1 引 言

1.1 论文的研究背景和选题依据

随着因特网的迅猛发展，网络资源信息量每8个月要翻一翻，如今的网页以10亿来计算，而且其信息量仍在以指数形式飞速增长。因特网已成为人们获取信息的重要来源，由于因特网的广泛性和开放性，在因特网上发布信息极为容易而且不受限制，无论任何单位、团体、个人只要具备上网条件便可以自由地在因特网上发布信息；从而加剧了因特网信息的急剧膨胀。而因特网已成为人们获取信息的重要来源；因此，如何快速、准确地从浩瀚的信息资源中寻找所需的信息已经成为困绕用户的一个难题。

搜索引擎技术要用到信息检索[1-7]、人工智能、计算机网络、分布式处理、数据库、数据挖掘、数字图书馆、自然语言处理等多个领域的理论和技术，所以具有综合性和挑战性。伴随互联网的普及和网上信息的爆炸式增长，它越来越引起人们的重视。用户在使用搜索引擎进行信息搜索时，有时并不十分关注返回结果的多少，而是看检索结果是否符合自己的需求。对于一次普通查询，传统的搜索引擎动辄几十万、几百万篇文档，这样的搜索结果是没有多大意义的，用户一般只能挑选在前面的几十条信息进行察看，尽管许多搜索引擎利用相关性的技术在这方面做了改进，也允许用户在前次检索结果的基础作进一步检索，但是查看检索结果的不便的问题始终未能得到彻底的解决，用户常常为此苦恼和困惑，这个问题也被称做“太多数据但没有足够的信息”，其主要缺陷就是没有对检索结果分类和按用户查询意图来组织。

本文首先利用人类语言中普遍存在的Zipf定律，对大量文本集进行局部和全局的特征提取，提取更少的特征来尽可能的表达出文本蕴含的信息，得到一种有效的

文本特征提取方法，其中也对文本特征进行约简从而得到少而精的特征信息。接着，将概念格理论基础和粗糙集理论、语义本体理论结合起来，改进了基本Tversky相似度模型。从形式概念的结构相似和语义相似两个层次上衡量概念间的相似度，并将两者结合起来构成最终的形式概念相似度衡量方法以体现形式概念的真实相似度。然后针对分布式环境，提出并构建一个协作式信息检索框架，实现从结构和语义两个层次上的概念匹配，获得更符合用户需求的检索结果。最后在对检索结果排序过程中，再一次利用语义本体WordNet对用户历史检索记录进行分析，从而构建用户兴趣模型，以便更精确的表达出用户兴趣所在，并按照此模型对初始检索结果重新排序，体现用户个性化搜索过程。

1.2 论文的研究意义及主要的研究内容

概念格是对概念以及概念之间关系的描述，是形式概念分析（Formal Concept Analysis，FCA）的核心数据结构。它在一定程度上是对客观世界的一种高度简化的描述形式。这种简化的最大优点是其具有良好的数学性质。Wille基于这种简化，开创了形式概念分析领域。但是，正是由于这种简化，使得概念格不能简单地被理解为客观世界的部分模型，而更多的是被看作一个人造的从一些数据集派生出来的表示。

国内外针对信息检索领域中提出了很多方法及理论[8-10]，其中，使用形式概念分析理论来进行信息检索也有很多深入的研究，如，使用 FCA 构建表示信息模型以更确切的表示原始检索模型，并使用构建出来的概念格进行信息检索，基于抽象具体程度的不同进行检索可获得独特层面上的检索结果。另外，在分布式环境下，使用 FCA 理论和技术进行协作信息检索，这可大大提高检索时间和效率。

协作信息检索在当今的网络体系结构下，更加符合分布式信息检索需要并在时空效率和准确度上有一定程度的提高。Web 信息与日俱增，其种类结构也是丰富多样，尤其文本信息更是巨大，为了能在大数据集中更快速、更有效的检索出更符合用户需求的结果，使用基于语义的一系列策略。如：提取大数据集中有意义的特征；基于概念格的理论和技术，构建多个概念格，并根据概念

间的相似度进行匹配，并可在分布式环境下进行检索；通过对用户的历史查询记录进行提取，建立用户兴趣模型(User Interest Model)并利用该模型对检索结果重新排序；这样形成了一整套即具有时空效率又能符合用户个性的信息检索方案。

在此基础上，本文具有以下几个研究内容：

1) 根据文本各个特征自身和类别信息的统计特性，提取更少的特征来尽可能的表达出文本蕴含的信息，得到一种有效的文本特征提取方法，其中也对文本特征进行约简从而得到少而精的特征信息。
2) 基于概念格理论，结合粗糙集理论和语义本体理论，从形式概念的结构相似和语义相似两个层次上衡量概念间的相似度形式一种有效确切的概念相似度衡量方法。
3) 针对分布式环境，提出并构建一个协作信息检索框架，实现从结构和语义两个层次上的概念匹配，获得更符合用户需求的检索结果。
4) 在对检索结果排序过程中，利用语义本体 WordNet 对用户历史检索记录进行分析，从而构建用户兴趣模型，以便更精确的表达出用户兴趣所在，并按照此模型对初始检索结果重新排序。

1.3 论文研究目的及创新点

本文基于概念格理论并结合粗糙集、语义本体、文本特征提取等理论和技术，从信息检索的文档简洁表示、相似度模型、分布式搜索框架及个性化搜索这几个方法进行研究。本文主要目标是给出利用概念格的特性实现一套完整的协作信息检索框架系统。本文主要的创新点如下：

1) 利用人类语言中普遍存在的 Zipf 定律和大文本集中各个特征的自身统计规律，进行特征提取，去除冗余特征和噪声，精简原检索文档集，使得在尽可能不损失确切含义的基础上能更少的保留文档特征以降低文档特征维数。
2) 利用粗糙集和语义本体理论，对 Tversky 相似度计算公式进行改进，实现从形式概念的结构层次和语义两个层次上进行形式概念相似度的衡量。
3) 提出有效的协作信息框架并给出相关算法。
4) 提出用户兴趣模型的构建方法，并通过此模型来实现个性化搜索结果排序方法，从而达到多个用户针对同一搜索关键词，但由于个人兴趣的不同使

获得的检索结果条目以不同的顺序返回。

1.4 论文的内容组织

本文共分为七章，各章节内容如下：

第一章 主要介绍论文的研究背景、选题依据、研究意义，以及论文选题的创新点和主要研究内容。

第二章 关于概念格、语义本体、协作信息检索的相关理论和技术。主要介绍了本体理论、概念格的模型及其常见的构造算法以及协作信息检索的基本概念。

第三章 基于类别分析及有效特征提取的文本分类方法。为了提取更少的特征来尽可能的表达出文本蕴含的信息，通过分析各个特征自身的统计特性，根据 Zipf 定律进行全局特征提取，不提取文本特征空间中普遍存在的特征和噪声特征；其次，在对特征的类别信息进行统计分析后，计算出每个特征词的类别贡献程度，进行类内局部特征提取；在选取有效的特征之后，提出新的特征权重计算公式 TF-IDF-CF 并通过实验验证此特征提取策略是准确有效的。

第四章 结构语义相似度计算。基于概念格的数学理论基础，结合粗糙集理论和语义本体理论，提出了一个形式概念相似度计算方法，改进了基本 Tversky 相似度模型。从形式概念的结构相似和语义相似两个层次上衡量概念间的相似度，并将两者结合起来构成最终的形式概念相似度衡量方法，这种方法在不同层面上度量了形式概念的真实相似度。

第五章 基于FCA的协作信息检索框架。在协作信息检索中，针对多个子文本数据库，利用渐进式构格算法构建各个子形式概念格，然后分别利用形式概念相似度的衡量方法对已有形式概念进行匹配，在找到临时概念集之后采用合并的方式，获得新的概念，再对新的概念进行相似度的匹配，最终获得满足用户需求的结果集合。整个系统充分体现了协作的思想，便于在分布式环境中部署实施，并且在匹配

查询关键词时使用了不精确的方式，从结构和语义两个层次上进行衡量，更好的体现了人性化需求。

第六章 基于WordNet的检索结果个性化排序方法。人们通过搜索引擎去寻找Web上的信息资源。虽然传统的信息检索技术满足了人们一定的需要，但由于其通用的性质，不能满足不同用户的特定查询请求。在用户输入查询关键字后，搜索引擎返回的结果非常之多，以致用户还得手工筛选出自己感兴趣的检索结果，这个过程需要花费大量的时间和精力。根据不同用户检索的历史记录(如检索关键字，在检索结果中的点击情况，在各个网站的访问情况等)，返回更适合这个用户的检索结果。利用语义本体WordNet对用户历史检索记录进行分析，从而构建用户兴趣模型。其中，不同的本体词汇采用得分机制并根据不同词汇类型赋予不同数值，以便更精确的表达出用户兴趣所在，进一步为初始检索结果的重新排序提供一个度量依据。

第七章 展望与结束语。

2 协作信息检索及相关理论

由于目前的网络上的信息资源与日俱增，这些繁而杂的信息已经给用户造成了严重的“信息过载”问题。通常传统的信息检索模型的效果却远远令人满意，所以，在信息检索领域渐渐提出了信息检索的分布协作策略以替代原本的集中式信息检索方式。而在这个改进的信息检索模型中，集合了很多优秀的理论基础来达到更用户友好的检索体验，从而提高检索效率。比如，在信息检索模型结合语义本体，可以揣测用户的语义信息，也可以使用语义上的，而不是单一结构上的信息衡量资源的匹配程度。同样的，利用形式概念分析改进了传统的信息检索模型，可以很方便的设计出一个分布协作式的Web信息检索系统。语义本体，形式概念分析可以从不同的角度上扩展信息检索的范畴，并能够在一定程度上解决传统信息检索的匮乏问题，因此在本文中将结合语义本体和形式概念分析来对传统的信息检索进行改进，提出一系列行之有效的理论及方法。其中，概念格是形式概念分析中的核心基础，由德国的R.Wille教授于1982年提出[11]。发展至今已经被应用于许多领域，如软件工程[12,13]、web挖掘[14]等。

本章分别针对这三个方面进行详细的介绍，主要包含本体、概念格的基本内容，涵盖这些理论中的基本概念、数学理论，以及协作信息检索的理论技术上的相关研究。

2.1 语义本体

最初，语义本体Ontology[15]的概念起源于哲学领域，是“对世界上客观存在物的系统地描述”。在人工智能界，最早给出本体定义的是Neches等人，他们将本体定义为“给出构成相关领域词汇的基本术语和关系，以及利用这些术语和关系构成的规定这些词汇外延的规则的定义”。 后来在信息系统、知识管理等领域，越来越多的人研究本体，并给出了许多不同的定义。例如：Gruber的定义[16]强调了本体是知识表示的元级描述；Wielinga和Schreiber的定义[17]强调了ontology在知识级的形式化，表示应用于可知识化的Agent中的知识；而人工智能的某些领域中，Ontology作为术语学的同义词，表示术语的语义解析，Alberts的定义[18]主要面向强调应用领域的概念术语分类。其中Gruber的定义被引用最多的：本体是共享概念化的形式

化、显式的说明[19]。随着人们对本体理解的不断加深，本体的概念也被不断修改，但是许多定义共享了关于本体最核心的部分：本体概念以及本体概念之间的IS－A关系。Fensel对这个定义进行分析后认为本体的概念包括四个主要方面：

概念化（conceptualization）：客观世界中现象的抽象模型；

明确（explicit）：概念及它们之间联系都被精确定义；

形式化（formal）：精确的数学描述；

共享（share）：本体中反映的知识是其使用者共同认可的。

虽然不同研究者对本体有不同的描述，但是从内涵上来看，他们对本体的认识是一致的，都是把本体当作某个领域内（可以是特定领域的，也可以是更广的范围）不同主体（人、代理、机器等）之间进行交流（对话、互操作、共享等）的一种语义基础，即由本体提供明确定义的词汇表，描述概念和概念之间的关系，作为使用者之间达成的共识。因此，本体的用途包括交流、共享、互操作、重用等等。

目前，Ontology已经被广泛应用于知识工程、自然语言处理、数字图书馆、信息检索和Web异构信息的处理、软件复用、面向对象技术和语义Web等领域。

在本文中，引入语义本体理论正是由于它自身具有的构建知识库，而且此知识库是可以推导出语义涵义的特性。在信息检索过程中，考虑到需要采取何种方式去更确切的描述用户的原始意图，基于语义本体能够很好的揣测出用户的兴趣及意愿，进而更好的进行搜索。

2.1.1 本体分类

由于研究ontology的机构和组织很多，各种ontology定义抓住了ontology各方面的特性，因此存在着不同的ontology的分类方式[20]。这里介绍三种典型的分类方式，一种是根据ontology的通用性级别，在建立一个系统的过程中，按所实现的不同功能确定ontology的不同类型；第二种是根据ontology按照概念化的结构数量和类型进行分类；另一种是根据ontology所刻画和描述的现实世界的不同方面进行ontology分类。

第一种分类方式把ontology分为以下四种类型[21]

领域ontology：针对特定的应用领域抽象领域知识的结构和内容，包括各种领域知识的类型、术语和概念，并对领域知识的结构和内容加以约束，形成描述特定领域中具体知识的基础。图2.1描述了关于出版物ontology的两种表示形式。

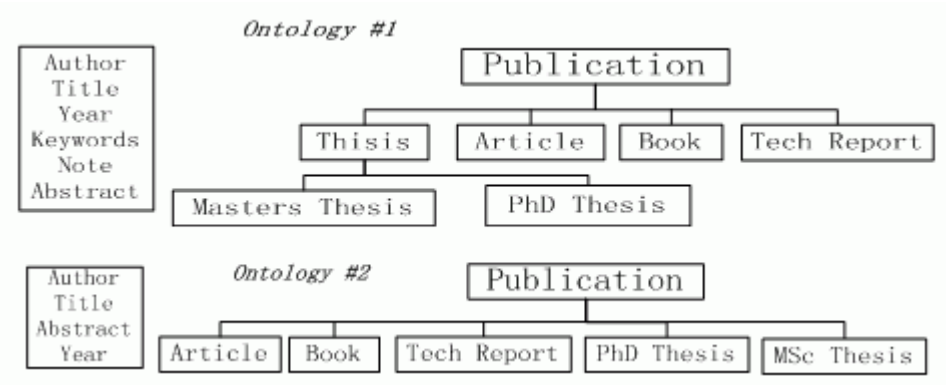

图2.1 出版物ontology的两种表示形式

通用ontology：针对获取关于世界的通用性知识，提供基本的观念和概念，如时间、空间、状态等。用来描述知识对象，包括描述知识对象的基本概念与属性，如标题、作者、关键字、日期等，主要用于对知识对象进行标注。目前，都柏林核心元数据非常适用于对知识对象（如电子文档、数据库中的记录、问题解决方案、web页面）进行描述，因此该模型用包含十五个描述属性的都柏林核心集描述信息ontology,如图2.2。

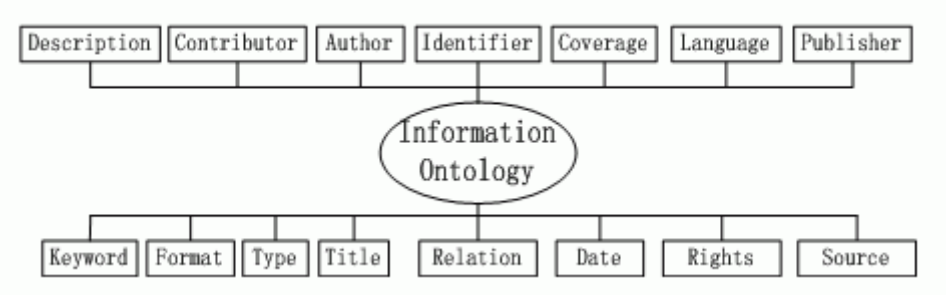

图2.2 都柏林核心集

应用ontology：针对特定应用领域知识建模的抽象定义。通常，应用ontology是一种概念的混合，这些概念来自领域ontology和通用ontology。

表示型的ontology：主要描述在知识表示形式化背后的概念化，而不致力于任何特定的领域，这种ontology提供表示性的中性实体，即提供表示的框架，而不描述什么应该被表示以及怎样表示。

第二种方式把ontology分为以下三种类型:

术语学ontology：类似于词典，定义了从不同方面表示知识的术语。在医学领域，这种ontology的一个示例是UMLS中的语义网络[22]；

信息ontology：定义了数据库的记录结构，数据库模式是这类ontology的一个示

例。在医学领域，这类ontology一个典型示例是PEN&PAD模型的Level1，一种针对病人医学记录建模的框架[23]。在Level 1上，模型提供了一种框架记录对病人的基本观察，但没有在症状、信号、治疗等方面进行区分；

知识建模ontology：定义了知识的概念化。与信息ontology相比，知识建模ontology通常具有更加丰富的内部结构，进而，这类ontology通常适用于一些特定的知识。在医学领域，Rector的Level 2描述是这类ontology的一个示例，在Level 2级别上，把Level 1对病人的基本观察分组，描述做决策的过程。

第三种分类方式把ontology分为以下四种类型[24]

静态的ontology：描述世界中静态方面的特征，即，存在的事物、它们的属性以及它们之间的关系；

动态的ontology：描述世界中不断变化的方面，典型的原语概念包括了状态、状态转换和过程描述世界；

意念型的ontology：包括动机、意图、目标、信念、选择等，典型的原语概念包括论题、目标、支持、否决、子目标、主体等；

社会型的ontology：包括社会结构、组织结构、联盟等，社会型的ontology通常用行为者、位置、角色、权威、承诺等原语概念进行刻画。

2.1.2 本体描述模型

自上个世纪90年代以来，一些基于AI的本体实现语言陆续被提出，如KIF[25]、Ontolingua[26]、Loom[27]、FLogic[28]。后来，随着Web的发展，又出现了一系列基于Web的本体语言，也叫做本体标记语言，如SHOE[29]、OML[30]、XOL[31]、RDF[32]、RDF-S[33]。 这些语言在给出了本体的描述方法的同时给出了本体描述模型。

目前由W3C 主持制定的RDF(resource description of framework)和RDF Schema是建立在XML 语法上，以语义网为理论基础，对信息资源进行语义描述的语言规范。RDF 采用“资源”(resources) 、“属性”(properties) 以及“声明”(statements) 等三元组来描述事物，RDFSchema 则做进一步扩展，采用了类似框架的方式，通过添加原语，对类、父子类、父子属性以及属性的定义域和值域等进行定义和表达。这样RDF成为一个能对本体进行初步描述的标准语言。

然而本体描述语言要走向通用，还需解决一些重要问题，如对推理的有效支持(包括计算复杂性和可判定性等)、正规和充足的语义表示机制以及标准化问题。这

些问题用描述逻辑可以得到部分解决。描述逻辑[34,35]是近二十多年来人工智能领域研究和开发的一个相当重要的知识表示语言，它以数学中的逻辑表示为基础。目前正被积极应用于本体描述，或者作为其他本体描述语言的基础。表2.1对各种不同的本体描述语言和描述框架做了比较。

表2.1 各种本体描述框架的比较

	SHOE	OML/CKML	RDF	OIL	DAML+OIL	OWL
语法	HTML/XML	XML	XML	RDF/XML	RDF/XML	RDF/XML
正规语义	有	有	无	有	有	有
类的层次	支持	支持	支持	支持	支持	支持
Horn逻辑	是	否	否	否	否	否
描述逻辑	否	否/是	否	是	是	是
谓词逻辑	否	否	否	否	否	否
类的相等	支持	支持	不支持	不支持	支持	支持
实例相等	不支持	不支持	不支持	不支持	支持	支持
本体分部定义	支持	支持	支持	支持	支持	支持
本体扩展	支持	不支持	支持	支持	支持	支持
本体版本修订	支持	不支持	不支持	不支持	不支持	支持
计算特性区分	无	有	无	有	无	有

2.1.3 语义本体 WordNet

WordNet[36]是一个覆盖范围宽广的英语词汇语义本体。它是以同义词集合(synset)作为基本建构单位进行组织的。除了用同义词集合的方式罗列词汇概念之外，同义词集合之间以一定数量的关系类型相关联，这些关系包括上下位关系，整体部分关系和继承关系等。它对词汇概念的层级描述，以及关于名词间同义，反义，上下位关系，整体部分关系等的描述在解释语言行为方面发挥着重要作用，很多心里语言学家将其解释成认知过程。

WordNet 中的基础语义关系是同义关系。同义词集合(synset)构成了 WordNet 的基本构建单位，其间的关系包含多种级别:

1) 同义关系(Synonym)，也就是同义词集。如 dog 的同义词集为 domestic dog，Canis familiaris 等；

2) 上下位(Hyponym/Hypernym)关系。如 dog 的上位关系表示成 something is a kind of dog，而其下位关系表示成 dog is a kind of something；

3) 整体部分(Meronym/Holonym)关系。如 dog 的整体关系表示成 parts of dog，而其部分关系表示成 dog is a part of something；

4) 反义(Antonym)关系。如 bad，badness 的反义关系为 good，goodness；

5) 多义名词的相似意义/近义。

WordNet 有效的描述了词汇间的多种关系，这些关系能够在一定程度上反映了人类自然语言中固有的多种关系，所以，在本文中，一方面，把它作为基础来构建用户兴趣模型。由于同一个词对应每种关系的相关词汇会有很多个，精简起见，只考虑词汇间的前 3 种关系。另一方面，在相似度的衡量上也采用了 WordNet 作为体现语义信息的依据。

2.2 形式概念分析

概念[37]是人类进行思维的最基本的单位，是用来组织成为诸如判断、结论等更为复杂思想的基础，是人类进行知识表述的一种有效手段，是一个哲学的范畴。对概念的这种理解源于古希腊哲学，发展于十七世纪的近代学院派，进一步发展成德国标准，最终成为了世界标准。在知识表示、知识管理、机器学习、专家系统等不同的领域，研究者们从不同的角度和观点来分析概念，形成了对概念的不同形式化描述方法。这是形式概念分析的核心理论。

在信息检索领域中，可以使用形式概念分析理论创建一种新颖的信息检索模型。由于形式概念分析中的概念格能够方便的进行分布式构建，这也为协作信息检索提供了一个理论上的新思路。

2.2.1 形式概念分析的基本概念

形式概念分析（Formal Concept Analysis，FCA）的数学基础是由Birkhoff[38]提出的，Birkhoff阐明了格结构和偏序间的相互对应关系。他指出一个格可以根据对象集和属性集间的每一个二元关系来构建，通过所得到的格可以洞察原始数据关系的结构。下面简单介绍下相关数学基础。

定义2.1 集合X上的关系R如果是自反的、非对称的、传递的，则称R在X上是偏序的或称R是集合X上的偏序关系，而称集合X为R的偏序集。一般，我们用符号“≤”表示偏序，而序偶“$\langle A,\leq\rangle$”表示偏序集。

定义2.2 设集合X上有一个偏序关系“≤”且设Y是X的一个子集，如果存在一个元素

$x \in X$，对每个$y \in Y$均有$y \le Y$，则称x是Y的上界，如果均有$x \le y$，则称x是Y的下界。

定义2.3 设集合X上有一个偏序关系“≤”且设Y是X的一个子集，如果$x \in X$是Y的上界，且对每一个Y的上界a均有$x \le a$，则称x是Y的最小上界(或称上确界supremum)，记作sup(Y)；如果$x \in X$是Y的下界，且对每一个Y的下界b均有$b \le x$，则称x是Y的最大下界(或成为下确界infimum)，记作inf(Y)。

定义2.4 设集合X上有一个偏序关系“≤”且设Y是X的一个子集，如果$x \in X$是Y的上界，且对每一个Y的上界a均有$x \le a$，则称x是Y的最小上界(或称上确界supremum)，记作sup(Y)；如果$x \in X$是Y的下界，且对每一个Y的下界b均有$b \le x$，则称x是Y的最大下界(或成为下确界infimum)，记作inf(Y)。

定义2.5 格是一个偏序集，其中任意两个元素所构成的子集都有上确界和下确界。记x，y的上确界为$x \vee y = \sup(\{x, y\})$，下确界为$x \wedge y = \inf(x, y)$。集合P上的偏序关系“≤”所构成的偏序集如它是格，可写成为$(P, \vee, \wedge)$。若P中元素有限，则称P为有限格。

代数格中以偏序关系为基础，集合元素之间存在上确界和下确界所构成的代数结果称为格，可以用Hasse图表示这种具有上、下确界的代数结构。

基于Birkhoff对格理论的贡献，德国的R.Wille教授在1982年作为一种数学理论首先引入了概念格(Concept Lattice)，奠定了FCA的理论基础，将哲学的概念进行数学化的描述，实现了概念的一种形式化描述方法，R.Wille首先提出根据二元关系系统来构造相应概念格(Galois格)的思想，也称为形式概念分析，就是以格中的每个节点表示一个形式概念，其中概念的外延代表相应的一组对象，内涵则为这组对象所具有的公共特征(属性)[39,40]。与概念格所对应的Hasse图则形象地揭示了概念间的泛化和例化的关系，反映出一种概念层次结构(Concept Hierachy)，实现了对数据的可视化[41,42]，非常适用于从数据库中进行知识发现的描述，从而成为数据分析和规则提取的一种有效工具。

FCA中的一种重要关系就是超概念和子概念关系，定义为一种自顶向下的推进，即从具有较大外延、较小内涵的更为广泛的概念到具有较小外延、较大内涵的相对例化的概念的次序。形式概念间通过这样的超概念和子概念关系相互关联构成了一种层次结构。称为概念格。换句话说，概念格是所给定的形式上下文的全部形式概念的有序集。概念格理论是FCA理论的核心数据结构，被认为是知识发现和数

据分析的有力数学工具。

在形式概念分析中，数据集是以形式背景(Context)的形式给出的，有关概念格的详细描述参见文献[43,44]，本节简要介绍一些基本概念。

已知形式背景C=(O,D,R)，其中O是对象集合，D是属性集合，R是O和D之间的一个二元关系，则存在唯一的偏序关系与之对应，并且每个偏序关系产生一个格结构，这种由C所诱导的格就称为概念格或Galois格，简记为Galois Concept Lattice(GCL)。

定义2.6 GCL中的每个结点(A,B)是一个二元组(称为概念)，其中$A \in P(O), B \in P(D)$，P(O)和P(D)分别表示对象和属性的幂集，A和B按如下运算关系建立连接：

$$A' = \{m \in D | gRm, \forall g \in A\} \tag{2.1}$$

$$B' = \{g \in O | gRm, \forall m \in B\} \tag{2.2}$$

并且A'=B，B'=A，A和B分别称为概念的外延(Extension)和内涵(Intension)，分别用Extent(C)和Intent(C)来表示。

设$C_1=(A_1,B_1)$和$C_2=(A_2,B_2)$是格中的两个概念，其中偏序关系“≤”定义为$C_1 \le C_2 \Leftrightarrow B_2 \le B_1$。此时称$C_1$是$C_2$的子概念(Sub_concept),$C_2$是$C_1$的超概念(Superconcept)。并且如果不存在概念C=(A，B)使得$A_1 \subset A \subset A_2$成立，定义$C_1 \le C_2$为直接子概念(Immediate_subconcept)或直接超概念(Immediate_superconcept)。

每一概念(A,B)描述了一组对象及其公共的特征。属性$b \in B$称为该概念所支持的属性，对象$a \in A$称为该概念所覆盖的对象。而且，每个概念$(A,B) \in P(O) \times P(D)$(A,B)对于关系R是完备的，即概念必须是最大扩展的，这是概念格的完备性。

概念格是一个完全概念格，因此对于概念格中的任意结点子集，都存在唯一的最大下界和最小上界。给定概念格L中的一个概念族$C = (A_1, B_1)(t \in L)$有最大下界和最大上界，我们分别用

$$\inf(C) = \left(\bigcap_{t \in T} A_1, \left(\bigcup_{t \in T} B_1 \right)'' \right) \tag{2.3}$$

和

$$\sup(C) = \left(\left(\bigcup_{t \in T} A_1 \right)'', \bigcap_{t \in T} B_1 \right) \tag{2.4}$$

来表示。

概念格中概念的外延集合A和内涵集合B之间存在对偶关系，给定$C_1=(A_1,B_1)$和$C_2=(A_2,B_2)$。则有 $C_1 \le C_2 \Leftrightarrow B_2 \subset B_1$；又 $B_2 \subset B_1 \Leftrightarrow A_1 \subset A_2$，从而 $C_1 \le C_2 \Leftrightarrow A_1 \subset A_2$。因此，一个概念格可以看作是相互联系的两个概念格。

一般来说，由一个规范形式背景及其概念格可以得到大量的关联规则。用图形方式表示概念格是传播知识和建立透明的高层次的有效方法。知识的各种连接和解释可以通过各种概念格的Hasse图来实现可视化。概念格的Hasse图是根据这种概念格偏序关系产生的。

如果 $C_1 \le C_2$ 并且概念格中不存在另一个元素C_3使得 $C_1 \le C_2 \le C_3$，则从C_1到C_2就存在一条边。下面表2.2和图2.3分别给出了一个形式背景及其对应的Hasse图。

表2.2 形式背景示例

	a	B	c	d
1	×	×		×
2	×		×	
3		×	×	
4	×	×		×
5	×			

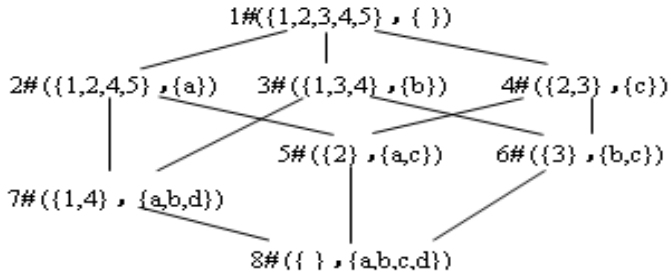

图2.3 形式背景2.2所对应的Hsse图

2.2.2 概念格及其构造算法

概念格的构造算法很多，但可分为两大类：批处理算法和渐进式算法。现有的批处理算法大多都是先产生形式背景所对应的所有概念，然后再决定概念之间的相互关系，从静态形式背景中采用批处理算法来构造概念格是有效的，但在应用中，

形式背景并不是永远不变的，时常会增加或者删除部分记录，如超市顾客交易数据库中总是随着交易的发生而不断的增加，当形式背景发生变化时，利用批处理算法就要将构造概念格的过程重新再处理一次，如果一次仅增加1条记录，按照批处理算法，也需要重新扫描整个数据库来进行重新计算，效率是低下，不适应于动态形式背景的处理要求。概念格的渐进式生成算法就是为了满足形式背景的渐增更新而发展起来的。

批处理算法不同，渐进式构造算法基本思想是将当前要插入的对象和格中所有的概念交，根据交的结果采取不同的行动。其中最典型的算法是Godin R.等在1995年提出的概念格的渐进式生成算法，通常称为Godin算法[45-47]。下面将简单介绍Godin算法的基本思想。

渐进式构造概念格就是在给定原始形式背景K=(O,D,R)所对应的原始概念格L以及新增对象x*的情况下，求解形式背景 $K^* = \left(O \cup \left(x^*\right), D \cap \left(\left(x^*\right)\right), R\right)$ 所对应的概念格L^*。

在渐进式生成概念格的求解过程中，要解决的问题主要有三个：

1. 所有新节点的生成
2. 避免已有格节点的重复生成
3. 边的更新

为了有效地解决这三个问题，对于原有概念格中的每个节点，根据它和新增对象的内涵描述之间的关系，可以定义它们的不同类型：

定理2.1 如果一个格节点C满足 $Intent(C) \subseteq f\left(x^*\right)$，则C被称为是一个更新格节。显然，如果C是一个更新格节点，则L^*中C应当被更新为 $\left(Extent(C) \cup \left(x^*\right), Intent(C)\right)$。

定理2.2 如果某个格节点C_1=（O_1,D_1）满足：

1. $Intersection = D_1 \cap f\left(x^*\right)$，而对L中任意的节点$C_2$都有Intent($C_2$) ≠ Intersection
2. 对于L中任意满足C_3>C_1的节点C_3，有Intent(C_3)∩f(x*)≠Intersection

则C_1被称为是一个产生子格节点。

一个更新格节点C显然不是一个产生子格节点，因为 $Intersection = D_1 \cap f\left(x^*\right) = D_1$。如果L中的一个概念节点既不是更新格节点也不是产生子格节点，则它被称为是一个不变节点。

对于L^*中的任意一个节点C，如果不存在L中的某个节点满足Intent(C_1)=Intent(C)，则C被称为一个新生节点；否则它被称为一个继承节点。如果

C_1是一个产生子格节点，节点$(Extent(C_1)\cup(X^*), Intent(C_1)\cap f(X^*))$，显然是中的一个新生节点，它被称为是由产生的新生的节点。

基于以上两条定理，那么Godin的完整算法过程如下：

1. 初始化格L为一个空格；
2. 从G中取一个对象g；
3. 对于概念格L中的每个概念$C_1=(A_1,B_1)$，如果$B_1\subseteq f(g)$，则把g并到A_1中
4. 如果同时满足：$B_1\cap f(g)\neq\phi$，$B_1\cap f(g)\neq B_1$和不存在(A_1,B_1)的某个父节点(A_2,B_2)满足$B_1\cap f(g)\subseteq B_2$，则要产生一个新节点；
5. 把新产生的节点加入到L中，同时调整节点之间的链接关系；
6. 重复(2)到(5),直至形式背景中的对象处理结束；
7. 输出概念格L。

概念格的渐进式生成算法在产生所有概念节点的同时，还产生了概念之间的父概念－子概念连接关系，同时它非常适合于处理动态数据库，被认为是一种生命力很强的概念格生成算法。

人们对Godin算法的改进也没有停止过。谢志鹏[48]等提出了一种利用字典索引树的快速概念格渐进式构造算法，该算法利用一个辅助索引树来快速判断概念节点的类型，并根据概念节点的类型来决定概念格的渐进修改策略。在新增对象时，该算法遍历整个索引树，利用索引树中节点的父子关系来判断原概念格中概念的种类，分别生成新概念或更新原概念。算法利用树状结构对格节点进行索引，对格节点的访问是通过遍历索引树来实现的，从而能有效地缩小新生格节点的父节点和子节点的搜索范围以及产生子的搜索范围，最终达到加速概念格渐进式更新过程的作用。

2.3 协作信息检索

文本信息检索技术[49]是一项成熟的处理文本数据的技术。这些文档所具有的信息量是非常巨大的，因此得通过有效的信息组织方式将大量文档集合转换成可表示形式。文本信息检索的基本任务就是对于任意一个用户查询，在给定的文档集合中找到一个与用户查询相关的文档子集。这里首先明确一下文本信息检索中的几个基本概念。

定义2.6 (文档（Document）)是检索系统的检索对象，一般是自然语言描述的

非结构化的自由文本或半结构化的文本。一般通常用D 来表示。在检索之前，文档一般要依据检索模型进行处理。如：对网页需要进行正文的抽取，去除广告等无关噪音。在垂直搜索中还需要提取文档中的每个域（标题、作者、时间等）信息。

定义2.7(文档集（Collection）)是一组文档的集合或数据集。一般用C 来表示。

定义2.8 (查询（Query）)是用户对查询需求的自然语言描述。一般用Q 来表示。用户对查询的描述基本上都是自然语言的词或句子。在检索系统进行查询之前，需要将用户输入的查询转化为查询表达式，如：简单的与或非的布尔表达式或Lemur 系统中的Indri 查询语言。

定义2.9(相关性（Relevance）)是用户查询与查询结果文档的匹配程度。一般用R 来表示。这是信息检索中最难定义的一个概念，这是因为实际上我们很难将用户的查询需求形式化，即使用户输入了一些相关查询词汇，但文档和查询是否相关以及相关程度还依赖与用户自身的经验知识；同时由于自然语言处理还没有完全达到文本理解的能力，也无法判断一个文档与查询的真实相关程度。

定义2.10 (用户需求（User Need）)，存在于用户的内心，文本描述是否能确切表达出用户的内心。

基于上面的基本概念，一个基本的文本检索过程如图2.4 所示。首先，检索系统会对文档集C 中的每一个文档D 进行分析，并将其文档信息建立索引数据对象。一般包括文本内容以及与文本相关的元信息，如：作者，标题等。然后，当一个用户查询需求到来，通过将用户输入的自然语言表示成检索系统所识别的查询表达式。第三，通过检索模型匹配查询表达式和索引对象，找到相关文档，并将其作为检索结果。最后，通过查询反馈过程从查询结果的前几篇文档中找到与查询词相关的词汇，并加入到前面的查询过程及索引对象以提高检索效果。从这个过程，可以清楚的看到在输入（用户查询需求Q，文档集C 中的文档D）和输出（检索结果）之间，最为关键的步骤就是检索模型。

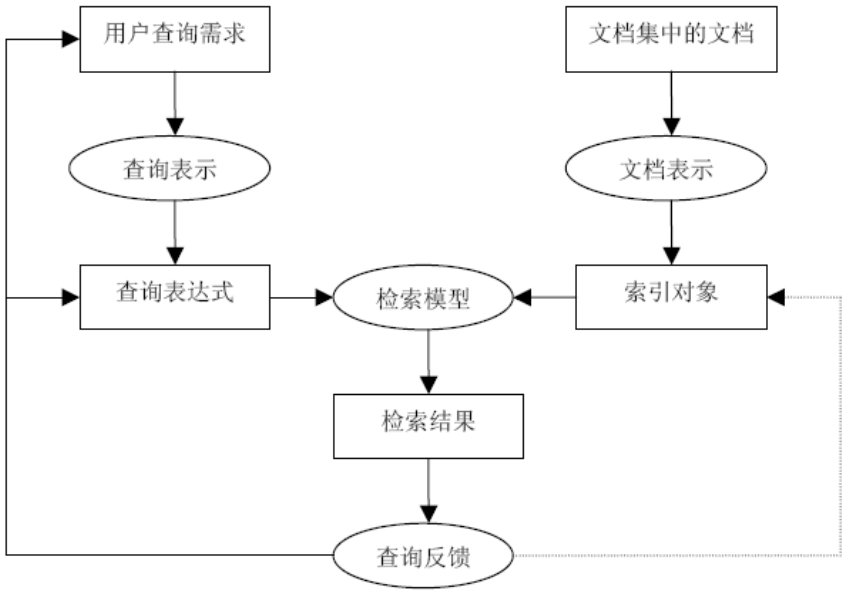

图 2.4 文本信息检索一般过程

文本信息检索一般包含如下几个核心部分：

1) 用户接口(User Interface)：人机接口，输入查询(Query)，返回排序后的结果文档(Ranked Docs)并对其进行可视化(Visualization)，支持用户进行相关反馈(Feedback)
2) 文本处理(Text Operations)：对查询和文本进行的预处理操作，如中文分词(Chinese Word Segmentation)、词干还原(Stemming)、停用词消除(Stopword removal)
3) 查询处理(Query operations)：查询的内部表示(Query Representation)，查询扩展(Query Expansion，利用同义词对查询进行扩展)，查询重构(Query Reconstruction，利用用户的相关反馈信息对查询进行修改)
4) 文本标引(Indexing)：对经过文本处理后的文本进行进一步处理，得到文本的内部表示(Text Representation)，通常基于标引项(Term)来表示。向量化、概率计算、倒排表存贮
5) 搜索(Searching)：从文本中查找包含查询中标引项的文本
6) 排序(Ranking)：对搜索出的文本按照某种方式来计算其相关度
7) 逻辑视图Logical View：查询或文本的表示，通常采用一些关键词或标引项

(index term)来表示一段查询或文本。

在协作信息检索中，主要是分别针对多个子文档集合进行检索，再通过某种策略将所有检索结果汇总表示。Web 信息与日俱增，其种类结构也是丰富多样，尤其文本信息更是巨大，如今信息检索领域已成为各位专家学者研究的热门领域。国内外针对信息检索领域中提出了很多方法及理论，其中，使用形式概念分析理论来进行信息检索也有很多深入的研究，如，使用 FCA 构建表示信息模型以更确切的表示原始检索模型，并使用构建出来的概念格进行信息检索，基于抽象具体程度的不同进行检索可获得独特层面上的检索结果。另外，在分布式环境下，使用 FCA 理论和技术进行协作信息检索，这可大大提高检索时间和效率。

为了能在大数据集中更快速、更有效的检索出更符合用户需求的结果，使用基于语义的一系列策略。在协作信息检索中，针对多个子文本数据库，利用渐进式构格算法构建各个子形式概念格，然后分别利用形式概念相似度的衡量方法对已有形式概念进行匹配，在找到临时概念集之后采用合并的方式，获得新的概念，再对新的概念进行相似度的匹配，最终获得满足用户需求的结果集合。整个系统充分体现了协作的思想，便于在分布式环境中部署实施，并且在匹配查询关键词时使用了不精确的方式，从结构和语义两个层次上进行衡量，更好的体现了人性化需求。

2.4 本章小结

本章介绍了概念格、本体和协作信息检索的理论和技术知识。在信息检索领域中，可以将FCA联合相关方法及理论，如文本特征提取和约简方法，语义本体理论，可以构建出一个完整的信息检索系统，可成为一个实际的搜索引擎。

3 基于类别分析及有效特征提取的文本分类方法

现代网络不断发展的同时，资源信息尤其是文本信息，也日益膨胀，人们需要投入更多的时间对信息进行组织和管理。对这些庞大的信息进行人工分类是相当耗费精力的，所以利用计算机进行信息的自动分类技术已成为数据挖掘领域中一个的重要研究方向，并且具有很高的商业价值。简单来说，文本分类[50]的任务是：在给定的分类体系下，根据文本的内容自动地确定文本关联的类别。现有的大多数文本分类系统都使用了向量空间模型对文本进行表示，即是把一个文本看成是特征词序列，并计算这些特征词的权重，将文本表示成这些权重的向量形式，然后再对这个向量空间进行处理。

由于一个文本包含的特征词往往非常多，所得到的向量空间的维数非常大，这么高维的向量空间使得文本分类的时间、空间消耗急剧上升。因此，一般在进行文本分类之前，需要对文本进行特征提取和特征降维，那么，如何做到在没有失去可用信息的情况下提取出更少的特征，尽可能得让特征维数更小呢？很多文献[50~57]提出了不同的方法来特征提取和特征降维。文献[50~52]对传统的 TF-IDF 计算方法进行改进，其中文献[52]提出一种基于词频差异的特征提取方法，以提高特征提取质量和文本分类的准确度。文献[53]对 CHI 公式进行改进来更好的表示特征对类别的贡献程度和相关程度。文献[54]主要集成多种特征提取方法对关系密切的特征进行合并以达到降维的目的。文献[55~57]在对类别信息进行分析之后，根据特征的类别区分能力进行特征选择，选择出对文本分类中具有较高类别区分能力的特征。从中可以看到，为了能够得到更好的特征向量空间，可以通过以下 2 个方法：1）根据各个特征自身的统计特性，过滤掉普遍的特征，去除噪声，以提取出更有意义的特征；2）可以对文本类别信息进行分析，计算出每个特征词的类别贡献程度，从而能够在文本自动分类中得到更准确的效果。然而，文献[50~57]中大多是从单一角度出发，即或是对传统公式进行改进，或仅仅利用特征类别贡献度，又或是从词频特性中进行特征提取而对后续的分类过程没有处理，而整个过程需要经过前续的特征提取和后续的分类两个步骤。如果特征提取过程中提取的特征非常符合后续分类的需要，那么总的效果就会很好，反之效果则相对较差。同样，如果仅对后者改进而前者没有，那么总的结果也不会很理想，所以应该将两者结合起来统一考虑。

基于这个思想，本章首先根据 Zipf 定律[58-61]进行全局特征提取，不提取整个文本特征空间中普遍存在和特别稀少的噪声特征；其次，对特征的类别信息进行统计分析后，进行类内局部特征提取；在选取有效的特征之后，利用类别频率对传统的 TF-IDF 公式进行改进，最终得到一个全面统一的分类方法。本章最后在数据集 NewsGroup 上进行测试，证明采用的特征提取方法是合理的，分类方法也能够得到较高的分类准确性。

3.1 相关知识

3.1.1 齐普夫定律

1932 年，哈佛大学的语言学专家 Zipf[58]在研究英文单词出现的频率时，发现如果把单词出现的频率按由大到小的顺序排列，则每个单词出现的频率与它的名次的常数次幂存在简单的反比关系：

$$P(r)=\frac{C}{r^{\alpha}} \tag{3.1}$$

这种分布就称为 Zipf 定律（Zipf's law）。其中，C 为一个正的常数，通常为 1，α 也为一个正的常数，称为 Zipf 指数，大小仅取决于具体的分布，与其他参数无关，在英语中约为 1。这一规律表明在语言中经常被使用的词汇只占词汇总量的很少一部分，而绝大部分词汇则很少被使用。它的频率-名次曲线[62]如图 3.1 所示。如果对词频及其序号做对数-对数（log-log）曲线[63]，将会出现一条斜率约为 $-\alpha$ 的直线。

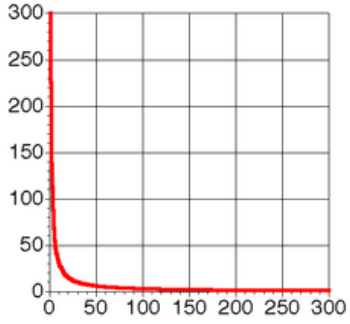

图 3.1 zipf 定律频率-名词曲线示意图

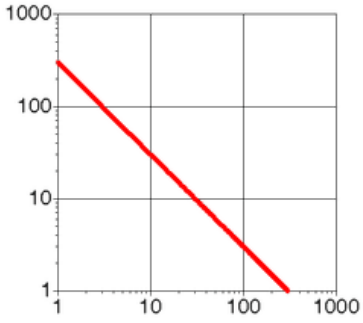

图 3.2 zipf 定律对数-对数曲线示意图

Zipf 定律表明在英语单词中，只有极少数的词被经常使用，而绝大多数词很少

被使用。实际上，许多人类语言都有符合这个定律。从 Zipf 定律分布曲线图中，可以看到，频率高的词的个数不多，大多数的词很少被使用。本章正是利用文本词集中的这种特性来进行特征提取的。

3.1.2 传统特征提取方法

目前有很多种特征提取算法[53]，如文档频率（DF），信息增益（Information Gain，IG），互信息（Mutual Information，MI），χ^2 统计法（CHI）。下面分别简单介绍下各个方法：

1) 文档频率（Document Frequency，DF）。 文档频率是指在训练文本集中某一特征词出现文本数。采用 DF 作为特征抽取基于如下基本假设：DF 值低于某个阈值的词条是低频词，它们不含或含有较少的类别信息。将这样的词条从原始特征空间中删除，不但能够降低特征空间的维数，,而且还有可能提高分类的精度。

DF 的优点在于计算量很小，而在实际运用中却有很好的效果。缺点是稀有词可能在某一类文本中并不稀有，也可能包含着重要的判断信息，简单舍弃可能影响分类器的精度。

2) 信息增益（Information Gain，IG）。信息增益是一种基于熵的评估方法，涉及较多的数学理论和复杂的熵理论公式，定义为某特征项为整个分类所能提供的信息量，不考虑任何特征的熵与考虑该特征后的熵的差值。它根据训练数据，计算出各个特征项的信息增益，删除信息增益很小的项，其余的按照信息增益从大到小排序。信息增益计算公式如下：

$$IG(t) = P(t)\sum_{i=1}^{M} P(C_i \mid t)\log\frac{P(C_i \mid t)}{P(C_i)} + P(\overline{t})\sum_{i=1}^{M} P(C_i \mid \overline{t})\log\frac{P(C_i \mid \overline{t})}{P(C_i)} \tag{3.2}$$

其中 t 表示特征项, P(t)表示训练集中包含特征项 t 的文本的概率，$P(C_i)$表示类别 C_i 在训练集中出现的概率，$P(C_i|t)$表示文本包含特征项 t 时属于 C_i 类的条件概率。$P(\overline{t})$ 表示训练集中不包含特征项 t 的文本的概率，$P(C_i \mid \overline{t})$ 表示文本不包含特征项 t 时属于 C_i 类的条件概率。显然，某个特征项的信息增益值越大，类别区分能力就越强。

从信息论角度出发，信息增益方法的本质是用各个特征值来划分训练样本空间，根据所获信息增益的多少来选择相应的特征。不足之处在于，它考虑了词未出现的情况。虽然某个词不出现可能对判断文本类别也有贡献，但实验证明这种贡献

往往小于考虑词不出现情况所带来的干扰，因为一篇文本仅能包含特征空间中的很少一部分特征，此时信息增益大的特征主要是信息增益公式中后一部分（代表单词不出现情况）大，而非前一部分（代表单词出现情况）大，信息增益的效果就会大大降低。

3) 互信息（Mutual Information ，MI）。互信息是一种广泛用于建立特征项关联统计模型的标准，它体现了特征项与类别的相关程度。对于特征项 t 和某一类别 C_i（i = 1，2，…，m），在 C_i 中出现的概率高，而在其他类别中出现的概率低的特征项 t 将获得较高的互信息，也就有可能被选取为类别 C_i 的特征。互信息的估计值定义为：

$$\text{Mutual Info Txt}(t,\ C) = \log \frac{A \times B}{(A+C)(A+B)} \tag{3.3}$$

其中，A 表示特征 t 与类 C_i 同时出现的次数，B 表示特征 t 不在类 C_i 中出现的次数，C 表示类 C_i 中没有出现特征 t 的文本数。

互信息的一个好处是，既能反映类与特征之间的关系（相对于 DF），计算又不至于太复杂（相对于 CHI 和 IG），缺点是没有考虑 C 特征发生的频率，这造成了互信息评估函数经常倾向于选择稀有单词，从而淘汰了很多高频的有用词条。

4) χ^2 统计法（CHI）。χ^2 统计法用于度量特征和类别之间独立性的缺乏程度，它同时考虑了特征存在与不存在的情况。χ^2 越大，独立性越小，相关性越大；反之独立性越大，相关性越小。

χ^2 统计量表示为：

$$\chi^2(t,\ C) = \frac{N(AD-BC)^2}{(A+C)(B+D)(A+B)(C+D)}$$

$$\chi^2_{AVG}(t) = \sum_{i=1}^{m} P(c_i)\chi^2(t,c_i)\text{(对所有类别求和)} \tag{3.4}$$

$$\chi^2_{Max}(t) = \max\nolimits_{i=1}^{m}\{\chi^2(t,c_i)\}\text{(取词条对所有类别的CHI最大值)}$$

其中，A 为 t 和 c 同时出现的次数，B 为 t 出现而 c 没有出现的次数，C 为 c 出现而 t 没有出现的次数，D 为 t 和 c 都没有出现的次数，N 为训练集中所有实例文本数。

χ^2 统计量可以用来度量类 c 和特征 t 的关联性，使得它在特征削减中十分有用。对每一对 c 和 t 都计算 χ^2 的值，并按照降序排列，去除排在后面的特征。χ^2 统计

量与互信息的差别在于它是归一化的统计量，但是它对低频特征项的区分效果也不好。

3.1.3 特征权重计算

特征权重的计算方法对应各种特征选择方法也有很多种不同的算法，常用的有布尔值法、词频法、TF-IDF 法。基于向量空间模型中的特征权重计算方法一般采用 TF-IDF 公式。

传统的 TF-IDF 公式，即如下所示：

$$\omega(t,d)=\frac{TF(t,d)\times\log(IDF(t)+\delta)}{\sqrt{\sum_{t\in d}(TF(t,d)\times\log(IDF(t)+\delta))^2}} \tag{3.5}$$

其中，TF（t，d）为特征 t 在文本 d 中的频率，IDF（t）为逆文档频率，文档频率 DF（t）的倒数，即 $IDF(t)=\frac{1}{DF(t)}$，δ 为调节系数，一般为 0.01。用 TF-IDF 算法来计算特征的权重值是表示当一个特征词在这篇文档中出现的频率越高，同时在其他文档中出现的次数越少，则表明该特征对于表示这篇文档那个的区分能力越强，所以其权重就应该越大。也可以从这个公式中看到特征 t 的权重只根据词频和文档频率，没有考虑到它的类别频率，而往往类别频率在文本分类中也起着一定作用，所以本章提出一个改进的特征权重计算方法。

3.2 特征提取策略

由于特征提取的效果好坏直接制约着后续的文本分类效果，而现有的特征提取仅从单一层面出发，没有从全局角度进行提取。所以本章提出了一种新颖的特征提取方法，分为两个步骤，分别是全局特征提取和局部特征提取，前者针对整个训练文本集（Training Corpus，简记为 TC），利用 Zipf 定律去除普遍存在的和特别稀少的噪声特征，实现在全局范围内的特征提取；而后者是在对每个特征词的类别信息分析之后，根据得到的类别频率进行类内的局部特征提取。在这个过程中，除了计算每个特征的词频 TF，文档频率 DF 外，还需计算特征的全局频率（Overall Frequency，OF）用于全局特征提取和特征的类别频率（Category Frequency，CF）用于局部特征提取。

3.2.1 全局特征提取

由于人类自然语言中普遍存在 Zipf 定律，即只有极少数的词被经常使用，而绝大多数词很少被使用。对于普通文本集中，同样呈现这种规律，那么在提取特征时，不提取频率很高的少数特征，同时对于那些频率极低的特征，很可能是噪声的特征，将其从原始特征空间中删除。

定义 3.1：在整个训练文本集 TC 中，特征词 t 在 TC 中出现的频率称为特征词 t 的全局频率 OF，定义为：

$$OF(t)=\frac{t\text{在TC中出现的次数}}{\text{TC中所有词数}} \tag{3.6}$$

在计算出所有特征的全局频率之后，按照全局频率从大到小进行排序得到一张频率-位置分布图，大多数情况下它是符合 zipf 定律的。因此我们可以通过设定两个阈值：全局低频特征百分比 z-low，全局高频特征百分比 z-high，它们分别用于删除在频率-位置分布图中所有特征个数×z-low 个较低频率特征和所有特征个数×z-high 个较高频率特征，即只保留整体频率在频率-位置分布图中位于两个阈值之内的特征。另外，经过计算每个特征的文档频率 DF 之后，如果某个特征的 DF 值低于一定阈值，也在这个步骤中将其从特征空间中删掉。这个过程可以由算法 3.1 来描述：

算法 3.1: Overall_Selected(TC，z-low，z-high，df-threshold)

输入: 训练词集 TC，全局低频特征百分比 z-low，全局高频特征百分比 z-high，DF 阈值 df-threshold

输出: 有效特征集 TSet

描述:

Step 1：根据 TC，获得所有特征集 TSet 并计算每个特征 t 的全局频率

Step 2：对 TSet 根据全局频率大小，从高到低排序

待删除的全局高频特征集

high_OF = TSet 中从位置 1 到位置$\lfloor |TSet| \times z-high \rfloor$的特征集

待删除的全局低频特征集

low_OF = TSet 中从位置$\lfloor |TSet| \times z-low \rfloor$到位置$|TSet|$的特征集

特征集 TSet 中删掉 high_OF 和 low_OF 中的特征

Step 3：# 在文本向量空间中删除全局高频/低频特征, 并计算每个特征的 DF

for TC 中的每个文本 file 依次进行：

for 文本 file 中的每个特征 term 依次进行：

if term in high_OF or term in low_OF：

将 term 从文本 file 中删除

计算 DF

for 特征集 TSet 每个特征 t：

if 这个特征 t 在该文本 file 中：

对应特征 t 的频数增 1

Step 4：# 去除 DF 低的 term

根据 TSet 中每个特征 t 的频数转换计算 DF

在文本空间中删除 DF 小于 df-threshold 的特征

返回最终的特征集 TSet

这个算法是从全局出发，在整个文本特征空间中过滤出现次数非常频繁的，或出现次数极少（很有可能是噪声）的特征，能够减少原始文本特征空间中的无效特征个数。

3.2.2 局部特征提取

每个特征在文本特征空间中具有特定的类别区分能力，这些类别信息对文本特征提取也是非常有意义的。因此，可以统计每个特征词在各类别中的出现频率，这里称为类别频率，其定义如下：

定义 3.2：在整个训练文本集 TC 中，其类别集合 $C=(c_1,c_2,...c_m),m$为类别数，每个文本已指明类别信息 $c_i \in C$，对于特征词 t 在文本对应的类别 $c_i \in C$ 中出现的频率称为特征词 t 的类别频率 CF，定义为：

$$CF(t,c_i)=\frac{t\text{在类}c_i\text{中的出现次数}}{\text{类}c_i\text{中的所有词数}} \tag{3.7}$$

对于特征 t，如果它在较少的类别中出现且在较少的类别中出现的频率较高，表示这个特征在这些类别中是特定含义的，它相对于其他特征来说具有较大的区分能力，所以像这样的特征应该尽可能保留下来；但如果它在大多数的类别中都出现并且它在各个类别中的类别频率都非常接近，那么说明特征 t 对类别区分能力不

大，所以可以把该特征从特征空间中删除。最终保留下来的是类别频率差异较大的特征，在此提出算法 3.2 进行特征空间的局部提取。

算法 3.2: Category_Selected(TC，cf-rate，cf-threshold)

输入: 训练词集 TC，类别出现率 cf-rate，词频差异阈值 cf-threshold

输出: 有效特征集

描述:

Step 1：# 计算特征 t 的类别频率 CF

for 对于 TC 的每个文本 file 依次进行:

获得该文本 file 所在的类别 c

for 对于文本 file 中的每个特征 t 依次进行:

CF(t,c)增 1

根据每个特征 t 在各类别中的频数转换计算得到特征的 CF

Step 2： # 即是类间频率阈值

CF_min = 总的类别数*cf-rate

for CF 中的每个特征 t 依次进行:

如果特征 t 在大多数类别中出现

并它的类别频率值相差不大，就把 t 删除

if |特征 t 的类别数|≥

CF_min and _IsInThreshold(特征 t 的类别频率，cf-threshold):

删除特征 t

Step 3： 返回最终的特征集

_IsInThreshold(cf，cf-threshold)

输入: 一系列类别频率值 cf，词频差异阈值 cf-threshold

输出: 数值波动是否在 cf-threshold 内，如果是返回 True，否则 False

描述:

Step 1: # 寻找出类别频率中的最大值和最小值

cf_min 为类别频率 cf 中的最小值，初始化为 cf 中的第一个数值

cf_max 为类别频率 cf 中的最大值，初始化为 cf 中的第一个数值

for 每个剩余 cf 中的 ocf 依次进行:

if otc < tc_min:

```
            tc_min = otc
            continue
        if otc > tc_max:
            tc_max = otc
```

Step 2： # 判断波动是否在给定词频差异阈值内

```
# 如果满足则返回 True，否则返回 False
If tc_max-tc_min < CF_THRESHOLD:
    Return True
Return False
```

文本特征空间在经过算法 3.2 的处理之后，得到的是具有类别区分能力较大的特征集，而且这个特征集中的特征个数要比原始的特征集中的要小的多。

这样，经过全局特征提取和局部特征提取之后，能够在尽可能保留原始文本特征信息的情况下，最大化的降低特征维数，从而提高文本自动分类效率。

3.3 基于类别分析和有效特征的文本分类算法

3.3.1 基于类别频率的特征权值计算公式

依据 2.3 的传统 TF-IDF 公式，编程很容易实现且计算方便，效率也较高，所以被广泛应用于很多实际分类和检索系统中，这是 TF-IDF 的最大优点。但是从中也可以看到，它的特征权重只根据词频和文档频率，没有考虑自身的类别频率，而往往类别频率在文本分类中也起着一定作用，所以本章提出一个改进的特征权重计算公式，记为 TF-IDF-CF：

$$\omega(t,d)=\frac{TF(t,d)\times\log(IDF(t)+\delta)\times\max\limits_{c\in C}CF(t,c)}{\sqrt{\sum\limits_{t\in d}(TF(t,d)\times\log(IDF(t)+\delta)\times\max\limits_{c\in C}CF(t,c))^2}} \tag{3.8}$$

其中，TF（t，d）为特征 t 在文本 d 中的频率，IDF（t）为逆文档频率，δ 为调节系数，一般为 0.01。CF（t，c）为特征 t 在类 c 中的类别频率， 其计算方法和算法 3.2 中相同，由于特征 t 在类别集合 C 中对应多个类别可能有多个类别频率，为了更好的反映出特征 t 的类别贡献度，如果特征 t 的某个类别频率越大，就表示其类别贡献度越大，反之越小，所以应该在所有的类别频率中取最大值来加重

特征 t 在文档 d 中的权重。从全局看，公式中加入类别频率 CF 之后，可以看到如果特征 t 的类别频率越高，相应的特征权重也就越高，表示特征 t 越有意义。反之，如果特征 t 的类别频率越低，所得到的特征权重就越小。

3.3.2 改进 kNN 分类算法

使用改进的权值计算公式TF-IDF-CF对算法3.2中提取的有效特征进行权重计算之后，提出一种类似于传统分类算法kNN[64]的算法3.3来进行文本分类。主要思想是，利用算法3.2，对整个训练文本集提取所有有效的特征之后，根据4.1中提出的特征权值计算公式计算特征权重，从而构成整个文本向量空间。接着，就对未知文档以同样的方式提取有效特征并用TF-IDF-CF计算特征权重得到一个文本向量，在整个文本向量空间中寻找出K个最相似的文本，统计这些文本所属的类别，并用出现最多的类别号作为该未知文本的类别。其中，利用文本向量之间的余弦相似度[65]来度量文本之间的相似性。整个过程可以由算法3.3来描述：

算法3.3： Classify(TC，Test Corpus，TSet， K)

输入：训练词集TC，测试集Test Corpus，有效特征集TSet，近邻个数K

输出：测试集中各文档的类别

描述：

```
Step 1:   # 对算法3.2中获得的有效特征集TSet，计算每个特征t的权重
          for 对于TSet中的每个特征t:
              for TC中的每个文本d：
                  根据公式TF-IDF-CF计算w(t, d)
Step 2:   # 对测试集进行分类
          for 对于每个测试集Test Corpus中的文档file：
              计算file中有效特征的权重
              遍历整个训练空间, 使用余弦相似度，
              计算该文档file和训练文档的相似度, 找到K个最相邻的文档
              从这K个最相邻的文档中, 选择类别最多的作为该文档的类别
```

在算法3.3中，根据TF-IDF-CF公式对特征空间进行权重计算，是体现了每个特征词对特定类别的贡献程度，从而能够在自动计算简便的同时，得到更准确的文本分类效果。

3.4 实验及其分析

3.4.1 数据集

本章实验采用 NewsGroup[66]这个国际上通用的数据集。它是互联网用户在 Usenet 上张贴的 19997 条消息组成的。这些消息均匀分布在 20 个不同的新闻组中，每个新闻组约有 1000 条消息，每个新闻组对应着一个文本类别。本章采用完整的 20 个类别作为数据集，其中将 12000 个文本作为训练集，剩余的 7997 个文本作为测试集。

3.4.2 性能评估标准

评价分类器性能好坏的两个常用性能评价指标为召回率 r（recall）和准确率 p（precision），其分别定义为：

$$r = \frac{a}{a+c},\ if\ a\ +\ c\ >\ 0;\ otherwise\ r\ =\ 1,$$

$$p = \frac{a}{a+b},\ if\ a\ +\ b\ >\ 0;\ otherwise\ p\ =\ 1,$$

其中 a 表示被正确分到该类的文本个数，b 表示被误分到该类的文本个数，c 表示属于该类但被误分到其他类别的文本个数。

另一个常用的评估指标是 F-指标，它定义为：

$$F_{\beta}(r,p) = \frac{(\beta^2+1)pr}{\beta^2 p + r} \tag{3.9}$$

其中参数β用来为召回率（r）和准确率（p）赋予不同的权重，当β取 1 时，准确率和召回率被赋予相同的权重。

本章中，使用准确率、召回率和 F-指标这三个指标来衡量分类器的分类性能。

3.4.3 实验结果及其分析

实验过程中使用到的参数设置如下：

全局低频特征百分比 z-low 为 1%；

全局高频特征百分比 z-high 为 1%；

DF 阈值 df-threshold 为 1%；

类别出现率 cf-rate 为 90%；

类别频率差异阈值 cf-threshold 为 0.0005；

权值公式中的调节系数δ 为 0.01；

K 近邻算法中 K 为 500；

F-指标中的β为 1。

针对 NewsGroup 数据集，测试得到有 88336 个特征词，在经过全局特征提取和局部特征提取之后，总共约有 70000 个特征词，平均每篇文档包含 56 个有效特征词。可以看出，每个文本包含的特征词是大大减小了。

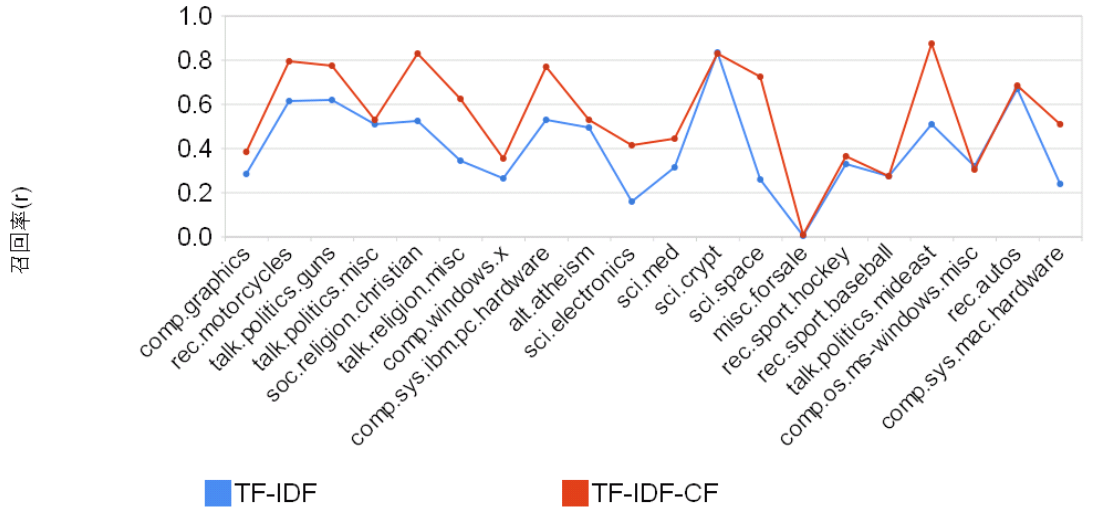

图 3.3 召回率比较

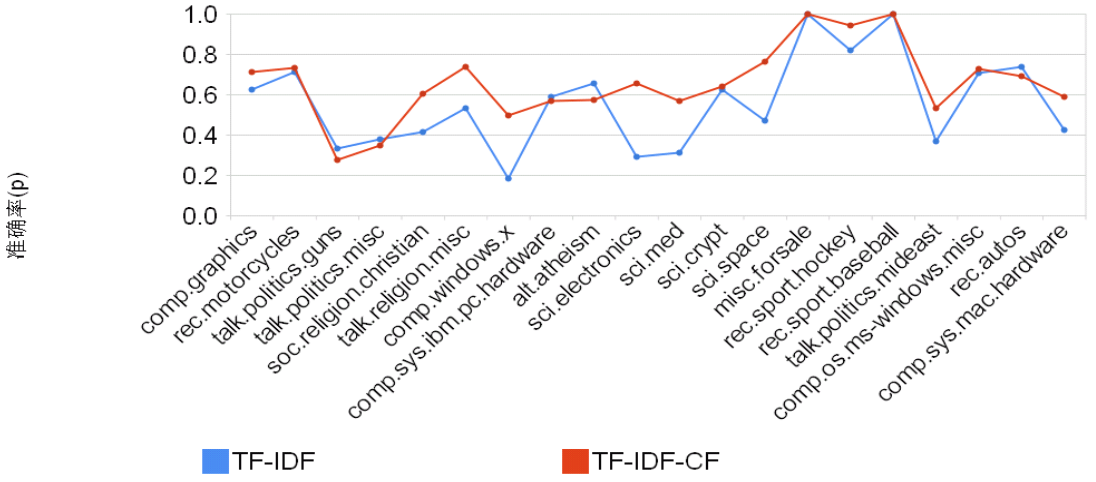

图 3.4 准确率比较

图 3.3 和图 3.4 分别显示在同一个特征空间中，使用本章提出的 TF-IDF-CF 和传统 TF-IDF 后，文本分类的召回率（r）和准确率（p）的不同结果。可以看到，在大多数文本类别中，都能取得较高的召回率和准确率，所以本章算法是优于 TF-IDF 方法的。

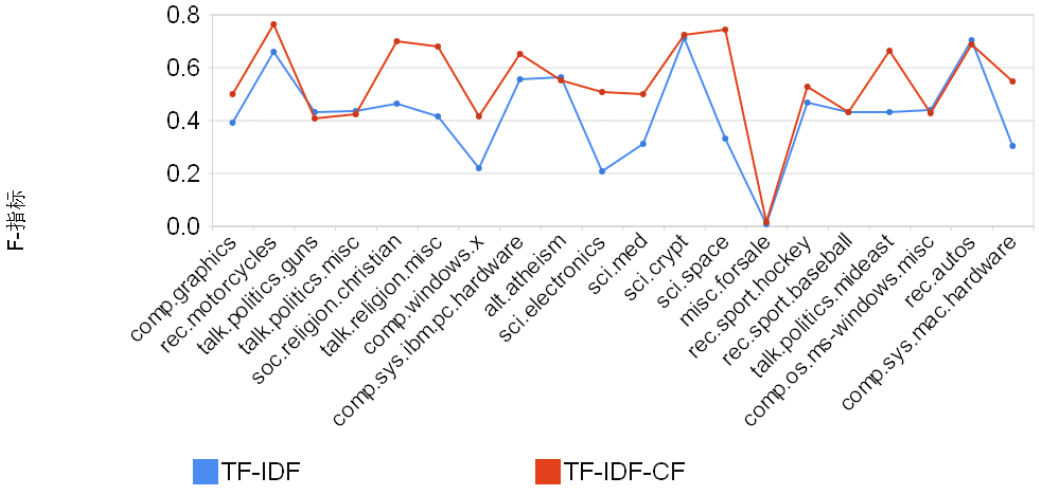

图 3.5 F-指标测试

图 3.5 显示了两种方法的综合评价，F-指标的测试结果。其中，β等于 1，即表示赋予召回率（r）和准确率（p）相同的权重。从图中也可以看到本章算法的综合评价也是高于传统方法的。

3.5 本章小结

本章根据特征空间中特征本身的统计规律分别进行全局特征提取和类内局部特征提取。在得到有效特征之后，利用类别频率对传统的 TF-IDF 公式进行改进提出一种新的特征权重计算方法 TF-IDF-CF，在此基础上进行文本分类。从实验测试结果中看到，本章所采用的特征提取方法是有效的，根据改进的权重公式获得的分类准确度也是较高的。

4 结构及语义层次上的概念相似度衡量方法

概念作为人的思想和知识的基本单元，一直以来，深受哲学界和科学界的重视，很自然地，也就成为了人工智能学科的重要研究对象，这主要体现在知识表示和机器学习等领域。将概念格理论应用到信息检索中，更能提取出概念之间的本质联系，对检索准确度有一个更深层次的提高。概念格是对概念以及概念之间关系的描述，是形式概念分析（Formal Concept Analysis，FCA）的核心数据结构。它在一定程度上是对客观世界的一种高度简化的描述形式。这种简化的最大优点是其具有良好的数学性质。Wille基于这种简化，开创了形式概念分析领域。但是，正是由于这种简化，使得概念格不能简单地被理解为客观世界的部分模型，而更多的是被看作一个人造的从一些数据集派生出来的表示。

基于概念格的数学理论基础，结合粗糙集理论和语义本体理论[67]，提出了一个形式概念相似度计算方法，改进了基本Tversky[68]相似度模型。从形式概念的结构相似和语义相似两个层次上衡量概念间的相似度，并将两者结合起来构成最终的形式概念相似度衡量方法，这种方法在不同层面上度量了形式概念的真实相似度。

4.1 相关知识

4.1.1 粗糙集

粗糙集理论[69]是一种刻画不确定性和不完整性知识的数学工具。粗糙集理论提供了一整套方法从数学上严格地处理数据分类问题。根据粗糙集理论的方法，知识推理就是给定知识表达系统的条件属性和结论（决策）属性，求出所有符合该知识的最小决策算法。粗糙集理论仅仅分析隐藏在数据中的事实，并没有带入人为的模糊性，是采用精确的数学方法分析不精确系统的一种理想方法。

定义 4.1 （信息系统）信息系统用一个列表的形式给出研究对象的信息，表的行对应于研究对象，表的列对应于对象的属性。信息系统可表示为一个四元组 $S=(U,A,V,f)$，其中 U 是对象的非空有限集合，称为论域；A 是属性的非空有

限集合，$A=C\bigcup D, C\bigcup D=\varnothing$，Va 是属性 a 的值域；$f:U\times A\to V$ 是一个信息函数，即 $f(x,a)\in Va,\forall a\in A,x\in U$。

信息系统也称为知识表达系统，为了简化符号，通常也用 $S=(U,A)$ 来代替 $S=(U,A,V,f)$)。

表 4.1 给出了一个关于某些病人的信息系统，其中 U = { X_1 , X_2 , X_3 , X_4 , X_5 , X_6}，A ={头痛，肌肉痛，体温}

表 4.1 一个关于某些病人的信息系统

U	头痛	肌肉痛	体温
X_1	是	是	正常
X_2	是	是	高
X_3	是	是	很高
X_4	否	是	正常
X_5	否	否	高

定义 4.2（决策表）决策表是一类特殊而重要的信息系统，多数决策题 都 可以用决策表形式来表达。设 $S=(U,A,V,f)$ 为一个信息系统，$P\subseteq A$，C 称为条件属性，D 称为决策属性。具有条件属性和决策属性的信息系统称为决策表。

一个决策表中决策属性有时是唯一的，称为单一决策；有时是不唯一的，称为多决策。对于具有多个决策属性的决策表我们可以变换成为单一决策表，这样有利于问题的简化和求解。

定义 4.3 （不可区分关系）令 $P\subseteq A$，定义属性集 P 的不可区分关系 $ind(P)$ 为 $ind(P)=\{(x,y)\in U\times U\mid\forall a\in P, f(x,a)=f(y,a)\}$ 如 $(x,y)\in ind(P)$，则称 x 和 y 是 P 不可区分的。$\forall P\in A$，不可区分关系 $ind(P)$ 是 U 上的等价关系，符号 U / ind(P)（简记为 U / P）表示不可区分关系 $ind(P)$ 在 U 上导出的划分，$ind(P)$ 中的等价类称为 P 基本集。符号 $[x]_P$ 表示包含 $x\in U$ 的 P 等价类。

例如，在表 4.1 中由属性集 P ={头痛，肌肉痛}划分的所有等价类为：

$U/P=\{\{x1,x2,x3\},\{x4,x6\},\{x5\}\}$

一个近似空间可以看成是一个信息系统 $S=(U,A,V,f)$，设 $X\subseteq U$，如果集合 X 能用一个不可区分关系 R 下的等价类的并集表示，那么称集合 X 是可定义的，否则称集合 X 是不可定义的，需要通过逼近来刻画集合 X。

定义 4.4 (下近似) 给定信息系统 $S=(U,A,V,f)$,对于每个子集 $X\subseteq U$ 和一个等价关系 $R\subseteq A$，X 相对于 R 的下近似定义为

$$\underline{R}(X)=\{Y\in U/R \mid Y\subseteq [x]_R\} \tag{4.1}$$

$\underline{R}(x)$ 实际就是那些根据已有知识判断肯定属于 X 的对像所组成的最大集合，也称为在关系 R 下 X 的正区域(Positive Region)，记作 $pos_R(X)$。事实上 $\underline{R}(x)$ 为 X 中的最大可定义集。

定义 4.5(上近似) 给定信息系统 $S=(U,A,V,f)$，对于每个子集 $X\subseteq U$ 和一个等价关系 $R\subseteq A$，X 相对于 R 的上近似定义为：

$$\overline{R}(X)=\{Y\in U/R \mid Y\bigcap [x]_R!=\Phi\} \tag{4.2}$$

$\overline{R}(x)$ 实际上就是那些根据已有知识判断可能属于 X 的对象所组成的最小集合。事实上 $\overline{R}(x)$ 为含有 X 的最小可定义集。由上近似的定义知 neg_R=U- $\overline{R}(x)$ 就是那些根据已有知识判断肯定不属于 X 的集合，称为在关系 R 下 X 的负区域(Negative Region)

定义 4.6 (边界域) 设 $X\subseteq U$,定义集合 X 相对于 R 的边界域为：

$$bn_R(X)=\overline{R}(x)-\underline{R}(x) \tag{4.3}$$

边界域就是在关系 R 下，即可能属于 X 也可能不属于 X 的对象的集合，也是集合 X 的不确定体现。如果 $bn_R(X)\neq\Phi$，说明不存在不确定性，在关系 R 下，X 可以

被精确定义：$bn_R(X) \neq \Phi$，说明存在不确定性，在关系 R 下，X 不可以被精确定义，这时 X 为关于 R 的粗糙集或者非精确集。

上近似和下近似可以形象地用下图 4.1 表示：

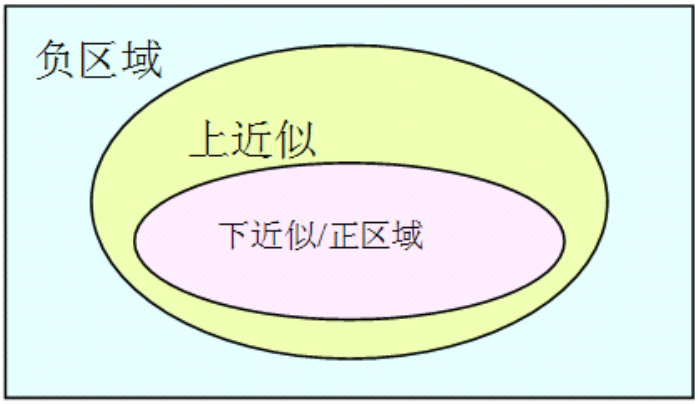

图 4.1 粗糙集描述图

在粗糙集理论中，集合的不精确性是由于边界区域的存在而引起，集合的边界区域越大，其精确性则越低。为了更精确的表达这一点，引入精度的概念。

定义 4.7 (近似精度) 由等价关系 R 定义的集合 X 的近似精度为，

$$a_R(X) = \frac{|\underline{R}(x)|}{|\overline{R}(x)|} \tag{4.4}$$

其中 $X \neq \Phi$，|X|表示集合 X 的基数

精度用来反映人民根据现有知识对集合 X 的了解程度。显然，对于每一个 R 和 $X \subseteq U$ 有 $0 \leq a_R(X) \leq 1$。当 $a_R(X) = 1$ 时，集合 X 的 R 边界域为空集，集合 X 为 R 可定义的；当 $a_R(X) < 1$ 时。集合 X 有非空 R 边界域，集合 X 为 R 不可定义的。

集合 X 的 R 粗糙度 $\rho_R(X)$ 定义为：

$$\rho_R(X) = 1 - a_R(X) \tag{4.5}$$

集合 X 的 R 粗糙度与精度恰好相反，它表示的是集合 X 的知识的不完全程度

粗糙集理论和形式概念分析是进行数据挖掘的两种常用方法,它们从不同的观

点进行概念的建模和研究。粗糙集理论的主要特点是利用对象的不可辨识关系推导近似算子,形式概念分析则是构造形式概念和概念格。粗糙集理论的发展是以对象的等价关系为基础的,通过对象集和属性集构成的二元关系来提取规则,而二元关系恰恰是形式概念分析中的基本概念,这就提供了研究粗糙集和概念格的公共基础。

数据集在粗糙集中用信息系统(U,A,F)表示，在形式概念分析中用形式背景(U,A,F)来表示，对于一个形式背景(U,A,F)，可以表示为只有 0 和 1 的表格。因此，有以下关系：$\forall x \in \bigcup, \forall a \in A.$ 若$(x,a) \in I$，则 x 在 a 上的取值为 1，否则 x 在 a 上的取值为 0。即：$(x,a) \in I \Leftrightarrow f_a(x)=1; (x,a) \notin I \Leftrightarrow f_a(x)=0$。因而，此形式背景就可以看作是一种特殊的信息系统，其对象在属性上的取值只有 1 和 0。另一方面，对于一个信息系统(U,A,F)来说，它实际上是 FCA 中的多值背景，可以转换为一般的形式背景。

4.1.2 Tversky 相似度模型

Tversky 在基于集合理论的基础上，提出了一种相似度衡量方法：

$$S(a,b)=\frac{f(B_1 \cap B_2)}{f(B_1 \cap B_2)+\alpha f(B_1-B_2)+\beta f(B_2-B_1)} \tag{4.6}$$

其中，$\alpha,\beta \geq 0$，f是相似度的一个衡量对象B_1和B_2的函数，B_1和B_2分别对应属性集a,b。这种方式是将一个对象用它具有的属性来表示的，那么，相似度的计算是根据这些属性的匹配来进行的。

在公式(7)基础上，Rodriguez 和 Egenhofer[70]提出了一种衡量不同本体间实体类的语义相似度，其公式如下：

$$S(a,b)=\frac{|B_1 \cap B_2|}{|B_1 \cap B_2|+\alpha(a,b)|B_1-B_2|+(1-\alpha(a,b))|B_2-B_1|} \tag{4.7}$$

使用了集合的势来衡量相似度，其中，$depth(a), depth(b)$分别代表了a,b在本体层次中的深度，那么，如果$depth(a) \leq depth(b)$，$\alpha(a,b)=\frac{depth(a)}{depth(a)+depth(b)}$，否则，$\alpha(a,b)=1-\frac{depth(a)}{depth(a)+depth(b)}$。

Zhao[71]对 Tversky 相似模型进行了进一步的改进，并结合了形式概念分析和粗糙集理论来计算对象间的相似度：

$$S(a,b)=\frac{|(B_1\vee B_2)_{LA}|}{|(B_1\vee B_2)_{LA}|+\alpha|B_{1LA}-B_{2LA}|+(1-\alpha)|B_{2LA}-B_{1LA}|} \tag{4.8}$$

其中，$B_{LA}=\text{intent}(\wedge\{(x,y)\in\zeta\mid y\subseteq B\}),\zeta$ 是形式概念格中所有概念的集合。

Wang[72]提出了一个更为复杂的形式概念相似度计算公式：

$$\begin{aligned}S^{\wedge}_{LA}((A_1,B_1),(A_2,B_2))=&\omega\frac{|(A_1\bigcap A_2)^{\wedge}_{LA}|}{|(A_1\bigcap A_2)^{\wedge}_{LA}|+\frac{1}{2}|A^{\wedge}_{1LA}-A^{\wedge}_{2LA}|+\frac{1}{2}|A^{\wedge}_{2LA}-A^{\wedge}_{1LA}|}\\&+(1-\omega)\frac{|(B_1\bigcap B_2)^{\wedge}_{LA}|}{|(B_1\bigcap B_2)^{\wedge}_{LA}|+\frac{1}{2}|B^{\wedge}_{1LA}-B^{\wedge}_{2LA}|+\frac{1}{2}|B^{\wedge}_{2LA}-B^{\wedge}_{1LA}|}\end{aligned} \tag{4.9}$$

该公式即考虑了形式概念的内涵，又考虑了形式概念的外延，使得计算复杂。

本章主要结合形式概念格和粗糙集理论、进一步改进了 Tversky 相似度公式，以便更准确的计算形式概念间结构相似度。

4.1.3 信息熵

信息熵最主要的思想是：如果有一个系统 S 内存在多个事件 S=(E1，E2，…，En)，每个事件的概率分布 P=(P1，P2， …，Pn)，则每个事件本身的信息量为 $I(E)=-\log Pi$(即当事件 Ei 出现后，给予我们的信息量)，整个系统的平均信息量为：

$$H(S)=\sum_{i=1}^{n}PiI(E)=-\sum_{i=1}^{n}Pi\log Pi \tag{4.10}$$

这个平均信息量就是信息熵，简称熵。通常规定如果 Pi=0，则 I(E)=0，这与数学上极限定理的定义是一致的。

本章中主要是用来计算每个属性自身具有的信息量，以此来计算概念间的语义相似度量。

4.2 结构相似度量

形式概念格是一个完备格，其格结构也具有一些可用信息，比如形式概念的抽

象具体程度，处在越上层的父亲节点越是抽象，其代表的信息相对越少；而处在越下层的孩子节点则越是具体，其代表的信息相对较多。当在进行相似度衡量时，假如一个形式概念 C_1 和其他形式概念进行相似度比较，若计算出该形式概念 C_1 和某两个形式概念 C_2，C_3 相似度相同时，如果形式概念 C_1 和 C_2 所处的层次比形式概念 C_1 和 C_3 的层次更靠近，那么，很显然的，层次相对靠近的形式概念 C_2 应该比形式概念 C_3 优先匹配，也就是说形式概念 C_1 和 C_2 真正内在的相似度应该大于形式概念 C_1 和 C_3 的内在相似度，因为它们之间有更类似的抽象具体程度。因此，在衡量形式概念的相似度时，利用形式概念的结构层次上的信息进行相似度的计算是非常必要的。

其次，在考虑形式概念层次结构的基础上，结合粗糙集理论，使用内涵集合的下近似而不是内涵集合本身来计算，这有助于包含更具体的信息。基于这两点，本章改进了 Tversky 相似度计算模型，加入层次结构和近似信息来更有效的计算形式概念间的相似度。

定义 4.8 形式概念间的结构相似度：形式概念 C_1 (A_1， B_1)和 C_2 (A_2， B_2)，是形式概念格中的任意两个形式概念，其形式概念间的结构相似度定义为

$$StructSim(C_1,C_2)=\frac{|(B_1\vee B_2)_{LA}|}{|(B_1\vee B_2)_{LA}|+\alpha(C_1,C_2)|B_{1LA}-B_{2LA}|+(1-\alpha(C_1,C_2))|B_{2LA}-B_{1LA}|} \quad (4.11)$$

其中，$(B_1\vee B_2)_{LA}=\text{intent}(\wedge\{(x,y)\in\zeta\mid y\subseteq B\}),\zeta$ 为所有形式概念，||表示集合的势，$|B_{1LA}-B_{2LA}|$表示属于 B_1 的下近似集但不属于 B_2 下近似集的内涵个数，类似的，$|B_{2LA}-B_{1LA}|$表示属于 B_2 的下近似集但不属于 B_1 下近似集的内涵个数；

$$\alpha(C_1,C_2)=\begin{cases}\dfrac{depth(C_1)}{depth(C_1)+depth(C_2)},\text{如果}depth(C_1)\le depth(C_2)\\ 1-\dfrac{depth(C_1)}{depth(C_1)+depth(C_2)},\text{如果}depth(C_2)\le depth(C_1)\end{cases}$$

，$depth(C_1)$ 表示形式概念 C_1 在形式概念格中的层次号，越是父层所具有的层次号就越小，形式概念格的最父节点层次号记为 0，那么α 系数则代表了形式概念之间的层次关系，能更准确的计算形式概念间的相似度。

本章使用文献[73]中的形式背景来举例说明形式概念之间是如何计算结构相似度的。形式背景如图 4.2 所示，根据形式概念格生成算法 Godin 生成的形式概念格如图 4.3 所示：

	cartoon	real	tortoise	dog	cat	mammal
Garfield	x				x	x
Snoopy	x			x		x
Socks		x			x	x
Greyfriar's Bobby		x		x		x
Harriet		x	x			

图4. 2 形式背景

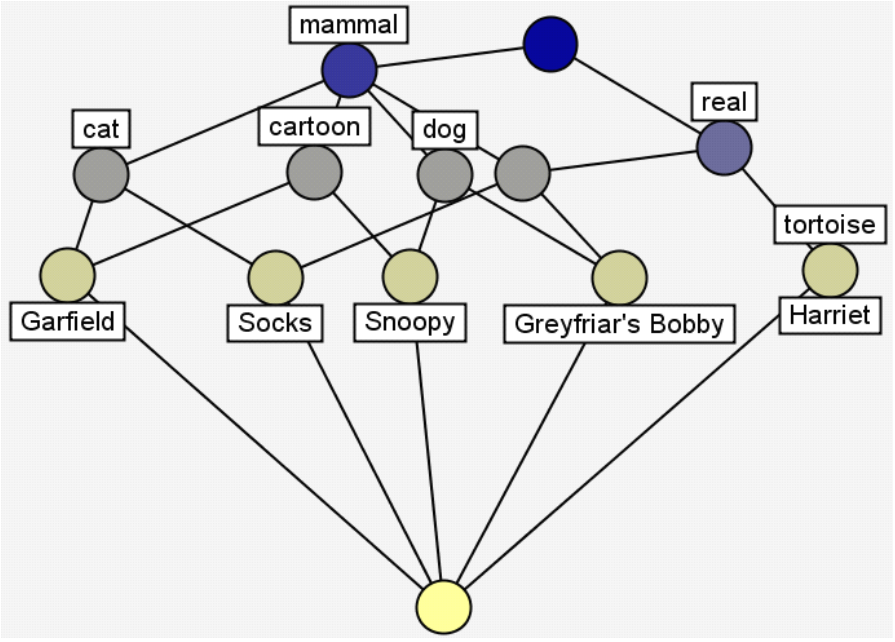

图4. 3 对应的形式概念格

考虑形式概念格中的形式概念 C_1((Garfield，Socks)，(mammal，cat))，C_2((Grayfriar's Bobby，snoopy)，(dog，mammal))， C_3((Grayfriar's Bobby)，(real，dog，mammal))，根据公式(12)可以得到 C_1 和 C_2，C_1 和 C_3 之间的结构相似度，分别为：

$$StructSim(C_1, C_2) = \frac{1}{1+\frac{1}{2}*1+(1-\frac{1}{2})*1} = \frac{1}{2}$$

$$StructSim(C_1, C_3) = \frac{1}{1+\frac{2}{5}*1+(1-\frac{2}{5})*2} = \frac{5}{13}$$

可以看到，形式概念 C_1 和形式概念 C_2 之间的相似度是比 C_1 和 C_3 的相似度大的，这和实际情况是相符的。

4.3 语义相似度量

为了能有效地从语义上衡量属性之间的相似度，采用 WordNet 作为基础词汇信息库，提取词汇间的 IS-A 关系。然后在具体词集中，统计各个词汇出现的概率，得到一个加权的 IS-A 层次关系。接着通过计算各个词汇的信息量得到语义相似度，从而得到词汇语义相似表。

WordNet 是一个覆盖范围宽广的英语词汇语义网，本章中主要使用 WordNet 中词汇之间的 IS-A 关系和同义词集 SynSet。对于具体领域的文集，统计各个词汇出现的概率，并加入 IS-A 层次关系中，得到一个加权的 IS-A 层次关系如图 4.3 所示。

定义 4.9 加权 IS-A 层次关系：给定一个英文词汇库 ε，利用每个词汇概率及其间的 IS-A 关系构成加权 IS-A 层次关系 $H(\varepsilon)$。其中词汇概率定义为：

$$p(n) = \frac{freq(n)}{M} + \sum_{i \in sub(n)} p(i) \tag{4.12}$$

其中，$freq(n)$ 为词汇 n 在文集中出现的次数；M 为文集中所有词汇的数目；$\sum_{i \in sub(n)} p(i)$ 为词汇 n 的所有直接下层词汇的概率之和，若词汇 n 是底层节点，即没有下层节点，则该值为 0。同时，定义一个层次关系的最顶层的节点，记为 Top，其 $p(Top) = 1$。

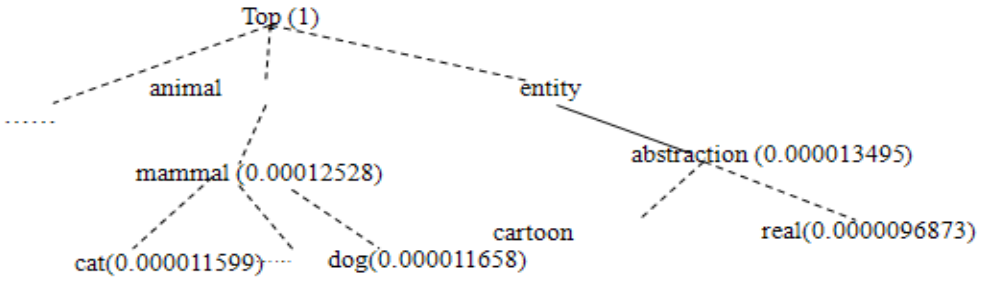

图 4.4 加权 IS-A 层次关系

从上述加权 IS-A 层次关系图中，根据信息量的计算公式 $I(E)=-\log Pi$，得到各个词汇的信息量，比如：

$I(cat)=-\log 0.000011599=4.935579$，

$I(dog)=-\log 0.000011658=4.933376$，

$I(mammal)=-\log 0.00012528=3.902118$，

可以看到在加权 IS-A 层次关系图中越上层的词汇所包含的信息量越少。并且对于两个词汇，若它们的共同父节点词汇的信息量越大，就表示它们共享的信息就越多，从而表示它们相似度越大。本章采用文献[74]中提出了信息量相似度作为词汇相似度量，以此进一步衡量概念间的语义相似度。

定义 4.9 信息量相似度 ics(n1，n2)：给定一个英文词汇库 ε 及加权 IS-A 层次关系 $H(\varepsilon)$，对于文集中的任意两个词汇 n1，n2，若 n1= n2 或者 n1 和 n2 是同义词，则 $ics(n_1,n_2)=1$；否则 $ics(n_1,n_2)=\dfrac{2I(n')}{I(n_1)+I(n_2)}$；其中，n’为 n_1、n_2 的最大公共父节点词汇，即 $I(n')=\max\limits_{n\in S(n_1,n_2)}\{I(n)\}$，$S(n_1,n_2)$ 是所有 n_1、n_2 的公共父节点词汇。

那么，根据这个定义，可以得到任意两个词汇间的相似度。比如：

$$ics(cat,dog)=\frac{2I(mammal)}{I(cat)+I(dog)}=\frac{2*3.902118}{4.935579+4.933376}=0.790786$$

由于形式概念的本质是由它所拥有的属性来表现的，所以只需要对形式概念所拥有的属性的不同进行衡量，即在这里只考虑形式概念的内涵。

定义 4.10 形式概念间的语义相似度：两个形式概念 $C_1(A_1，B_1)$和 $C_2(A_2，B_2)$ 的语义相似度：

$$SemanticSim(C_1,C_2)=\frac{M(B_1,B_2)}{\max\{|B_1|,|B_2|\}} \tag{4.13}$$

其中，$M(B_1,B_2)=\max\{\sum\limits_{b_1\in B_1,b_2\in B_2} ics(b_1,b_2)\}$，$b_1$ 和 b_2 只能在每一种组合中出现一次。

比如考虑两个形式概念 C_1((Garfield，Socks)，(mammal，cat))， C_2((Grayfriar’s Bobby，snoopy)，(dog，mammal))，它们的语义相似度为：

$$SemanticSim(C_1,C_2)=\frac{M((mammal,cat),(dog,mammal))}{\max\{|(mammal,cat)|,|(dog,mammal)|\}}=0.895393$$

而形式概念 C_1((Garfield，Socks)，(mammal，cat))，C_3((Grayfriar's Bobby)，(real，dog，mammal))的语义相似度为：

$$SemanticSim(C_1, C_3) = \frac{M((mammal, cat), (real, dog, mammal))}{\max\{|(mammal, cat)|, |(real, dog, mammal)|\}} = 0.596929$$

4.4 形式概念的相似度计算公式

在得到两个形式概念的结构相似度和语义相似度之后，为了在结构层次和语义层次上能准确的衡量形式概念的相似度，本章综合考虑这两个层次，得到一个新的形式概念相似度计算公式。

定义 4.11 Sim(C_1，C_2)：两个形式概念 $C_1(A_1，B_1)$和 $C_2(A_2，B_2)$的最终相似度：

$$Sim(C_1, C_2) = \omega StructSim(C_1, C_2) + (1-\omega) SemanticSim(C_1, C_2) \tag{4.14}$$

其中，ω 为权重系数，取值范围为 0～1，用于调节结构相似度和语义相似度的重要程度。

比如，仍然计算形式概念 C_1((Garfield，Socks)，(mammal，cat)) 和 C_2((Grayfriar's Bobby，snoopy)，(dog，mammal))的相似度为：

$$Sim(C_1, C_2) = 0.5 * \frac{1}{2} + 0.5 * 0.895393 = 0.697697$$

而形式概念 C_1((Garfield，Socks)，(mammal，cat))和 C_3((Grayfriar's Bobby)，(real，dog，mammal))的相似度为：

$$Sim(C_1, C_3) = 0.5 * \frac{5}{13} + 0.5 * 0.596929 = 0.490772$$

可以看出，形式概念 C_1 和 C_2 的最终相似度是高于形式概念 C_1 和 C_3 的，这是符合实际情况的。

4.5 本章小结

本章提出了一个形式概念相似度计算方法，改进了基本 Tversky 相似度模型，从形式概念的结构相似和语义相似两个层次上衡量概念间的相似度，使得能够在不

同层面上，更大程度上的反映出形式概念的真实相似性。若将此形式概念相似度衡量方法应用到本体工程，如本体映射，本体合并等，用于衡量本体概念的相似程度；或信息检索领域，用于匹配关键词，这种相似度衡量方法都可以作为一种基础来使用。

5 基于 FCA 的协作信息检索框架

Web 信息与日俱增，其种类结构也是丰富多样，尤其文本信息更是巨大，如今信息检索领域已成为各位专家学者研究的热门领域。国内外针对信息检索领域中提出了很多方法及理论[75-76]，其中，使用形式概念分析理论来进行信息检索也有很多深入的研究，如，使用 FCA 构建表示信息模型以更确切的表示原始检索模型，并使用构建出来的概念格进行信息检索，基于抽象具体程度的不同进行检索可获得独特层面上的检索结果。另外，在分布式环境下，使用 FCA 理论和技术进行协作信息检索，这可大大提高检索时间和效率。

为了能在大数据集中更快速、更有效的检索出更符合用户需求的结果，使用基于语义的一系列策略。在协作信息检索中，针对多个子文本数据库，利用渐进式构格算法构建各个子形式概念格，然后分别利用形式概念相似度的衡量方法对已有形式概念进行匹配，在找到临时概念集之后采用合并的方式，获得新的概念，再对新的概念进行相似度的匹配，最终获得满足用户需求的结果集合。整个系统充分体现了协作的思想，便于在分布式环境中部署实施，并且在匹配查询关键词时使用了不精确的方式，从结构和语义两个层次上进行衡量，更好的体现了人性化需求。

将形式概念分析的理论知识应用到信息检索中，可以进一步提高检索的时间和空间效率。而对于多个数据库中进行协作式的检索，尤其是可以利用形式概念格中富含的信息进行。基于文献[77]中提到的协作概念信息检索系统，它基于形式概念分析，提出了三个不同的检索系统，主要是对各个子形式背景进行约简，在约简对象的同时，获得等价对象集合，之后在对各个子形式背景检索得到一个临时结果集，再在此基础上，利用合并算法对这些临时结果进行合并，最终根据等价对象集获得最终检索结果返回给用户。但其中，有多点不足之处，比如，在子形式背景匹配时使用了精确的查找满足条件的对象集，但在实际情况下更需要的是一种模糊的，不精确的对象匹配过程。同样的，在对临时结果集进行合并时，合并组合各个临时结果，其匹配的过程也是精确的。另外一点不足之处就是，在使用形式概念格进行对象约简时，构造的形式概念格是一个完全格，这需要消耗非常大的时间空间资源，很多情况下构造完全格是没必要的。

5.1 基于 FCA 的协作信息检索

5.1.1 协作信息检索结构框图

本章提出了一种改进之后的协作概念信息检索系统，针对多个子形式背景，分别构成子形式概念格，并同时生成等价对象集。对用户查询特定的关键词，在各个子形式概念格上，利用本章之前提出的概念相似度计算公式，获得满足一定概念相似度阈值的临时形式概念集。然后将各个临时形式概念集根据合并算法进行合并，并同样适用前文提出的概念相似度对合并之后的形式概念进行衡量，符合概念相似度阈值的形式概念加入到最终结果集合。最终，在最终结果集合中每个形式概念的对象，包括与这些对象具有等价关系的对象即为最终检索结果返回给用户。整个过程可以由图 5.1 描述：

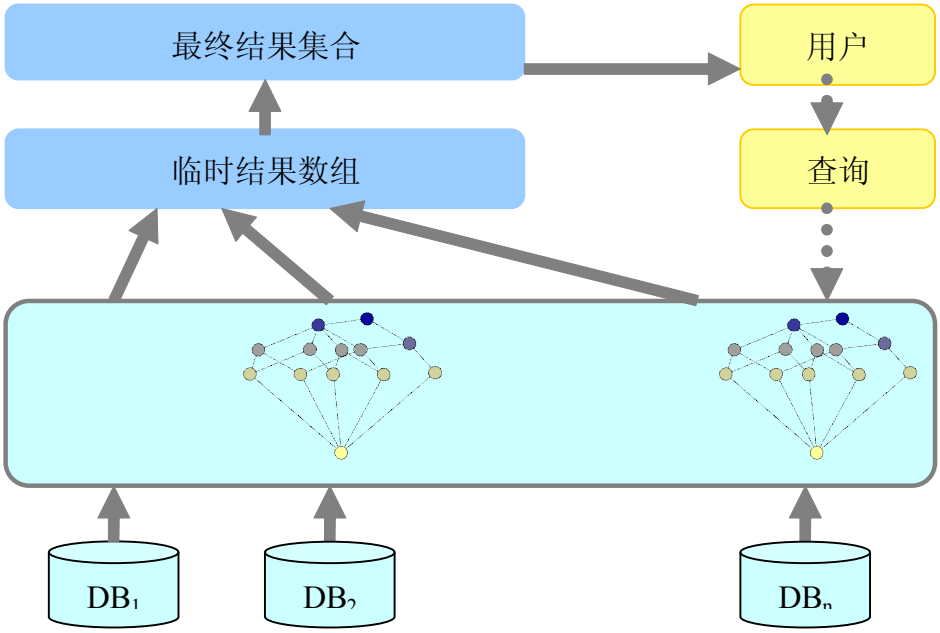

图 5.1 协作概念信息检索系统框图

从上图中，可以看到，对于各个不同的子数据库，构成各自独立的子形式背景，进一步的，根据相关构格算法构造子概念格，并在此过程中获得等价对象集合，相关的等价对象可以被约简，这形成了一个基本结构。当用户输入查询的各个关键词，分析得到对应的相关属性集，针对多个子形式概念格，在各自格上匹配满足一定相似度的形式概念集合并放入临时结果数组。这个过程中，由于各个子格上概念匹配的操作是独立的，所以非常便于分布式实现。当各个子过程完成后，再对

存放在临时结果数组中的概念进行组合合并，并再一次进行相似度的匹配，最终获得的结果返回给用户。基于上述的整体思想，本章提出相关策略并设计分布式算法，实现了具体的细节。

5.1.2 等价对象集的约简

定义 5.1 closure[77,78]：假设 $x \in G$ 中的一个对象，$A \in G$ 是一个对象集合，I 为 $G \times M$ 上的一个关系，那么，$closure(x) = g(f(x)), closure(A) = g(f(A))$。

定义 5.2 等价对象集[76]：假设 $x \in G$ 中的一个对象，$A \in G$ 是一个对象集合，I 为 $G \times M$ 上的一个关系。对象 x 等价于对象集 A，当且仅当，$\{x\} \cup A$ 是 I 上形式概念的外延，$closure(x) = closure(A) = \{x\} \cup A, x \notin A$。

比如，如下表 5.1 中的形式背景

表5.1 示例形式背景

	a	B	c
o1	×	×	×
o2	×		×
o3	×		
o4		×	×
o5			×

其中，对象 o5 等价于对象集(o1，o2，o4)的，由于包含对象 o5 的形式概念是 CP((o1，o2，o3，o4)，(c))，而相反地，包含对象集{o1，o2，o4}的形式概念也是 CP。符合上述等价对象集的定义，所以我们称对象 o5 是等价于对象集{o1，o2，o4}的。

既然对象 o5 可以由对象集(o1，o2，o4)等价表示，那么，可以在原来的形式背景中将对象 o5 约简，即对形式背景做对象层次上的约简，整个约简过程可由算法 5.1 描述：

算法 5.1：等价对象约简算法(FC)

输入：FC，原始形式背景

输出：经过对象约简后的形式背景

步骤：For 每个对象 o in FC：

p 为 o 对应的属性集合

查找形式背景中除对象 o 外的对象子集 sub，并得到其属性集 sub_p

If p == sub_p：

表示对象 o 可用对象子集 sub 来表示，即 sub 和 o 是等价的
将对象 o 从形式背景中删除，或者约简了对象 o 之后的形式背景 FC’

Return FC’

算法 5.1 是针对概念对象进行的约简算法，而在属性层次上没有进行约简，这可在属性提取时进行，形成初步形式背景之后再根据此算法进行对象层次上的约简。这样从整体上就可以得到比较精简的形式背景。

5.2 协作信息检索过程

基于文献[76]中提出的协作信息检索系统的基本思想，本章提出改进了的协作信息检索系统，其过程可以分为三个步骤：

1) 针对各个子数据库 DB_i，经过相关预处理之后，形成初步形式背景 FC_i，并根据算法 5.1 对初步形式背景进行对象约简并得到等价对象集合。之后对约简后的形式背景使用渐进式构格算法 Godin 生成子形式概念格 L_i。

2) 对于特定的查询词，抽取得到相关的查询属性集 T_1，T_2，…，T_n，其形式概念可记为((),(T_1，T_2，…，T_n))，即外延为空，在计算形式概念相似度时，可把它的层次号记为 0。然后分别在各个子形式概念格 L_i 上进行形式概念相似度的匹配，各个子临时结果集 SubConceptSet 保存入临时概念结果集 TempConceptSet。该详细过程可由如下算法 5.2 表示。

3) 在得到临时概念结果集之后，使用合并算法 5.3 将这些临时结果概念集进行组合，在进行相似度匹配之后，获得最终的概念集，其中各个形式概念的外延集合中的对象 FinalObjectSet，即为最终结果返回给用户。

其中，在各个子形式概念格上进行查找目标概念，得到子临时结果集 SubConceptSet，该算法是一个典型的可分布式执行的算法，时间和空间上都有一个很好的效果。

算法 5.2：子形式概念格概念匹配算法(L_i，(T_1，T_2，…，T_n)， sim_1)

输入：L_i 子形式概念格，(T_1，T_2，…，T_n) 查询属性集，
sim1 为形式概念相似度阈值

输出：子形式概念结果集 $SubConceptSet_i$

步骤：子临时结果集 $SubConceptSet_i=\varnothing$

根据查询属性集$(T_1，T_2，…，T_n)$，找到一个子形式概念格 L_i 中，和形式概念$((),(T_1，T_2，…，T_n))$ 具有最大相似度的形式概念作为目标概念 C_i。

For 所有子形式概念格 L_i 中除 C_i 之外的形式概念 C_j 进行：

利用公式(4.14)计算 C_i 和 C_j 的概念相似度 $Sim(C_i，C_j)$

If $Sim(C_i，C_j) > sim$：

将形式概念 C_j)加入到子临时结果集 $SubConceptSet_i$ 中

return 子临时结果集 $SubConceptSet_i$

假如有 n 个子形式背景，分别构成 n 个子形式概念格，那么，利用算法 5.2 可以得到 n 个子临时结果集 $SubConceptSet_i$，该算法精简易懂更易于分布式的实现，所以在时间空间上都将会有较好的效率。

算法 5.3：概念合并算法($SubConceptSet_i$， $(T_1，T_2，…，T_n)$， sim_2)

输入：子临时结果集 $SubConceptSet_i$，i 为子形式概念格数，

$(T_1，T_2，…，T_n)$ 查询属性集，sim2 为形式概念相似度阈值

输出：最终概念对象集 FinalObjectSet

步骤：所有概念结果集 AllConceptSet=∅

For 每个子临时结果集 $SubConceptSet_i$ 进行：

\# 将各个子临时结果集并入所有概念结果集

AllConceptSet=AllConceptSet∪ $SubConceptSet_i$

For 所有概念结果集 AllConceptSet 中的每个概念 C_i 进行：

\# 进行两两组合得到新的形式概念

For 所有概念结果集 AllConceptSet 中除 C_i 外的其他形式概念 C_j 进行：

\# 新的概念的内涵为 C_i 内涵和 C_j 内涵的并集，

\# 外延为 C_i 外延和 C_j 外延的交集

NewConcept(i，

j)=$((\text{extent}(Ci)\cap\text{extent}(Cj),\text{intent}(Ci)\cup\text{intent}(Cj)))$

计算得到新的形式概念 NewConcept 和形式概念 T((), $(T_1，T_2，…，T_n)$)的概念相似度 Sim(NewConcept，T)。

If Sim(NewConcept，T)大于 sim2：

将 NewConcept 的外延加入到最终概念对象集

FinalObjectSet 中

return 最终概念对象集 FinalObjectSet

通过算法 5.3，我们能够获得满足一定相似度阈值的最终概念集合，这些概念集合中每个形式概念的外延即为满足用户查询条件的对象集。其中，算法 5.2 和算法 5.3 中都涉及到两个相似度阈值，分别为 sim_1 和 sim_2，在对子形式概念格进行相似度的匹配时，由于子形式概念格中具有的内涵总数可能不多，可能也不完全，所以对于 sim_1 应该设定的小一点以便不会遗漏可用的形式概念，而在合并时，形式概念的合并，由于是通过其内涵的合并进行的，相对来说其内涵信息丰富，所以设置 sim_2 应该稍大一点以便过滤掉不匹配的形式概念。

5.3 简单实例描述

为了更好的说明本章提出的协作信息检索系统的有效性，以下进行简单实例描述。

假设，现有三个形式背景，分别如表 5.2 所示。

表 5.2 三个子形式背景

	A	b	c
o1		×	
o2	×		×
o3		×	×
o4			×

	a	b	d
o1		×	×
o2	×		×
o5			×
o6	×		
o7		×	×

	a	c	D
o2	×	×	×
o5			×
o6	×		

三个子形式背景经过算法 5.1 可以得到约简后的子形式背景如下表 5.3，并可以得到等价对象集合为：o4={o2，o3}，o1=o7。

表 5.3 进行对象约简后的三个子形式背景

	A	b	c
o1		×	
o2	×		×
o3		×	×

	a	b	d
o1		×	×
o2	×		×
o5			×
o6	×		

	a	c	D
o2	×	×	×
o5			×
o6	×		

接着对这些子形式背景使用 Godin 算法，构成子形式概念格，如图 5.2 所示：

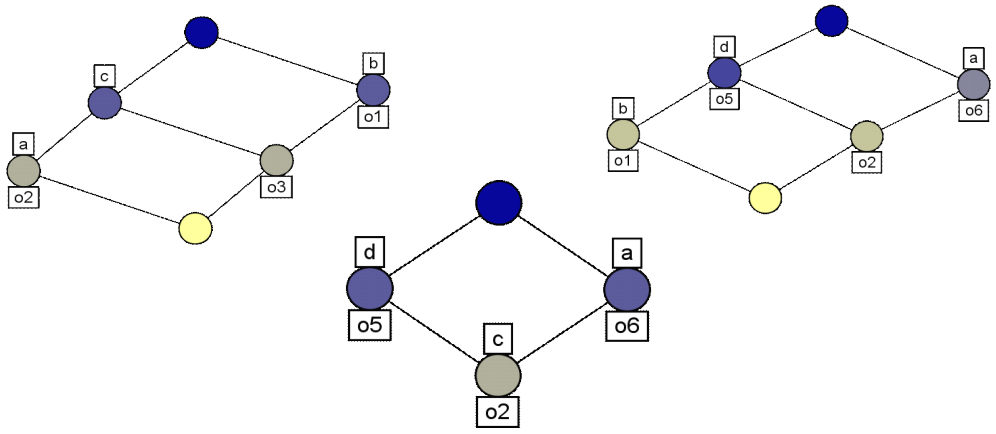

图 5.2 生成的各个子形式概念格

假如，用户需要查询属性集为(a，d)，针对各个子形式概念格，按照算法 5.2 的思想进行查找，其中设定形式概念相似度 sim_1 为 0.5。这里由于属性的语义无法衡量，所以在利用公式(4.14)时，设定 ω 为 1，即只考虑形式概念的结构相似度，但在实际过程中考虑形式概念的语义相似度是非常有必要的。

在第一个子形式概念格中，找到和形式概念(()，(a，d))最相似的形式概念为 C((o2)，(a，c))， 它们之间的相似度为 0.5，并把这个形式概念 C 作为该子形式概念格中的目标形式概念，接着使用这个目标形式概念和格中其他形式概念进行相似度的计算，满足 sim_1 的形式概念加入到子临时结果集 $SubConceptSet_1$，可以找到，形式概念((o2，o3)，(c))和形式概念 C 的相似度为 0.75，形式概念((o3)，(b，c))和形式概念 C 的相似度为 0. 5，它们都满足相似度阈值，所以并入子临时结果集 $SubConceptSet_1$={((o2)，(a，c))， ((o2，o3)，(c))， ((o3)，(b，c))}。按照同样的思想，分别对第二，三个子形式背景进行计算，得到 $SubConceptSet_2$={((o2)，(a，d))， ((o1)，(b，d))， ((o1，o2，o5)，(d))， ((o2，o6)，(a))}和子临时结果集

$SubConceptSet_3$={((o2)，(a，c，d))， ((o2，o6)，(a))， ((o2，o5)，(d))}。这样得到所有可能的概念集合为{((o2)，(a，c))， ((o2，o3)，(c))， ((o3)，(b，c))，((o2)，(a，d))， ((o1)，(b，d))， ((o1，o2，o5)，(d))， ((o2，o6)，(a))， ((o2)，(a，c，d))， ((o2，o6)，(a))， ((o2，o5)，(d))}。经过算法 3 进行合并成新的形式概念，总共有{((o2)，(a，c，d))， ((o3)，(b，c)， ((o1)，(b，d))}。假定 sim_2 为 0.6，计算这些概念和形式概念(()，(a，d))相似度，可以分别得到相似度为 0.666，0.5，0.5，那么，满足相似度阈值的形式概念为((o2)，(a，c，d))，它的外延就是 o2，再结合等价对象集，形成最终对象集返回给用户，即在这个例子中仍然是 o2。

5.4 复杂度分析

假设现有 n 个子形式背景，通过 Godin 算法得到子形式概念格，其时间复杂度为 $O(2^{2K}|G|)$，G 为对象的个数，K 一般情况下为一个固定值。等价对象约简算法，即算法 5.1 需要的时间复杂度为 $O(|C|2^M)$，C 为形式概念格中的所有形式概念，M 为属性个数。子概念格上形式概念的匹配算法，即算法 5.2，仅需要遍历整个形式概念集即可找到满足一定相似度阈值的形式概念，即时间复杂度为 $O(|C|)$。对于合并算法 5.3，由于需要对形式概念进行两两合并，如果临时概念结果集中有 N 个形式概念，那么合并的时间复杂度为 $O(\frac{N^2}{2})$。综合考虑，整个过程需要的时间复杂度为 $O(2^{2K}|G|+|C|2^M+\frac{N^2}{2})$。

但从整体上考虑的话，本章提出的整个系统能够很方便的部署在分布式计算环境中，那么，整个系统性能会在时间和空间上都将有一个较大的提升。

5.5 本章小结

在协作信息检索中，采用对各个子形式背景分别利用形式概念相似度的衡量方法对已有形式概念进行匹配，在找到临时概念集之后采用合并的方式，获得新的概念，再对新的概念进行相似度的匹配，最终获得满足用户需求的结果集合。整个系统充分体现了协作的思想，便于在分布式环境中部署实施，并且在匹配查询关键词时使用了不精确的方式，从结构和语义两个层次上进行衡量，更好的体现了人性化

需求。但本章虽然提出了整个系统框架和具体的算法，但由于硬件条件的限制，使得整个系统无法在真正分布式的环境下实现，所以在最后的评测分析中无法得到现实数据来分析性能，而仅仅在理论上分析时间空间复杂度。这会在以后的工作中继续完善，得到更加详细的数据结果报告以便做进一步的改进。

6 基于 WordNet 的检索结果个性化排序方法

Web 上的信息资源与日俱增，人们通过搜索引擎去寻找信息。虽然传统的信息检索技术满足了人们一定的需要，但由于其通用的性质，不能满足不同用户的特定查询请求。在用户输入查询关键字后，搜索引擎返回的结果非常之多，以致用户还得手工筛选出自己感兴趣的检索结果，这个过程需要花费大量的时间和精力。如何有效的返回更有针对性的检索结果，进一步提高检索效率和用户体验已成为信息检索中比较迫切的问题。个性化检索(Personalized Search)就是针对此问题提出的，它根据不同用户检索的历史记录(如检索关键字，在检索结果中的点击情况，在各个网站的访问情况等)，返回更适合这个用户的检索结果。通过对用户的历史查询记录进行提取，建立用户兴趣模型(User Interest Model)并利用该模型对检索结果重新排序成为个性化检索中的一个有效解决方案。本章提出了一种基于语义本体 WordNet 建立用户兴趣模型，再对初始检索结果分析并根据此模型重新排序的算法，使其按照用户兴趣匹配程度返回检索结果。

近年来，有很多关于个性化和语义检索的工作，其中，文献[79-80]通过构建用户本体(User Ontology)和一系列语义推理规则进行个性化检索结果返回。文献[81-84]根据用户行为构建的用户描述文件和用户本体(User Ontology)不够准确，且提取的语义关系也是最基本的，这导致系统检索效果不是很理想，而文献[82]提出的系统，虽然完整系统，但比较庞大和复杂，在检索过程中，仅基于概率上的规律，并没有利用关键词间的语义关系和用户历史记录进行分析，所以很难说检索系统返回的结果就是用户所感兴趣的。在构建用户本体时，文献 Kumar[85]使用用户浏览记录，文献 Kim[86]利用用户历史记录来构建的。文献[87]分别使用矢量空间模型和概率模型设计并实现了一个完整的个性化检索系统。利用语义本体 WordNet 对用户历史检索记录进行分析，从而构建用户兴趣模型。其中，不同的本体词汇采用得分机制并根据不同词汇类型赋予不同数值，以便更精确的表达出用户兴趣所在，进一步为初始检索结果的重新排序提供一个度量依据。

6.1 用户兴趣模型

构建用户兴趣模型，首先对用户历史检索关键词进行分析。对于某个关键

词，利用 WordNet 词汇之间的语义关系，提取与这个关键词具有语义关系的词汇并作为词汇节点加入到用户兴趣模型中。根据不同的得分策略不断更新词汇节点的得分值，最终形成类似于一张词汇语义关系网络，只是这个网络具有得分并且是针对特定用户的，而且比 WordNet 中的词汇网络是大大精简规模。随着检索关键词的不断加入，网络中某块区域节点中的得分普遍都很大时就表示用户对这类词汇比较感兴趣，进一步反应了对词汇所代表的类别的感兴趣程度。

在传统检索过程中，用户输入特定的关键词，搜索引擎对该关键词进行检索，返回初始检索结果序列之后，根据用户兴趣模型对初始结果进行重新排序，最后返回给用户。在用户输入新到的检索关键词之后，需要不断更新用户兴趣模型。根据用户兴趣模型进行检索结果重排的整个过程可以由图 6.1 来表示：

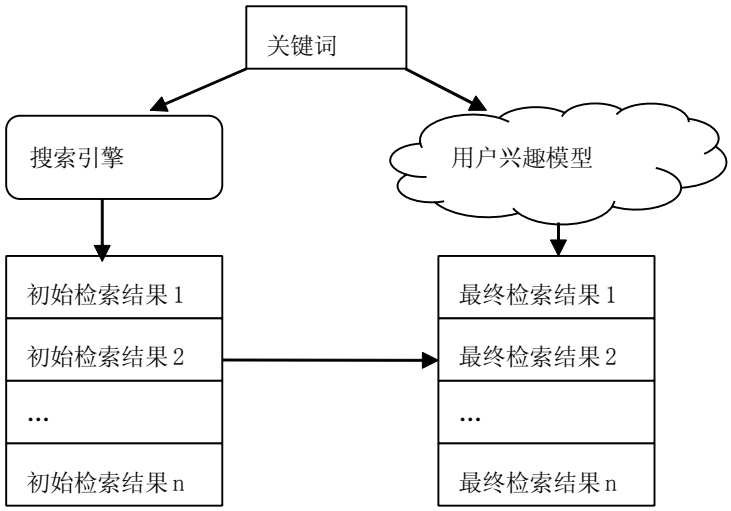

图 6.1 检索结果重排过程

6.1.2 用户兴趣模型表示

检索系统在用户每次查询之后要保存查询关键字，这条历史记录可以看成是一条数据流。随着关键词源源不断的到来，使用每个新到的关键词来更新原有的用户兴趣模型，本章采用的是渐进更新策略，为了更好的表示出用户对某些词汇的兴趣程度，对每个相关词汇赋予得分值(Score)，对于不同关系的词汇，根据关系的远近赋予不同等级的得分值来更好地体现其重要程度，这样一来，用户查询频率越高的词汇，其得分值就越高。由于历史关键词是有先后顺序的，最新查询的关键词总是比历史查询的关键词更有意义，所以在递增得分值时采取将某个词汇原先的得分

值乘上一个衰减系数β后再增加得分。另外，随着关键词的不断加入，使得词汇网络规模越来越大，所以在一定时间后，需要删掉某些陈旧的节点，以便在较好的反映用户兴趣的同时又能精简网络规模，加快查询效率。整个思想可以描述如下:

1) 对于一个新来的关键词 keyword，在原有用户兴趣模型中查找是否已经包含该词，若存在，直接增加相关词汇节点的得分，即在原有得分基础上乘上衰减系数β后递增 5 分; 若不存在，则新建词汇节点并赋予得分 5 分。

2) 依据 WordNet 查找与关键词相关的以下三种关系:

* 同义关系，获得同义词集 synonymset，依次将每个同义词插入到原有用户兴趣模型中，若已存在该词汇节点，则原有得分乘上衰减系数β后递增 4 分; 若不存在，新建词汇节点并赋予得分 4 分，同时增加一条新的无向边，连接关键词和该同义词，标明边的关系为 synony。

* 上下位关系，获得该关键词的上位关系词集 hyponymset 和下位关系词集 hypernymset，依次将每个词汇插入到原有用户兴趣模型中，和上述同义关系类似，只是赋予得分为 2 分，如果是新加节点，需增加一条新的有向边并标注 hyponym 或 hypernym 关系。

* 整体部分关系，获得该关键词的整体关系词集 meronymset 和部分关系词集 holonymyset，和上述同义关系类似，只是赋予得分为 1 分，如果是新加节点，需增加一条新的有向边并标注 meronym 或 holonym 关系。

3) 为了精简模型，在一定时间后，删除得分值较小的 N 个词汇节点。

经过以上步骤之后，可以得到一个丰富的词汇语义网络，如图 6.2， 为了让数值看起来更清晰数值，图中衰减系数β设置为 1， 即是依次累加，没有衰减， 但在实际中可以设置β为 0～1 中的小数。节点所代表的特定词汇表达了用户历史检索关键词相关的词集，边表示词汇之间的关系。假设说未来的另一个检索关键词正好对应某个节点，那么就能累加地更新其语义相关的词汇集，其得分值在对初始检索结果进行重新排序时提供了一个兴趣度衡量依据。随着用户历史记录的不断增加，这个网络也不断庞大，所附带的信息也越来越丰富，这样更好的表达了用户的兴趣所在。

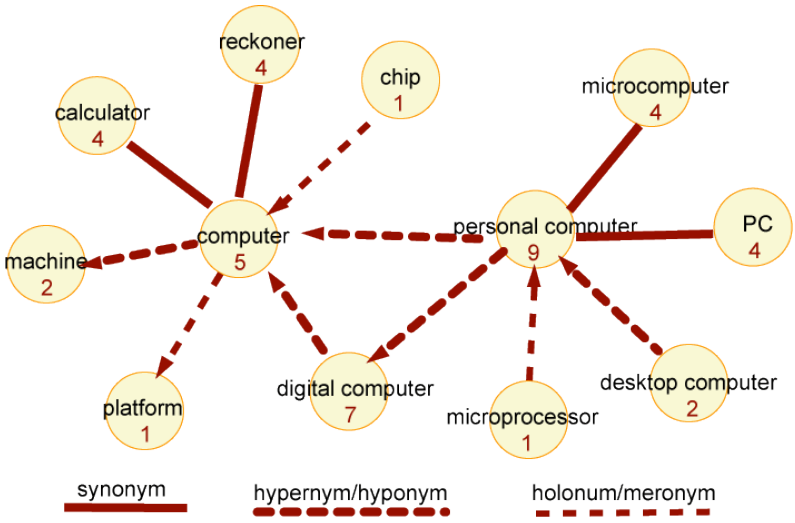

图 6.2 用户兴趣语义网络

6.1.3 用户兴趣模型构建

根据前面的思想，提出具体的用户兴趣模型渐进更新算法 6.1 ：

算法 6.1: UpdateUIM(keyword，U，β，N)

输入: 新到的检索关键词 keyword，原有的用户兴趣模型 U，
衰减系数β，删除个数 N

输出: 更新后的用户兴趣模型 U'

描述:

Step 1: # 将当前关键词加入到原有用户兴趣模型 U 中
If U 中不存在 keyword:
新建一个节点，并赋予得分值 5
Else:
已经存在该关键词汇节点
更新该词汇节点的得分值，即在原有得分基础上乘上衰减系数
β后递增 5 分

Step 2: # 查找 WordNet，获得相关词汇集并更新

Step 2.1: # 根据 WordNet 获得同义词汇集
synonym_set = GetSynonymSet()
for 每一个同义词 w in synonym_set:

if U 中不存在 w:

新建一个节点，并赋予得分值 4

增加一条新的无向边 keyword--w，并标注该边的特征为 synony 关系

else:

已经存在该词汇节点

更新该词汇节点的得分值，即在原有得分基础上乘上衰减系数β后递增 4 分

Step 2.2: # 根据 WordNet 获得上下位词汇集

hyponym_hypernym_set = GetHyponym_HypernymSet()

for 每一个上/下位词 w in hyponym_hypernym_set:

if U 中不存在 w:

新建一个节点，并赋予得分值 2

增加一条新的有向边，如果 w 是上位词则增加 w->keyword，并标注该边的特征为 hypony 关系;

如果是下位词则增加 keyword->w，并标注该边的特征为 hyperny 关系

Else: # 已经存在该词汇节点

更新该词汇节点的得分值，即在原有得分基础上乘上衰减系数β后递增 2 分

Step 2.3: # 根据 WordNet 获得整体/部分词汇集

meronym_holonym_set = GetMeronym_Holonym Set()

for 每一个整体/部分词 w in meronym_holonym_set:

if U 中不存在 w:

新建一个节点，并赋予得分值 1

增加一条新的有向边，如果 w 是整体关系词则增加 w->keyword，并标注该边的特征为 meronym 关系;

如果是部分关系词则增加 keyword->w，并标注该边的特征为 holonym 关系

else:

已经存在该词汇节点

更新该词汇节点的得分值，即在原有得分基础上乘上衰减系数β后递增 1 分

Step 3: 在一定时间后，如一个月之后，选出得分值较小的 N 个词汇节点，将节点和节点对应的边删除

Step 4: 结束更新，返回新的用户兴趣模型

根据这个算法，可以不断更新用户兴趣模型，而且随着关键词的不断增加。如果用户检索兴趣改变了，该模型也能通过词汇的得分值动态地体现出来。此模型的构建是为接下来的工作做了一个重要基础。

6.2 检索结果重排

当用户输入检索关键字，提交给检索系统之后，返回初始检索结果。这个是针对大众的，普通的，数量具大的结果条目。这时，可以通过之前构建的用户兴趣模型对这些庞大的检索结果进行筛选并根据兴趣程度重新排序，最终返回给用户。整个过程可以由图 6.3 所示的过程来表示:

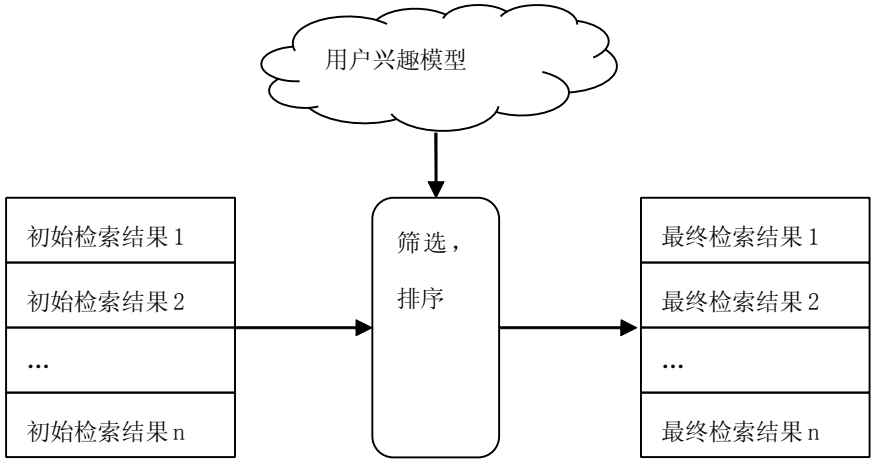

图 6.3 初始检索结果重新排序过程过

其中，最为关键的问题是对兴趣度的衡量。假设给定一条记录 X，可以看成是由词汇组成的序列(x1，x2，… ，xn)，如果其中有些词汇正好是用户兴趣模型中的某些节点，那么把它们对应的得分值累加作为该条记录 X 的兴趣度值，记为 Interest(X)。很明显的，如果这个用户兴趣网络足够大，词汇很容易命中网络节点并能够参与兴趣度的计算。如果都没有命中，那么记 Interest(X)=0。

计算得到每条记录的兴趣度，根据这个兴趣度的大小进行重新排序。若两条记录的兴趣度相同，那么还是以原来排序的先后顺序排列。这样，在最终的返回结果中可以看到，某条结果匹配用户兴趣模型程度越大，它就越靠前。根据这个思想，提出算法 6.2。

算法 6.2: ReSort(初始搜索结果条目，U)

输入: 初始检索结果条目(X1~Xn)，用户兴趣模型 U

输出: 重新排列后的检索结果条目(Y1~Yn)

描述:

```
Step 1:  # 依次遍历初始检索结果条目
         For Xi in (X1~Xn):
             Interest (Xi) = 0
             # 计算 Xi(x1，x2，… ，xn)的兴趣度
             for xj in Xi(x1，x2，… ，xn):
                 # 在 U 中查找是否含有 xj 节点，如果有则累计 xj 的得分
                 if U 包含 xj:
                     Interest (Xi) += Score (xj)
             # 将当前 Xi 插入到 Y 中，Y 是按照兴趣度由大到小排列的
             For Yk in Y:
                 If Interest(Xi) > Yk:
                     # 将 Xi 插在 Yk 之前的位置
                     Break
             # 若 Xi 都小于 Y 中的任意一个，则直接追加到 Y 的末尾，
             # 即以原来的顺序插入
         返回重新排列后的检索结果 Y
```

根据以上算法对初步检索结果重新排序，即某条检索结果中包含用户经常访问的关键词或相关词汇会在结果序列中比较靠前的位置出现，反之则会出现在相对靠后的位置，因此返回的检索结果更加符合用户兴趣，进一步提高检索效率和用户体验。

6.3 实验分析

由于个性化搜索依赖于用户兴趣模型，此用户兴趣模型是通过获取 Google Web History 中的用户历史检索关键词集合，根据 WordNet2.0 获得关键词的词汇语义关系来构建的，具体过程是通过算法 6.1 来实现。接下来利用 Google Search API 对用户输入的关键词检索获得 100 个检索记录，根据用户兴趣模型，对这 100 条检索记录进行重新排序并返回给用户，最后再把该新的检索关键词更新用户已有的兴趣模型中。整个实验使用 Google App Engine[88]作为部署平台，实现一个符合特定用户兴趣的搜索引擎，如图 6.4 所示。

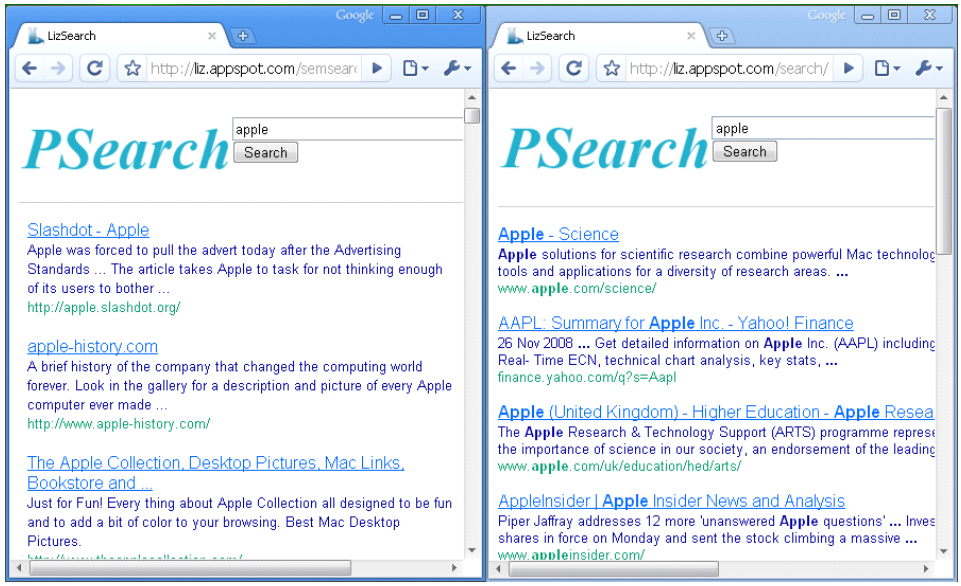

图 6.4 结果截图（注：左图为使用UIM后的结果，右图为未使用UIM后的结

为了验证本章方法的有效性，采用 Google Search API 返回的 100 条记录和经过本章算法 6.2 返回的记录做比较，根据返回的前 20 个记录中用户满意的网页个数进行统计，得到如图 6.5 的结果。

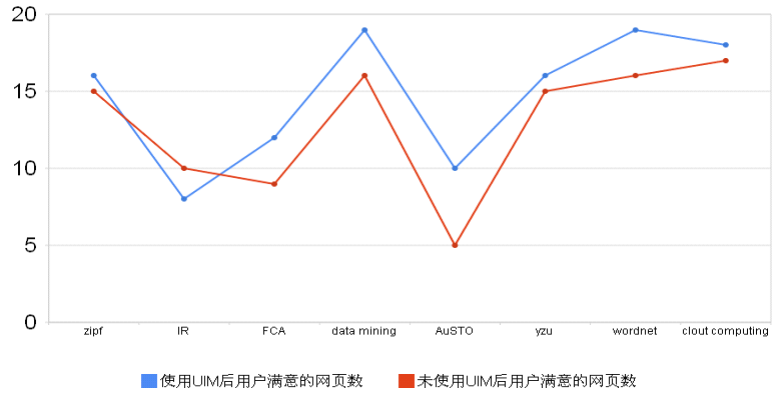

图 6.5 用户满意度测试结果比较

从图 6.5 中可以看到，在大多数情况下，本章方法得到的结果要比未使用用户兴趣模型的好，返回的网页条目也能更好的符合用户的需求。由于满意度是根据用户的主观意愿来衡量的，上图中的测试结果只能在某种程度上反映此方法有优势。但是，如果随着用户兴趣模型的不断迭代更新，返回的结果会越来越符合用户的需求。

6.4 本章小结

本章利用语义本体 WordNet 对用户历史检索记录进行分析，并构建用户兴趣模型，然后根据此模型对初始检索结果进行重新排序，已返回更符合用户需求的结果。实验证明本章的方法是可行有效的。

7 结束语

7.1 论文总结

随着网络的飞速发展，网上的信息量在以指数形式飞速增长，由于因特网的广泛性和开放性，因特网已成为人们获取信息的重要来源；因此，如何快速、准确地从浩瀚的信息资源中寻找所需的信息已经成为困绕用户的一个难题。如何提取有用的信息，如何匹配关键字，如何加快检索效率，又如何将结果更人性化的呈现给用户都是可以进一步深入研究的课题，本论文在一定程度上，针对这几个问题提出了相关实际可用的方法，具体主要在以下几个方面展开了研究：

1） 为了提取更少的特征来尽可能的表达出文本蕴含的信息，通过分析各个特征自身的统计特性，根据Zipf定律进行全局特征提取，不提取文本特征空间中普遍存在的特征和噪声特征；其次，在对特征的类别信息进行统计分析后，计算出每个特征词的类别贡献程度，进行类内局部特征提取；在选取有效的特征之后，提出新的特征权重计算公式TF-IDF-CF并通过实验验证此特征提取策略是准确有效的。

2） 基于概念格的数学理论基础，结合粗糙集理论和语义本体理论，提出了一个形式概念相似度计算方法，改进了基本Tversky相似度模型。从形式概念的结构相似和语义相似两个层次上衡量概念间的相似度，并将两者结合起来构成最终的形式概念相似度衡量方法，这种方法在不同层面上度量了形式概念的真实相似度。

3） 在协作信息检索中，针对多个子文本数据库，利用渐进式构格算法构建各个子形式概念格，然后分别利用形式概念相似度的衡量方法对已有形式概念进行匹配，在找到临时概念集之后采用合并的方式，获得新的概念，再对新的概念进行相似度的匹配，最终获得满足用户需求的结果集合。整个系统充分体现了协作的思想，便于在分布式环境中部署实施，并且在匹配查询关键词时使用了不精确的方

式，从结构和语义两个层次上进行衡量，更好的体现了人性化需求。

4） 人们通过搜索引擎去寻找Web上的信息资源。虽然传统的信息检索技术满足了人们一定的需要，但由于其通用的性质，不能满足不同用户的特定查询请求。在用户输入查询关键字后，搜索引擎返回的结果非常之多，以致用户还得手工筛选出自己感兴趣的检索结果，这个过程需要花费大量的时间和精力。根据不同用户检索的历史记录(如检索关键字，在检索结果中的点击情况，在各个网站的访问情况等)，返回更适合这个用户的检索结果。利用语义本体WordNet对用户历史检索记录进行分析，从而构建用户兴趣模型。其中，不同的本体词汇采用得分机制并根据不同词汇类型赋予不同数值，以便更精确的表达出用户兴趣所在，进一步为初始检索结果的重新排序提供一个度量依据。

7.2 进一步研究工作

基于上面的研究结果，将来可以在下列方向上进行进一步的研究：

(1) 针对形式概念相似度匹配时，语义相似的相关策略和方法可以进一步研究。由于目前是基于英文 WordNet 本体，另外一个可扩展方向就是自行构建适合特定需求的领域本体以表达出更加适合应用需要的情况。

(2) 在实际应用中，可以采用更加详细的技术和方式提高写作信息检索系统的性能。

(3) 根据用户搜索历史记录，构建用户兴趣模型时，可以采用更好的策略来维护用户不断变化的兴趣模型，以更加有效的反映用户当前的兴趣及行为。另外一方面，用户模型的构建是基于现成的 WordNet 本体，对于特定领域中的检索任务，比如图书检索时，可以构建一个图书领域本体，然后脱离 WordNet 直接根据这个本体来构建和更新用户兴趣模型，这样在检索结果排序时更能确切的反映用户的检索兴趣。另外，如何设置用户兴趣模型中词汇得分值的更新策略更能反映出用户兴

趣的不断变化是一些扩展工作。

(4) 目前所有成果都是基于英文词集，所以未来的另一个关键扩展可以将现有理论和技术转移到中文词集上。比如说对于中文文本的特征提取，其中涉及到中文分词；中文词集本体的构建，用于构建中文词语之间的关系；中文词语的语义解析方法的进一步研究等。

参考文献

[1] Jiawei Han, Kamber M.数据挖掘——概念与技术.范明, 孟小峰译.北京:机械工业出版社, 2001.

[2] J.Han and M.Kamber. Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers, San Fransisco, CA, 2001.

[3] Carpineto C, Romano G. A lattice conceptual clustering system and its application to browsing retrieval. Machine Learning, 1996, 24(2): 95-122.

[4] R.J. Bayardo, Efficlently mining long patterns from databases, In: Proceedings of SIGMOD'98, 1998, pp 85-93.

[5] Han J.Kamber M, Tung A K H．Spatila Clusteirng Methods in Data Mining：Survey[c]．Gographic Data Mining and Knowldege Discovery, 2001

[6] Buckley C, Voorhees EM. Retrieval evaluation with incomplete information[C]// Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. New York: ACM, 2004:25-32.

[7] 李荣陆,王建会,陈晓云,等. 使用最大熵模型进行中文文本分类[J].计算机研究与发展,2005,42(1):94~101

[8] Carpineto C, Romano G. Information retrieval through hybrid navigation of lattice representations. International Journal of Human-Computer Studies, 1996, 45: 553-578.

[9] Cole, R., Eklund, P.: Application of formal concept analysis to information retrieval using a hierarchically structured thesauris. In: Supplementary Proceedings of International Conference on Conceptual Structures, ICCS '96：1–12

[10] 丁国栋.基于统计语言建模的信息检索及相关研究[D]. 中国科学院研究生院（计算技术研究所）, 2006.

[11]Ganter.B, Wille.R. Formal concept analysis: mathematical foundations. Proceedings of Berlin: Springer, 1999

[12]Han.jia-wei, Pei.Jian, Yin.Yi-wen. Mining Frequent Patterns Without Candidate Generation.the 2000 ACM SIGMOD Internal Conference on Management of Data, 2000. 1–12

[13]Xie.Z, Liu.Z. Research on classifier based on lattices structure. Conference on

Intelligent Information Processing, 16thWorld Computer Congress, Beijing, China, 2000. 333–338.

[14]梅馨,邢桂芬.文本挖掘技术综述[J].江苏大学学报,2003,9.

[15] Nicola Guarino, Pierdaniele Gieratta: Ontologies and Knowledge.Bases-Towards a Terminological Clarifrcation, In N. Mars (ed.): Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing, I995: 25-32；

[16] T.R. Gruber. A Translation Approach to Portable Ontology Specifications, Knowledge Acquisition,5（2）,1993:199-221

[17] G. van Heijst, A.Th. Schreiber, B.J. Wielinga.Using Explicit Ontologies in Development, J. of Human and Computer Studies, 46(2/3): 183-292, 1997:

[18] L.K.Alberts: An Ontology for Engineering Design. PhD thesis, University of Twente, 1994

[19] T. R. Gruber: Towards principles for the design of ontologies used for knowledge sharing. In Guarino, N.&Poli, R., editors, Formal Ontology in Conceptual Analysis and Knowledge Representation. Kluwer, 1994：907-928

[20] Guarino, N., Some Ontological Principles for Designing Upper Level Lexical Resources. In: Proceedings of First International Conference on Language Resources and Evaluation. Granada, Spain, ELRA- European Language Resources Association: 527一534, 1998

[21] G. van Heijst, A.Th. Schreiber, B.J. Wielinga.Using Explicit Ontologies in KBS Development, J. of Human and Computer Studies, 46(2/3): 183-292, 1997:

[22] D.A.Lindberg, B.L.system, Methods of Information in Humphreys, The unified medical language Medicine, 1993, 32: 281-291

[23]A. I.Rector, W. A. Nowlan. Methods of Information Kay: A framework for modeling the electronic in Medicine, 1993, 32: 109-119

[24] Igor Jurisica, Jolm Mylopoulos, Eric Yu: Using Ontologies for Knowledge Management: An Information Systems Perspective, Proc. Of Annual Conference of the American Society for Information Science, Washington,1999：436-454

[25] M R Genesereth , R E Fikes. Knowledge interchange format version reference manual Stanford University, Tech Rep : Logic29221 , 1992

[26] T R Gruber. ONTOLINGUA : A mechanism to support portable ontologies Stanford University, Tech Rep : KSL291266 , 1992

[27] R MacGregor, R Bates. The loom knowledge representation language USC Information Sciences Institute, Tech Rep: ISI/RS2872188, 1987

[28] L Farinas , A Herzig Interference logic conditional logic frame axiom International Journal of Intelligent Systems, 1994:119-130

[29] J Heflin, J Hendler Searching the web with SHOE In: Artificial Intelligence for Web Search Menlo Park, CA: AAAI Press, 2000: 35-40

[30] E. K. Robert. Conceptual knowledge markup language: The central core The 12th Workshop on Knowledge Acquisition, Modeling and Management (KAW99), Banff, Canada, 1999

[31] P D Karp, V K Chaudhri, J Thomere XOL : An XML based ontology exchange language AI Center, SRI International, TechRep: 559,1999

[32] Dave Beckett, Brian McBride1 RDF/ XML Syntax Specification (Revised) World Wide Web Consortium http://www.w3.org/tr/rdf2syntax2grammar/, 2004-02-10

[33] D Brickley, R V Guhal RDF Vocabulary Description Language 110 : RDF Schema1 World Wide Web Consortiuml http://www.w3.org/tr/rdf2schema/,2004-02-10

[34] F Baader, D Calvanese, D McGuinness, etal The Description Logic Handbook : Theory, Implementation and Applications Cambridge : Cambridge University Press, 2003

[35] Donnifm, et al. Reasoning in description logics. Studies in Logic, Language and Information.CLSI Publications, 1996. 193 - 238.

[36] WordNet 2.0: http://wordnet.princeton.edu/wn2.0.shtml

[37]R.Wille. Knowledge Acquisition by Methods of Formal Concept Analysis. Proceedings of In: Diday E(ed), Data Analysis, Learning Symbolic and Numeric Knowledge,P365-380,New York: Nova Science Publishers, Inc, 1989.

[38]Birkhoff.G. Lattice Theory. Proceedings of RI: American Mathematical Society, 1967

[39]R.Wille. Restructuring Lattice Theory: An Approach Based on Hierachies of Concepts.Proceedings of Ordered Sets. Dordrecht:Reidel,P445-470, 1982.

[40]R.Wille. Knowledge Acquisition by Methods of Formal Concept Analysis. Proceedings of In: Diday E(ed), Data Analysis, Learning Symbolic and Numeric

Knowledge,P365-380,New York: Nova Science Publishers, Inc, 1989.

[41]Botorg G, H.Kuchen. Using algorithmic skeletons with dynamic data structures. Irregular,Lecture Notes in Computer Science, 1996, 1117:263–276.

[42]Emilion.R, Lambert.G, Levy.G. Algorithms for general Galois lattice building. Proceedings of Technical report CERIA,university PARIS IX Dauphine, 2001.

[43]R.Agrawal, R.Srikant, Quoc.Vu. Mining Association Rules with Item Constraints. Proceedings of In Proc 3rd Int. Conf. Knowledge Discovery and Data Mining (KDD97),P67-73,Newport Beach, California, 1997

[44]Ingo.Schmitt, Gunter.Saake. Merging Inberitance Hierarchies for Scheme Integration Based on Concept Lattice. 1997

[45]Godin.R, Mineau.G.W, Missaoui.R. Incremental structing of knowledge bases. Proceedings of International knowledge retrevial, use, and storage for efficiency Symposium(KRUSE.95), Santa Cruz, 1995

[46]R.Godin, H.Mil, G.Mineau, et al. Design of class hierarchies based on concept Galois lattices. Proceedings of TAPOS 4(2),117-134, 1998

[47]Petko.Valtchev,Rokia.Missaoui,Robert.Godin.Generating frequent itemsets incrementally:two novel approaches based on Galois lattice theory. Proceedings of ournal of Experimental and Theoretical Artificial Intelligence 14(2-3),115-142, 2002

[48]谢志鹏, 刘宗田. 概念格节点的内涵缩减及计算. 计算机工程.2001

[49] Nafkha I., Elloumi S., Jaoua A., Using Concept Formal Analysis for Cooperative Information Retrieval. Concept Lattices and their applications Workshop (CLA'04), VSB-TU Ostrava, September 23th-24th, 2004.

[50] 王美方,刘培玉,朱振方. 基于TFIDF的特征选择方法. 计算机工程与设计,2007,12,Vol.28,No.23

[51] 冯长远,普杰信. Web文本特征选择算法的研究. 计算机应用研究,2005,7,Vol.22,No.7:36~38

[52] 罗欣,夏德麟,晏蒲柳. 基于词频差异的特征选取及改进的TF-IDF公式. 计算机应用,2005,9,Vol.25,No.9

[53] Zipf Curves and Website Popularity. http://www.useit.com/alertbox/zipf.html

[54] 呼声波,刘希玉. 网页分类中特征提取方法的比较与改进. 山东师范大学学报

(自然科学版),2008,9,Vol.23,No.3

[55] 褚力,张世永. 基于集成合并的文本特征提取方法.计算机应用与软件,2008,10,Vol.25,No.10

[56] 吴迪,张亚平,殷福亮,李明. 基于类别分布差异和VPRS特征选择的文本分类方法. 电子与信息学报,2007,12,Vol.29,No.12

[57 徐燕,李锦涛,王斌,孙春明. 基于区分类别能力的高性能特征选择方法. 软件学报 Journal of Software,Vol.19,No.1,2008,1,pp.82-89.

[58] Qiang Wang,Yi Guan,XiaoLong Wang,Zhiming Xu. A Novel Feature Selection Method Based on Category Information Analysis for Class Prejudging in Text Classification. IJCSNS International Journal of Computer Science and Network Security, VOL.6 No.1A, January 2006.

[59] G.Zipf,Human Behavior and the Principle of Least-Effort (Cambridge, Mass, 1949; Addison-Wesley,1965);

[61] Xiaojin Zhu.CS838-1 Advanced NLP: Words, Zipf's Law, Miller's Monkeys. 2007 . http://pages.cs.wisc.edu/~jerryzhu/cs838/words.pdf

[62] 季燕江. Zipf定律及其应用. http://www.qiji.cn/eprint/abs/4.html

[63] 韩筱璞. 网络信息搜索中的Zipf 定律. http://www.qiji.cn/eprint/abs/840.html.

[64] 张宁, 贾自艳, 史忠植. 使用KNN 算法的文本分类. 计算机工程. 2005, 4, Vol. 31, No. 8(171~173)

[65] Yiming Y. An evaluation of statistic approaches to text categorization[J]. Information Retrieval, 1999, 1(1/2):69-90.

[66] NewsGroup. 1999. http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/www/data/news20.html

[67] Y. Zhao, X. Wang, W.A. Halang, Ontology mapping based on rough formal concept analysis, in: Proceedings of the Advanced International Conference on Telecommunications and International Conference on Internet and Web Applications and Services, AICT/ICIW, 2006.

[68] A. Tversky, Features of similarity, Psychological Review 84 (1977) 327-352.

[69]Pawlak Z, Rough Sets. International Journal of computer and information Sciences 1982, II: 341-356

[70] M.A. Rodriguez, M.J. Egenhofer, Determining semantic similarity among entity classes from different Ontologies, IEEE Transactions on Knowledge and Data Engineering 15 (2003) 442-456.

[71] Y. Zhao, W.A. Halang, Rough concept lattice based ontology similarity measure, in: Proceedings of the First International Conference on Scalable Information Systems, Hong Kong, 2006.

[72] Lidong Wang, Xiaodong Liu. A new model of evaluating concept similarity. Knowledge-Based Systems 21(2008), 842-846.

[73] Uta Priss. Formal Concept Analysis in Information Science. Knowledge-Based Systems archive Vol. 21 , Issue 1 (February 2008) Pages 80-87

[74] Anna Formica. Concept similarity in Formal Concept Analysis: An information content approach. Knowledge-Based Systems archive Volume 21, Issue 1 (February 2008)

[75] Uta Priss. Formal Concept Analysis in Information Science. Knowledge-Based Systems archive Vol. 21 , Issue 1 (February 2008) Pages 80-87

[76] B.Ganter , R.Wille. Formal Concept Analysis: Mathematical Foundations. Springer-Verlag, 1999.

[77] Jaoua A., Bsaies Kh., and Consmtini W. : May reasoning be reduced to an Information Retrieval problem. Relational Methods in Computer Science, Quebec, Canada, (1999).

[78] Jaoua A., Al-Rashdi A., AL-Muraikhi H., Al-Subaiey M., Al-Ghanim N, and Al-MisaifriS.: Conceptual Data Reduction, Application for Reasoning and Learning. The 4th Workshop on Information and Computer Science, KFUPM, Dhahran, Saudi Arabia, (2002).

[79] Hirst and D. St-Onge. 1998. Lexical chains as representations of context for the detection and correction of malapropisms. In C. Fellbaum, editor, WordNet: An electronic lexical database, pages 305–332. MIT Press.

[80] 赵仲孟,袁薇,何世丽.个性化搜索引擎中用户模型智能调整算法的研究[J].计算机工程与应用,2005,41(24):184-187.

[81] 卢林兰,李明.用户 Ontology 的构建及其在个性化检索中的应用.计算机应用,2006,11,Vol.26,No.11

[82] 曾春,邢春晓,周立柱.基于内容过滤的个性化检索算法.软件学报 Journal of

Software,2003,14(05)0999:1000-9825.

[83] Hirst and D. St-Onge. 1998. Lexical chains as representations of context for the detection and correction of malapropisms. In C. Fellbaum, editor, WordNet: An electronic lexical database, pages 305–332. MIT Press.

[84] Li Zhengwei, Xia Shixiong, Niu Qiang and Xia Zhanguo. Research on the User Interest Modeling of personalized Search Engine. Wuhan University Journal of Natural Sciences, 2007, 9, Vol. 12, No. 5:893-896.

[85] Harshit Kumar, Sungjoon Park and Sanggil Kang. A Personalized URL Re-ranking Methodology Using User's Browsing Behavior. Agent and Multi-Agent Systems: Technologies and Applications, 2008, 4, 3, Vol. 4953/2008: 212-221

[86] Taehwan Kim, Hochul Jeon, Joongmin Choi. Personalized Information Retrieval using the User History. Proceedings of the 2008 International Conference on Multimedia and Ubiquitous Engineering (mue 2008) - Volume 00, Pages: 229-232

[87] Zeng Chun, Xing Chunxiao, Zhou Lizhu. Personalized Search Algorithm Using Content-Based Filtering. Journal of Software,2003,14(05)0999:1000-9825.

[88] Google App Engine: http://code.google.com/appengine/

致 谢

“上善若水，虚若怀谷”。三年的研究生生涯让我深深体会到这八个字的内在涵义。

这三年的时间就像水一样从我整个人生之河中流过，平静而又必不可少。 最好的善心也正如水一样，它可以流淌到任何地方，滋养万物，洗涤污淖。它处于深潭之中，表面清澈而平静，但却深不可测。它源源不断的流淌，去造福于万物却不求回报。在这平静无大风大浪的理论研究生涯中，不断追求这种平静，平凡，默默努力但不求回报的境界。

虚若怀谷，若人能够达到一种谦虚、放眼世界的境界，犹如自己的心胸可以放下所有的东西，那已足矣。这两者是我过去及现在所追求的东西，未来也将继续如此。

三年中的种种，周围的人和事，让我自身不断进步。不止在理论技术上获得提高，更在生活，做人做事方面让我有了一个质的改变。所以非常感谢我们这个并行实验室的所有。

感谢导师李云教授。三年多来，李教授坚持给我在学业上精心的指导， 使得我能够在学术方面取得满意的结果。

感谢栾鸾，王姝，张珊，戴彩艳，孙艳，席艳秋，孙粮磊，潘舟金，陈伯伦等师弟师妹们的一起相互讨论问题，一起学习。让我的研究生生活有了很好的伙伴。

感谢我的父母，不断支持我，也要谢谢他们不断培养我独立行事的能力，让我形成一个不依赖于其他人的性格。

我的研究生涯虽然即将结束，但对于我漫漫人生路还仅仅是一个开始。希望未来平静中渗透着淡淡的精彩，也继续追求“水”一样的境界。

最后，感谢所有关心和帮助过我的人！谢谢～

作者攻读硕士学位期间所发表的文章

1. Yun Li， Yunhao Yuan, Xin Guo, Yan Sheng, Ling Chen. A Fast Algorithm for Generating Concepts, Proc. of international conference on Information and Automation(ICIA 2008)：1728-1733, (EI:084411663016)
2. 栾鸾，李云，盛艳.多关系频繁项集的并行获取, 微电子学与计算机,Vol.25,NO.10,2008:94-96（DPCS2008）
3. 盛艳, 李云, 李拓, 袁运浩.一种基于概念格的本体合并方法, 微电子学与计算机,Vol.25,NO.9,2008:34-36（DPCS2008）
4. 盛艳,李云,李拓,栾鸾.基于概念格模型的本体映射, 南京师范大学学报（工程技术版）,Vol.8,NO.4,2008:91-94（JSCC2008）,
5. Yun Li, Yan Sheng, Sufang Tian and Luan Luan .A Rough Concept Lattice Model of Variable Precision, Proceedings of International Symposium on Intelligent Information Technology Application (IITA 2008) ：162-166 EI:20091411996929
6. Yun Li,Yunhao Yuan, Yan Sheng, Xin Guo and Ling Chen. Mining Self-adaptive Sequence Patterns Based on Sequence Fuzzy Concept Lattice, Proceedings of International Symposium on Intelligent Information Technology Application (IITA 2008) ：167-171, EI:20091411996930
7. Yun Li, Yan Sheng, Luan Luan and Ling Chen .A Text Classification Method with an Effective Feature Extraction based on Category Analysis, Proceedings of the 6th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD'09),Vol.1:95-99 (EI: 20100712709114)
8. Yan Sheng,Yun Li, Luan Luan and Ling Chen .A Personalized Search Results Ranking Method Based on WordNet , Proceedings of the 6th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD'09),Vol. 7 :500 – 504, 14-16 Aug. 2009 (EI: 20100712715597)
9. Yun Li,luan luan, Yan Sheng, Yunhao Yuan. Multi-relational classification based on the contribution of tables, The 2009 International Conference on Web Information Systems and Mining and the 2009 International Conference on Artificial Intelligence and Computational Intelligence (WISM'09-AICI'09), Vol.4:371-374

10. Yan Sheng, Yun Li, Luan Luan.A Concept Similarity Method in Structural and Semantic Levels. Second International Symposium on Information Sciences and Engineering (ISISE 2009) :620-623,26-28

作者攻读硕士学位期间参加的科研项目和学术会议

1. 基于语义的协作信息检索框架研究, 2009 年江苏省普通高校研究生科研创新项目
2. 面向本体的形式概念分析扩展模型及算法（60575035），国家自然科学基金项目
3. 序列概念格扩展模型及其序列模式挖掘算法研究（08KJB520012）， 江苏省教育厅高校自然科学研究计划项目
4. 基于概念格模型的本体构建与运算，扬州大学信息工程学院研究生创新基金项目
5. 第三届江苏计算机大会（JSCC 2008）,2008.11.14~16,南京.
6. 2008 全国开放式分布与并行计算学术年会（DPCS2008）2008.10.25~26, 扬州
7. The 5th International Conference on Natural Computation(ICNC'09) and The 6th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD'09), Tianjin, China;
8. The 2009 International Conference on Web Information Systems and Mining (WISM’09) and the 2009 International Conference on Artificial Intelligence and Computational Intelligence (AICI'09). ShangHai, China;

扬州大学学位论文原创性声明和版权使用授权书

学位论文原创性声明

本人声明：所呈交的学位论文是在导师指导下独立进行研究工作所取得的研究成果。除文中已经标明引用的内容外，本论文不包含其他个人或集体已经发表的研究成果。对本文的研究做出贡献的个人和集体，均已在文中以明确方式标明。本声明的法律结果由本人承担。

学位论文作者签名：

签字日期： **2010** 年 月 日

学位论文版权使用授权书

本人完全了解学校有关保留、使用学位论文的规定，即：学校有权保留并向国家有关部门或机构送交学位论文的复印件和电子文档，允许论文被查阅和借阅。本人授权扬州大学可以将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、汇编学位论文。同时授权中国科学技术信息研究所将本学位论文收录到《中国学位论文全文数据库》，并通过网络向社会公众提供信息服务。

学位论文作者签名： 导师签名：

签字日期： 年 月 日 签字日期： 年 月 日

(本页为学位论文末页。如论文为密件可不授权，但论文原创必须声明。)