先，在基于内容的推荐系统中，初始用户模板的准确性对后面的推荐精度有很大影响。因此，在系统初始时，必须从少量用户信息中准确地提取出用户兴趣模板，尽可能的减少噪声的引入。否则会在后期更新模板时产生偏移性问题，造成推荐的不准确。本文提出了一种基于 TextRank 算法建立初始模板的方法，首先对所拥有的少量用户感兴趣文本进行预处理并确定词义项，然后进行聚类，接下来对聚类得到的每个类别分别以义项为单位构建 TextRank 模型，并引入各种影响力因子对 TextRank 模型中的概率转移矩阵进行改进。迭代之后选取每个类中最为关键的若干义项进行综合，得到最终的初始用户模板。

其次，在用户模板更新方面，为了让用户模板尽快的达到较为精确的状态，避免只是在小范围内推荐造成的推荐效果不佳的问题。本文在使用用户提供的新数据更新模板的同时将信息检索中的伪相关反馈的概念引入推荐系统，更新用户模板。同时，为了避免出现类似信息检索中的查询偏移问题，本文通过聚类，添加其他影响因子等一系列操作对伪相关反馈文档进行重新评分，挑选最优文档进行反馈，由此对 Rocchio 算法进行改进。在挑选反馈文本中的关键词方面，本文更多的考虑了候选词与原模板词的共现性，以减少噪声词的引入。最终更新用户模板。

1.4 论文结构

本论文共分为七章，各章内容结构组织如下：

第一章为绪论，对全文进行简要介绍。首先描述了论文的研究背景及意义，表明推荐系

统的研究具有学术价值和商业应用价值。然后介绍国内外对推荐系统相关工作的研究现状。在本章的最后阐述论文的主要工作内容以及本文的结构安排。

第二章介绍了基于内容推荐算法所用到的相关理论知识。

第三章主要介绍了基于内容的自适应推荐系统的整体架构，并且分别介绍了其中的基于TextRank 算法初始模板建立模型以及基于伪相关反馈的用户模板更新模型架构。

第四章提出了一种基于 TextRank 算法建立初始模板的方法，详细介绍了图模型的建立步骤，各种影响力因子的引入原因，以及最后初始用户模板的生成方法。

第五章提出了一种基于伪相关反馈的用户模板更新方法，按照需求拆分用户模板，通过一系列操作提高伪相关反馈文档质量，挑选相关关键词，减少噪声引入，更新用户模板。

第六章对上文提到的几种算法分别进行准确度实验，对比试验。

第七章总结本论文的研究工作，并给出了后续研究方向以及改进点。

2 相关基础理论介绍

本章首先介绍了基于内容推荐系统中文本处理特定步骤的理论基础，为本文后面的算法描述打下基础。然后介绍信息检索中相关反馈的概念，本文会将相关反馈的概念引入内容推荐系统，以更新用户模板。

2.1 中文分词技术

2.1.1 中文分词概述

中文分词技术作为中文处理的基础，指的是将一个中文字符序列切分成一个个有实际意义的词项，是一个按照一定的规范将连续的字序列重新组合成词序列的过程，广泛应用于自然语言处理，搜索引擎，文本挖掘等领域。不同于英文中词与词之间有着天然分界符，中文中不存在这种情况。因此，在分词难度上，中文分词比英文要大的多[17]。

之所以要进行分词操作，主要是因为词是最小的有意义的语言成分[18]，并且是能够独立活动的。分词作为自然语言处理过程中最初一环，其结果直接影响着后期分析的精度和效率。信息检索和信息抽取，文本挖掘，文本分类等都会使用分词这一基本模块。因此，分词技术的研究是自然语言处理的基础，也是基于内容的文本推荐算法的根基。

2.1.2 中文分词算法简述

细化的中文分词算法种类繁多，概括起来可对应于三类：基于词典的分词算法，基于知识理解的分词算法以及基于统计的分词算法。三类算法有着不同的思想以及工作方式，下面对这三类算法进行大致描述。

1. 基于词典的分词方法

该算法首先需要一个足够大的机器词典。待分析的字符串按照一定策略与该词典中的词项逐条匹配。如果在词典中找到该词，则匹配成功，顺利分出词项。该类算法按照扫描方向或者不同长度优先匹配的情况细分如下：

(a) 正向最大匹配算法

取词典中的最大词长最为初始截取长度，从左到右对待处理字符串截取该长度的字符串，与词典中的词依次进行匹配操作，若匹配到词典中对应字符串，则截取的串就是一个词项，继续处理剩下的待处理子串。否则，向前缩短截取字符串，继续进行匹配操作，直到匹配成功。

(b) 逆向最大匹配算法

与正向最大匹配法类似，只是扫描时从右向左扫描，与词典进行匹配操作。该方法的提

出是因为有实验证明逆向匹配有着比正向匹配更高的准确率。

(c) 双向匹配算法

该算法结合了不同扫描方向的优点，是上面两个算法的结合。

(d) 最佳匹配算法

对词典进行词频统计，优先匹配那些词频较高的词项，同时也提高了效率。

2. 基于统计的分词算法

基于统计的分词算法主要依赖于字与字之间的共现性。显然，如果邻居字的共现性越高，这些字越可能组词。这种算法并不依赖于词典，只需要训练文本。与基于词典的分词方法比较，首先，这种方法省去了加载词典的时间损耗。由于用于匹配的词典一般比较大，加载时会损耗一定资源。其次，统计的方法不依赖于词典，那么一些新词或者特有词就容易被识别出来。常用于统计分词算法的统计模型有最大熵模型，隐马尔科夫模型等。

3. 基于知识理解的分词算法

该算法主要是基于对待处理字符串含义的理解。基于知识理解分词系统会对字符串进行语法分析，上下文分析，试图让计算机模拟人类的理解能力，以此改善系统性能。在处理歧义问题时，该算法比前面两类算法有更好的性能，但是鉴于中文的复杂性，该算法在中文分词领域应用还不是很广泛。

2.2 文本表示模型

在自然语言处理过程中，要想让计算机“理解”待处理文本，需要把文本转变为计算机可以处理的形式。文本表示就是指将实际的文本内容变成机器内部表示结构的过程。文本的表示可以用字、词、短语、n-Gram 等形成向量或树等结构。其主要包括两个方面：表示和计算。表示特指特征的提取，计算指的是特征重要程度的量化以及相似度计算方式的定义。下面对几种主要的文本表示模型分别进行介绍。

2.2.1 布尔模型

布尔模型是建立在布尔变量，布尔操作，布尔表达式基础之上的一种简单模型。其对特征项进行严格匹配，假定对应特征项要么在文档中，要么不在文档中，对应的将特征项的取值用布尔变量 true 或者 false 表示。最终将文档表示为一个布尔表达式，特征项与特征项之间用 AND，NOT，OR 等布尔操作连接。

布尔模型中文本与文本之间的相似度以及文本与查询之间的相似度通过布尔运算法则运算，将查询表示为析取范式(disjunctive normal form DNF)，进而计算相似度。

布尔模型发展较早，最初的检索系统大部分是基于布尔模型。其优点主要体现在简单、速度快，直到现在类似 google 的很多搜索引擎中仍然包含着布尔模型的思想。于此同时，布尔模型的缺点也十分明显。首先，布尔模型的检索类似于精确匹配的数据库检索，不能近似或部分匹配，多个结果无法排序，缺乏灵活性。其次，用布尔表达式表示文本并不精确，缺

乏定量分析，并且表达式构造十分复杂，不好的表达式会造成匹配结果的过多或过少。

2.2.2 向量空间模型

向量空间模型（VSM）最初的提出者是康奈尔大学的 Salton 等人，其成功地将 VSM 应用于 SMART 系统[19]中。该模型认识到了布尔模型在量化分析上的局限性，将文本表示为欧式空间中的一个向量，进而，向量空间中的许多运算都可以被引入到文本与文本之间。目前，该模型广泛应用于信息检索，文本分类等众多领域，是最高效简便的文本表示模型之一。

在向量空间模型中，文档以标引项（Term）以及其权重组成的向量表示，并且假设标引项在文章中独立出现，互相不影响。这样，一个文档就可以视为欧式空间中的一个点。在向量空间模型中，主要有两个问题：标引项的选择，权重计算。标引项必须能体现文档的特征。由于目前自然语言处理方面的研究水平还不足以理解文本的含义，所以为了大批量处理文本，主要采用统计的方法获取文本信息，以标引项标识文本，并辅之语义，句法等知识。当前使用的标引项的种类主要有字，词，短语，N-gram，语义单元等。一般情况下，标引项的语言层次越高，对应包含的信息越多，越有利于表征文档特征，但是，层次越高会使分析的代价变大。由于词汇是一篇文档中最小的能够独立活动的有意义的语言成分，很多领域都以词为单位并结合统计规律表征文档。还有一些情况下，一些系统会以字为标引项，一般而言，虽然字是汉语的最基本单元，但是其并不能完整的表示一个语义概念，所以用其对文档进行表示是不合理的。可是，实验表明，用字作为特征相较于用词汇，语义单元等作为特征在文本分类，信息检索等应用中性能并没有下降，反而，由于省去了分词等操作，效率有所上升。这可能与汉语的特殊性有关。此外，短语由于其更强的表现能力，也会被用于标引项。特别是在汉语中，很多短语拆分成词时会造成信息量锐减或者产生歧义等问题。比如“向量空间”这个短语，被拆分为“向量”和“空间”两个词分别标识时，显然脱离了原来的语义。近年来，有一些新的研究把语义单元作为标引项表征文本，特别是一些类似于 WordNet 等资源的出现，把许多词汇按照含义及之间的语义关系以树状结构等很好的组织了起来，这样就给语义单元作为特征的研究提供了便利。在汉语中，由于有大量的一词多义和同义词现象，用语义单元作为特征项有助于更精确地表示主题。但是，由于引入了大量确定语义等操作，系统效率也会相应下降。本文中主要使用了以词为单位的向量空间模型和以语义单元为单位的向量空间模型。

在确定使用何种标引项之后，最简单的方法就是使用全文标引，即使用文本中出现的所有字，词或者语义单元作为特征。但是，全文标引的缺点明显。首先，在基于统计的情况下，一些常用词在文本中有着较高的频率却有着较低的区分能力，即与文章的主题关联不大，把这些噪声引入向量显然效果不是很好。其次，采用全文标引，文本向量的维数很大，即使是一个中等规模的文本数据集也具有几万维。很多机器学习算法无法处理这么高的维度，只有少数几种神经网络的算法能处理这么高的维度，并且系统效率会急剧下降。因此，在选择标引项时，通常会伴随使用一些降维的策略，比如去除停用词，词干还原，对词性进行筛选等

等。

权重计算也是向量空间模型中一个重要的方面。其最主要的方法就是著名的 tf-idf[20]公式计算。设 $Freq_i$ 表示第 i 个特征项在文档中出现频率，则 $freq_i$ 可以从一个侧面反映该特征项的重要程度，由于不同文档的长度不同，需要对 $freq_i$ 进行归一化处理。归一化方法式（2-1）：

$$tf_i = \frac{Freq_i}{MaxFreq_j} \tag{2-1}$$

除了特征项频率之外，特征词的重要程度还与文档频率有关。若包含该特征的文档越多，那么这个词的区分度就应该越低。对应的得到特征项的 IDF 公式（2-2）如下：

$$idf_i = \log\frac{N}{n_i} \tag{2-2}$$

其中 N 为文档集中的文档数目，n_i 为其中包含该特征的文档个数。由此得到计算标引项的 tf-idf 权重计算公式（2-3）如下：

$$w_i = tf_i * idf_i = tf_i * \log\frac{N}{n_i} \tag{2-3}$$

一般情况下，需要对上面的式子进行平滑处理。得到处理后的公式（2-4）如下：

$$w_i = (1+\log(tf_i)) * \log\frac{N}{n_i} \tag{2-4}$$

传统的权重计算方法并没有考虑特征词出现在不同位置上的差异性问题。比如相同的词出现在标题或者摘要上时，显然比出现在正文中更有价值。根据这个思想，文献[21]提出了一种 N 层向量空间模型，对传统的 tf-idf 公式进行改进。其基本思想是将一篇文档按照其组织结构划分为若干不同等级的区域。对不同的区域采用不同权重计算方法，各个等级之间的差异用不同的比例系数体现，对权重计算的过程进行调整，使向量更加贴合于文本。具体的计算过程为：首先，对文档进行区域划分，如一级标题，二级标题，摘要等，并根据不同区域的重要程度确定不同的比例系数。其次，记录每个特征项在各个区域的出现频率，最终结合区域系数，得到该特征项的修正后 tf 值，见式（2-5），其中 K 为划分区域总数，h_n 为对应区域的重要程度比例系数，D_i 与 d_i 分别表示特征在文档集和当前文本的出现频率。最终的到权重计算公式见（2-6），本文中一些章节使用了这种改进的权重计算公式。

$$tf_i = \sum_{n=1}^{K} h_n * tf_{in} * \log(\frac{D_i}{d_i}) \tag{2-5}$$

$$w_i = (1+\log(\sum_{n=1}^{K} h_n * tf_{in} * \log(\frac{D_i}{d_i}))) * \log\frac{N}{n_i} \tag{2-6}$$

经过特征选取及权重计算之后，一篇文档就用n维空间中的一个向量来表示一个文本，表示为 $D = D(t_1, w_1; t_2, w_2; .....; t_n, w_n;)$，其中 t_i 是特征项，w_i 是 t_i 对应的权重，$1 \le i \le n$。

如图1所示：

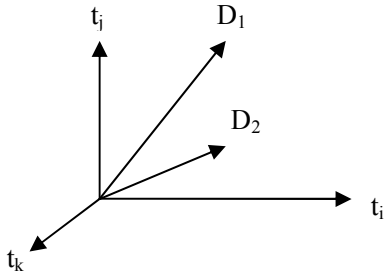

图1 向量空间模型示意图

Figure 1 The Graph of Vector Space Model

文本之间的相似度可以通过余弦相似度计算，见公式（2-7）。

$$sim\left(D_i,D_j\right)=\frac{\overrightarrow{D_i}\overrightarrow{D_j}}{\left|\overrightarrow{D_i}\right|\left|\overrightarrow{D_j}\right|}=\frac{\sum_{k=1}^{n}w_{ki}*w_{kj}}{\sqrt{\sum_{k=1}^{n}w_{ki}^2}*\sqrt{\sum_{k=1}^{n}w_{kj}^2}} \quad (2\text{-}7)$$

其中 D_i 和 D_j 为对应的文本向量，$\left|D_i\right|$ 和 $\left|D_i\right|$ 为向量的模（norms），可见相似度运算结果满足 $0\le sim\left(D_i,D_j\right)\le 1$ 。

2.2.3 概率模型

概率模型是由上世纪九十年代Roberston和Spack Jones提出的经典模型[22]，用以克服文本表示的模糊性及相关性判断的不确定性。概率模型以概率排序为基础，文献与文献之间的相关程度通过两者的相关概率体现。

假设用户查询为q，并且设D为相关文档的集合，$\overline{D}$ 为不相关文档的集合。那么文档 d_i 与q的相关概率就可以表示为 $P(D|d_i)$，对应的 d_i 与q的不相关概率表示为 $P(\overline{D}|d_i)$。那么就可以定义出文档 d_i 和q的相似度见式（2-8）。

$$sim(d_i,q)=\frac{P(D|d_i)}{P(\overline{D}|d_i)} \quad (2\text{-}8)$$

显然，上面的式子中 $P(D|d_i)$ 与 $P(\overline{D}|d_i)$ 并不便于计算，于是，用贝叶斯定理对上面的式子进行处理，得到相似度计算公式（2-9）如下：

$$sim(d_i,q)=\frac{P(d_i|D)*P(D)}{P(d_i|\overline{D})*P(\overline{D})} \quad (2\text{-}9)$$

其中 $P(\mathrm{d_i}|D)$，$P(\mathrm{d_i}|\overline{D})$，$P(D)$，$P(\overline{D})$ 分别表示相关文档集合中任选出 $\mathrm{d_i}$ 的概率大小；不相关文档集合中任选出 $\mathrm{d_i}$ 的概率大小；总文档集合中任意选择出的文档为相关文档的概率；总文档集合中任意选择出的文档为不相关文档的概率。在该模型中，文档依然会由若干标引项表示，那么当假设特征之间相互独立的情况下，上式可以得到式（2-10）：

$$\mathrm{sim(d_i,q)}=\frac{\prod P(\mathrm{t_i}|D)*\prod P(\overline{t_i}|D)}{\prod P(\mathrm{t_i}|\overline{D})*\prod P(\overline{t_i}|\overline{D})} \qquad (2\text{-}10)$$

其中 $P(\mathrm{t_i}|D)$，$P(\overline{t_i}|D)$，$P(\mathrm{t_i}|\overline{D})$，$P(\overline{t_i}|\overline{D})$ 分别表示特征 $\mathrm{t_i}$ 随机出现在D中的概率；$\mathrm{t_i}$ 不出现在D中的概率；$\mathrm{t_i}$ 随机出现在不相关文档集中的概率；$\mathrm{t_i}$ 不出现在不相关文档集中的概率。由上式即可计算出文档 $\mathrm{d_i}$ 和q的相似度。

2.2.4 基于模糊集的模型

模糊理论[23]是由美国加州大学的控制论专家 L.A. Zadeh 与上世纪六十年代创立。在经典数学中，集合通常是以明晰集的形式存在的。对于集合 S，可以表示为 S={w |K(w)}，其中性质 K 起到了区分作用。任何一个对象要么属于集合 S（满足条件 K），要么不属于集合 S（不满足条件 K），二者必取其一。由于对于是否从属于该集合的界限是分明的，不承认在是否满足条件 K 这个问题上存在有中介状态，所以传统的集合理论是一种对象的简单汇聚法则。然而，在日常生活中却存在有许多传统集合论无法表征的问题，在这些问题中，是否属于集合的界定相对模糊化，无法准确的判定对象是否从属于某个类别。以“跑的快”这个概念为例，由于标准的模糊化，很难确定某个对象时候属于这个类别。这样就引出了模糊集的概念如下。

定义 2.1 对于给定论域 U，可以定义模糊子集 H，该模糊子集由一个函数 $\mu_H(u)$ 确定，该函数的值域为[0,1]。

$$\mu_H : U \to [0,1]$$

该函数体现了论域上的元素 u 到模糊集 H 的隶属程度。也就是说，一个对象之于某个集合不再是非是即非的从属关系，而是由一个 0 到 1 之间的数决定归属性的大小。当隶属函数的取值非 0 即 1 时，此时对应的就是传统的明晰集。

模糊集在信息检索中有所应用，首先，可以列出所有的文档集合 $D=\{d_1,d_2,d_3...d_n\}$ 和所有的特征项集合 $\mathrm{T}=\{t_1,t_2,t_3...t_k\}$，并计算不同特征项之间的相关程度（通过共现性表现）。然后，以 D 为论域，每一个特征项为一个模糊集合，以此定义特征项与文档之间的关系。现在需要定义出论域中单个文档 d 对于模糊集的隶属度，定义见式（2-11）。

$$\mu_H(d_i)=1-\prod_{t_k\in d_i}(1-c_{HK}) \qquad (2\text{-}11)$$

其中模糊集 H 对应于特征项 $\mathrm{t_h}$，模糊集 K 对应于特征项 $\mathrm{t_k}$，c_{HK} 表示特征项 $\mathrm{t_h}$ 和 $\mathrm{t_k}$ 之

间的相关性，取值范围为 0 到 1。由上面的式子可见在文档 d_i 中，与特征项 t_h 相关的特征越多，该文档对于模糊集 H 的隶属度越大。然后，就可以确定查询与文档之间的关系了。将查询 q 同样的也产生一个模糊集，论域仍然为整个文档集合，现在只要确定每个文档对于查询模糊集的隶属度大小，就可以确定该篇文档与查询的相关程度。由于该查询对应的模糊集可以视为查询中关键词模糊集的并集或交集等集合运算的结果，引入模糊集运算法则后就可以通过计算文档对于查询中每个关键词模糊集隶属度得到最后的相关性结果。

在文本推荐以及文本分类方向，模糊理论同样有所应用。文献[24]就提出了一种基于模糊理论的文本分类模型。该模型以文档集中的所有的特征项为论域，将文本以及类别分别以模糊子集的形式表示出来。首先，计算特征项对于单个文档的隶属度，引入 TF-IDF 公式参与运算，归一化之后即可求得隶属度大小。然后，计算特征项对于类别模糊子集的隶属度大小，该隶属度的求法依然是基于训练集的统计情况，即统计每个特征在各个类别中出现概率的大小，归一化之后用文本频率数进行修正，即可得到特征项对于类别模糊子集的隶属度大小。在模糊理论中，模糊子集与模糊子集之间的相关性是可以计算的，定义如下：

定义 2.2 若模糊子集 C 和 D 有着相同的论域 U，其中 $u_1, u_2, u_3 \dots u_n$ 为论域内的元素。设 $\mu_C(u_i)$ 与 $\mu_D(u_i)$ 分别表示 U 中元素对于模糊子集 C，D 的隶属函数，那么 C 与 D 的相关系数 ρ_{CD} 定义如下：

$$\rho_{CD}=\frac{\sum_{i=1}^{n}(\mu_C(u_i)-\overline{\mu_C})(\mu_D(u_i)-\overline{\mu_D})/(n-1)}{S_c * S_D} \tag{2-12}$$

$$\overline{\mu_C}=\frac{\sum_{i=1}^{n}\mu_C(u_i)}{n} \qquad \overline{\mu_D}=\frac{\sum_{i=1}^{n}\mu_D(u_i)}{n} \tag{2-13}$$

$$S_C=\sqrt{\frac{\sum_{i=1}^{n}(\mu_C(u_i)-\overline{\mu_C})^2}{n-1}} \qquad S_D=\sqrt{\frac{\sum_{i=1}^{n}(\mu_D(u_i)-\overline{\mu_D})^2}{n-1}} \tag{2-14}$$

其中 $\overline{\mu_C}$，$\overline{\mu_D}$，S_c，S_D 分别表示模糊子集 C 的平均隶属度，D 的平均隶属度，C 的样本标准方差，D 的样本标准方差。

根据上面的式子计算文本模糊集与类别模糊集的相关程度，最终将文本归属于相关程度最大的那个类别，这样就完成了基于模糊集的分类任务。在推荐系统中，文本的推荐可以转化为分类任务，按照上述的思想，将所有的文档化为两个类别：用户感兴趣文档集，用户不感兴趣文档集。只要将用户提供的感兴趣文档作为训练集，为两个类别建立模糊集，再将待推荐文本表示为模糊集，计算相关程度即可判定是否推荐该文本。目前，还有研究将 vague 集的概念引入推荐系统[25]，达到更加精确地推荐效果。

2.3 信息检索与相关反馈

信息检索[26]是一个帮助用户获取对其有价值信息的过程。他将用户的需求具体化，将对用户有用的文档返回给用户，起到了信息过滤的作用。其中用户的需求由查询表示，该查询可以用上文中提出的各种模型进行表征。模型化之后，将查询与文档集通过某种匹配策略进行匹配，就可以得到相关文档，这些相关文档包含用户所需要的信息。

匮乏的信息一直是信息检索中的一个挑战。研究表明，查询的平均长度很短，而且，由于语言中存在着一词多义和同义词问题，歧义问题随之而来，这给查询效果带来了不好的影响。查询扩展作为一种信息检索系统中的重要技术，能显著提高检索性能。目前的扩展方法可以区分为全局分析方法和局部分析方法两大类[27]。相关反馈就属于查询扩展中的局部分析技术。

全局分析方法主要基于在查询中添加新的查询词之类的操作，对查询进行重构，从而对查询的结果产生影响。采取全局分析最简单的方法是使用某种词典或者知识库，通过这些资源对查询中的每个词用同义词进行扩展。扩展词的权重可以通过在外部资源中体现出的与关键词关系强弱动态指定，一般小于原关键词。比如，使用 WordNet[28]进行查询扩展，将查询中的每个词分别对应于 WordNet 的节点，进行同义词扩展。根据扩展词与原特征词节点的远近情况，可以将扩展词进行分级处理，不同的等级对应不同的比例系数，再结合比例系数求出新词的权重，这样就得到了一个更长更精确的新查询。

局部分析方法主要指的就是相关反馈技术。按照不同的特点，相关反馈技术可以分为三种：显式反馈，隐式反馈，伪相关反馈。下面对这三者分别进行描述，然后介绍经典的查询更新算法 Rocchio 算法。

2.3.1 显式反馈

显式反馈的特点是需要用户参与，对于检索结果中的文档是否相关由用户确定，由用户来提供相关性信息。

首先系统对初始查询执行检索操作，并进行自动的相关性判断，比如通过设定阈值等操作进行文档过滤。检索的结果会得到一批文档，这些文档中往往会包含一些脱离用户需求的低品质文档。接下来，用户需要对该文档集进行标注，标识出与自己的需求相关的部分以及不相关的部分。根据用户的标识情况，系统会对原始查询进行更新，这个过程可能会重复多次，以得到一个尽可能精确的查询。最后，系统就可以通过这个新的查询得到检索结果，从而在更加大的范围内得到更加精确地检索结果。

显然，显示反馈的优点是比较精确，能反映用户的真实需求。但是，由于需要用户的直接参与，会造成系统的检索效率降低，同时用户体验也会随之下降。

2.3.2 隐式反馈

隐式反馈的特点是不需要用户的直接参与。取而代之，系统通过用户的种种行为判断检索结果是否为相关文档。比如，系统可以通过捕捉用户鼠标，键盘的使用情况了解哪些文档曾经被用户点击过，哪些文档用户在其驻留的时间比较长，从而了解检索结果是否满足用户的需求，对查询进行更新。这种策略有时会用于相关 Web 搜索引擎中，在 Web 应用场景中，假如一个用户多次浏览一篇文档，那么这篇文档极有可能是相关文档。这样就可以通过统计页面点击率取代用户对文档的直接判定。

隐式反馈的优点在于不需要用户的参与，结果也较为精确。缺点是用户的隐私会随之被侵犯，可能会造成系统用户的流失。

2.3.3 伪相关反馈

伪相关反馈目前在信息检索中十分常用，主要是因为其不需要向显式反馈那样让用户参与相关性的判断，也不会像隐式反馈那样侵犯用户的隐私。其基本思想是以初始查询进行检索，对检索得到的文档集进行重要度排序，提取排名靠前的 N 个文档视为相关文档，对原始查询进行更新以及扩展，新查询用于检索。在伪相关反馈中系统的判断代替了用户的参与，由于这个特点，结果必然会有所偏差。但是，由于没有用户参与判断，使得系统完全的处于自学习状态，这就使伪相关反馈有很高的研究价值，只需要设计一定的算法，使得用于反馈的文档质量上升，就可以提高系统的准确率以及效率。

伪相关反馈中一个显著的问题就是容易产生查询偏移现象。所谓的查询偏移是指由于系统所使用的反馈文档并不贴合于主题，使得更新后的查询中有效词比例及其权重大大下降，造成查询结果越来越偏移主题的现象。该现象毫无疑问会给系统带来负面的影响。目前，有很多的关于如何减少伪相关反馈中查询偏移的研究，比如在反馈文本集中进一步筛选，挑选更加相关的子集参与反馈等。

2.3.4 Rocchio 算法

上文中简述了相关反馈技术的分类以及不同类别之间的区别。在描述的过程中多次提及了使用相关文档集更新查询的概念。本节中要介绍的 Rocchio 算法就提供了一种用反馈文档更新查询的方案。

Rocchio 算法[29]是一种主要用于向量空间模型的经典算法。它公式化的标明了由旧查询得到新查询的过程。其主要基于以下思想：对于一个最优的查询，其应该与相关文本的相似度最大，与不相关文本的相似度最小。也就是说，我们希望找到一个查询，该查询之于相关文档集的平均相似度与其之于不相关文档集平均相似度之差尽可能的大，如图 2。

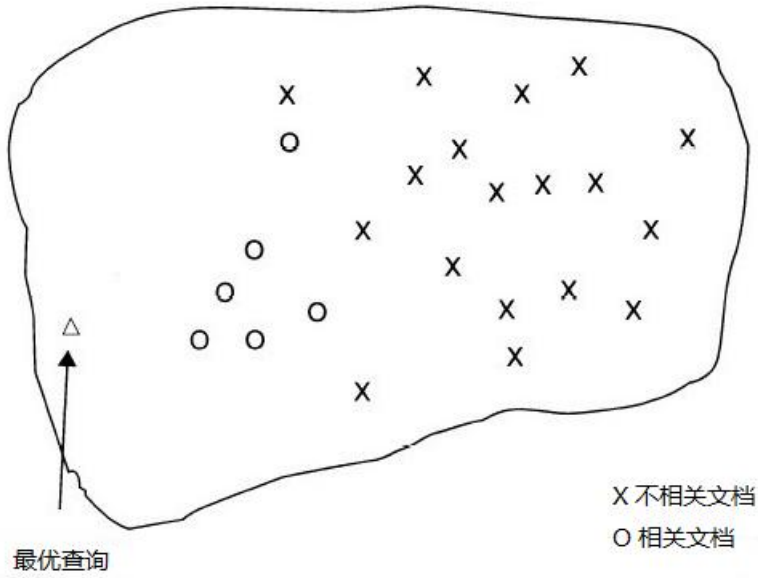

图2 最优查询与相关文档，不相关文档的关系

Figure 2 The optimal query, the related documents and unrelated documents

根据这个思想，可以定义出最优查询 $\overrightarrow{q_{best}}$ 见式（2-15）。

$$\overrightarrow{q_{best}}=\frac{1}{r}\sum_{i=1}^{r}\overrightarrow{d_{rel}}^{(i)}-\frac{1}{n}\sum_{j=1}^{n}\overrightarrow{d_{irrel}}^{(j)} \tag{2-15}$$

其中 r 为相关文档数目，n 为不相关文档数目，$\overrightarrow{d_{rel}}^{(i)}$ 为第 i 篇相关文档对应的向量，$\overrightarrow{d_{irrel}}^{(j)}$ 为第 j 篇不相关文档对应的向量。该式的意义为最有查询为相关文档集与不相关文档集质心之差。由于查询初始时相关文档的划分是未知的，上面的式子实际意义并不大。为了灵活使用，上式可以推广为式（2-16）

$$\overrightarrow{q_{new}}=\alpha\overrightarrow{q_{old}}+\beta\frac{1}{r}\sum_{i=1}^{r}\overrightarrow{d_{rel}}^{(i)}-\gamma\frac{1}{n}\sum_{j=1}^{n}\overrightarrow{d_{irrel}}^{(j)} \tag{2-16}$$

其中 $\overrightarrow{q_{new}}$ 表示新查询，$\overrightarrow{q_{old}}$ 表示旧的查询，α，β，γ 为用于调节平衡的系数因子，为了调节方便，通常式 α=1。该式子体现了如何由反馈信息更新查询的过程，其本质为使查询更加靠近相关文档的中心，更加远离不相关文档中心，见图 3，以此提升查询的质量，提高检索精度。上式给自学习的过程提供了指导思想，本文将使用改进的 Rocchio 算法更新用户模板。

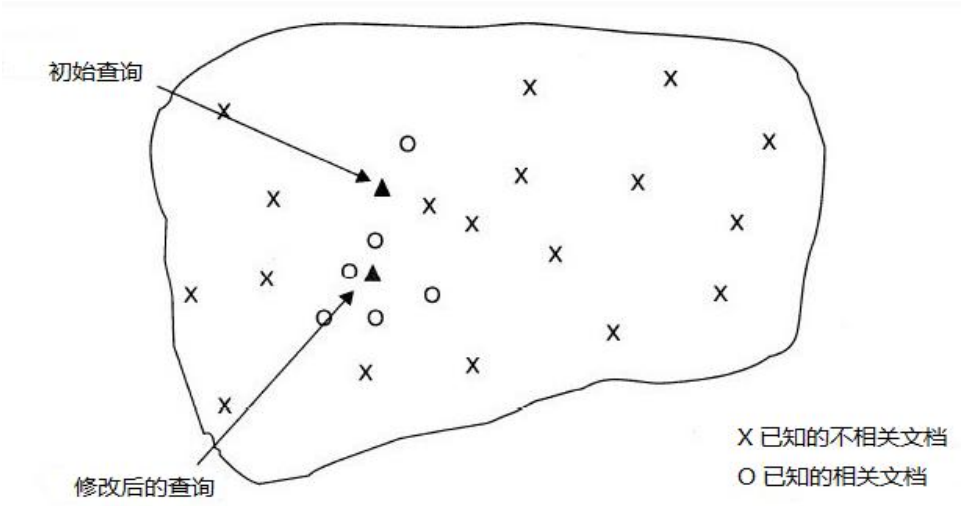

图3 新旧查询与相关文档，不相关文档的关系

Figure 3 The new query, the old query, the related documents and unrelated documents

2.4 本章小结

由于本文主要研究的是基于内容的文本推荐。本章首先研究了推荐系统中文本处理特定步骤的理论基础。对于分词步骤来说，主要的方法包括基于词典的分词方法等三大类。对于推荐系统中的文本表示，可以使用的模型有布尔模型，向量空间模型，概率模型，基于模糊集的模型等，这几种模型在不同的推荐系统都曾有过应用。本章对上述内容进行了简要分析。其次，本章介绍了信息检索中相关反馈的概念，该概念为推荐系统中的用户模板自学习提供了思路，并且介绍了 Rocchio 算法，公式化的给出了模板更新的过程。

3 基于内容的自适应推荐系统模型

本章首先给出了基于内容的自适应文本推荐系统整体架构图，然后详细的描述了其中的基于 TextRank 算法的初始模板建立模型和基于伪相关反馈的用户模板更新模型。对于基于 TextRank 算法的初始模板建立模型来说，涉及的主要操作有：预处理，基于《同义词词林》的语义分析与确定，层次聚类，构建 TextRank 图模型并计算，义项综合与拓展，模板生成等。对基于伪相关反馈的用户模板更新模型来说，涉及的主要操作有：用户模板划分（划分为用户需求模板和反馈部分模板），用户需求模板更新，反馈部分模板更新。其中反馈部分基于伪相关反馈，步骤包括：引入多种因子挑选反馈文档，基于共现性挑选拓展特征词，用改进的 Rocchio 算法更新模板等。

3.1 系统整体架构图

本文中的基于内容的自适应文本推荐系统主要功能是：

（1）在系统添加新用户时，由用户提供的少量感兴趣文档为用户建立起一个精确的用户模板，对待推荐文本集中的文档进行推荐，改善新用户带来的冷启动问题。

（2）为了实现精确推荐，用户的模板不能是一成不变的，该系统会动态的对模板进行更新。

总的来说，该系统就是通过一系列操作实现推荐系统在各个时间状态下的精确推荐，包括在系统初始时以及系统稳定时。

本文所设计的自适应文本推荐系统输入为少量用户感兴趣文档，待推荐文档集等。对应的资源为作者通过网络爬虫从相关新闻网站的各个板块获取并筛选。输出为用户模板，待推荐文档集中的推荐文本。该系统主要包含两个部分：基于 TextRank 算法的初始模板建立模型，基于伪相关反馈的用户模板更新模型。这两个部分会在 3.2 节和 3.3 节分别进行详尽的描述。系统的整体架构图见图 4。

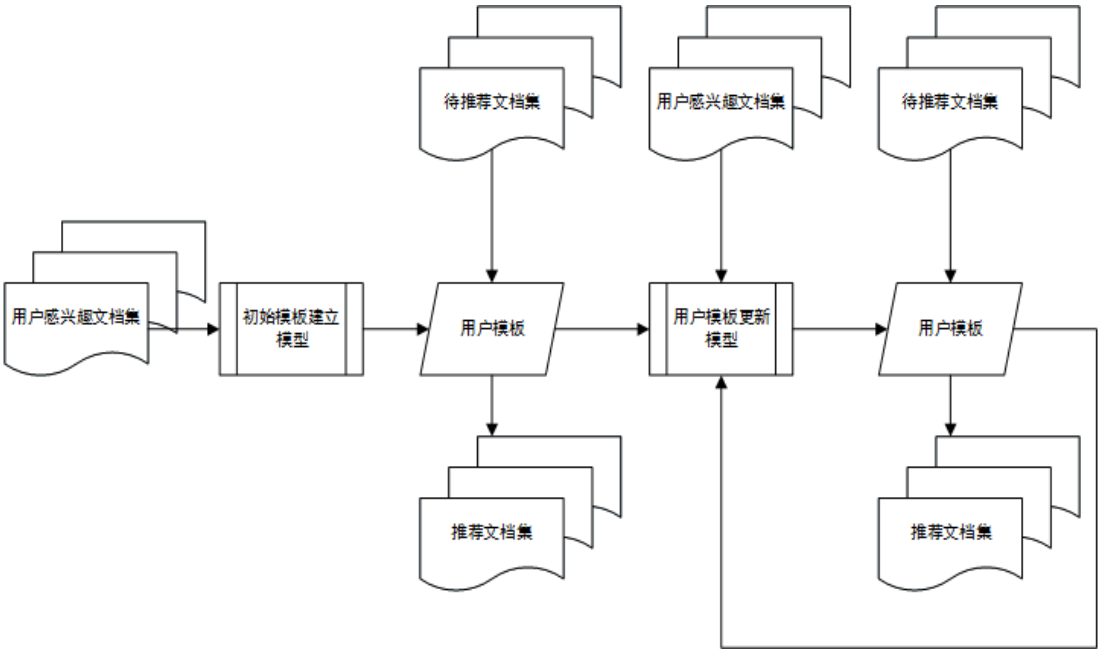

图4 系统整体模架构图

Figure 4 Architecture of the entire system

3.2 基于 TextRank 算法的初始模板建立模型

在基于内容的推荐系统中，当系统添加新用户时，会有冷启动的问题，此时新用户已标识的感兴趣文档数量很少，建立一个准确的用户模板相对困难但却至关重要。因为如果引入大量的用户不感兴趣的噪声词，接下来系统必然会推荐相关用户不感兴趣文档，模板在自学习更新时必然会造成偏移，影响整个系统的推荐效果。因此，在系统初始时，必须从少量用户信息中准确地提取出用户兴趣模板，尽可能的减少噪声的引入。否则会在后期更新模板时产生偏移性问题，造成推荐的不准确。

为了解决这一问题，实现文本推荐系统在初始状态时的精确推荐，本文提出了一种基于 TextRank 算法的初始模板建立方法，以减少模板中的噪声。该算法的提出是基于一种对实际情况的考察：当用户在搜取感兴趣文档时，每次往往会以集簇的形式获取感兴趣文档。比如当用户观看体育方面新闻时，对世界杯方面感兴趣，他往往会阅读不止一篇关于世界杯的文章。这样，尽管系统初始时用户标识的感兴趣文章不多，但由于其具有集簇性，利用此特性，我们就可以建立起一个相对准确的用户初始模板。该算法主要使用了 TextRank 算法的思想，以义项为单位建立图模型。由于在中文中存在着大量的一词多义和同义词现象，传统的通过字符串匹配不涉及上下文的方法很难确定两个词之间的关系。所以算法中使用《同义词词林》这个外部资源，确定词语的词义，并且计算义项之间相似度，由此建立 TextRank 模型。最终建立用户模板。主要步骤如下：

（1）对拥有的少量用户文本进行预处理，包括分词，去停用词，词性标注操作。

（2）按照词性对词语进行进一步筛选。

（3）确定每个词的义项。这里使用《同义词词林》确定语义。

（4）聚类处理预处理后的文档，由于预先不知道目标信息集合内到底包含多少类别，本文采用自底向上的层次聚类方法。

（5）对聚类得到的每个类别分别以义项为单位构建 TextRank 模型，并引入相似度影响因子，共现度影响因子，类权重影响因子对 TextRank 模型中的概率转移矩阵进行修正。迭代之后得到每个类中最为关键的 N 个义项。

（6）对每类的关键义项进行综合，计算权重，得到一个由词义项组成的用户模板。

（7）将义项通过《同义词词林》拓展为关键词，最终的到初始用户模板。

整个模型的架构图如下：

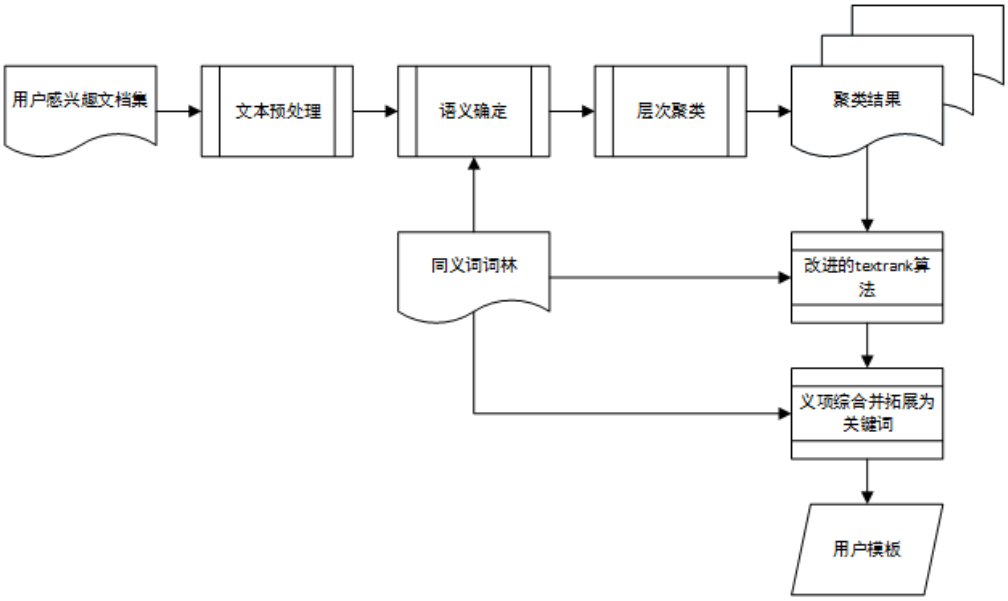

图5 基于TextRank算法的初始模板建立模型架构图

Figure 5 Architecture of the model of building initial profile based on TextRank

3.3 基于伪相关反馈的用户模板更新模型

在初始用户模板建立之后，系统便可以通过计算待推荐文档集中文本与模板的相关程度实现个性化推荐。但是，用户兴趣是会随着时间变化而变化的，也就是说，用户感兴趣文档集中的文档数目会逐渐增加。假如不对用户模板进行及时的更新，系统显然滞后于用户需求，产生偏移性现象，随之带来不好的用户体验。同时，由于系统初始时用户信息较少，尽管前面已经建立了较为精确的初始用户模板，但也仅仅是依赖于用户提供的少量文档，这样就往往会造成推荐范围过窄。比如说由初始文档集提取的用户模板中包含“裁判”这个特征词，虽然一些文章中以“犯规”这个词为主要特征，但是由于不包含“裁判”这个词，这些文档极大可能不被推荐。而“犯规”这个词与“裁判”这个词的相关性主要是由共现性体现出来

的，用外部词典对模板进行扩展的方法无法实现该词的添加。所以，需要在更新用户模板的过程中引入伪相关反馈技术，以扩大推荐范围。

在基于内容的推荐系统中，用户模板的更新和用户模板的建立一样是系统核心部分之一。目前有许多这方面的研究。比如有学者研究在更新用户配置文件的过程中使用了自适应过滤技术。还有一些研究专注于自适应更新模板是阈值的设定问题。本文中的用户模板更新模型主要使用了伪相关反馈技术更新用户模板，实现用户模板的自学习更新。

在信息检索中，由于初始查询所含信息较少，产生的结果之一就是检索范围较窄。这时候需要引入查询扩展技术对原始查询进行扩展，相关反馈就是查询扩展技术中主要方法之一。在相关反馈中，显示相关反馈需要用户的直接参与，伪相关反馈则不需要用户的参与，可以实现查询的自学习扩展。在推荐系统中，我们可以把用户模型在某种程度上视为一个查询，推荐文本视为查询的检索结果。这样，我们可以通过相关反馈技术对用户模板进行扩展，使其更加精确。基于伪相关反馈的用户模板更新算法的主要步骤如下：

（1）将用户模板划分为两部分：用户需求模板、反馈部分模板。其中用户需求模板由用户自己提供的感兴趣文档集确定。反馈部分模板由伪相关反馈产生与更新。

（2）确定两部分模板的加权方式。在用户模板中，用户需求模板和反馈部分模板所占比重应该是随着系统状态逐渐变化的。在系统的初始阶段，由于用户提供的感兴趣文档数目较少，此时反馈部分模板有着较大的作用；在系统逐渐稳定后，此时系统拥有的用户信息越来越多，用户感兴趣文档集中的文档数逐渐增大。此时用户需求模板已经趋于完整与准确，反馈部分模板所占比重应该越来越小，否则可能会产生偏移性问题。

（3）用户需求模板更新。这部分的模板更新可以视为显式相关反馈，更新模板。

（4）挑选用于伪相关反馈的文档。执行聚类操作，引入多种影响因子综合评定反馈文档的质量，挑选评分最高的作为用于更新模板的反馈文档。

（5）挑选反馈文档中的关键词。为了减少伪相关反馈中的偏移性问题，这里挑选关键词主要考虑与模板中特征项的共现性。

（6）用改进的 Rocchio 算法更新反馈部分模板。

（7）加权得到新的用户模板。

整个模型的架构图如下：

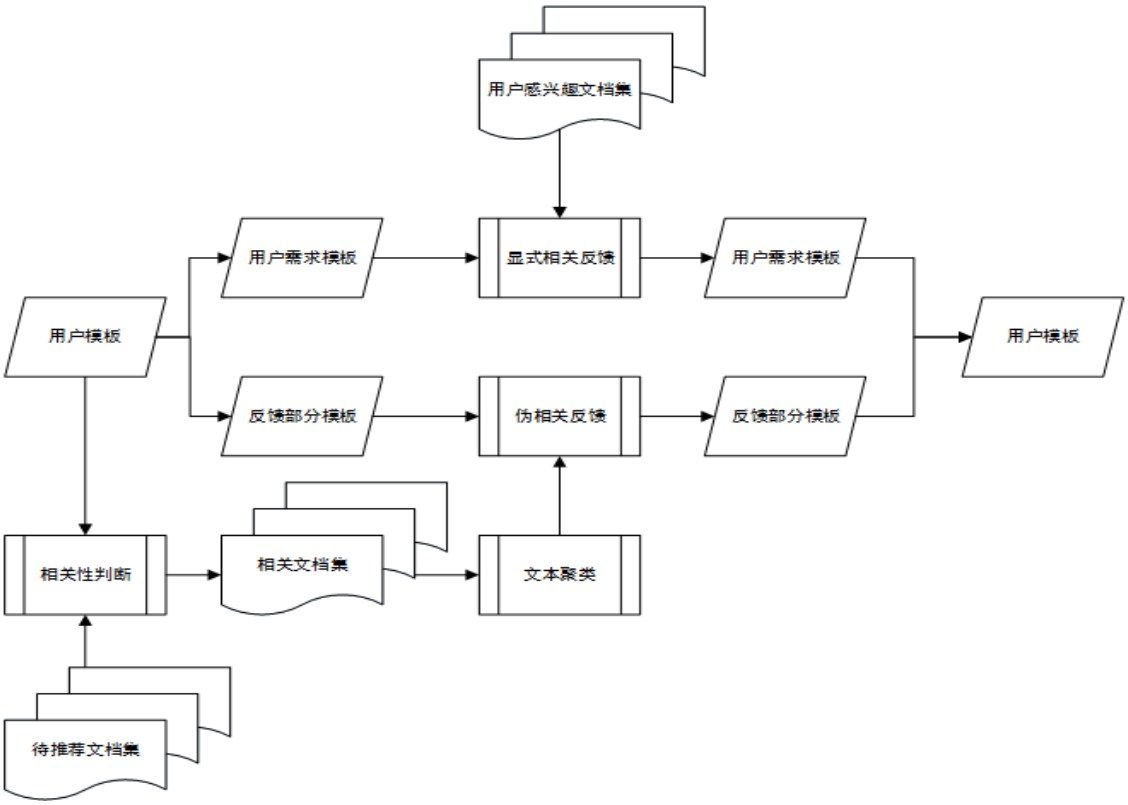

图6 基于伪相关反馈的用户模板更新模型架构图

Figure 6 Architecture of the model of updating user profile based on pseudo feedback

3.4 本章小结

本章首先给出了基于内容的自适应文本推荐系统整体架构图，其中主要包括基于TextRank 算法的初始模板建立模型和基于伪相关反馈的用户模板更新模型。前者负责在系统的初始阶段生成一个能反映用户真实需求的用户模板；后者负责在用户兴趣变化时动态的更新用户模板，保证系统推荐持续精确。3.2 节主要介绍了基于 TextRank 算法的初始模板建立模型，模型涉及预处理，基于《同义词词林》的语义分析与确定，层次聚类，构建 TextRank 图模型并计算，义项综合与拓展，模板生成等。3.3 节介绍了基于伪相关反馈的用户模板更新模型，涉及用户模板划分（划分为用户需求模板和反馈部分模板），用户需求模板更新，反馈部分模板更新。反馈部分模板的更新步骤包括：引入多种因子挑选反馈文档，基于共现性挑选拓展特征词，用改进的 Rocchio 算法更新模板。

第四章，第五章，第六章将分别对基于 TextRank 算法的初始模板建立模型及基于伪相关反馈的用户模板更新模型进行详细的讨论。

4 基于 TextRank 的内容推荐系统用户模板构建方法

本章主要对基于 TextRank 的内容推荐系统用户模板构建方法进行详细介绍。首先，对该算法的主题思想进行描述。然后按照算法的流程分别介绍了预处理，基于《同义词词林》的语义分析与确定，层次聚类，构建 TextRank 图模型与计算，义项综合与拓展，模板生成等模块。在预处理部分，介绍了去停用词，分词及词性标注等步骤，并介绍了一种基于无词典的分词方法。在语义确定模块，主要介绍了《同义词词林》的背景知识以及如何使用它来确定词义项。在聚类模块，介绍了主要的聚类算法，以及此处使用层次聚类的原因。最后，描述了 TextRank 图模型的建立方法，分析了各种影响因子被引入的原因以及如何在图模型中发挥作用。

4.1 算法概述

在推荐系统添加新用户以及添加新产品时，冷启动问题会随之而来。所谓的冷启动问题是指由于初始信息匮乏造成的无法推荐或者推荐不精确问题。新用户没有或者只有少量产品选择记录，新产品没有被选择过或被评分的记录都会造成冷启动问题。协同过滤推荐算法面对这种问题很难处理，对于基于内容的推荐来说，冷启动问题的改善由于工作机理的不同已经优于协同过滤算法。但是，当系统添加新用户时，基于内容的推荐系统很容易产生推荐偏差。此时新用户已标识的感兴趣文档数量很少，建立一个准确的用户模板相对困难但却至关重要。假如在模板中引入大量的用户不感兴趣的噪声词，接下来系统必然会推荐相关用户不感兴趣文档，就会产生推荐偏差。同时，由于相关反馈依赖于相关文档的质量，一个差的用户模板产生的相关文档集必然是质量不高的，这样模板在自学习更新时必然会造成偏移，影响整个系统的推荐效果。因此，在系统初始时，必须从少量用户信息中准确地提取出用户兴趣模板，尽可能的减少噪声的引入。

本章中的基于 TextRank 的内容推荐系统用户模板构建方法是基于一种对实际情况的考察：当用户在搜取感兴趣文档时，每次往往会以集簇的形式获取感兴趣文档。这样，尽管系统初始时用户标识的感兴趣文章不多，但利用集簇性，我们就可以建立起一个相对准确的用户初始模板。首先，对拥有的少量用户文本进行预处理并且确定每个词的义项，这里使用《同义词词林》确定语义。之所以要确定词义项是因为在中文中由于存在着大量的一词多义和同义词现象，传统的通过字符串匹配不涉及上下文的方法很难确定两个词之间的关系，这样以词为节点构建的图并没有以义项节点建立的图模型合理。其次，聚类处理预处理后的文档。这个操作就是要找出少量文档中的集簇性。由于预先不知道目标信息集合内到底包含多少类别，本文采用自底向上的层次聚类方法。接下来，对聚类得到的每个类别分别以义项为单位构建 TextRank 模型，并引入相似度影响因子，共现度影响因子，类权重影响因子对 TextRank 模型中的概率转移矩阵进行修正。迭代之后得到每个类中最为关键的 N 个义项。最后，将每类的关键义项进行综合，通过《同义词词林》扩展为关键词，计算权重，得到最终的初始用

户模板。下面几节将分别对算法中的各个步骤进行详细描述。

4.2 预处理

预处理是文本处理的基础，它可以将原始字符串进行转换，生成一个由有意义词条组成的词条串。同时，对词条串的内容进行过滤，减少噪声，提高文本处理系统的效率。在本章介绍的算法中，预处理主要包括：分词与词性标注；词串过滤。

4.2.1 分词及词性标注

对于推荐系统中的文本信息，需要对其进行分词以及词性标注操作，这是整个算法的基础。对于分词部分，本文使用 stanford segmenter 完成；对于词性标注部分，本文使用了 stanford postagger。其中 stanford segmenter 由 java 实现，分词基于 CRFs，可以达到较好的分词效果。对于 stanford postagger，可以实现词性标注的功能。以“今天天气相当不错。”这个短句为例，经过上述工具的分词以及词性标注操作之后可以得到的结果为“今天#NT 天气#NN 相当#AD 不错。#VA”。其中部分词性标注集见下表：

表 1 词性标注集

Table 1 POS tag set

标注	含义	标注	含义
NN	常用名词	NR	固有名词
NT	时间名词	PN	代词
VV	动词	VA	表语形容词
AD	副词	AS	内容标记
DP	限定词短语	VRD	动补复合语
QP	量词短语	PP	介词短语
NP	名词短语	VP	动词短语
LCP	方位词短语	AS	内容标记

4.2.2 词串过滤

在完成分词及词性标注之后，需要执行过滤操作。假如使用全文标引，即使用文本中出现的所有词作为特征，参与下一步的操作，缺点十分明显。首先，一些常用词在文本中有着较高的频率却有着较低的区分能力，即与文章的主题关联不大，把这些噪声引入显然对后面的操作毫无益处。其次，在使用向量空间模型表示文本时，全文标引的文本向量的维数很大，使系统效率会急剧下降，甚至根本无法处理。因此，必须要使用一些过滤策略。

首先，系统对分词结果实施去停用词操作。类似“啊”，“哎”，“不但”，“那么”等词对

模型中文本的表示贡献不大，过滤掉这些词有助于提高系统效率以及精度。其次，由于本章算法的主要目的是建立一个能准确的体现用户兴趣的初始用户模板，那么我们可以通过词性对分词结果进行进一步的筛选。在用户模板中，动词和名词辨识度较大，价值高，所以保留其中的动词和名词，其余剔除。这样就完成了词串过滤。

4.2.3 一种基于无字典的快速分词方法

在 4.2.1 节中介绍的分词方法是基于字典匹配的分词方法。由本文在第二章的讨论可知，使用这种方法必须要加载词典，并且在某些特有名词的识别上效果并不是很好。所以本系统还提供了一种分词的备选方案。这种分词方案主要基于统计，可以快速实现分词的目的。

当需要对一篇文档进行分词处理时，我们的目的实际上就是要提取文章中的高频子串。利于基于统计的思想，如果一个短字符串在该篇文档中多次出现，那么这个字符串极有可能为一个词或词组。利用这个思路，我们只需要对全文进行若干次扫描，即可分解出一些高频词串，达到分词的目的。这样做尽管那些出现频率较少的词会被忽略，但是由于我们在挑选一篇文档特征词时，频率是主要的考虑方向，所以这样做是符合我们需求的。

该方法的第二个依据源于汉语的一些特点。在汉语中，除了一些标点符号天然的将文本分为多个小部分之外，对于分词需求来说，还存在着另一些特别的分界点，这些分界点就是一些特定的字，比如“的”，“也”等等，这些字基本上都是以单独的形式出现在文本中的，很少以词的形式出现。那么利用这些分界点，就可以把文章切分为许多的小块，实现分词操作。

该方法的第三个依据体现在扫描的实现难易上。汉语中的所有字可以按照其 GBK 编码哈希到一个二维数组中，坐标可由高低字节计算得到，计算方法如式（4-1），（4-2）。这样，对于扫描过程中一些数据的存储就会十分的方便。

$$\text{row} = high\text{-}0xa0 \tag{4-1}$$

$$column = low\text{-}0xa0 \tag{4-2}$$

其中 row 为行号，column 为列号。high 表示汉字编码的高字节，column 表示汉字编码的低字节。

该算法的流程图见图 7，图 8。

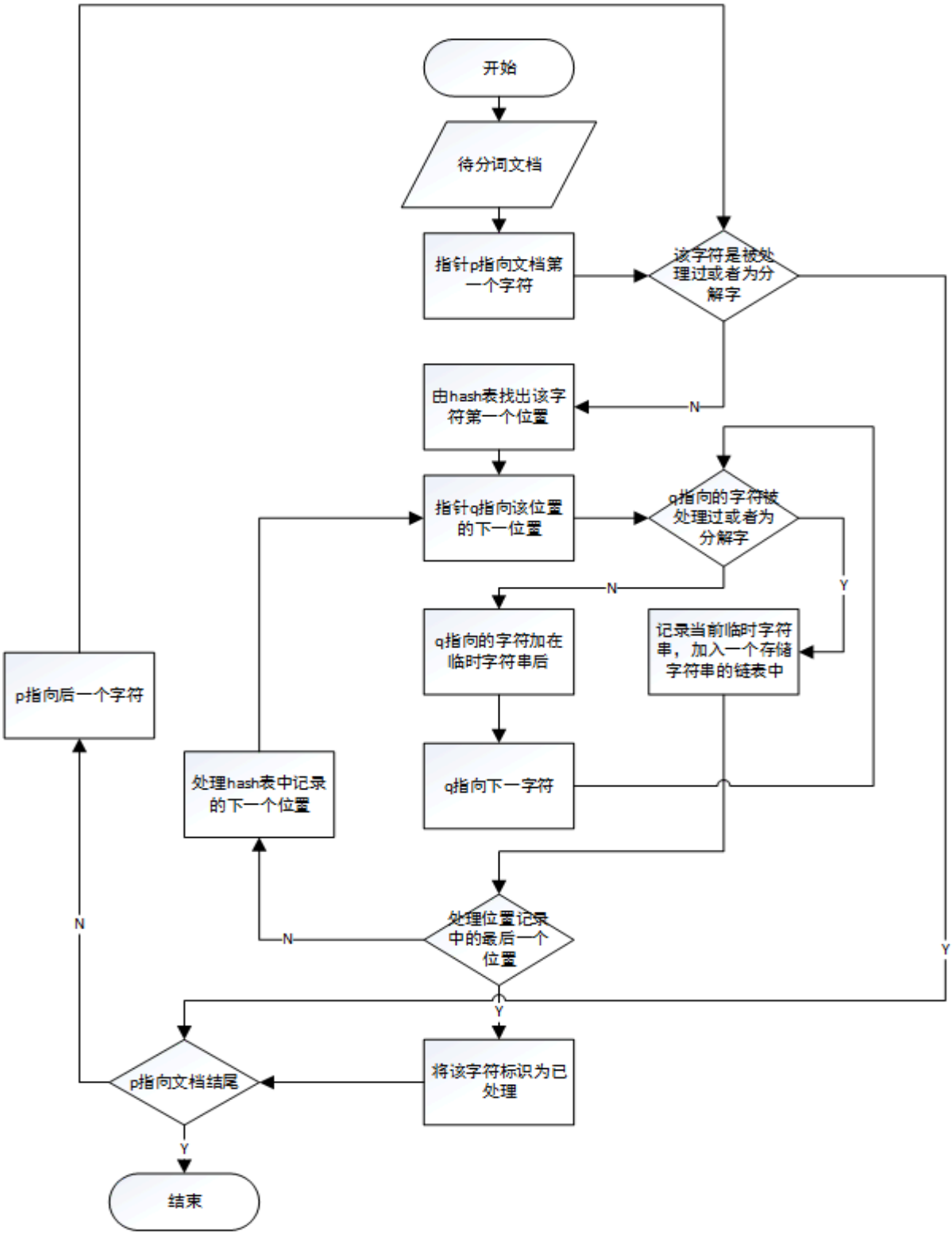

图7 第三次扫描全文对应流程图

Figure 7 Flow chart of the third time to scan the full text

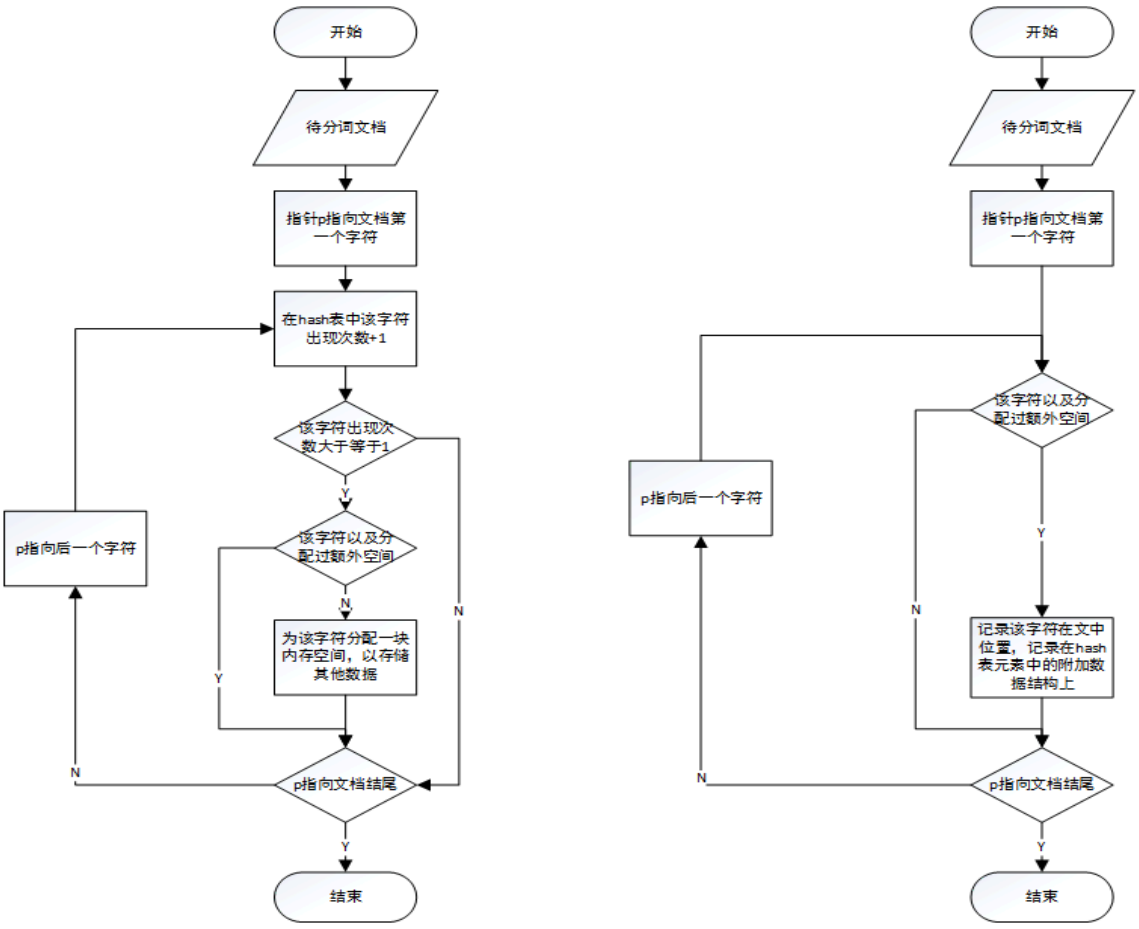

图8 第一次与第二次扫描全文对应流程图

Figure 8 Flow chart of the first and second time to scan the full text

经过三次扫描后，可以得到大量的词串，对这些词串进行词频统计，去除停用词。然后结合删除同频子串等尾处理操作对所得到的词串进行进一步的筛选，即可得到分词结果。

4.3 基于《同义词词林》的语义确定方法

在本章中的基于 TextRank 的用户模板构建方法中，理论上需要通过词与词之间的相似度建立图模型。但是，在自然语言处理中，消除歧义一直是一个重要方面。特别是在中文中，由于存在着大量的一词多义和同义词现象，传统的通过字符串匹配不涉及上下文的方法很难确定两个词之间的关系。比如“红领巾”在指代小学生的时候很明显和“衣服”没关系，但是不为指代时就与“衣服”有关系。所以要想正确的建立词语间的关系图，必须要先由上下文确定词语的词义，以义项节点建立的图模型，这比单独的以词为节点的图模型要合理。在建立初始模板时，所拥有的用户感兴趣文本并不多，所以确定词义的工作量并不大，却能对

提高精度有较为明显的效果。本节介绍通过《同义词词林》这个外部资源确定词语的词义的方法。下面几节会通过计算义项之间相似度，建立 TextRank 模型。

4.3.1 《同义词词林》介绍

《同义词词林》[30]最初是由梅家驹等编纂的，以同义词，相关词的形式组织起来的词典。哈尔滨工业大学信息检索研究室综合了多种资源，对该词典进行了更新，完成了词典《同义词词林扩展版》。该词典对原来的词语进行筛选，并进行了相关扩展，最终收录大约七万条词语，并且按照这些词语的意义进行组织，属于一部大的类义词典。

《同义词词林扩展版》整体为一个树状结构，其将词汇分为三类：大类，中类，小类。数量分别为 12 个，97 个，1400 个。小类由许多词汇构成，按照其中词汇的词义关系，又可以将小类划分为许多词群。对于词群中的词汇还可以进一步划分，按照词义相关性及相似性，词群被划分为若干行。比如“计算机”与“电脑”就在同一行。这样，词林就可以划分为五层。每一层粒度逐渐细化，直到划分至原子词群。见图 9。

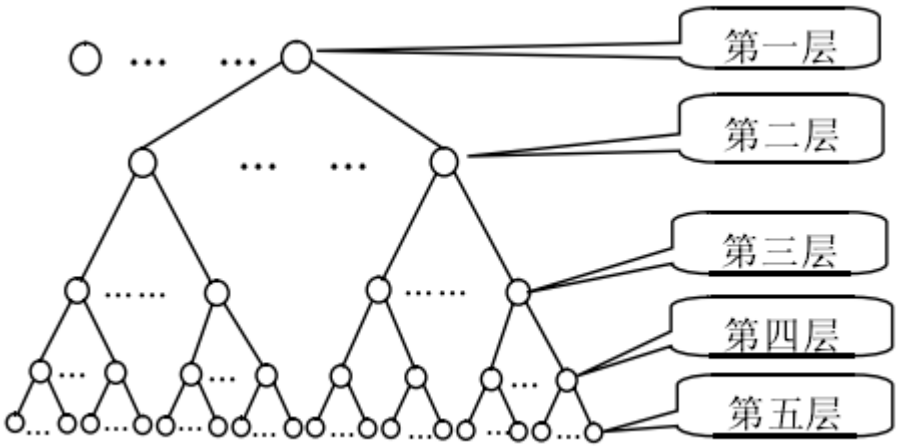

图9 《同义词词林》层次结构

Figure 9 Hierarchical structure of TongYiCi CiLin

《同义词词林扩展版》的结构是由其编码方式体现的。其总共提供了五层编码，这也给我们表述词的义项提供了便利。以“Ab03A07# 男孩子，少男”为例，其中“Ab03A07#”为该义项的编码，义项中包含的词有“男孩子”，“少男”。其中“A”为第一层编码，“b”为第二层编码，“03”为第三层编码，“A”为第四层编码，“07”为第五层编码，“#”表示该义项内的词语为相关关系。具体的编码对应关系见表 4.2。对于一次多义的情况，一个词语会有多个义项编码。如词“八角”为一个多义词，其对应的编码有“Bh09A12=”，“Br07A02=”。

表 2 《词林》编码对应关系

Table 2 Code meaning in CiLin

级别	第一层	第一层	第一层	第一层	第一层	附加位
性质	大类	中类	小类	词群	行	词汇关系
表示方式	大写英文字母	小写英文字母	两位十进制整数	大写英文字母	两位十进制整数	“=”或“#”或者“@”

4.3.2 文本词义项确定方法

在使用《同义词词林》确定词义项的方法中，会使用到文献[31]中的词林义项相似度计算方法。由上一节可以知道，词林是以树状结构组织的，故词林中的义项相似度可以通过节点在树结构的分布情况处理得到。

设 M_i 与 M_j 分别为同义词词林中的两个义项，那么 M_i 与 M_j 的相似度 $Sim(M_i, M_j)$ 计算方法如下：

（1）若 M_i 和 M_j 在第一层编码出现不同：

$$Sim(M_i, M_j) = s_1 \tag{4-3}$$

（2）若 M_i 和 M_j 在第二层编码出现不同：

$$Sim(M_i, M_j) = 1 \times s_2 \times (\frac{n-k+1}{n}) \times \cos(\mathrm{n} \times \frac{\pi}{180}) \tag{4-4}$$

（3）若 M_i 和 M_j 在第三层编码出现不同：

$$Sim(M_i, M_j) = 1 \times 1 \times s_3 \times (\frac{n-k+1}{n}) \times \cos(\mathrm{n} \times \frac{\pi}{180}) \tag{4-5}$$

（4）若 M_i 和 M_j 在第四层编码出现不同：

$$Sim(M_i, M_j) = 1 \times 1 \times 1 \times s_4 \times (\frac{n-k+1}{n}) \times \cos(\mathrm{n} \times \frac{\pi}{180}) \tag{4-6}$$

（5）若 M_i 和 M_j 在第五层编码出现不同：

$$Sim(M_i, M_j) = 1 \times 1 \times 1 \times 1 \times s_5 \times (\frac{n-k+1}{n}) \times \cos(\mathrm{n} \times \frac{\pi}{180}) \tag{4-7}$$

其中 n 表示两节点的公共层上节点的总数，而 k 表示了两个义项之间的距离，这两者体现了在计算相似度时对树结构的考虑。举例来说，假如 M_i 与 M_j 在第二层编码出现不同，第一层的编码均为“A”，那么 n 的取值为所有编码首字母为“A”的义项个数，k 为 M_i 与 M_j 在

第二层上的距离。式中的s_1，s_2，s_3，s_4，s_5为常数影响因子，取值分别为：0.1，0.6，0.8，0.9，0.96。最后计算的得到的$Sim(M_i,M_j)$满足取值范围为[0，1]。

义项相似度可以计算之后，就可以通过《同义词词林》结合文本上下文确定词语的义项了。算法步骤如下：

(1) 对用户感兴趣文档进行预处理，包括分词，去停用词，词性标注。

(2) 因为在用户模板中，动词和名词辨识度大，价值高，所以保留其中的动词和名词，其余剔除，对于一词多义中包含多个词性的，只要含动词词性或名词词性则保留，进入下一步。

(3) 确定保留词 W 的义项。

其中（1）和（2）的处理方法见本章中的 4.2 节。对于（3）分为两种情况：W 在正文中；W 在标题中，下面对这两者情况分别讨论。

若 W 在正文中，步骤如下：

(a) 在同义词词林中搜索 W 的所有义项，组成集合$Q=\{s_1,s_2,s_3..s_n\ldots\}$，其中$s_n$为 W 的义项（如 W=“爱慕”，则 Q={Gb09A01=，Gb09B01=，Gb13A01=}，其中 Gb09A01=为同义词词林中的语义编码）。若 Q 中只有一个元素，则义项已确定，否则进入下一步。

(b) 因为语义的关联不能脱离句子，此处以句子为单位，r=R 为半径做一个窗口，即取 W 所在句子的前 r 个句子和后 r 个句子，挑选出窗口内所有的词，构成集合$P=\{W_1,W_2,W_3,W_4\ldots W_k\ldots\}$，其中$W_k$为 W 的邻近词。

(c) 在同义词词林中搜索 P 中每个词的语义，每个词 W_i 对应一个语义集合$Q_i=\{s_{i1},s_{i2}\ldots s_{in}\ldots\}$。

(d) 计算集合 Q 中每个义项与集合 Q_i 中每个义项的相似度$sim(s_j,s_{ik})$，计算方法在上面已经叙述过。得到 W 的义项s_j和集合 Q_i 的相似度定义如下：

$$sim(s_j,Q_i)=\ Max\ sim(s_j,s_{ik}) \tag{4-8}$$

(e) 对 W 的每个义项计算得分：

$$\mathrm{Score}(s_j)=\frac{\sum_{i=1}^{T}\lambda_i*\mathrm{sim}(s_j,\ Q_i)}{T} \tag{4-9}$$

其中 T 为集合 P 中词总数。其中λ_i为修正系数，代表着词 W 与 W_i 有语义关联的可能性大小。λ_i有如下特点，在 W 所在句子中出现的所有词以及窗口中半径为 1 处的句子中出现的所有词应该具有相同的 λ 值，出现在窗口其他句子中的词随着距离的增大，与 W 有语义关联性的可能性减小。为计算方便，在这里把距离 W 大于 N 的句子视为没有语义相关性，这是因为与每个词最相关的词往往出现在本句以及相邻句子中。所以得到λ_i的表达式如下：

$$\lambda_i = \begin{cases} \dfrac{\log\dfrac{N+2}{2}}{\sum_{j=2}^{N} L_j * \log\dfrac{N+2}{j+1} + \sum_{j=0}^{1} L_j * \log\dfrac{N+2}{2}} & 0 \leqq r_i \leqq 1 \\ \dfrac{\log\dfrac{N+2}{r_i+1}}{\sum_{j=2}^{N} L_j * \log\dfrac{N+2}{j+1} + \sum_{j=0}^{1} L_j * \log\dfrac{N+2}{2}} & 1 < r_i \leqq N \end{cases} \quad (4\text{-}10)$$

其中 N 为前面选取窗口的半径。r_i 表示词 w_i 在窗口中距离中心的位置，$r_i = 0$ 表示该词汇与 W 在同一句中，L_j 表示距离中心为 j 的句子中的词汇总数量，注意除了距离为 0 的情况，其他距离中心为某长度的句子均有对称的两句。

(f) 从 W 的语义集合 Q 中选取得分最高的义项作为 W 在此处的义项。

若 W 在标题中，确定义项方法如下：

(a) 与 W 在正文中时操作（a）相同，在同义词词林中搜索 W 的所有义项，组成集合 $Q = \{s_1, s_2, s_3 .. s_n \ldots\}$，其中 s_n 为 W 的义项。若 Q 中只有一个元素，则义项已确定，否则进入下一步。

(b) 对正文中的所有保留词汇进行词频统计，保留频率最高的前 M 个词，组成集合 $P = \{W_1, W_2, W_3, W_4 \ldots W_M \ldots\}$。

(c) 在同义词词林中搜索 P 中每个词的语义，每个词 W_i 对应一个语义集合 $Q_i = \{s_{i1}, s_{i2} \ldots s_{in} \ldots\}$。

(d) 得到 W 的义项 s_j 和集合 Q_i 的相似度，义项与集合的相似度定义见式（4-8）。

(e) 对 W 的每个语义计算得分：

$$Score(s_j) = \frac{\sum_{i=1}^{M} sim(s_j, Q_i)}{M} \quad (4\text{-}11)$$

(f) 取得分最高的最为标题中 W 的义项。

经过上面的操作，文本中无论是正文还是标题中的候选词均已经确定了其含义，这样就可以执行下一步操作了。

4.4 用户已标识文本聚类

在本章的开头时，我们曾描述过一种普遍现象：当用户在搜取感兴趣文档时，每次往往会以集簇的形式获取感兴趣文档。这样，为了建立起一个相对准确的用户初始模板，我们可以利用这种集簇性。尽管系统初始时用户标识的感兴趣文章不多，我们仍然可以通过聚类等操作建立精确用户模板。经过上一节的一些操作，文本中每个保留词的义项已经确定，本节将介绍对这些处理后的文本如何进行聚类操作。

4.4.1 聚类算法

聚类是文本处理中的一个重要应用。其可以对众多文本进行分析，无指导的将这些文档聚集为若干集簇。这些集簇满足同一个集簇中的文档相似度很高，不同集簇中的文档相似度很低。现有的聚类算法可以划分为几类，如层次聚类算法，基于划分的聚类算法，基于网络和密度的聚类算法等，下面对其中几个进行介绍。

4.4.1.1 层次聚类算法

层次聚类算法[32]的主要思想是采用某种数据连接规则，反复将文档集进行聚合或者分裂，形成一个层次性问题，最终达到聚类的效果。该算法很适合数据量较少时的聚类问题。相较于其他聚类算法，它的一个显著特点就是不需要指定划分类别数以及指明初始划分。

层次聚类算法具体可以划分为凝聚式以及分裂式。凝聚式层次聚类采用自底向上的策略，其在算法初始时将每个对象作为一个集簇，按照相关性逐渐将集簇合并，直至满足终止条件。分裂式层次聚类基于自顶向下的策略，它与凝聚式正好相反，初始时会将所有文本放于一个大的集簇中，然后按照相关条件逐渐对该集簇进行分解，得到小集簇，直至满足终止条件。示意图见图 10。常用的层次聚类算法由 ROCK 算法，CHAMELEON 算法，CURE 算法等。

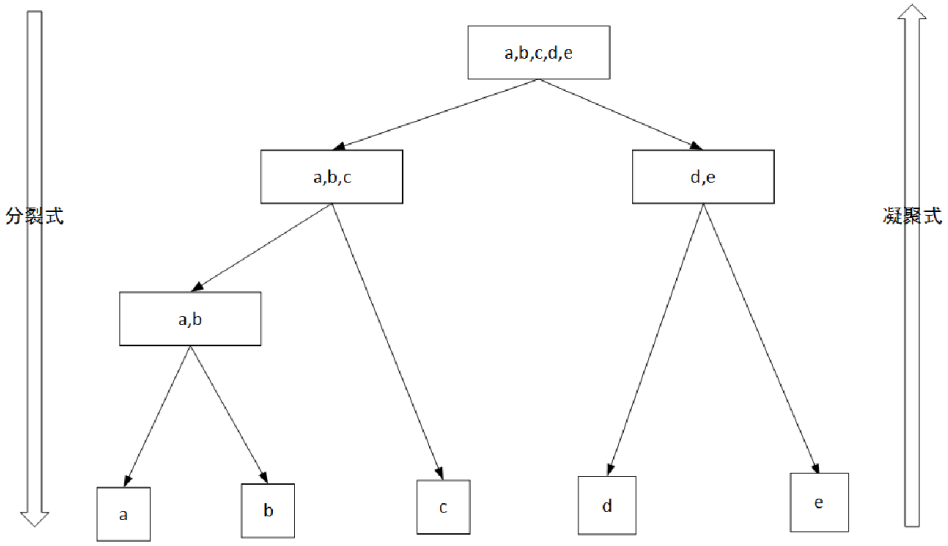

图10 层次聚类

Figure 10 Hierarchical clustering

4.4.1.2 基于划分的聚类算法

划分式聚类算法与层次聚类方法不同，在聚类之前需要预先指定最终生成多少个类别或聚类中心。著名的 K 均值算法[33]就属于划分聚类算法。其基本思想就是在提供聚类数以及初始的 K 个聚类中心之后，将剩下的文档与初始集簇的质心比较相关程度，将相关程度大的文档归入对应的集簇，调整集簇的质心，迭代这一过程，直到得到稳定的聚类结果。具体过程如下：

（1）指定聚类结果的类别数 K，并且指定 K 个文档作为初始聚类中心，这 K 个点称为聚类种子。此时，系统中有 K 个集簇，每个集簇中有一篇文档，记为$C=\{c_1,c_2...c_K\}$。

（2）对于其余文档，计算该文档与 C 中每个集簇质心的相似度或距离。若文档d_i满足求得的相似度或距离最大，则将d_i并入对应的集簇，得到新的聚类$C=\{c_1,c_2...c_K\}$。其中相关性计算方法有欧氏距离，余弦相似度等等。

（3）调整前面被改变的类别的质心。

（4）计算偏差值$Q=\sum_{i=1}^{n}[\min_{j=1...K} dis(d_i,c_r)^2]$，若 Q 收敛，则聚类结束，否则从前面的第二个步骤开始迭代操作。

除了 K 均值算法之外，常见的划分聚类算法还有 K-Medoids 算法等。

4.4.2 用户文档聚类

由上一节可知层次聚类可以在预先不知道目标集合内到底包含多少类别的情况下，将所有的信息组成不同的类。所以根据本文的应用场景，此处使用自底向上的层次聚类方法。其基本步骤为：

（1）将文本以义项为单位，每篇文章用一个向量表示。由于当某个义项出现在标题中时显然比出现在正文中更有价值，为了聚类更加准确，这里采用 2.2.2 节介绍的改进 tf/idf 公式计算向量中每个义项的权重，对出现在标题中义项的贡献度进行调整。

（2）对于文档集$D=\{d_1,d_2\ldots d_n\}$，生成一个初始聚类$C=\{c_1,c_2\ldots c_n\}$，其中类别c_i由文档d_i组成，这样每个类别都只有一篇文档。

（3）计算每两个类别之间的相似度$sim(c_i,c_j)$，这里类别相似度计算通过把两个集合中的向量两两的相似度全部放在一起求一个平均值得到。其中向量与向量之间的相似度使用余弦相似度，计算公式如下：

$$sim(d_i,d_j)=\frac{\overrightarrow{d_i}\overrightarrow{d_j}}{\left|\overrightarrow{d_i}\right|\left|\overrightarrow{d_j}\right|}=\frac{\sum_{k=1}^{n}w_{ki}w_{kj}}{\sqrt{\sum_{k=1}^{n}w_{ki}^2}\times\sqrt{\sum_{k=1}^{n}w_{kj}^2}} \quad (4\text{-}12)$$

（4）选取相似度最大的两个类，若这两个类的相似度大于等于阈值，合并这两个类，构成一个新的聚类$C=\{c_1,c_2\ldots c_{n\text{-}1}\}$，类别的数量少 1，然后跳转至（3）；否则跳转至（5）。

（5）返回聚类结果。

聚类处理后,用户表示的感兴趣文本被分成多个小文本集，文本集中每个文本的保留词语义已经确定。

4.5 用改进 TextRank 算法提取关键义项

4.5.1 TextRank 算法

TextRank算法[34]源于著名的PageRank算法[35]。PageRank是Google用于衡量特定网页价值的著名算法。其主要思想为：如果有大量网页链接到该网页或者有一个很重要的网页链接到该网页，则该网页价值会很高。一个页面的得分是由所有链向它的页面的重要性经过迭代得到的，最后得到一个页面的等级。以图11为例，图中共有4个页面1，2，3，4。箭头表示该页面有指向页面的超链接。显然，页面4有最多的页面指向它，所以最有价值。对于页面2和页面3均有两个页面指向其网页，由于指向2页面的包括前面所说的最有价值的4页面，所以页面2比页面3要更有价值一些。为了体现这种对价值排序的过程，需要根据整个页面的连接情况构建连接矩阵，然后迭代求出矩阵的特征向量，从而完成排序过程。

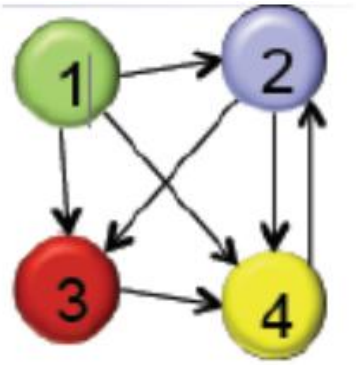

图11 页面连接图

Figure 11 Diagram of page connection

当PageRank算法被引入到自然语言处理领域，就诞生了TextRank，并且被广泛的应用于文本关键词提取。其通过将文本视为若干实体单元组成的集合，建立图模型，然后利用类似PageRank中投票机制排序文本中的候选单元，可以实现无指导关键词抽取。TextRank算法中构建图的方式也多种多样，比如一些研究中，词为图模型中的一个节点，在原文中以该词为中心以一定长度为半径做区间，区间内的词汇视为与该节点有边，这样就可以完成图模型的构建。本文利用了TextRank算法提取出聚类后每个类别的关键义项，组成用户模板。

4.5.2 改进 TextRank 算法提取关键义项方法

经过前面的聚类，现在需要对聚类结果中的每一类别单独进行处理，最终目的是挑选出每个类别中最具价值的那些义项，这些义项可以很好的代表这一类别，这样最后融合的用户模板就能很好的反映用户兴趣之所在。本节将介绍一种用改进的 TextRank 算法提取义项的方法。

在本节中，我们需要提取一个类别中的关键义项。显然，在一个类别中最具有概括性的

义项会和该类别中很多其他义项有相关关系。并且只有很多有价值的义项和该义项相关，该义项对于该类别才有价值。基于这个思想，本文使用 TextRank 算法提取义项是合理的。TextRank的迭代模型在理论上支持带权运算，但是传统的TextRank模型是基于影响力均分的，显然，这是不合理的[36]。如果图模型中一个相连的词语越重要，或者图中的一个链接对于最后的评估越有价值，那么对应的词理应分取更多的价值。本文中用各种影响力因子对此情况进行改进。下面对每个类别同样地采用下面的操作。

4.5.2.1 核心义项提取图模型构造

(1) 为了降低维数以及减少噪声，首先对每篇文章中的义项通过前面使用过的改进 tf/idf 公式进行排序，剔除排名最后的 K 个义项。

(2) 将该类别中每篇文章剩下的义项进行综合，取并集，得到集合 $Q=\{s_1,s_2\ldots s_i\ldots\}$。其中 s_i 为该类别中出现过的义项。

(3) 以集合 Q 中每个元素作为节点做图。和前面一样，计算每个节点之间的义项相似度，如果相似度大于阈值，则在节点之间建立一条边[37]。这样就得到了一个无向图模型 G=(V，E)。其中 $V=\{s_1,s_2\ldots s_i\ldots\}$，若 $sim(s_i,s_j)>T$，则 $<s_i,s_j>\in E$。其中 T 为相似度阈值。

4.5.2.2 建立评分转移概率矩阵

在该模型中，评分转移概率矩阵可以反映一个义项的价值传递给相邻每个节点的概率大小。该矩阵记为 S。

$$\boldsymbol{S}=\begin{bmatrix} p_{11} & p_{21} & p_{31}\cdots p_{n1} \\ p_{12} & p_{22} & p_{32}\cdots p_{n2} \\ p_{13} & p_{23} & p_{33}\cdots p_{n3} \\ \vdots & \vdots & \vdots \quad \vdots \\ p_{1n} & p_{2n} & p_{3n}\cdots p_{nn} \end{bmatrix} \tag{4-13}$$

其中 p_{ij} 即为义项 i 的价值转移给义项 j 的概率。所以矩阵每一列的和为 1。修正后的矩阵记为 $\boldsymbol{G}$，见等式（4-14）。其中 α 为阻尼系数，此处取 0.8。n 为图中节点的总数，这里即图中所有义项的总数。$\boldsymbol{U}$ 是一个全 1 矩阵。现在要寻找 $\boldsymbol{G}$ 的特征向量，即满足 $\boldsymbol{q}=\boldsymbol{G}\boldsymbol{q}$ 的向量。求法为任意选取一个 $\boldsymbol{q}^{current}$ 开始，迭代计算等式（4-15），当 $\boldsymbol{q}^{next}$ 和 $\boldsymbol{q}^{current}$ 足够接近时结束迭代。所求向量 $\boldsymbol{q}$ 的每一维的权重就代表这该义项的得分多少。

$$\boldsymbol{G}=\alpha\boldsymbol{S}+(1-\alpha)\frac{1}{n}\boldsymbol{U} \tag{4-14}$$

$$\boldsymbol{q}^{next}=\boldsymbol{G}\boldsymbol{q}^{current} \tag{4-15}$$

由此可见，首先我们要计算出矩阵 $\boldsymbol{S}$ 中每个概率 p_{ij} 的大小。传统方法见等式（4-16），其

中 M 为与节点 s_i 相连的所有节点的总数。但是这种均分的策略并没有考虑到每个节点的特殊性。首先，我们是基于义项间的相关性建立起的边，显然，如果义项之间相关性越大，那么 p_{ij} 就应该越大。其次，在两个义项 A 和 B 之间相关性没有那么大的情况下，如果义项 B 在该类别的很多文章都有很高的权重，那么就说明 B 很可能是足以表达该类别主题的义项，那么为了能够提取出这种概括性义项，我们人为的希望 A 尽可能多的给 B 投票，那么概率 p_{AB} 就应该对应的取大一些。最后，同样假设 A 和 B 在语义上相似度并不大，但是在类文档中 A 和 B 经常一起出现，那么就说明在该类的特定情况下，A 和 B 极有可能是相关的，那么如果 A 是关键义项的话，B 也应该被选出，也就是说如果 A 和 B 共现性很大的话，p_{AB} 的值也应该增大。综上，本文将转移概率 p_{ij} 拆分为三部分，见等式（4-17）。其中 p_{ij} 为节点 s_i 价值传递给 s_j 概率的大小。Q_{ij} 为相似度影响概率。R_{ij} 为类权重影响概率。K_{ij} 为共现性影响概率。式中 $a+b+c=1$ 且 $0 \le Q_{ij,}R_{ij,}K_{ij} \le 1$ 。

$$\mathrm{p}_{ij}=\begin{cases}0, <\mathrm{s}_i,\ \mathrm{s}_j>\notin E\\ \dfrac{1}{M}, <\mathrm{s}_i,\ \mathrm{s}_j>\in E\end{cases} \tag{4-16}$$

$$\mathrm{p}_{ij}=a\mathrm{Q}_{ij}+\ b\mathrm{R}_{ij}+\ c\mathrm{K}_{ij} \tag{4-17}$$

现在分别计算每个部分。

（1）相似度影响概率

$$\mathrm{Q}_{\mathrm{ij}}=\frac{\mathrm{sim}(\mathrm{s}_{\mathrm{i}},\ \mathrm{s}_{\mathrm{j}})}{\sum_{<\mathrm{s}_{\mathrm{i}},\ \mathrm{s}_{\mathrm{k}}>\in \mathrm{E}}\mathrm{sim}(\mathrm{s}_{\mathrm{i}},\ \mathrm{s}_{\mathrm{k}})} \tag{4-18}$$

其中 $sim(\mathrm{s}_{\mathrm{i}},\ \mathrm{s}_{\mathrm{j}})$ 为 s_i 和 s_j 的相似度，计算方法和前面确定语义时方法相同。

（2）类权重影响概率

此处我们要为每个义项计算出一个针对类别的权重，该权重可以从出现频率和分布上大体反映某些义项对于类别的重要程度。这里我们考虑到篇文章中标题，正文中出现义项代表性的差别，采用类似 TF*PDF[38]的算法。每个义项 s_i 的类权重 w_i 如下：

$$\mathrm{w}_{\mathrm{i}}=\left|\mathrm{F}_{\mathrm{i\_body}}\right|\exp\left(\frac{\mathrm{n}_{\mathrm{i\_body}}}{\mathrm{N}}\right)+\beta\left|\mathrm{F}_{\mathrm{i\_title}}\right|\exp\left(\frac{\mathrm{n}_{\mathrm{i\_title}}}{\mathrm{N}}\right) \tag{4-19}$$

$$\left|\mathrm{F}_{\mathrm{i\_body}}\right|=\frac{\mathrm{F}_{\mathrm{i\_body}}}{\sqrt{\sum_{\mathrm{k=1}}^{\mathrm{K}}\mathrm{F}_{\mathrm{k\_body}}^{2}}} \tag{4-20}$$

$$\left|\mathrm{F}_{\mathrm{i\_title}}\right|=\frac{\mathrm{F}_{\mathrm{i\_title}}}{\sqrt{\sum_{\mathrm{l=1}}^{\mathrm{L}}\mathrm{F}_{\mathrm{l\_title}}^{2}}} \tag{4-21}$$

其中 $\beta>1$.这里 N 为文档的总数，$\mathrm{F}_{\mathrm{i\_body}}$ 为义项 s_{i} 在 N 篇文档的正文中出现的频率，$\mathrm{n}_{\mathrm{i\_body}}$

为在 N 篇文档的正文中出现过义项 s_i 的文档数量，K 为 N 篇文档中正文出现义项的总数，同理，F_{i_title} 为义项 s_i 在 N 篇文档的标题中出现的频率，n_{i_title} 为在 N 篇文档的标题中出现过义项 s_i 的文档数量，L 为 N 篇文档中标题出现义项的总数。式（4-19）的左边部分反映了该义项在类别正文中的影响力，右边反映了该义项在标题中的影响力。通常标题更具有概括性，所以此处取 β>1。由此，可以计算出类权重影响概率如下，其中 w_j 为 s_j 的类权重。

$$R_{ij}=\frac{w_j}{\sum_{<s_i,\ s_k>\in E} w_k} \tag{4-22}$$

（3）共现性影响概率

另 K_{ij} 表示共现性影响概率，则：

$$K_{ij}=\frac{X_{ij}}{\sum_{<s_i,\ s_k>\in E} X_{ik}} \tag{4-23}$$

其中 X_{ij} 表示义项 s_i 和义项 s_j 的共现性得分。此处 X_{ij} 计算公式如下：

$$X_{ij}=\frac{\sum_{d\in D}\log\left(tf\left(si,d\right)+1\right)*\log\left(tf\left(sj,d\right)+1\right)}{\log(N)} \tag{4-24}$$

其中 N 为该类中的文档总数，D 代表这些文档的集合，d 代表其中一篇文章，$tf(si,\ d)$ 与 $tf(sj,\ d)$ 分别表示 s_i 和 s_j 在文档 d 中的归一化频率。

需要注意的是 K_{ij} 分母为 0 的情况，此时为了保证不影响 p_{ij}，使概率转移矩阵每列和仍为 1，取 $K_{ij}=1/T$。其中 T 为 s_i 的边的个数。

最终，我们就可以得到转移概率 p_{ij}：

$$p_{ij}=\begin{cases}0, & <s_i,\ s_j>\notin E\\ aQ_{ij}+bR_{ij}+cK_{ij}, & <s_i,\ s_j>\in E\end{cases} \tag{4-25}$$

由此，如前面所述，我们可以得到修正后的转移矩阵 $\boldsymbol{G}$，计算特征向量 $\boldsymbol{q}$，$\boldsymbol{q}$ 每一维的权重就是图模型中每个语义的最后影响力得分，接下来，我们对权重进行排序，挑选出前 M 个作为关键语义用于构建用户模板。

4.6 用户初始模板生成

经过前面的几步，我们现在已经得到了聚类后每个类的关键义项。现在把这些义项进行综合，取并集，得到一个体现用户真实兴趣的义项集。然后我们需要对该义项集进行处理，得到一个用向量空间模型表示的用户模板。

在用户模板中，这里我们可以选择用义项作为向量中的特征项，计算每个义项在模板中的权重。但是，这种选择会降低系统的推荐效率。因为假如模板以义项为单位，那么在计算模板与待推荐文档相关程度时，文档也必须表示为以义项为特征的向量形式。对于每一篇待

推荐文档来说，都需要附加确定关键词语义的操作。而待推荐文档集通常较大，累计附加操作带来的效率上的降低是相当可观的。所以，这里建立初始模板时，我们不选择以义项为单位。

在信息检索中，查询中信息量较少带来的查询偏移问题可以通过查询扩展的方式得到改善。其中的一种方法就是通过外部资源引入同义词或者相关词等等。这里，我们不使用义项为单位而是以词为单位建立模板。由于汉语中的一次多义问题，系统也有可能产生偏移性问题。但是，假如我们将义项替换为《同义词词林》中该义项包含的所有词，就可以起到查询扩展的效果，不会出现大的偏移性问题。所以这里建立最终的初始用户模板步骤如下：

（1）针对用户感兴趣义项集中的每个义项，在初始文档集的每篇文章中用改进tf/idf公式求该义项权重，最后求平均，得到该义项在模板中的权值，此时会得到一个义项向量。

（2）查找《同义词词林》，对义项向量中的每个义项用对应的词汇集替换，词汇集中每个词的权重与原义项相同，这样就得到一个关键词向量，这就是初始用户模板。得到该模板后就可以通过计算其与其他文本相似度，确定是否应该推荐该文本。推荐后，也可以对该模板进行动态更新。

4.7 本章小结

本章给出了基于 TextRank 的内容推荐系统用户模板构建方法。针对在基于内容的推荐系统中添加新用户时，由于此时新用户已标识的感兴趣文档数量很少，建立一个准确的用户模板相对困难的问题，系统首先对用户文档集进行预处理，并且提供了基于词典和不基于词典的两种分词方式，然后用《同义词词林》确立保留词的语义，为建立合理的图模型打下基础，接着用层次聚类充分挖掘文档集中体现出的用户兴趣，最后建立 TextRank 图模型，引入各种影响力因子对模型进行改进，建立一个较为准确的初始用户模板，减少了噪声的引入。该算法可以为系统的初始阶段提供好的推荐效果。

5 基于伪相关反馈的用户模板更新方法

本章主要对基于伪相关反馈的用户模板更新方法进行详细介绍。首先，对该算法的主题思想进行描述。然后按照算法的流程分别介绍了用户模板划分的方法及原因，用户需求模板更新方法，反馈部分模板更新方法。对于用户模板划分的原因，主要是基于系统的实际情况以及对准确度的要求。对于用户需求模板部分的更新，本章将其视为显式相关反馈实现更新操作。对于反馈部分模板的更新，本章主要研究反馈文档质量如何提高，以及如何在这些文档中挑选关键词，以减少噪声的引入，避免出现偏移性问题。

5.1 算法概述

在内容推荐系统中，用户模板更新和初始用户模板的建立一样重要。因为这决定着系统能否持续的向用户进行精确推荐。从用户的角度来说，其兴趣是会随着时间变化而变化的，提供的用户信息也会随之增多。对应的，用户感兴趣文档集中的文档会越来越多。假如不对用户模板进行及时的更新，系统显然会滞后于用户需求，产生偏移性现象，随之带来不好的用户体验。同时，由于系统初始时用户信息较少，尽管前面已经建立了较为精确的初始用户模板，但也仅仅是依赖于用户提供的少量文档，这样就往往会造成推荐范围过窄。所以，需要在更新用户模板的过程中引入伪相关反馈技术，以扩大推荐范围。

由上面的描述可以看出，我们需要的更新用户模板的操作包含有两个部分。第一个部分的更新操作主要源于用户提供信息的逐渐增多，我们需要使用用户提供的新的感兴趣文档对模板进行更新。而第二部分，我们实际上并不需要用户的直接参与，目的是为了通过查询扩展更加丰富我们的模板，避免系统推荐范围过窄。这样，我们就可以将用户模板划分为两个部分，每一部分用不同的策略进行更新操作。对于第一个部分，由于是由用户直接提供信息的，模板的更新可以视为信息检索中的显式相关反馈，不同的是这里需要更新的是用户模板，而不是查询。对于第二个部分，我们不需要用户的直接参与，对应的在相关反馈技术中，伪相关反馈也不需要用户直接参与。那么我们可以将该部分模板在某种程度上视为一个查询，将伪相关反馈技术引入该部分模板的更新。最后实现整体用户模板的更新。具体的，该算法包含的主要内容如下：

（a）将用户模板划分为两部分：用户需求模板；反馈部分模板。其中用户需求模板由用户自己提供的感兴趣文档集确定。反馈部分模板由伪相关反馈产生和更新。

（b）确定两部分模板的加权方式。在用户模板中，用户需求模板和反馈部分模板所占比重应该是随着系统状态逐渐变化的。

（c）用户需求模板更新。这部分的模板更新可以视为显式相关反馈，更新模板。

（d）挑选用于伪相关反馈的文档。执行聚类操作，引入多种影响因子综合评定反馈文档的质

量，挑选评分最高的作为用于更新模板的反馈文档。

（e）挑选反馈文档中的关键词。为了减少伪相关反馈中的偏移性问题，这里挑选关键词主要考虑与模板中特征项的共现性。

（f）用改进的 Rocchio 算法更新反馈部分模板。

（g）加权得到新的用户模板。

5.2 用户模板划分

由上一节可知，我们将用户模板划分为两部分分别进行更新是合理的。对于完全由用户自己提供的感兴趣文档集决定的部分，我们称之为用户需求模板，并用 p_{user} 表示。对于使用由伪相关反馈得到的那部分模板，我们称之为反馈部分模板，并用 p_{back} 表示。那么总的用户模板表达式如下。其中 p_{total} 表示总的用户模板。d 与(1-d)分别表示两部分模板在总模板中所占的比重。

$$p_{total} = d * p_{user} + (1-d) * p_{back} \quad (5\text{-}1)$$

现在，我们需要确定 d 值的大小，最简单的办法是取常数 0.5，但是这样显然是不合理的。在总的用户模板中，用户需求模板和反馈部分模板所占比重应该是随着系统状态逐渐变化的。在系统的初始阶段，由于用户提供的感兴趣文档数目较少，此时反馈部分模板有着较大的作用；在系统逐渐稳定后，此时系统拥有的用户信息越来越多，用户感兴趣文档集中的文档数逐渐增大。此时用户需求模板已经趋于完整与准确，反馈部分模板所占比重应该越来越小，否则可能会产生偏移性问题。

由上面的描述，我们对 d 要满足的条件进行分析。首先，d 的初值应该为 0.5，也就是说，在系统最初的时候，两部分模板起着同样的作用。其次，随着用户感兴趣文档集中用户文本数目的逐渐增加，d 的数值应该逐渐增大，当文本数量无穷大时，d 的数值应该无限逼近于 1。根据这些条件画出 d 函数的示意图如下：

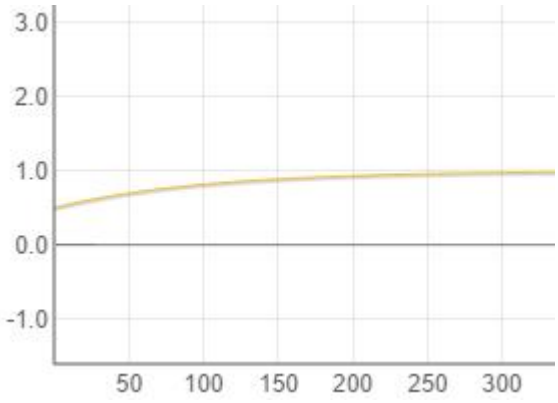

图12 d函数示意图

Figure 12 Diagram of function d

按照上面的示意图，我们确定 d 函数的取值为：

$$d = 1 - 0.5 * e^{-an} \quad (5\text{-}2)$$

其中 n 为用户感兴趣文档集中的文档数目，a 为常数，当 a 越大时，d 靠近 1 的速度越快。这样就实现了用户模板的划分。

5.3 用户需求模板更新

用户需求模板完全由用户感兴趣文档集中的文档决定。用户会随着时间的推移逐渐增加感兴趣文档。该模板的更新过程有如下特点：一次更新时，只有少量的正例文档用于更新模板。这是由于我们为了保证系统推荐的持续准确，必须要对该模板进行实时的更新，而单次添加的感兴趣文档数目不会太多。针对这种情况，我们按照下面的步骤完成用户需求模板的更新，其间使用了显式相关反馈的原理。

5.3.1 预处理

在更新用户需求模板之前，首先需要对新添加的感兴趣文档进行预处理，步骤如下。

（1）对这些文档进行分词处理，方法见 4.2 节。

（2）对分词结果进行去停用词操作。

（3）通过对用户模板中采用不同词性的关键词进行试验，从推荐效果来看，名词和动词作为关键词时有着更好的推荐效果，所以这里使用词性标注工具进行标注，并保留动词和名词。词性标注方法见 4.2 节。

（4）对这些文档中剩下的词进行特征选择。常用的方法有基于文档频率，互信息法，信息增益法，χ^2 统计量法等。在这里的应用场景中，首先因为不存在训练集，所以不能使用互信息法，信息增益法，χ^2 统计量法。其次，由于前面描述过单次添加的感兴趣文档数目不会数目太多，有时可能只有三到四篇，使用基于文档频率的方法效果也不是很好。所以这里对每一篇文档进行词频统计，并且用文档的总词数进行归一化处理。然后设定一个阈值，对每一篇文档低于阈值的词进行过滤，这样对剩下的词取并后就可以得到特征词集合。

5.3.2 模板更新操作

通过前面的预处理操作后，需要在用户模板中添加的特征词已经确定了，现在要做的是确定这些词在模板中的权重，或者当有些特征词以已经存在于模板中时，我们需要增加的权重。按照 Rocchio 算法，对于下面的式子，取α=1，$\gamma = 0$，也就是说忽略负反馈部分，只需调节参数β。

$$\overrightarrow{q_{new}} = \alpha \overrightarrow{q_{old}} + \beta \frac{1}{r} \sum_{i=1}^{r} \overrightarrow{d_{rel}}^{(i)} - \gamma \frac{1}{n} \sum_{j=1}^{n} \overrightarrow{d_{irrel}}^{(j)} \quad (5\text{-}3)$$

对于$\overrightarrow{d_{rel}}^{(i)}$，每一维代表前面处理后特征词集合中的关键词。一般情况下，这些关键词的

权重只需用 tf-idf 公式即可。但是，由于这里用于反馈的文档很少，可能 tf-idf 具有的定义能力不够准确。于是，我们使用下面的式子求出特征词权重。

$$w(t)=w_{avg}(p_{user})\times\sqrt{tf(t)} \quad (5\text{-}4)$$

其中 w(t) 为特征 t 在某篇文档的权重，$w_{avg}(p_{user})$ 为用户需求模板中所有特征的权重平均值，$tf(t)$ 为特征 t 在该篇文档中的 tf 值。这里求 tf 时，可以参照第二章中的内容，对文章中的不同区域按照重要程度的大小分配比例因子，然后各个区域的词频乘以对应区域的比例因子，求和之后得到总的 tf 值。这样的求法考虑了标题与正文的差异性，更加合理。

按照上面的式子，将 $\overrightarrow{q_{old}}$ 与 $\overrightarrow{q_{new}}$ 分别视为原用户需求模板和新的用户需求模板，即可实现用户需求模板的更新。

5.4 反馈部分模板更新

由前面的描述可知，我们需要用反馈技术对用户兴趣模板进行补充，避免推荐范围过窄，这一过程不需要用户直接参与。对应的，在相关反馈技术中，伪相关反馈也不需要用户直接参与，这样我们可以将反馈部分模板在某种程度上视为一个查询，将伪相关反馈技术引入该部分模板的更新。在伪相关反馈的过程中，相关文档的选取至关重要。因为更新过程没有用户的参与，用于反馈的相关文档都是由系统选择的，所以有一些文档可能并不是用户感兴趣文档。假如引入了大量用户不感兴趣文档进行反馈，必然会产生偏移性问题，这时不但没有起到预想的模板扩展作用，推荐效果反而大大降低。另一方面，对于相关文档中关键词的选取同样重要，假如引入了大量噪声词，推荐效果同样会受到很大影响。因此我们必须对这两部分进行严格控制，确保反馈的确能起到扩展用户模板的作用，尽可能减少偏移性问题。下面对这两部分分别进行讨论。

5.4.1 反馈文档选取

通过计算用户模板与待推荐文档集中每个文档的相关程度，可以得到哪些文档应该推荐给用户。一般情况下，反馈文档的质量可以通过与用户模板的相关程度衡量。即我们可以使用与模板的相关程度大小对文档进行排序，取靠前的若干篇文档作为反馈文档，更新模板。但是这么做会有如下一些问题，这些问题可能会造成偏移性问题。

（1）有一些文档可能只有部分段落符合用户兴趣，但其主题并不符合。假如这类文档作为相关文档更新用户模板必然会产生偏移性问题。举例来说，一篇文档主要讲的是在战后的 1954 年瑞士世界杯德国队最终奇迹夺冠的历程。但是这篇文档的前几段可能会介绍德国在二战之后的经济困难局面。这样，当用户模板涉及到德国经济时，这篇文档可能会被判断为相关，参于反馈。显然一篇体育方面的文章参与主要涉及经济方面模板的更新时，会造成偏移性问题。

（2）对于一些包含较少模板中关键词，却的确符合用户兴趣的文章，用这种方法可能不会被判定为相关文档参与反馈。比如模板中有关键词“计算机”、“调制解调器”，文档 A 包含关键词“计算机”、“调制解调器”、“显示器”、“触摸板”，文档 B 包含关键词“电脑”、“Modem”、“显示器”、“触摸板”。虽然文档 A 和 B 基本上类似，但是 A 被判定为相关，B 被判定为不相关。这样 B 中一些有价值的关键词就无法添加到模板中，这显然是不合理的。

（3）对于有一些文档，可能仅仅涉及到模板中的单个方面，将这些文档用于反馈文档，显然没有用涉及到多个方面的文档效果好。比如模板中主要关注世界杯中的巴西队。文档 A 主要说的是世界杯，文档 B 主要说的是世界杯中的巴西队。这样，将 B 用于反馈比将 A 用于反馈有更好的效果。

针对上面的情况，我们按照下面的步骤对文档质量进行重新排序，挑选质量最高的文档进行反馈。

5.4.1.1 文本聚类

考虑到前面描述的问题一和问题二，我们需要首先对初始筛选后文档进行聚类处理。对于问题一，假如某个文档只有局部与模板相关，那么它在聚类后会体现为孤立点。因为初始候选集中的文档是由与模板的相关程度筛选出来的，他们之间应该会有相关性。以问题一中给的例子来说，首先按照与模板的相关程度得到一个初始候选集，该文档集合中大部分文章都是关于德国经济的，进过聚类之后，那一篇关于世界杯德国队的文章就会被孤立，我们可以通过剔除这些孤立点完成对初始侯选集的筛选。对于问题二，由上面给出的例子就可以看出，文档 A 和文档 B 显然聚类后会归为一类，这样可以通过聚类修正原相关程度排名，选出用于反馈的文档。具体步骤如下：

（1）将文本集中的文本表示为向量形式，并计算与用户兴趣模板的相似度，挑选大于阈值 M 的文档，组成初始候选集。

（2）对初始候选集中的文档进行层次聚类。

（a）对这些文档进行特征选择。需要注意的是，由于这些文档都是通过模板选出，所以如果特征选择仍然包括原模板的高权重关键词，聚类效果会收到影响。可能会出现聚类结果类别数目很少的现象。所以这里挑选关键词时在完成分词，去停用词，词性选择之后，首先剔除原文档中包含的高权重关键词，然后通过文档频率挑选出特征词。

（b）将这些文档按照挑选的关键词表示为向量形式，权重用第 2.2 节中改进的 tf-idf 公式计算。

（c）对这些文档向量进行层次聚类，聚类方法见第 4.4 节。

（3）聚类之后初始候选集被划分为若干个子集。

5.4.1.2 文档重排序

聚类处理之后，我们需要采取一些措施对初始候选集中的文档进行重排序与筛选。我们用 $Score(d_i)$ 表示文档 d_i 的质量评分，最后通过对质量得分进行排序，挑选最靠前的几篇文

档用于反馈。计算 $Score(d_i)$ 的步骤如下。

（1）对前面聚类之后得到的初始候选集进行处理。对每个类别中文档数目进行统计，若数目小于阈值 K，则剔除这个类别中的文档。这样就解决了前面的问题一，排除了局部相关的文档。

（2）根据前面的描述，这里将 $Score(d_i)$ 分为三个部分，对文档质量进行全面的评定。首先一篇文章与模板的相关程度依然可以作为一个重要的评定标准，这里我们用余弦相似度体现这种相关程度，这部分对应的得分记为 $Score_{\text{sin gle}}(d_i)$。其次，对于前面描述的问题二，我们应该用聚类带来的特性对那些含有较少模板中特征词的文档进行修正得分，这部分得分记为 $Score_{\text{class}}(d_i)$。最后，我们需要考虑前面描述的问题三如何解决。显然，如果一篇文档中包含模板中的关键词越多，其越可能是涉及到多个方面的文档，这部分得分记为 $Score_{multi}(d_i)$。下面分别计算这三部分。

（a） $Score_{\text{sin gle}}(d_i)$ 用文档与模板的余弦相似度求取，公式如下：

$$Score_{\text{sin gle}}(d_i)=\frac{\overrightarrow{d_i}\,\overrightarrow{p_{total}}}{\left|\overrightarrow{d_i}\right|\left|\overrightarrow{p_{total}}\right|}=\frac{\sum_{k=1}^{n} w_{ki}w_{ktotal}}{\sqrt{\sum_{k=1}^{n} w_{ki}^2}\times\sqrt{\sum_{k=1}^{n} w_{ktotal}^2}} \tag{5-5}$$

（b） $Score_{\text{class}}(d_i)$ 用文档 d_i 所在类别的质心与模板的余弦相似度求取[39]，公式如下：

$$Score_{\text{class}}(d_i)=\frac{\overrightarrow{c_i}\,\overrightarrow{p_{total}}}{\left|\overrightarrow{c_i}\right|\left|\overrightarrow{p_{total}}\right|}=\frac{\sum_{k=1}^{n} w_{ki}w_{ktotal}}{\sqrt{\sum_{k=1}^{n} w_{ki}^2}\times\sqrt{\sum_{k=1}^{n} w_{ktotal}^2}} \tag{5-6}$$

其中 $\overrightarrow{c_i}$ 表示文档 d_i 所在类别的质心。

（c）对于一篇文档的范围得分，我们用 $S(d_i)$ 表示。按照上面的描述，$S(d_i)$ 的大小与文档包含的模板中关键词数目有关[40]，则其表达式如下：

$$S(d_i)=\sum_{t_j\in d_i\wedge t_j\in p_{total}} idf(t_j) \tag{5-7}$$

该式含义为：首先求出在 d_i 和模板中同时存在的关键词的 idf 值。求 idf 时，文档集为剔除了局部相关文档的初始候选集。将这些 idf 值求和后得到 $S(d_i)$。

由上面的 $S(d_i)$ 公式，可以得到 $Score_{multi}(d_i)$ 表达式如下：

$$Score_{multi}(d_i)=\frac{S(d_i)}{\sum_{k=1}^{n} S(d_k)} \tag{5-8}$$

其中 n 为剔除了局部相关文档的初始候选集中文档的总数量。

（3）利用（2）中给出的式子，对剔除了局部相关文档的初始候选集中每一篇文档计算质量得分 $Score(d_i)$，公式如下：

$$Score(d_i)=\alpha*Score_{\text{single}}(d_i)+\beta*Score_{\text{class}}(d_i)+\gamma*Score_{multi}(d_i) \quad (5\text{-}9)$$

其中 α，β，γ 为常数因子，满足 $\alpha+\beta+\gamma=1$。

（4）根据 $Score(d_i)$ 对初始候选集剩下的文档进行排序，选择前 N 个作为相关文档，用于更新反馈部分模板。这样就完成了反馈文档的选取。

5.4.2 反馈文档特征词选取

经过 5.4.1 节的操作，用于反馈的相关文档已经确定，下面就需要研究在这些文档中选取哪些关键词用于更新模板。传统的方法是基于这些文档中词语的统计情况对关键词进行筛选。但是这里我们不用这种方法。因为我们用伪相关反馈对用户模板进行更新是为了将模板进行扩展，避免在小范围内进行推荐。所以，我们通过共现性从相关文档中挑选与模板相关的词更合理一些。那么，我们使用下面的步骤对前面得到的反馈文档进行特征词挑选。

（1）对反馈文档去停用词，保留名词和动词，组成备选词集合。

（2）定义用户模板中关键词 T_{pi} 和备选词集合中关键词 T_{bj} 的共现性得分如下：

$$\text{co}(T_{\text{pi}},T_{bj})=\frac{\sum_{\text{d}\in\text{D}}\log\left(\text{tf}\left(T_{\text{pi}},\text{d}\right)+1\right)*\log\left(\text{tf}\left(T_{\text{pi}},\text{d}\right)+1\right)}{\log(\text{N})} \quad (5\text{-}10)$$

其中 D 表示反馈文档集，d 表示一篇反馈文档，N 表示 D 文档集中的文档数目。

（3）上面的式子只是针对模板中的单个关键词，现在需要考察备选词与整个模板之间的关系，则定义备选词集合中关键词 T_{bj} 与用户模板 p_{total} 之间的共现性得分如下：

$$\text{co}(T_{bj},\text{p}_{total})=\sum_{t\in\text{p}_{total}}\text{w(t)}*\text{co}(T_{bj},t) \quad (5\text{-}11)$$

其中 t 表示模板中的关键词，w(t)表示关键词在模板中的权重。$\text{co}(T_{bj},t)$ 为关键词 t 与备选词 T_{bj} 的共现度得分。

（4）对备选词与模板的共现性得分进行排序，挑选靠前的 X 个关键词作为用于反馈的关键词，完成特征筛选。

5.4.3 模板更新

前面两节完成了反馈文档的选取以及用于反馈的特征选取。本节利用这些信息完成反馈部分模板的更新操作。这里我们使用改进的 Rocchio 算法。在传统的 Rocchio 算法中，对于相关文档集中的每一篇文章都视为平等，可实际上这些文档对于更新模板来说质量是不同的。

显然得分高的文章内的信息往往更加贴近于用户兴趣，得分低的文章可能多少会偏离用户的真实兴趣，所以对于不同质量的文档分而视之会使更新后的模板更加准确，质量高的文档提供较多的信息，质量低的文档提供较少的信息有助于避免偏移性问题[41]。根据这个思想，我们用下面的改进 Rocchio 公式实现反馈部分模板的更新。

$$\overrightarrow{p_{back\text{-new}}}=\alpha\overrightarrow{p_{back\text{-old}}}+\beta\frac{1}{\mathrm{r}}\sum_{\mathrm{i}=1}^{\mathrm{r}}s(d_i)\overrightarrow{d_i} \qquad (5\text{-}12)$$

其中 $\overrightarrow{p_{back\text{-new}}}$ 为新的反馈部分模板；$\overrightarrow{p_{back\text{-old}}}$ 为原反馈部分模板；r 为前面筛选的用于反馈的文档总数量；$\overrightarrow{d_i}$ 为文档 d_i 的向量表示，其中每一维对应前面筛选的反馈文档中的特征词，特征词权重的求法可以参考 5.3.2 节；$s(d_i)$体现了文档 d_i 的质量，即与用户兴趣的契合度；α,β 为常数，为了调节方便，另 α=1，只需调节 β 即可。

根据前面筛选反馈文档的过程，我们这里采用 $Score(d_i)$ 度量文档与用户兴趣的契合度，即式（5-12）中的 $s(d_i)$满足下式：

$$s(d_i)=Score(d_i) \qquad (5\text{-}13)$$

由此，本章已经介绍了用户需求模板的更新以及反馈部分模板的更新，按照式（5-1）对两部分模板进行加权，即可实现总的用户兴趣模板更新。

5.5 本章小结

本章首先介绍了基于内容推荐系统中用户模板更新操作的意义之所在，对提出的基于伪相关反馈的用户模板更新方法主题思想进行大致描述。基于多种需求，算法中对用户模板进行了划分，并且对划分出的两部分如何进行加权进行深入研究，结合图像，确定了加权方式。然后，对用户需求模板和反馈部分模板的更新分别进行了讨论。对于用户需求模板的更新，由于该模板完全依赖于用户提供的信息，所以可以将其视为查询扩展中的显式相关反馈进行操作，同时，在更新的过程中考虑到了每次更新只有少量的正例文档的情况合理的进行操作。对于反馈部分模板的更新，不需要用户的直接参与，所以该过程对应于相关反馈中的伪相关反馈。为了避免出现偏移性问题，在反馈文档的挑选上，本章讨论了会造成偏移性问题的三个问题，由此提出了一种新的对文档进行评分的方法，结合聚类操作，实现对高质量文档的挑选；在关键词的挑选上，本章主要考虑了备选词与模板的共现性得分，而不是其他统计信息，这是基于反馈部分模板的更新目的选择的；在更新方式上，本章对 Rocchio 算法进行了改进，使高质量文本提供更多的能量，从而让模板更加精确。综合上面的所有讨论，最终可以实现用户兴趣模板的更新。

6 实验与分析

本章主要对本文提出的基于内容的自适应推荐系统中核心的基于 TextRank 算法的初始模板建立模型和基于伪相关反馈的用户模板更新模型分别进行实验。实验结果都较为理想。

6.1 基于 TextRank 算法的初始模板建立方法实验

6.1.1 数据集及相关工具

本文的自适应推荐系统主要用于长文本的推荐。对应的，因为新闻类型的文本长度适中，包含标题与正文，层次清晰，所以本文搜集新闻文本作为数据集进行试验，检验算法效果。作者通过网络爬虫对新浪新闻等网站上的数据进行搜集。搜集的范围包括网站中的各个板块，涵盖军事类，经济类，科技类，体育类，旅游类，教育类。抓取的每条新闻均包含正文部分以及标题部分。表 3 显示了数据集中的类别与新闻数目的对应关系。

表 3 数据集中类别及对应新闻数

Table 3 The categories and the corresponding number of news in data set

类别	新闻数
军事类	1877
经济类	1100
科技类	1061
体育类	1600
旅游类	1005
教育类	1000

下面的试验将会从该数据集中按照一定策略取出少量文档训练用户初始模板，然后验证该模板的准确性。

除数据集外，本文使用了分词工具以及词性标注工具。其中，分词工具为stanford segmenter，词性标注工具为stanford postagger，当然分词部分也可以使用本文中曾经提到的基于无词典的方法。

6.1.2 评测标准

在推荐系统中，模板的精确与否可以由推荐结果反映出来，假如符合用户兴趣的推荐文本越多，那么该模板越精确。在系统中，常用的评价标准有 MAE 算法，召回率（Recall），准确率（Precision）。前者属于统计准确性标准，后两者属于推荐结果评价标准。介绍如下：

（1）平均绝对误差（MAE）

该方法通过计算预测评分和实际评分的偏差大小来评测系统推荐准确率，若 MAE 的值越大，则系统的推荐效果越差。计算公式如下：

$$MAE = \frac{\sum_{d \in \mathrm{D}} \left| d_p - d_t \right|}{\left| D \right|} \tag{6-1}$$

式中$|D|$是被系统预测评分的产品总数量，d_p是系统的预测评分，d_t为实际评分。

（2）召回率（Recall）和准确率（Precision）

召回率和准确率被广泛应用于信息检索和统计学分类等领域中，用于评估结果的质量。假如在一个数据集合中检索文档，则可以按照检索动作将文档分为四类：c_1 表示系统检索到的和查询相关的文档集；c_2 表示被系统检索到的但是与查询不相关的文档集；c_3 表示系统没有检测到的相关文档集；c_4 表示系统没有检测到的不相关文档集。则召回率表示相关文档中被检索出的比例大小；准确率表示检索文档中相关文档所占的比例大小。对应表达式如下：

$$\mathrm{Re}\,call = \frac{|c_1|}{|c_1|+|c_3|} \tag{6-2}$$

$$Precision = \frac{|c_1|}{|c_1|+|c_2|} \tag{6-3}$$

对应在推荐系统中，召回率的分母表示用户感兴趣的文档总数，分子表示系统推荐的确实符合用户兴趣的文本数量；准确率体现了系统推荐的文本中用户感兴趣文本所占比例。

针对本文的应用场景，由于不涉及评分，所以不能使用 MAE 作为评价标准。一般情况下，常规的试验思想为由用户提供感兴趣文档和不感兴趣文档，将感兴趣文档中的小部分用于训练模板，同时，将不感兴趣文档和剩下的感兴趣文档一起组成待推荐文本集，用训练的模板对测试集进行推荐，得到推荐结果，用该结果计算召回率和准确率。但是对于抓取新闻数据集的过程，无法直接获取用户感兴趣文档信息，即不知道哪些是用户感兴趣文档。面对这种情况，结合本节要测试的算法特点，提供了下面三种可选方案。

（1）将数据集划分出 K 个试验集合，每个试验集合用于单次试验。首先人工对每个试验集进行标注，注明该集合内哪些文档是自己感兴趣的，哪些是自己不感兴趣的，此时每篇文档均需要标注，剔除那些用户不确定是否感兴趣的文档。然后每个集合分别进行试验，用少量正例文档训练模板，对剩下的文档进行推荐，对结果进行分析。用这种方法可以计算出查全率和查准率。

（2）将数据集划分出 T 个试验集合，每个试验集合用于单次试验。对于每个试验集合取出一部分文档人工进行标记，主要标记哪些是自己感兴趣的文档。然后取少量感兴趣文档训练出模板，对原试验集合剩下的部分进行推荐，在推荐结果中再一次人工标注，看结果中哪些文档是自己感兴趣的。用这种方法可以节省标注的工作量，但是只能计算查准率。

（3）在本文提出的算法中，我们提到过用户的感兴趣文档具有集簇性，我们可以用这种特性模拟出感兴趣文档，基本可以体现推荐效果。举例而言，假如用于训练兴趣模板的文档只是

选自体育类，那么用户感兴趣文档也应该属于体育类。属于政治类，经济类的可能性很小。我们只需要对推荐结果中文档的类别进行统计，看有多少属于体育类，有多少属于其他类，就可以判断模板的精确程度。并且，对于没有改进的模板生成方法采用同样的试验过程，就可以通过比较确定本文算法是否有效。

由于第三种方法省去了繁重的人工标注操作，有效利用了数据集已经分好类的特性，基本上可以实现对算法的效果进行评测，所以这里我们采用该方法进行试验。度量标准以及描述如下：从类别集合 $\overline{S}$ 中选取少许文章作为用户感兴趣文档集，通过本文中建立初始模板的方法获取用户模板。再从类别集合 S 中选取大量文章作为待推荐文本，其中 $\overline{S} \subsetneq S$（如 $\overline{S}$={经济，旅游}，S={经济，旅游，科技，教育}）。计算文本与模板的相似度可以得到推荐文本集 Q。若 Q 中属于类别 $\overline{S}$ 的文档数越多，则推荐效果越好。则定义推荐准确度 R 如下：

$$R_{S\overline{S}} = \frac{|C'|}{|C|} \tag{6-4}$$

R 在一定程度上对应于上文中的准确率（Precision），其中|C|为推荐文本数目，$|C'|$ 为推荐文本集中属于类别 $\overline{S}$ 的文档数目。可知 $0 \le R_{S\overline{S}} \le 1$。

需要注意的是，由于使用的是层次聚类，本文的算法实际可以为每个大类别各自生成一个用户模板，对每一类别进行粒度更加细化的推荐，比如单独为体育类生成一个用户模板，从体育类文章中选择用户感兴趣项目进行推荐。此处，考虑到数据集，为了便于度量以及和其他算法的比较，只为几个大类生成一个用户模板。

6.1.3 实验及结果分析

6.1.3.1 系数选取试验

在建立初始模板的过程中，主要用到的系数及其取值分析如下：

（1）求一个词每个义项计算得分时，式子（4-10）中的 N。

若 N 取值过小，会漏掉一些有价值的句子，若取值过大，则与实际情况不符，也不利于计算。本实验取若干篇文章单独进行 4.3.2 节的义项确定的操作，对不同 N 值得到的结果抽样进行检验与对比，此处取 N=5

（2）确定标题中关键词义项时，对正文中的所有保留词汇进行词频统计，保留频率最高的前 M 个词。

由于之前已经进行过去停用词以及词性筛选工作，保留下来的词语比较有价值，所以此处经过试验，充分考虑到效率与准确性的平衡性，此处取 M=8。

（3）合成转移概率时，式（4-25）中的系数 a，b，c。

基于 a，b，c 的含义以及多次试验验证，此处将三种因子对转移概率的影响视为接近的，所以本实验中取 a=c=0.3，b=0.4。

（4）计算类权重影响概率时，式（4-19）中系数β。

β的取值体现了标题与正文的差别，其取值应该满足大于 1。通过对比试验，另β=2 足以满足试验要求。

（5）构建图模型时，若 $sim(s_i, s_j) > T$，则 s_i，s_j 有边相连，T 为相似度阈值。

若此处 T 的取值过大，会造成图模型过于稀疏，T 取值过小，则失去了构造模型的意义。本文分别对 T 取值为 0.5，0.55，0.6，0.65，0.7，0.75，0.8 进行试验，取 S={旅游类，经济类，军事类，体育类}，构建待实验类别集合 $\overline{S}_1$ ={旅游类，经济类}，$\overline{S}_2$={旅游类，军事类}。以 $\overline{S}_1$ 为例，从旅游类，经济类各取 20 篇构成 40 篇文档的训练集，用不同的 T 值训练用户模板，再从 S 的四个类别中随机各选 200 篇文档，构成 800 篇的待推荐文本集，计算待推荐文本与模板间的相似度，取相似度最高的 200 篇文档作为推荐集，计算推荐准确度 R。重复 10 次，取 R 的平均值作为 $\overline{S}_1$ 的准确度。计算 $\overline{S}_2$ 的准确度使用相同的方法。最终得到试验类别集分别为 $\overline{S}_1$ 和 $\overline{S}_2$ 时，T 取值与推荐准确度 R 之间的关系图 13 如下：

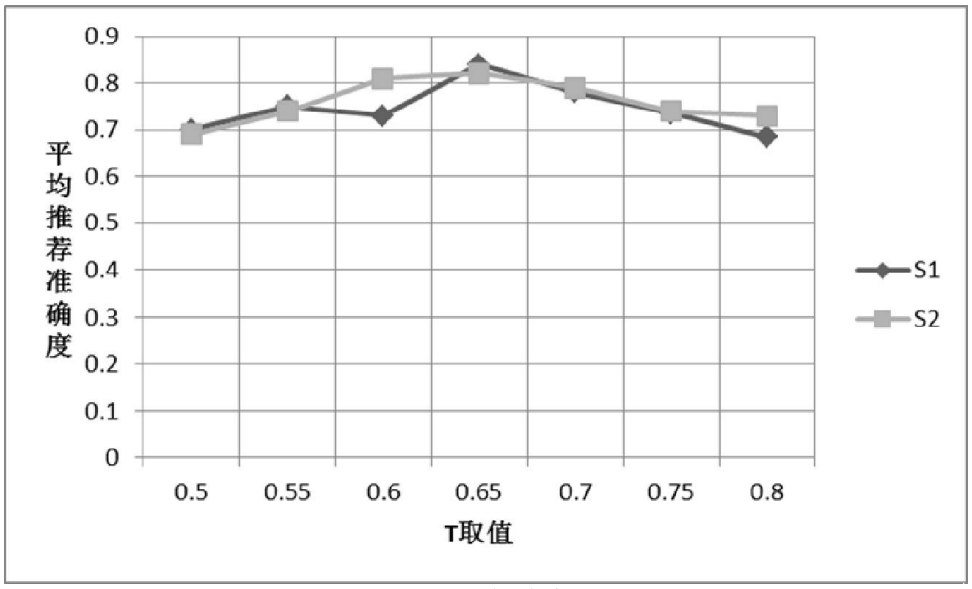

图13 T取值与平均推荐准确度的关系

Figure 13 The relationship between T and the average accuracy of recommendation

对上图进行分析，综合考虑 $\overline{S}_1$ 与 $\overline{S}_2$ 的试验结果，此处取T=0.65。

6.1.3.2 准确率实验

经过上一节，试验中的所用到的参数已经确定。本节基于这些参数进行准确率测试实验。设用户感兴趣文档集文本数目为 N，总类别集合为 S，感兴趣类别集合为 $\overline{S}$。

选取数据集中的旅游类，经济类，军事类，体育类构成 $S^{(1)}$。即 $S^{(1)}$={旅游类，经济类，军事类，体育类}。现以 N=40 为例，进行实验分析。构建待实验类别集合 $\overline{S}_1^{(1)}$ ={旅游类，经济类}，$\overline{S}_2^{(1)}$={旅游类，军事类}，$\overline{S}_3^{(1)}$={旅游类，体育类}，$\overline{S}_4^{(1)}$={经济类，军事类}，$\overline{S}_5^{(1)}$={经

济类，体育类}，$\overline{S}_6^{(1)}$={军事类，体育类}，$\overline{S}_7^{(1)}$={旅游类，经济类，军事类}，$\overline{S}_8^{(1)}$={旅游类，经济类，体育类}，$\overline{S}_9^{(1)}$={旅游类，军事类，体育类}，$\overline{S}_{10}^{(1)}$={经济类，军事类，体育类}。在进行$\overline{S}_1^{(1)}$的准确度测试时，40 篇文档平均的从旅游类，经济类各取 20 篇，训练用户模板，再从四个类别中随机各选 200 篇文档，构成 800 篇的待推荐文本集，计算待推荐文本与模板间的相似度，取相似度最高的 200 篇文档作为推荐集，计算推荐准确度 R。重复 10 次，取 R 的平均值作为$\overline{S}_1^{(1)}$的准确度。计算$\overline{S}_i^{(1)}$的准确度使用相同的方法。最终得到 N=40 时的准确率统计表如下。

表 4 N=40 时准确率统计表

Table 4 Accuracy statistics table when N=40

S	$\overline{S}_1^{(1)}$	$\overline{S}_2^{(1)}$	$\overline{S}_3^{(1)}$	$\overline{S}_4^{(1)}$	$\overline{S}_5^{(1)}$	$\overline{S}_6^{(1)}$	$\overline{S}_7^{(1)}$	$\overline{S}_8^{(1)}$	$\overline{S}_9^{(1)}$	$\overline{S}_{10}^{(1)}$
$\overline{R}$	0.84	0.835	0.86	0.9	0.91	0.82	0.98	0.96	0.98	0.97

使用相同的方法对N=60，N=80，N=100的情况分别进行试验，得到对应的准确率统计表如下：

表 5 N=60 时准确率统计表

Table 5 Accuracy statistics table when N=60

S	$\overline{S}_1^{(1)}$	$\overline{S}_2^{(1)}$	$\overline{S}_3^{(1)}$	$\overline{S}_4^{(1)}$	$\overline{S}_5^{(1)}$	$\overline{S}_6^{(1)}$	$\overline{S}_7^{(1)}$	$\overline{S}_8^{(1)}$	$\overline{S}_9^{(1)}$	$\overline{S}_{10}^{(1)}$
$\overline{R}$	0.86	0.84	0.85	0.915	0.93	0.86	0.98	0.98	0.97	0.97

表 6 N=80 时准确率统计表

Table 6 Accuracy statistics table when N=80

S	$\overline{S}_1^{(1)}$	$\overline{S}_2^{(1)}$	$\overline{S}_3^{(1)}$	$\overline{S}_4^{(1)}$	$\overline{S}_5^{(1)}$	$\overline{S}_6^{(1)}$	$\overline{S}_7^{(1)}$	$\overline{S}_8^{(1)}$	$\overline{S}_9^{(1)}$	$\overline{S}_{10}^{(1)}$
$\overline{R}$	0.9	0.905	0.85	0.9	0.92	0.91	0.98	0.97	0.98	0.99

表 7 N=100 时准确率统计表

Table 7 Accuracy statistics table when N=100

S	$\overline{S}_1^{(1)}$	$\overline{S}_2^{(1)}$	$\overline{S}_3^{(1)}$	$\overline{S}_4^{(1)}$	$\overline{S}_5^{(1)}$	$\overline{S}_6^{(1)}$	$\overline{S}_7^{(1)}$	$\overline{S}_8^{(1)}$	$\overline{S}_9^{(1)}$	$\overline{S}_{10}^{(1)}$
$\overline{R}$	0.905	0.9	0.89	0.89	0.93	0.925	0.99	0.97	0.99	0.98

对 N=40 表中的所有推荐准确率求平均，得到 N=40 的平均推荐准确率$R_{N=40}=0.9055$。使用相同的方法对 N=60，N=80，N=100 求取平均推荐准确率。接下来，选取数据集中的科

技类，教育类，军事类，体育类构成 $S^{(2)}$。即 $S^{(2)}$={科技类，教育类，军事类，体育类}。同时构建待实验类别集合 $\overline{S}_1^{(2)}$ ={科技类，教育类}，$\overline{S}_2^{(2)}$={科技类，军事类}，$\overline{S}_3^{(2)}$={科技类，体育类}，$\overline{S}_4^{(2)}$={教育类，军事类}，$\overline{S}_5^{(2)}$={教育类，体育类}，$\overline{S}_6^{(2)}$={军事类，体育类}，$\overline{S}_7^{(2)}$={科技类，教育类，军事类}，$\overline{S}_8^{(2)}$={科技类，教育类，体育类}，$\overline{S}_9^{(2)}$={科技类，军事类，体育类}，$\overline{S}_{10}^{(2)}$={教育类，军事类，体育类}。使用同样的方法计算出 N=40，N=60，N=80，N=100 的平均推荐准确率。则 $S^{(1)}$与 $S^{(2)}$的准确率统计表及分布图如下：

表 8　$S^{(1)}$与 $S^{(2)}$的准确率统计表

Table 8 Accuracy statistics table of $S^{(1)}$ and $S^{(2)}$

$S^{(i)}$ / 训练文本数量	$S^{(1)}$对应准确率	$S^{(2)}$ 对应准确率
40	0.9055	0.908
60	0.9155	0.9145
80	0.9305	0.929
100	0.937	0.9395

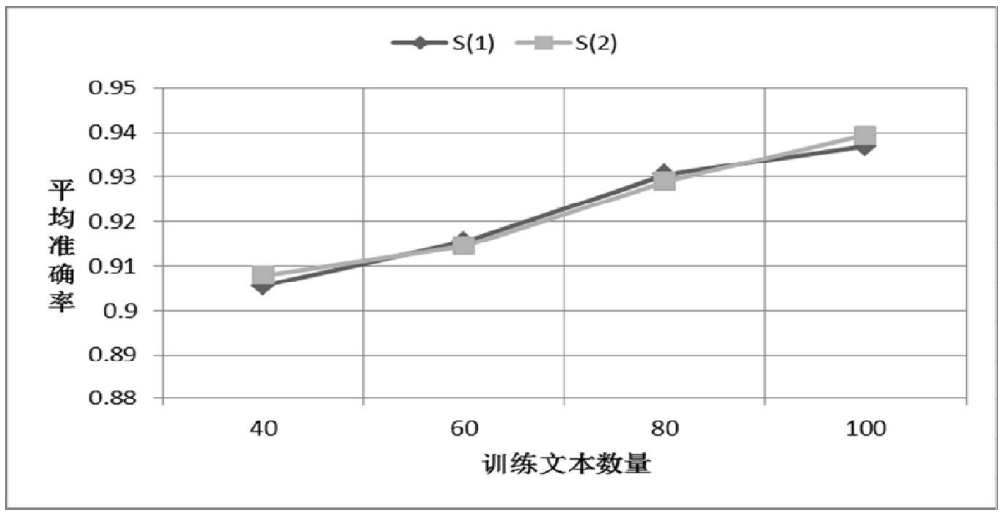

图14 $S^{(1)}$与$S^{(2)}$的准确率分布图

Figure 14 Accuracy distribution diagram of $S^{(1)}$,$S^{(2)}$

由分布图可见随着训练文本数量的增多，准确率有所上升。

6.1.3.3 对比试验

在有少量训练集的情况下，最初构建用户模板的方法为将整个文档集进行简单的关键词筛选，然后由tf/idf为每篇文章生成向量，以这些向量的质心作为用户模板。在此基础上，以语义为基本单位代替以词为基本单位提高了推荐效果。所以此处我们使用以语义为单位的模板进行对比试验。本文首先对文档集进行语义确定及筛选，然后为每篇文章以语义为单位生

成向量，并且计算质心得到初始用户模板。和6.1.3.2节相同，分别对$S^{(1)}$与$S^{(2)}$下的N=40，N=60，N=80，N=100的情况用该模板进行推荐，与本文所提出算法的准确率结果进行对比。比较结果见图15。可见本文中的算法对准确率的确有提升作用。

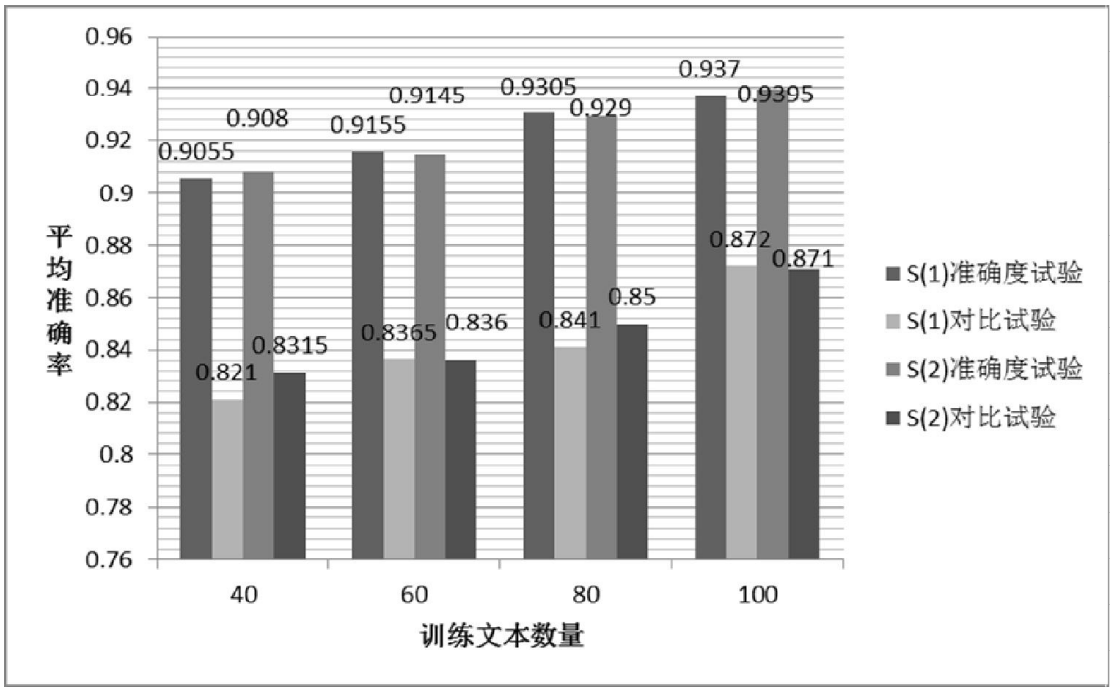

图15 推荐准确率比较图

Figure 15 The recommendation accuracy comparison diagram

6.2 基于伪相关反馈的用户模板更新方法试验

6.2.1 实验数据

和上一节一样，本次试验所用数据集源于新浪新闻等网站。作者通过网络爬虫从各个板块进行新闻获取，构建数据集。整个数据集包含军事类新闻 1877 篇，经济类新闻 1100 篇，科技类新闻 1061 篇，体育类新闻 1600 篇，旅游类新闻 1005 篇，教育类新闻 1000 篇。每条新闻均包含正文部分及标题部分。

6.2.2 评测标准

本次试验主要目的为考察对用户模板进行更新操作之后，推荐效果是否有所提升。如上节所述，对推荐效果常用的评价标准有 MAE 算法，召回率（Recall），准确率（Precision）。本次试验基于以下思路：将数据集按照类别分为 6 个部分，每一部分取小部分文档用于人工标记，从中得到感兴趣文档集 A；将 A 分为两部分 A_1 和 A_2，其中 A_1 用于使用上一节的方法训练初始用户模板，然后对除去 A 剩下的文档进行推荐，人工的对推荐结果进行重新标记，看

其中有多少是自己感兴趣的文档，这样可以计算出推荐准确率；然后用 A_2 对模板进行更新，重新进行推荐以及标记，计算出推荐准确率，将两部分结果进行比较。

根据上面的描述，由于不涉及评分，所以不能使用 MAE 作为评价标准。此外，为了节省标注的工作量，上面使用的方法不易对查全率进行评价，所以这里使用查准率足以描述推荐的效果。公式如下：

$$Precision=\frac{|D_{\text{intrest}}|}{|\text{D}|} \tag{6-5}$$

其中|D|表示推荐文本的总数，$|D_{intrest}|$表示推荐文本中用户感兴趣文本的总数。整个式子表示用户感兴趣文档在推荐文档中所占的比例大小。

6.2.3 实验及结果分析

6.2.3.1 系数选取试验

在更新模板的过程中，主要用到的系数及其取值分析如下：

（1）在划分模板时，（5-2）中 a 的取值。

我们对 a 取不同值时，式子 $d=1-0.5*e^{-an}$ 的函数图像进行观察与分析。图像如下：

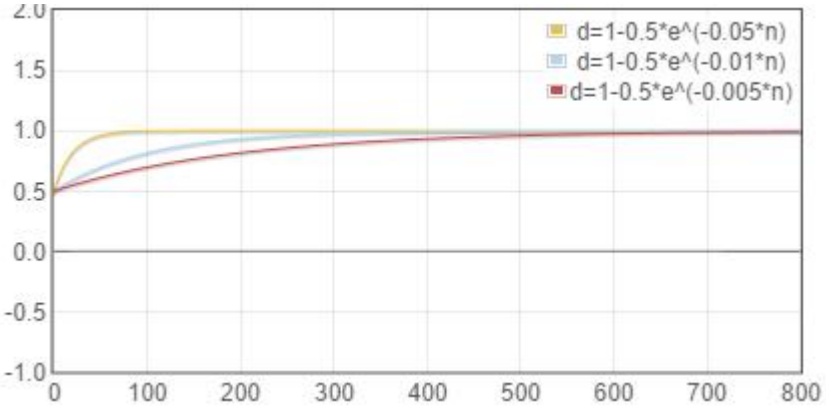

图16 a不同取值对应d函数图

Figure 16 The diagram of d function with different a value

由上图可以看出，在 a 取 0.005 时，当 n=550 时，d 取值趋于 1，符合实际，所以这里 a 的值取 0.005。

（2）更新用户需求模板时，式子（5-3）中 β 的取值。

对军事以及体育两个类数据进行部分标注后用 6.1 节中的方法建立初始模板，然后单独更新用户需求模板部分，取不同 β 值进行推荐测试，观察查准率变化情况。综合实验结果，取 β=0.7 。

（3）5.4.1.2 节中，对聚类结果进行分析，当类别中文档数目小于阈值 K，则剔除这个类别中的文档。

通过单独地对聚类结果进行观察和分析，确定此处 K 的取值为 3。

（4）在 5.4.1.2 节中，计算每一篇文档质量得分 $Score(d_i)$ 时，α，β，γ 的取值。

基于 α，β，γ 的含义以及多次试验验证，此处将三种得分的影响力视为接近的，这里取 $\alpha = 0.4$，$\beta = 0.3$, $\gamma = 0.3$。

（5）在 5.4.1.2 节中，按照文档得分 $Score(d_i)$ 排序后，取 N 个作为相关文档更新模板。

根据实验，对用户兴趣模板中的反馈部分模板单独进行更新，取不同 N 值执行反馈操作，更新后观察推荐的准确率，取准确率最高时 N 的取值。此处取 N=5。

（6）5.4.3 节中改进 Rocchio 公式（5-12）中 β 取值。

根据实验，比对不同 β 取值对推荐效果的提升情况，此处取 β=0.75。

6.2.3.2 准确率实验

本实验将整个数据集按照类别划分为六部分，包括军事类，经济类，科技类，体育类，旅游类，教育类。每个类别选取文档 1000 篇。以军事类为例，试验步骤如下：

（1）从 1000 篇文档中选出 500 篇用于人工标记，标注出感兴趣文档；然后从标注好的感兴趣文档中划分出 70 篇组成集合 S。

（2）在 S 中随机选取 60 篇文档构成集合 S_1，剩下的 10 篇文档构成集合 S_2。

（3）对于集合 S_1，利用基于 TextRank 的初始模板建立方法建立初始用户兴趣模板 p。

（4）利用 p 对除了 S 外的剩下 930 篇文档进行推荐，取相似度最高的 200 篇作为推荐集 R_1。

（5）人工的对 R_1 进行标记，考察 R_1 中自己感兴趣的文档数目。

（6）由人工标识的结果计算出查准率 $Precision_1$。

（7）将 S_2 部分视为新添加的用户感兴趣文档，更新用户需求模板，得到新的用户模板 p_2。

（8）用 p_2 对除了 S 外的剩下 930 篇文档进行推荐，取相似度最高的 200 篇作为推荐集 R_2。人工的对 R_2 进行标记，考察 R_2 中自己感兴趣的文档数目。

（9）由人工标识的结果计算出此时查准率 $Precision_2$。

（10）此时利用伪相关反馈再次对模板进行更新，得到总的用户模板 p_3。

（11）用 p_3 对 930 篇文档进行推荐，取相似度最高的 200 篇作为推荐集 R_3。人工的对 R_3 进行标记，考察 R_3 中自己感兴趣的文档数目。

（12）由人工标识的结果计算出此时查准率 $Precision_3$。

（13）比较 $Precision_1$，$Precision_2$，$Precision_3$ 大小。评测模板更新算法的效果。

按照上面的步骤，得到六个类别的 $Precision_1$，$Precision_2$，$Precision_3$ 结果见下表：

表 9 模板更新与查准率变化情况

Table 9 Precision changes when profile updating

类别 \ Precision	$Precision_1$	$Precision_2$	$Precision_3$
军事类	0.71	0.785	0.79
经济类	0.69	0.77	0.78
科技类	0.735	0.8	0.83
体育类	0.73	0.81	0.85
旅游类	0.68	0.785	0.845
教育类	0.74	0.8	0.835

对六个类别的 $Precision_1$，$Precision_2$，$Precision_3$ 值分别求平均，得到 $\overline{Precision_1}$=0.714，$\overline{Precision_2}$=0.792，$\overline{Precision_3}$=0.822。三者的变化示意图如下：

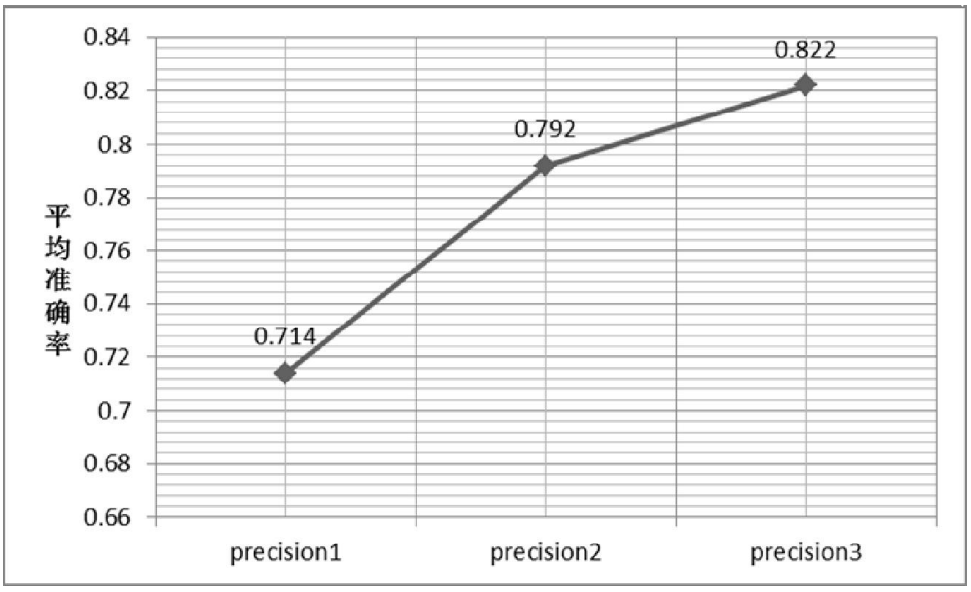

图17 模板更新与查准率变化图

Figure 17 Precision changing diagram when profile updating

有上图可以得出结论：（1）本文提出的模板更新算法对提高推荐准确率有明显的作用。（2）$Precision_1$ 到 $Precision_2$ 的涨幅比 $Precision_2$ 到 $Precision_3$ 的涨幅要大一些，其主要原因是计算 $Precision_2$ 时对应的模板是由用户提供数据更新得来的，符合用户的真实需求；而计算 $Precision_3$ 时，利用了伪相关反馈更新模板，没有用户直接参与，所以可能会稍有偏差，但总体还是对推荐效果有改善作用。

6.3 本章小结

本章分别对上文提出的算法进行了试验。首先，对于基于 TextRank 算法的初始模板建立方法进行检验。通过网络爬虫从相关新闻网站的各个模块搜集新闻，构建数据集。鉴于试验的实现难度及工作量大小，此处提出了恰当的评测标准，分别进行了系数选取试验，准确性试验以及对比试验。通过对比试验可以看出该算法的确对提高初始模板的精度有较大的作用。其次，本章对基于伪相关反馈的用户模板更新方法也进行了检验。该处使用和上面一样的数据集，主要采用准确率作为评测标准，在通过试验确定参数取值之后，本处进行了准确率试验：用初始模板进行推荐，计算查准率，然后分别对用户需求模板和反馈部分模板进行更新，计算查准率，与前面的结果进行比较和分析。试验结果显示，本文提出的模板更新算法能有效提高推荐精度。

7 总结与展望

7.1 本文工作总结

本文首先介绍了推荐系统的概念，对推荐系统在信息过载时代中的特殊意义进行了阐述。鉴于其在商业应用以及学术研究方面都有很大的研究价值表明本文的研究是有实际意义的。此外，简要介绍了现在常用的推荐算法以及他们之间的优缺点，由于自然语言处理和机器学习技术日益成熟，基于内容的推荐系统在文本推荐领域有着广泛的应用，本文研究的就是基于内容对相关长文本进行推荐。

在文本推荐过程中，需要涉及到一些自然语言处理方面的基本技术，这是整个内容推荐系统的基础。本文对其中的中文分词技术，文本表示模型进行了研究。文本分词技术本文主要介绍了基于词典的分词方法等三大类方法。本文中主要会用到基于词典和基于统计的分词方法。文本表示模型包括布尔模型，向量空间模型，概率模型，基于模糊集的模型等。本文的研究主要是建立在向量空间模型基础上，并使用了改进的权重计算方法。鉴于本文主要研究可以进行自学习的自适应推荐系统，本文详细描述了信息检索以及相关反馈技术，对其中的显式反馈，隐式反馈，伪相关反馈进行描述，这些在本文的系统中具有重要的作用。

在基于内容的推荐系统中，初始用户模板的准确性对后面的推荐精度有很大影响。因此，在系统初始时，必须从少量用户信息中准确地提取出用户兴趣模板，尽可能的减少噪声的引入。否则会在后期更新模板时产生偏移性问题，造成推荐的不准确。本文提出了一种基于 TextRank 算法建立初始模板的方法，有效的利用了用户信息中的集簇性，首先对所拥有的少量用户感兴趣文本进行预处理并确定词义项，然后进行聚类，接下来对聚类得到的每个类别分别以义项为单位构建 TextRank 模型，并引入各种影响力因子对 TextRank 模型中的概率转移矩阵进行改进。迭代之后选取每个类中最为关键的若干义项进行综合，得到最终的初始用户模板。该模板减少了噪声的引入，可以为系统的初始阶段提供好的推荐效果。

在内容推荐系统中，用户模板更新和初始用户模板的建立一样重要。因为这决定着系统能否持续的向用户进行精确推荐。假如不对用户模板进行及时的更新，系统显然滞后于用户需求，产生偏移性现象，随之带来不好的用户体验。本文针对用户模板更新的不同需求，提出了一种用户模板更新方法。将用户模板分解为两部分，一部分主要针对用户感兴趣文档的增多，另一部分致力于解决推荐范围过窄的问题，并且讨论了两部分的加权方式。对于第一部分，本文使用显式相关反馈的方法对其进行更新；对于第二部分，主要基于伪相关反馈。本文详细的讨论了反馈部分模板更新时改进的相关文档的选取方法、关键词选取方法、模板更新方法，以此减少伪相关中的偏移性问题。最后对整个用户模板更新。

第六章的实验表明，本文提出的主要由上面两个模块组成的基于内容的自适应推荐系统有较好推荐效果。

7.2 研究展望

本文提出的自适应文本推荐系统可以在如下两个方面进一步研究以及扩展，以提高推荐效果，使算法更加健壮：

1） 与其他推荐算法结合

基于内容推荐系统毫无疑问会受到信息提取技术的限制，所以现在主要还是使用在文本推荐方面。而协同过滤等算法基于找出用户之间的相似性，对特征提取要求并不高，并且现在对社区网络的研究有效的对协同过滤进行了补充。将这些技术与基于内容推荐系统算法结合，可以将社区网络研究与成熟的自然语言处理技术有效结合，进一步提升推荐准确率以及效率。

2） 基于 TextRank 算法的初始模板建立方法实验优化

该实验衡量了试验的易实现性和有效性，采用分类文本集作为测试集，大致的反映了算法对推荐效果的提升作用。但是，此实验还有可以改进的方面，这里我们为军事类，经济类，科技类，体育类，旅游类，教育类六个类别只建立了一个模板测试推荐效果。实际上，一般推荐系统面对这种情况会为每个类别各生成一个模板，比如，为体育类生成一个模板，专门对该类的文章进行推荐，这样使推荐粒度更加细化。所以，举例而言，后续试验可以为体育类专门建立模板，然后对体育类内的推荐进行准确度测试。

参 考 文 献

[1] 项亮. 推荐系统实践 [M]. 北京: 人民邮电出版社, 2012.
[2] BELKIN N J, CROFT W B. Information filtering and information retrieval: two sides of the same coin[J]. Communications of the ACM, 1992, 35(12): 29-38.
[3] ADOMAVICIUS G, TUZHILIN A. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions [J]. Knowledge and Data Engineering, IEEE Transactions on, 2005, 17(6): 734-49.
[4] 杨博, 赵鹏飞. 推荐算法综述 [J]. 山西大学学报(自然科学版), 2011, 03): 337-50.
[5] LAWRENCE R D, ALMASI G S, KOTLYAR V, et al. Personalization of supermarket product recommendations [M]. Springer, 2001.
[6] RICH E. User modeling via stereotypes* [J]. Cognitive science, 1979, 3(4): 329-54.
[7] 孙冬婷, 何涛, 张福海. 推荐系统中的冷启动问题研究综述 [J]. 计算机与现代化, 2012, 05): 59-63.
[8] RESNICK P, VARIAN H R. Recommender systems [J]. Communications of the ACM, 1997, 40(3): 56-8.
[9] PAZZANI M, BILLSUS D. Learning and revising user profiles: The identification of interesting web sites [J]. Machine learning, 1997, 27(3): 313-31.
[10] BALABANOVIĆ M, SHOHAM Y. Fab: content-based, collaborative recommendation [J]. Communications of the ACM, 1997, 40(3): 66-72.
[11] WEBB G I, BOUGHTON J R, WANG Z. Not so naive Bayes: aggregating one-dependence estimators [J]. Machine learning, 2005, 58(1): 5-24.
[12] 焦李成. 神经网络系统理论 [M]. 西安: 西安电子科技大学出版社, 1990.
[13] QUINLAN J R. Induction of decision trees [J]. Machine learning, 1986, 1(1): 81-106.
[14] CORTES C, VAPNIK V. Support-vector networks [J]. Machine learning, 1995, 20(3): 273-97.
[15] 崔春生, 吴祈宗. 基于 Vague 集的内容推荐算法研究 [J]. 计算机应用研究, 2010, 06): 2109-11.
[16] 刘建国, 周涛, 汪秉宏. 个性化推荐系统的研究进展 [J]. 自然科学进展, 2009, 01): 1-15.
[17] 孙铁利, 刘延吉. 中文分词技术的研究现状与困难 [J]. 信息技术, 2009, 07): 187-9+92.
[18] 王懿. 基于自然语言处理和机器学习的文本分类及其应用研究 [D]; 中国科学院研究生院（成都计算机应用研究所）, 2006.
[19] SALTON G. The SMART retrieval system—experiments in automatic document processing [J]. 1971,
[20] SALTON G, MCGILL M J. Introduction to modern information retrieval [J]. 1983,
[21] 高珊. 信息检索中的查询扩展及相关技术研究 [D]; 华中师范大学, 2008.
[22] ROBERTSON S E, JONES K S. Relevance weighting of search terms [J]. Journal of the American Society for Information science, 1976, 27(3): 129-46.
[23] ZADEH L A. Fuzzy sets [J]. Information and control, 1965, 8(3): 338-53.
[24] 娄娟. 模糊理论在文本分类中的应用研究 [D]; 重庆大学, 2011.
[25] 王伟. 基于 Vague 集理论的推荐与模糊决策相关算法研究 [D]; 西北大学, 2014.
[26] MANNING C D, RAGHAVAN P, SCH TZE H. Introduction to information retrieval [M]. Cambridge university press Cambridge, 2008.
[27] 胡保祥. 基于查询日志的查询扩展研究 [D]; 北京邮电大学, 2013.
[28] MILLER G A, BECKWITH R, FELLBAUM C, et al. Introduction to wordnet: An on-line lexical database* [J]. International journal of lexicography, 1990, 3(4): 235-44.
[29] ROCCHIO J J. Relevance feedback in information retrieval [J]. 1971,
[30] 梅家驹，等. 同义词词林 [M]. 上海: 上海辞书出版社, 1993.

[31] 田久乐, 赵蔚. 基于同义词词林的词语相似度计算方法 [J]. 吉林大学学报: 信息科学版, 2010, 6): 602-8.
[32] FRED A L, LEITAO J M. Partitional vs hierarchical clustering using a minimum grammar complexity approach [M]. Advances in Pattern Recognition. Springer. 2000: 193-202.
[33] 吴夙慧, 成颖, 郑彦宁, 等. K-means 算法研究综述 [J]. 现代图书情报技术, 2011, 27(5): 28-35.
[34] MIHALCEA R, TARAU P. TextRank: Bringing order into texts, F, 2004 [C]. Association for Computational Linguistics.
[35] LANGVILLE A N, MEYER C D. Google's PageRank and beyond: The science of search engine rankings [M]. Princeton University Press, 2011.
[36] 夏天. 词语位置加权 TextRank 的关键词抽取研究 [J]. 现代图书情报技术, 2013, 29(9): 30-4.
[37] 黄云平, 孙乐, 李文波. 基于上下文图模型文本表示的文本分类研究; proceedings of the 第四届全国信息检索与内容安全学术会议, 中国北京, F, 2008 [C].
[38] 迟呈英, 李红. 基于改进 TF* PDF 算法的网络新闻热点话题检测和跟踪 [J]. 计算机应用与软件, 2013, 30(12): 311-4.
[39] 李大高. 信息检索中的查询扩展算法研究 [D]; 江苏大学, 2008.
[40] MITRA M, SINGHAL A, BUCKLEY C. Improving automatic query expansion; proceedings of the Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval, F, 1998 [C]. ACM.
[41] 叶正. 基于网络挖掘与机器学习技术的相关反馈研究 [D]; 大连理工大学, 2011.

致　谢

首先，我要感谢我的导师刘功申副教授。刘老师严谨的治学态度和科学的工作方法给了我极大的帮助和影响。在两年半的时间内，刘老师给了我很多有用的建议和帮助，耐心的解答我在学习中遇到的困惑，为我指明了学习方向，使我不断进步。同时，刘老师对我的职业规划也给出了指导，让我受益匪浅。这些对于我以后的成长都有着深远影响。

其次，我要感谢和我同组的黄晨，陈博文，周志杰，吴蔚蔚。和他们的交流总是让我学到很多东西。在这两年半的时间内，他们也给予了我许多帮助。

同时，我要感谢我所有的同学以及朋友们，两年半以来，在生活中，我们互相帮助，互相包容，度过了许多快乐的时光。在学习中，我们互相交流，互相学习，开阔了视野。

最后，我要向我的父母致谢，他们一直无条件的理解和支持着我，给予我无微不至的关怀和帮助。在我遇到困难的时候，他们会耐心的安慰我，支持我，给予我鼓励，给予我信心，指导我如何坦然去面对，从困难与失败中汲取经验壮大自我。没有他们的培养，我不会这样快速成长。

在这即将毕业之际，谨向所有给予我帮助的朋友和老师们致以最诚挚的谢意，感谢你们为我的进步提供动力。

攻读硕士学位期间已发表或录用的论文

[1] 段准，刘功申. 基于 TextRank 的内容推荐系统用户模板构建方法. 计算机技术与发展（已录用）.

上海交通大学

学位论文原创性声明

本人郑重声明：所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的作品成果。对本文的研究做出重要贡献的个人和集体，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：段准

日期：2015 年 1 月 13 日

上海交通大学

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，同意学校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。本人授权上海交通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。

本学位论文属于

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：段准　　指导教师签名：刘功申

日期：2015年1月13日　　日期：2015年1月