基于Web的问答系统答案抽取的研究

计算机应用技术

研究生 唐娟　指导教师 杜亚军

摘　要

随着 Internet 的高速发展，网上的信息越来越多，如何在海量的信息中快速准确的找到所需要的信息成为目前的一个研究热点。新一代搜索引擎—问答系统于互联网丰富，开放的信息资源中应运而生，实现更快速，更智能，更准确的获取用户所需的信息。问答系统（Question Answering,简称 QA）即是能利用信息抽取，信息检索，自然语言处理等相关技术，用准确、简洁的答案回答用户用自然语言提出的问题。它主要由三个部分组成：问题理解，信息检索，答案抽取。如何在问题理解阶段充分理解用户的提问意图，如何在信息检索模块中把相关的文档找出来，如何在答案抽取模块中准确地把答案从相关文档中抽取出来，这三个问题是问答技术的核心问题。

本文的研究内容是问答系统的答案抽取部分。利用形式概念分析对以下两个部分做相应的研究：从常问问答集中抽取答案；从 Web 中抽取答案。本文采用基于 Web 和语料相结合的多策略方法。针对问答系统的结构复杂性，提出使用 FCA（形式概念分析）来抽取答案。对于不同类型的问题，使用不同的抽取模式。利用概念匹配完成答案抽取，特别地，对于定义型问题，提出了协作推荐的方法。

本文首先使用了 FCA 来处理问答系统的答案抽取。在抽取处理中，首先在 FAQs 中寻找问题，如果该问题相应的答案不能满足用户的需要，再通过搜索引擎从网上获取相关的文档，从而使用返回的最相关的前 n 个文档建立概念格。最后，利用概念匹配的在格中抽取答案。对于不同的问题，本文使用了不同的抽取策略。

为了提高问答系统的精确度，本文提出了一个新的结合形式概念分析的概

念化聚类用户日志的方法。由于日志信息是每天变化的，本文改进聚类算法获得更好的性能。首先使用改进的基于 DBSCAN 聚类算法聚类用户的日志。其次，这些聚类被用来构建形式背景，从而试图根据问题/查询词的内容和文档的点击信息来处理相似性问题。这里，主要利用聚类来建立更符合用户需求的概念格。最后，本文提出使用导航技术从 FAQs 中抽取答案。

在信息获取方面，本文介绍了一种新的基于 FCA 的个性化的元搜索引擎，MySearch。它获取用户的信息，通过重排结果，提供一个实时的响应。重排是通过使用用户的使用日志和源搜索引擎返回的结果共同组建的概念格实现的。最后，改进的重排通过 MySearch 返回给用户。

对于定义型的问题，本文利用基于形式概念分析的协作推荐来回答定义型的问题。在协作推荐中，本文应用文档和问题之间的关系作推荐。FCA 组建文档和查询为概念，通过概念格来排序。

最后，本文介绍了基于概念匹配的答案抽取，概念聚类日志与 FAQs 评估，基于元搜索引擎信息获取，利用协作推荐回答定义型问题等四个核心模块的实验方法，步骤，结果及其评价。

关键字：问答系统；形式概念分析；答案抽取；聚类分析；数据挖掘

Answer exaction of Question Answering Based on Web

Computer Application Technology

Master Degree Candidate Juan Tang **Supervisor** Yajun Du

Abstract

Along with the high speed development of internet, the information on the web is more and more..It is a hot research that how to find out the needed information quickly and exactly from very large amount of information source in general domain. At present, the internet not only has the abundant information source, but also the information is opened. How to retrieve the needed information the most of quickly, the most of intelligence and the most of exactly based on the each user have very large amount of information source is the new generation of search engine to develop. Thereby, Question Answering is produced.

Question Answering depends on applying information retrieval, information extraction, and natural language processing (NLP) to answer for given domain independent questions written in natural language exactly and simply.

Question answering (QA) systems typically consist roughly of question analysis, document/passage retrieval, and answer selection. There are three core questions of question answering: how to comprehend enough at the stage of question analysis, how to find out the related documents from module of information retrieves, and how to exact the answers from the related documents in the module of answer exaction.

We are research on the answer exaction in this paper. We utilize formal concept analysis to research the following two parts: to exact answers from the Frequently Asked Questions and to exact answers from the web.

We propose a new approach of conceptual clustering the user queries logs with formal concept analysis (FCA). Due to the log data change daily, our methods could be received better performance by use our clustering method. We use our cluster algorithm that is based on the DSBCAN to cluster the user logs firstly. Then these clusters are established the formal context. We attempt to deal with similar questions/queries according to their contents as well as the document click information (cross-references) in the formal context. We mainly use them to build the better concept lattices. In addition, we proposed that navigation can be used to extract answers from the FAQs.

We use Multi-strategy based on the web and corpus. Due to the complexity of the structure in the Question Answering, we proposed that use FCA to exact answers. For the different questions, we use the different strategies to exact. We achieved the answer exaction based on the concept matching. Specially, for the definition questions, we utilize the Collaborative Recommenders to select answers.

This paper introduced a new personal meta-search engine MySearch, which is based on formal concept analysis. It extracts user's information implicitly and provides real-time response by re-ranking the results. Re-ranking is done by using concept lattice that is built by user's usage logs and the results of source engine. Lastly, the improve re-rank is returned by the MySearch. Experimental results show our method have a significant improvement on satisfaction degree between the search result and user's requirement.

Finally, we introduce the experiment results and the evaluation.

Keywords: Question Answering; Formal Concept Analysis; Answer Exaction; Cluster Analysis; Data Mining

声 明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作及取得的研究成果。除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得西华大学或其他教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示谢意。

本学位论文成果是本人在西华大学读书期间在导师指导下取得的，论文成果归西华大学所有，特此声明。

作者签名 [illegible] 2017年 6月 1日

导师签名 [illegible] 2017 年 6月 1 日

第1章 引言

随着 Internet 的高速发展，网上的信息越来越多，如何在海量的信息中快速准确的找到所需要的信息成为目前的一个研究热点。目前的搜索引擎（Google[1], Yahoo[3], 百度[2], 中搜[4]等）已经取得了很大的成就，但是这些搜索引擎存在了一些不足：

1. 目前，搜索引擎主要是被设计用来获取与用户查询请求相关的文档，其查询的序列一般是一系列的关键字的组合，而不是自然语言形式提供的，用户不能仅仅通过几个关键字清楚的表达自己的检索意图，搜索引擎也就没有办法找到最符合用户心意的答案；

2. 搜索引擎返回的也是与关键字内容相关的网页(即关键字字符与网页字符相关)列表，并没有涉及到语义，检索的质量较差；

3. 另外，在返回的结果中只有一小部分是用户需要的，用户还必须从这些结果中自己寻找相关的具体信息。

目前，互联网不仅拥有丰富的信息资源，而且信息是开放的，每个用户都可以通过互联网访问这个海量的信息，因此如何最快速，最智能，最准确的获取自己所需的信息就成为了新一代搜索引擎的发展方向，问答系统便由此产生。

1.1 问答系统研究动态

随着网络和信息技术的快速发展，同时人们想更快的获取信息的愿望促进了自动问答系统技术的发展。最近越来越多的公司和科研院所参与了自动问答技术的研究。比如：微软和 IBM 等著名的跨国公司。

文本信息检索会议（TREC，Text Retrieval Conference, http://trec.nist.gov）是文本检索领域最为权威的组织，TREC1，TREC5，TREC6 等是世界公认的文本检索系统的标准化测试集。从 1999 年开始 TREC 推出了 TREC Qusetion Answering Track (http://trec.nist.gov/data/qa/html)，用于推动问答领域的工作。在 2000 年 10 月召开的 ACL2000 国际计算语言学学术会议上，有一个专题讨论会，题目是“Open Domain Question Answering”。自此，自动问答（Question

Answering Track）是最受关注的主题之一。越来越多的大学和科研机构参与了TREC 会议的 Question Answering Track。目前已有的问答系统：

(1) STAR：http://www.ai.mit.edu/projects/infolab/ 麻省理工（MIT）就开发出一个问答系统 STAR，从 1993 年开始发布在 Internet 上，可以回答一些有关地理，历史，文化，科技，娱乐等方面的简单问题。在 2000 年，MIT 开发的 START 是世界上最早基于 Web 的 QA 系统，返回段落或者句子。

(2) AnswerBus：http://misshoover.si.umich.edu/~zzheng/qa-new/ 是一个多语种的自动问答系统，它不仅可以回答英语的问题，还可以回答法语，德语，意大利语和葡萄牙语的问题，返回的是段落或者句子。

(3) AskJeeves：http://www.askjeeves.com 返回结果与普通的搜索引擎很相似，都是网页。特点是允许用户用自然语言句子提问，检索系统会自动分析用户的提问，然后通过人机交互方式，准确地辨识用户的意图，这样用户就能够充分表达他的检索需求。

(4)ASKMSR[5]：微软研究院研制开发，为了快速查找相关文档的能力，建立在 GOOGLE 搜索引擎之上，返回简短词语或短语。ASKMSR 是基于答案频率统计的问答系统。把答案用 Answer-Tiling 的方式组合，没有使用词性信息（但是，对中文来说，词性对答案很有用），简单使用出现频次和 N-GRAM 模型来匹配答案，N-GRAM 即是考虑前面 n-1 个词构成历史的模型，它有助于英文短语的提取。

(5)WENIWEN：http://www.weniwen.com 由 100 多个学生组织起来对 Internet 上的各个网页进行提问，这些提问被记录下来作为网页的索引，在实际使用时，如果用户的某个提问与作为索引的某些提问在语义上非常接近，那么就把与这些提问相连的网页返还给用户。但是它返回的结果仍然是网页，而不是真正的问题的直接答案。

另外，其他典型的问答系统还有 IONAUT，Webclopedia，MULAER 等等[6]。

自从文本检索会议（TREC）在 1999 年的 TREC—8 会议上引入了对问答系统的评测后，越来越多的基于自然语言的问答系统产生。问答系统进入一种积极的研究领域，由 TREC QA 上反应出，问答系统逐渐向问题多样化，问题复杂化，和评估的精确化方向发展。近年来，问答系统渗入学习机制，向多种语言，多领域发展。近期的研究也是集中在为问题获取答案，而不是获取与查询

词相关的文档或者最佳的匹配章节。问答系统的难点是应用信息检索，信息抽取，机器学习和自然语言处理等相关技术。

综上所述，国内外针对问答系统已经做了大量的研究，从理论和实践上都取得了一定的成果，但目前仍然处于研究的初期阶段，因此，有必要进行深入、系统的研究。

1.2 问答系统的研究简况

现有的问答系统所采用的方法主要包括：自然语言处理方法；冗余技术；基于频率统计；多策略方法。它的体系结构包括问题理解，信息检索，答案抽取。

1.2.1 问答系统的研究方法

自从文本检索会议（TREC）在1999年的TREC—8会议上引入了对问答系统的评测后，越来越多的基于自然语言的问答系统产生。问答系统进入一种积极的研究领域，由TREC QA上反应出，问答系统逐渐向问题多样化，问题复杂化，和评估的精确化方向发展[7]。由于管道结构成功的应用[8]，问答系统通过结合知识源，应用复杂的推理机制,逐渐开始处理各种类型的问题[9]，引入的外部数据源，比如：Web、百科全书、数据库等[10，11]。在元系统（meta-systems中也应用了多种代理和策略[12，13]。综上，问答系统的研究方法主要分为以下几类：

(1) 自然语言处理方法

TREC QA在近来的问答系统研究领域中，起了积极的推动作用。在最初的问答系统研究中，主要是基于事实的，短的问答，比如："Who killed Abraham Lincoln", "what was the length of the Wright brother's first flight?", "when did CNN begin broadcasting?"。问答系统使用NLP技术来标准化信息获取技术。系统使用IR技术来标识侯选段落，然后利用语言学详细的分析问题和匹配段落，以便找到明确的答案。在问答系统中使用了多种语言学方法，比如：词性标注（Part of Speech Tagging)；句子解析（Parsing)；命名实体抽取（Entity extraction)；

利用语义关系词典，如：WordNet[1]；使用相关语言学处理软件，如：LingPipe[2]，Minipar[3]等。由 Harabagiu 等人提出的 Falcon 系统便使用了语言学方法，在基准测试中证明具有极好的性能。在系统中，查询问题被解析为可识别重要实体，并提出一种可能的答案类型。一种应用词汇—语义资源的答案类型在 WordNet 中发展起来[14]。WordNet 在同义词集中描绘了大于 10 万的英语名词，动词，形容词和副词，并通过词典编纂者进行编码。侯选的匹配段落如果能与答案的类型匹配，那么它将会做相似的处理。常见的，相关的段落中含有的单词与查询中的单词不一样。在这类情况下，Falcon 系统使用 WordNet 来检查词性的可选性，词汇的可选性（例如：名词“killer”，“assassin”，“slayer”将会与动词“killed”匹配）和语义的可选性（例如：“cause the death of”）。另外，绑架的过程也会用来作为答案的选取依据，从而删除一些不正确的答案。

(2) 冗余技术

相对于这些丰富的自然语言方法，有的问答系统试图通过信息源中大量的冗余信息来解决匹配和抽取答案的问题。例如：AskMSR[15]。冗余信息，可以用来获得多样的，不同的提问方式的答案。目前，促进问答系统发展有两种趋势：第一，大量的信息源，信息越多，与问题越接近的答案越可能出现。例如：问题“Who killed Abraham Lincoln?”，本文可以在大量的冗余信息中，直接的获得“John Wilkes Booth killed Abraham Lincoln in Ford's theater.”。第二，就算没有直接的答案，冗余信息也可以通过频率统计比较容易的获得答案。有些研究者利用 Web 作为问答系统的信息源。例如：Mulder 系统[16]在很多方面与 AskMSR 相似。就每个问题，Mulder 向 Web 搜索引擎提交多个查询，然后从搜索引擎返回的结果中分析答案。Mulder 对问题进行解析，然后从获取的全文网页中识别侯选答案。Mulder 用了一个本地数据库来为每个侯选答案标识权重以抽取和选择答案。但是 Mulder 还没有被 TREC 查询评估，所以它的性能很难与其它的系统比较。Clarke 等人[17，18]在他们的问答系统中，对冗余的重要性进行了研究。他们发现，问答系统最好的评估权重的方法是同时使用统计频率（即

[1] WordNet：（Miller,1990）一个众所周知的英语语义关系词典，它覆盖了一大半的英语名词，动词，形容词，副词等，被广泛的应用在 NLP 和其它的 QA 系统中。

[2] LingPipe：由 Alias-I 开发，是一种开放式的 NLP 资源软件。它提供语言学上的分析处理功能，如：句子的发现，命名实体发现等。

[3] Minipar：由 Dekangling 博士于 1994 年开发，提供 NLP 功能，比如：词性标注，句子解析，命名实体分类等。

冗余）和一个标准权重。而且，他们发现分析更多的高权重的段落是很有帮助的。他们的系统为所有搜集的文档建立一个全文索引作为 TREC 测试搜集。它们的实现要求一个可靠的辅助性的 Web 文集。Kwok[16]和 Clarke[17，18]为问题和最佳匹配网页实现复杂的句子解析和命名实体的提取，这样可以控制它们，从而详细分析网页数量。他们同样在选取或者排序最佳匹配段落中要求辅助的数据结构。AskMSR 不同之处在于它的简单性和有效性。该系统只使用了简单的重写和句子匹配，且在查询到 Web 资源时直接的使用了摘要和简单的队列返回。这种数据驱动技术在 TREC 基准测试中表现出很好的性能[19]。

(3) 基于频率统计

基于统计学习的答案提取技术已经取得一定的成功。一个统计的 QA 系统[20]使用了极大熵来实现答案的正确性分类。[8]使用了简单的模式，采用直接的匹配来回答相关类别的问题。[21]提出一种噪声—引导方法（noisy-channel）来抽取答案，根据统计学的特点，来适应知识源的使用。使用同样的方法[22]提出一种完全的基于实例的，易于训练的统计方法。

(4) 多策略方法

近年来，问答系统出现多策略方法。[23]结合两种端到端的问答系统：基于知识的系统和基于统计的系统，在解析，获取和产生答案阶段共享中间结果。MITRE 公司[24]在 TREC2002 结合使用基于距离和单词及其性质的 n-grams 方法实现问答系统的标准输出。虽然目前没有更好的组合比这个组合更有效，但是，他们仍然建议使用几种系统结合的方法来使输出表现出更高的有效性，更多的训练数据集和更好的特征。基于组件水平的多策略方法已经通过反馈循环和极限集的应用，使他们在问答过程中重复的实现。[25]利用预先设定极限值来决定查询的重写和另外获取数据集的必要性。近几年来，LCC[9]构建了一个系统，尝试着证明一个问题理论上来自一个答案，使用一种基于来自 WordNet 的公理萃取的推理机制来证明。当证明失败后，系统增加了松弛条件。AskMSR-DT[11]是一个基于 Web 的问答系统，探测使用了决策树来为动态产生的查询重写策略提供答案质量预测。这个方法，即是使用质量预测作为一个启发式的规则来选择重写的顺序和提供重写的极限值。

1.2.2 问答系统体系结构

随着与多种自然语言处理工具与知识源的结合，问答系统的结构逐渐复杂

起来。传统的问答系统（如图 1.1）由三个部分组成：问题理解，信息检索，答案抽取。如何在问题理解阶段充分理解用户的提问意图，如何在信息检索模块中把相关的文档找出来，如何在答案抽取模块中准确地把答案从相关文档中抽取出来，这三个问题是问答技术的核心问题。

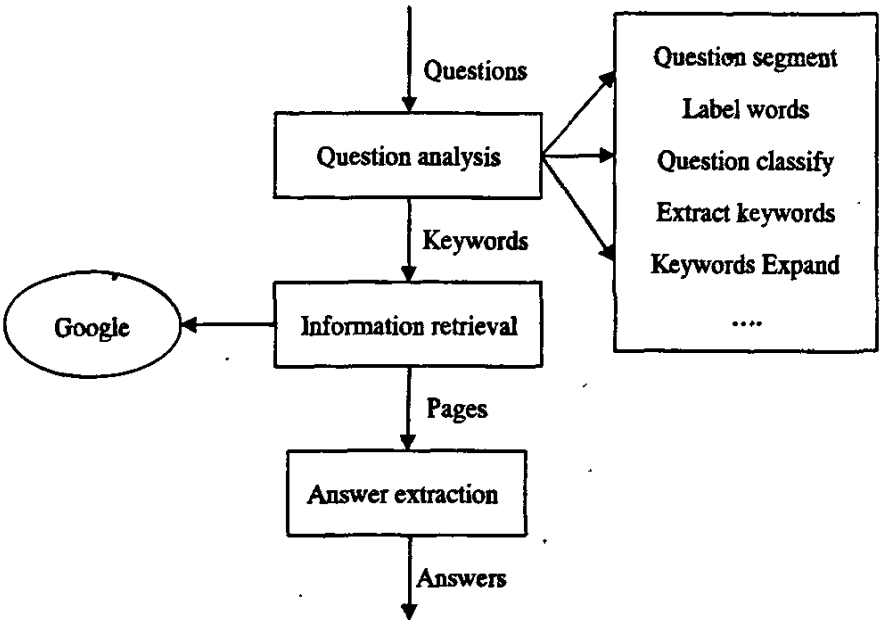

Figure1.1 The Structure of QA System

图 1.1 问答系统结构

(1) 问题理解

问题的分析是问答系统首先进行地工作，它的功能是将用自然语言描述的问题转化成一系列代表了问题语义的关键词。主要包括以下几部分工作：对问题进行分词以及词性标注，确定问题的类型，提取出问题的关键词，依据问题的类型等因素对关键词进行适当的扩展。

1. 问题分类

对不同类型的问题，将会用不同的处理方法。一般的问答系统都按照疑问短语来对问题进行分类。它的目的就是为了帮助用户缩小可能答案的范围。通过对大量问题的统计分析发现，用户提出的问题大概可以分为以下类型：（如图二）

2. 关键字提取

本文需要在用户的提问中，提取对后面检索系统有用的关键字，它不仅能

说明问题的主要内容，使问题意义清楚，而且突出强调了答案的类型。比如，

表 1.1 常见的问题类型

Table 1.1 Normal Types of Question

Type of question	question
Ask people	who
Ask time	What time/which year/....
Ask number	How many/how large/ how long....
Ask definition	What....
Ask location	Where ...
Ask reason	Why....
Others	—

疑问词和一些常用的“吧，了，的”等词就应该被过滤掉，为此，需要建立一个停用词表来过滤这些词。关键词主要由名词，动词，形容词，限定性副词等组成，它们将被赋予不同的权重，在检索段落时使用它们来计算段落的权重。

关键词被分为两种：一般性关键词和“必须含有”性关键词。所谓“必须含有”性关键词指的就是该关键词在答案段落中必须含有，主要由专用名词，限定性副词（如：最大，最高，最远，最快，第一等）、时间、数词等组成。而一般性关键词可以不被答案段落所包含。通常情况下，名词，“必须含有”性关键词，具有限定性作用的副词会有比较高的权重。

3. 关键词扩展

信息检索评价的标准是信息检索的精度和召回率。信息检索的精度为检索结果中相关信息文档与查询结果总数之比。信息检索的召回率为实际检索出的相关信息文档数与信息库中总的文档数之比。为了提高检索系统的召回率，需要对关键词进行扩展。在答案句子中某些词常常不是原来问题的关键词，而是这些关键词的同义词。对关键词进行扩展，可以提高关键词的查询成功率。对关键词的扩展主要采用以下方式：（1）名词同义词扩展和语义蕴涵扩展；（2）动词同义词扩展（减少出现歧义的情况）；（3）根据问题类型扩展（如：询问数量类型可以把一些表示数量的单位加入）。

(2) 信息检索

信息检索就是用前面提取出来的关键词，使用搜索引擎（如：google）来查找相关的文档并返回一些最相关的文档。系统实际分为两个阶段进行检索：文献检索和段落检索。

1. 文献检索

文献检索主要使用信息检索技术，根据问题处理部分产生的关键词分类进行查询递交出符合关键词的文献（一般的，取前 100 篇），并可以根据文献相关性对它们进行排序。

2. 段落检索

从文献检索中的文献中选取最相关的段落（一般的，取前 100 个段落），采用段落过滤方法，用尽可能少的摘录覆盖尽可能多的问题内容。

(3) 答案抽取

问答系统需要返回的是简短的，准确性高的答案。这样，通过信息检索模块搜索出来的相关文档就要提交给答案抽取模块来提炼答案。在答案抽取部分为了找到答案，系统主要依据关键词，命名实体，模式匹配。对不同的问题，系统确定不同的答案模式。答案可以是一句话，或者是几句话，也可以是几个词或短语。对于那些问时间地点的问题，就可以用很短的语句回答，而对于询问原因，事件的问题就需要较长的句子才能回答（比如：为什么，怎么样，等类型的问题）。

1. 以句子作为答案

此类答案的抽取步骤如下：

(1) 把检索出来的文档分成句子；

(2) 按照一定的算法，计算每个句子的权重；（需要考虑：句子中含有的关键词，和关键词有相同语义的词，句子中不包含的关键词以及侯选句子与问题之间的语义相似度。）

(3) 对句子按照权重进行排序；

(4) 根据问题的类型对侯选答案重新排序。

2. 以词或短语作为答案

对于问题类型为时间或者地点类的问题，答案就比较简短。所以希望直接把包含答案的词或短语提取出来。但是，目前很难准确的把答案抽取出来。

3. 以文摘作为答案

对于有些问题，简短的一个短语或者一句话很难说清楚，比如："地球是怎样形成的？"。这类问题，本文需要采用多文档自动文摘技术，它把信息检索模

块检索出来的相关文档做成文摘，再把这个文摘作为答案返回给用户。

1.2.3 问答系统评价方法

评价方法:

1)首先，需要建立一个测试集，可以是人工做出来的问题和答案对的集合。把这个测试集中的问题提交给问答系统，让问答系统自动的给出答案。然后把问答系统自动找出的答案和测试集中的答案，进行人工的对比。如果问答系统给出的答案通过人工的对比基本正确，则可以判断这个答案是正确的，否则可以判断这个答案是错误的。根据公式（1）可以计算出问答系统的准确率。

准确率=答对的问题数/ 问题总数 (1)

2)TREC 会议每年都会提供一个测试集，让参加 TREC 的研究人员评价自己的问答系统。TREC 允许对每个问题给出 5 个答案。如果问答系统返回的第一个答案是对的，那么这个问题就得 5 分，如果问答系统返回的第二个答案是对的，那么这个问题就得 4 分，如果问答系统返回的第三个答案是对的，那么这个问题就得 3 分，如果问答系统返回的第四个答案是对的，那么这个问题就得 2 分，如果问答系统返回的第五个答案是对的，那么这个问题就得 1 分。把每个问题所得得分加起来就可以得到问答系统所得总分。总分越高，该系统得到准确率越高。

1.3 形式概念分析

概念被理解为由外延和内涵两个部分所组成的思想单元。基于概念的这一哲学理解，德国的 Wille 教授提出了形式概念分析[26],[27],用于概念的发现，排序和显示。在形式概念分析中，概念的外延被理解为属于这个概念的所有对象的集合，而内涵则被认为是所有这些对象所共的的特征（或属性）集，这实现了对概念的哲学理解的形式化。而概念格作为形式概念分析中核心的数据结构，本质上描述了对象和特征之间的联系，表明了概念之间的泛化和例化关系，其相应的 Hasse 图则实现了对数据的可视化。作为序论和格论与实际应用结合的产物，概念格模型的研究具有重要的理论意义。

1.3.1 形式概念分析的理论基础

现实世界是由各种各样的对象组成，每个对象有自己的一组属性或者特征，概念格结构是反映对象与属性之间联系以及概念泛化与例化关系的一种完备的概念层次结构。While 于 1982 年提出[26]由二元关系来建造相应的概念格的基本思想，这种特殊形式的格结构及其相应的 Hasse 图就反映了一种概念层次结构。其建立在一个形式背景上，为研究信息系统的概念或知识提供了有力的工具。下面给出概念格的形式化描述。

定义 1 [26]一个形式背景(formal context)是一个三元组$T=(O,D,R)$，其中 O 是对象集合，D 是特征集合，R 是 O 和 D 之间的二元关系，即$R\subseteq O\times D$。其中 oRd 表示$o\in O$与$d\in D$之间存在关系 R。形式背景可以用一个数据表来表示，它描述了对象及其特征之间的自然分组和关系的有序集。

定义 2 [26]在形式背景中，对于对象集$A\in P(D)$和特征集$B\in P(O)$可以定义下面的两个函数 f 和 g:

$$B=f(A)=\{y\in D|\forall x\in A,xRy\}$$

$$A=g(B)=\{x\in O|\forall y\in B,xRy\}$$

简记为$A'=B$，$B'=A$通常称函数 f 和 g 为 D 的幂集 P(D)和 O 的幂集 P(O)之间的 Galois 连接。定义从形式背景中得到的每一个满足 B=f(A)和 A=g(B)二元组(A, B)为一个形式概念(formal concept)。其中 A 称为概念的外延(extent) B 称为概念的内涵(intent)。

显然，概念的内涵是概念外延中所有对象的共同属性的集合，而概念的外延是概念内涵可以确定的最大的对象集合，一个概念是一个完备的二元组。

定义 3 在概念节点之间能够建立起一种偏序关系。对于给定$C_1=(A_1,B_1)$和$C_2=(A_2,B_2)$，则$C_1>C_2\Leftrightarrow B_1\subset B_2\Leftrightarrow A_1\supset A_2$，领先次序意味着 C_1是 C_2的父节点或称泛化。若概念$C_1=(A_1,B_1)$和$C_2=(A_2,B_2)$)满足$A_2\subset A_1$，且不存在概念(A, B)使得$A_2\subset A\subset A_1$，则称 C_1是 C_2的直接超概念，C_2是 C_1的直接子概念，记为$(A_1,B_1)>(A_2,B_2)$。

根据偏序关系可生成概念格的 Hasse 图。如果有 $C_1>C_2$，在 Hasse 图中将存在一条边从 C_1到 C_2，C_1是 C_2的直接超概念，C_2是 C_1的直接子概念，形式背景 T=(O,D,R)中，满足直接子概念—超概念关系的所有概念节点的集合是一个完全格称之为 Galois 概念格，简记为概念格。

1.3.2 概念格的构造算法概述

在概念格的应用过程中，建格算法具有很重要的地位。概念格所具有的完备性是不受数据或属性排列次序的影响，不同的构造方法所生成的格形式是唯一的，完备性一方面是概念格的优点之一，另一方面即使对于适当大小的数据，也将产生庞大的格结构。理论上说，概念的节点个数会以形式背景对象个数和属性个数的指数倍增长。所以概念格的构造过程十分耗时和非常关键。

从构造方法的角度，概念格的建造算法分为三类：批生成算法，渐进式算法和并行算法。目前有关概念格生成的并行算法（parallel algorithm）研究不多，并行算法通常是以其他算法原理为基础提出。可以预测对于海量数据的概念格的生成和应用研究，并行算法必然是一个大的趋势。建立 Galois 格的过程实际上就是一个概念聚类的过程，因为它产生一种概念层次结构。迄今为止，国内外学者已经提出了许多不同的算法来构建概念格，但只有少数在构造的同时生成相应的 Hasse 图。

(1) 渐进式算法（incremental algorithm）

其基本的思想是假设形式背景中的前 i 个对象已经生成了概念节点的子集和概念格的子格，当第 i+1 号对象加入时，去更新原有的概念节点子集和概念格的子格。重复这一步骤，直到生成最终的格结构。典型的算法有 Godin 算法[28]和 Carpinet[29]算法。这些策略很容易应用于属性上，可以依据属性而不是对象来渐进式生成概念格。在渐进式生成概念格的求解过程中，将当前要插入的对象插入概念格图表中时，通常需要考虑三个问题：所有新节点的生成；避免已有的概念节点的重复生成；Hasse 图表的更新。

Godin 算法[28]采用渐进式算法来创建概念格，构造的关键在于找出所产生的子节点。其基本思想是在插入一个新对象之后，将格中的节点分为三类：不变节点。它们是新格 L'中所保留的格 L 中的节点，这些节点的内涵和新对象的内涵没有交集；更新节点。它们是对原来格 L 中的节点更新后的节点，这些节点的内涵包含在新对象的内涵中，因此只需将其外延更新包括新对象即可；新增节点。即所要插入的节点的内涵与原来格 L 中某个节点的内涵的交所产生的而在格 L 中没有出现过的集合。可以证明，新节点的父节点必然是某个新增节点或者更新节点。Carpineto 算法和 Godin 算法的主要不同点出现在连接过程中，Carpineto 的做法是找到新节点的最小上界和最大下界，删除它们之间的边，

并将其连接到新概念。

(2) 批生成算法（batch algorithm）：

主要完成两项任务:第一，生成所有概念节点集合，即形式概念的集合；第二，建立概念节点之间的直接前驱和直接后继关系。常见的有两种情况，其一是首先生成所有的概念节点的集合，再生成概念的图表结构；其二是每次生成概念节点的一小部分，同时将这些节点链接到概念格图表结构中。

目前已提出许多生成概念格的批生成算法。但是其中多数是仅仅生成概念节点的集合只有少数同时生成 Hasse 图。批生成算法根据其构造格的不同方式，可分为两类，即自顶向下算法和自底向上算法。自顶向下算法首先构造格的最上层节点即外延最大的概念节点，再逐渐往下生成较小外延概念。自底向上算法则相反，首先构造底部的外延最小的概念节点，逐步向上扩展生成较大外延的概念，如 Chein 算法。另外，本文也可以从属性的角度，依据属性和概念的内涵而不是外延来自顶向下或者自底向上的生成概念。

Bordat[70]在 1986 年提出的批处理算法采用的是自顶向下的策略，其基本思想是从格的最大下界概念（E，f（E））开始，并生成它的所有子概念，然后把这些概念添加到 GCL 中，并与其超概念连接，这一子概念生成过程对每个概念重复进行。Ganter 算法使用表示 X 集的特征向量来枚举格的 X 集，每个向量的长度是属性集的基数。若某个属性值在该向量中出现，则相应位置 1，否则为 0。给定完全对的向量 $X=\{x_1, x_2, \ldots, x_m\}$，它找到作为完全对一部分的下一个 X 向量（关于向量的字典序）。查找方式是按顺序将属性位置 1，并测试它是否是完全对来进行的。以此方式产生的向量的有序列表是按照 X 集的包含顺序拓扑排序的。基于 Bordat 的批处理算法[70]，Njiwoua 等提出了构造概念格的并行算法，文中给出了算法的正确性证明并研究了该算法的理论复杂性。

1.3.3 概念格的应用

作为近来引起广泛关注的一种数据分析和知识处理的形式化工具，概念格目前已经在机器学习，信息检索，数字图书馆，软件工程，知识分类和数据挖掘等领域显示出来一定的应用价值。作为数据分析和知识处理的形式化工具，形式概念分析已经获得了广泛而成功的应用。在软件工程领域，形式概念分析为再工程，软件重用，面向对象程序设计等领域中某些问题的解决提供了理论

支持，并已经取得了一系列的应用成果。随着计算机和数据库技术的发展以及各种电子设备的大量使用，人类收集数据的能力得到了极大的增强，数据信息日益膨胀，但是堆积如山的积累数据对于人类是难以处理的，真正有价值的是埋藏于数据中的知识，因此数据挖掘技术已经得到了广泛的研究。而形式概念分析以概念格的形式使数据有机的组织起来，概念格节点体现了概念内涵和外延的统一，因此非常适合于用来发现规则型知识。已有不少学者讨论了从概念格上提取规则或函数依赖的问题。和其它分类器相比，概念格上提取的规则具有相当或更好的分类效果[7][8][9]。

而在信息检索方面，Neuss 和 Kent[30]使用概念格进行 Internet 上文档信息的自动分类和分析。Cole 和 Eklund 将概念格方法应用于分析和可视化具有 1962 个属性和 4000 个处方摘要的医药数据库。Eklund 和 Martin[31]展示了概念层次进行 Web 文档索引和导航的能力。Lengnink[32]将 VSM 模型中的相似度和距离概念转换到格模型中，这样传统信息检索模型中的自动化的方法(自动分类，聚类)能应用到格模型中。Carpineto 和 Romano[33]将查询插入文档集的概念格中，用最短路径的方法来计算文档相似度，以解决词汇不匹配问题。

联机分析处理（OLAP）依赖于一个包含有数据的多维立方体，当维不是层次化结构的时候，用立方体作比喻为多位数据提供了一种良好的直观理解。但是 OLAP 维的一个重要的特征在于它们是层次排序的。通常它们是树，但也可能是任意的偏序集。由于立方体的比喻反映的是线形向量空间的直积的数学意义，在这种情况下，层次必须被强制转化为简单的线形形式，因此立方体并不适合用于表示具有层次维的数据。

概念信息系统由一个关系数据库和多个概念层次（概念标尺）组成，这些层次被用于支持对数据的导航。以概念信息系统为基础，文[1]通过使用嵌套线图来对概念层次的任意组合进行可视化。由于层次维大致地对应于概念标尺，因此 OLAP 分析工具可以被大致看作是一种特殊的概念信息系统。嵌套的线图被用于绘制维的直积，从而替代了在 OLAP 中被广泛使用的嵌套图。与传统的嵌套图相比，嵌套的线图占用了较多的空间，它的优点在于沿着最重要的维（被选为最外层的层次）的清晰的结构化，因此非常便于阅读数据，并且允许使用比 OLAP 中通常使用的树状维更加复杂的标尺（任意的偏序集）。为了实现对数据立方中数据进行存取的四种交互方法：切片，旋转，Drill-Down 和 Drill-Up，另外还研究了与这些操作相对应的嵌套线图的相应动作。

概念格还获得了其它的一些应用，比如，Wille 在[34]中将概念图和形式概念分析结合起来，从而得到了对初等逻辑的一种形式化，这对于知识表示和处理是非常有用的。Richards Debbie[35]利用概念格对 Ripple Down Rule 进行有机的组织；Cole 的 CEM 电子邮件管理系统[36]通过将 Email 存储在概念格中，而不是常用的树状结构中，从而在检索电子邮件时获得了更大的灵活性；文[5]则将概念格应用于智能帮助系统的领域建模；等等。

1.4 本文的主要研究内容

问答系统主要由三个部分组成：问题理解，信息检索，答案抽取。如何在问题理解阶段充分理解用户的提问意图，如何在信息检索模块中把相关的文档找出来，如何在答案抽取模块中准确地把答案从相关文档中抽取出来，这三个问题是问答技术的核心问题。

本文主要内容是对问答系统的答案抽取部分的研究。利用形式概念分析重点对以下部分做相应的研究：从常问问答集中抽取答案；和从 Web 中抽取答案等问题进行研究。

1、从常问问答集 (frequently asked questions, FAQs)中抽取答案

FAQs，即常问问题集。作为本文问答系统的一个组成部分，把用户经常提问的问题和答案保存起来。对于用户输入的问题，首先在常问问题集中查找，如果能够找到相应的问题，就可以直接将相应的答案返回给用户，而不需要经过问题理解，信息检索，答案抽取等相关的复杂的处理过程，这样不仅可以提高效率，而且也能提高精度。但是其中需要解决两个问题：人类编辑如何确定哪些问题/查询是常问问题，即那些是 FAQs？另外，一个系统如何判断两个问题/查询词是相似的？

本文采用的方法是使用一种概念化的查询聚类的方法，使用用户的日志信息来识别用户使用过的文档信息。对于计算推导两个查询词的相似度将使用查询词和交叉参照两个标准。

Principle 1（using query contents）：如果两个查询包含相同的或相似的项，它们表示相同或相似的信息需要。

显然，查询越长，原理 1 就越可靠。但是，在查询短的情况下，光使用这个原理是不够的。因此，第二个标准作为原理的补充。

Principle 2 (using document clicks)：如果两个查询导致选择了相同或相似的文档，那么它们是相似的。

而聚类处理的算法，考虑日志信息本身的特点：

—查询日志经常都是很大的，该算法应该有能力处理大的数据集，而且存在合理的时间和空间的约束。

—算法应该不需要人工的设置聚类的形式，例如，聚类的数量或者最大聚类的大小。根据本文的研究对象—庞大的日志，事先就要确定好聚类数量等参数是不切实际的。

—对于实际的日志信息，本文只要求找到出现频率高的 FAQs，对于低频信息，本文暂不关注。所以算法应该可以过滤低频率的查询。

—由于实际的日志数据是时刻变化的，算法必须是增量的。

所以，本文采用基于密度的聚类方法 DBSCAN 和它的增量的版本增量的 DBSCAN 作为基础算法，另外改进它，从而可以直接产生出常问问题类，方便抽取精确度更高的答案。

2、基于 Web 的答案抽取

采用基于 Web 和语料相结合的多策略方法。针对问答系统的结构复杂性，提出使用 FCA（形式概念分析）来抽取答案。对于不同类型的问题，使用不同的抽取模式。首先，从语料数据库中（即 FAQs）搜索与用户问题相关的问题，如果没有或者答案不满足用户的需求，再通过搜索引擎从 WEB 中搜索相关的文档；然后使用搜索引擎返回的前 n 个文档，构建概念格；最后，利用概念匹配来抽取答案。这里，本文针对不同类型的问题，使用不同的抽取模式。

抽取处理步骤：

1) 文档概念格的建立；

使用搜索引擎返回的文档作为形式背景，文档作为对象，描述文档的短语作为属性，文档与短语之间的关系作为值（布尔值）。主要涉及的问题：短语的选取。[37]

本文的步骤：

（1）在文档中选取所有出现的侯选短语；

（2）在侯选短语中根据特殊查询平衡策略（Query Specific Balanced strategy）[37]选取能更好描述文档的短语（得分高）。特殊查询平衡策略的特点是利用短语本身的特点和它在文档中出现的频率进行打分。

2)问题概念格的建立；对于一个用户提出的问题，本文主要是发现问题的主题和问题的焦点。这里，将问题的主题作为对象，焦点作为属性，进行概念格的构建。另外，考虑问题类型对答案的影响很大，所以，考虑将问题的类型也作为属性。

3)概念匹配：为每对问题-答案对赋权值，再利用相似度的计算方法作概念匹配。余弦相似度的计算方法如下 [38]：

$$similarity_{w\text{-}keyword}(p,q)=\frac{\sum_{i=1}^{k} cw_i(p)\times cw_i(q)}{\sqrt{\sum_{i=1}^{m} w_i^2(p)}\times\sqrt{\sum_{i=1}^{n} w_i^2(q)}}$$

Factoid questions：在答案选取过程中，使用单元匹配（object-attribute对）；利用相似度的计算方法，选择最相关（匹配的概念最多）的短语作为答案。

List Questions：使用单元匹配，但是选择所有匹配的概念作为答案。这里需要选择一个相似度的阈值来判断，该答案是否为 List Questions 的答案。

Definition Questions：使用单元匹配和关键字匹配（又称 Partial matching）相结合来获取答案选择所有的单元匹配和某些关键字匹配作为答案。这里，需要为关键字匹配选择一个支持度，即至少答案中包含问题主题和问题焦点两个内容或相似的内容。

1.5 论文结构

本文将结合形式概念分析的方法对基于 Web 的自然语言问答系统 Myaskanswer 进行设计与实现，重点阐述了系统中聚类用户日志形成 FAQs，利用概念匹配抽取答案，特别地，对于定义型问题，使用了基于形式概念的协作推荐方法来获取答案。以下的章节组织如下：

第二章：Myaskanswer 的总体设计。本章使用了 FCA 来处理问答系统的答案抽取。在抽取处理中，本文首先在 FAQs 中寻找问题，如果该问题相应的答案不能满足用户的需要，再通过搜索引擎从网上获取相关的文档，从而使用返回的最相关的前 n 个文档建立概念格。最后，利用概念匹配的在格中抽取答案。对于不同的问题，本文使用了不同的抽取策略。

第三章：概念聚类的方法，处理用户日志，智能化的构建 FAQs。在本章，采用了概念化的聚类用户的日志来构建 FAQs。这是本文问答系统的一个部分。

本文使用用户的日志构建形式背景，试图根据查询的内容信息和文档点击来聚类相似的问题/查询。假设查询/问题和点击文档之间存在相关性。本文的方法基于两个原则：（1）如果两个查询导致了相同的文档点击，就被认为他们是相似的或者相关的，从而合并他们作为内涵；（2）如果一个文档集合被频繁的由相同的查询选取，就认为这些文档是相关的，从而合并他们作为外延。这两个原则与传统的查询内容相结合，从而构建初始的概念格。由于日志信息的特点，本文使用基于 DSBCAN 聚类算法来获取更好的性能。由于概念格的特点，在形成相关 FAQs 的概念格中，本文使用导航来为问题抽取答案。

第四章：Myaskanswer 中对 Definition Questions 型问题进行答案抽取的算法和实现。本章利用基于形式概念分析的协作推荐来回答定义型的问题。推荐是计算机系统设计来帮助用户在一个大面积的可选择领域中找到更适合的项。在协作推荐系统中，本文应用文档和问题之间的关系作推荐。FCA 组建文档和查询为概念，通过概念格来排序，从而获取定义型问题的答案。

第五章：实验。本文主要对三个模块进行了实验。包括基于概念匹配的答案抽取；概念聚类日志与 FAQs 评估；还有利用协作推荐回答定义型问题。

第 2 章 Myaskanswer 的总体设计

近年来，随着自然语言处理，信息获取技术，信息抽取等技术的发展，问答系统的结构逐渐复杂化。传统的问答系统采用了典型的管道方法（Pipeline approach），主要包括问题理解，信息获取，答案抽取三个部分。Myaskanswer 是一个基于语料和 WEB 的多策略方法，试图回答更复杂的问题。首先，本文对问题进行分析，将问题基本分为三类：仿真陈述型(Factoid questions)，列表型(List questions)，定义型(Definition questions)。对于不同的问题类型，利用形式概念分析为基础，利用不同的抽取策略来抽取答案。

2.1 系统结构

总体系统结构图，如图 2.1。对于用户的问题输入，本文首先提取问题的重点和焦点，扩展查询词；将扩展后的查询词与 FQAs 中的查询词匹配，形成问题的答案，即概念匹配模块。而对于 FQAs 的形成，文本主要是基于对用户查询日志的概念化聚类方法，即概念化聚类模块。根据对聚类后的查询词构建概念格，采用基于导航的答案抽取策略形成 FQAs。如果用户对从 FQAs 中直接匹配的答案不满意，本文再根据查询词选取初始 URL 集合，从 Web 上获取网页集合，其中，本文对于获取网页集做了优化，即对特定的查询词，选取特定的爬行方向，从而获取的网页更能满足用户的需要，即爬行方向选取模块。对于，从 Web 中获取的网页集合，本文构建相应的概念格，即概念格的构建模块。对于答案的抽取，主要采用了两种不同的方式进行答案抽取。一种是基于概念匹配的答案抽取，一种是基于协作推荐的答案抽取。后者，主要应用于定义型问题。

图 2.1 描述了整个的系统。而对于本文的研究工作，主要集中在以下模块：基于概念匹配的答案抽取，概念化的聚类，基于导航的答案抽取，网页集的获取，基于协作推荐的答案抽取。本章主要是对基于概念匹配的答案抽取的研究。第 3 章主要介绍概念化聚类和基于导航的答案抽取。第 4 章是对网页集获取的研究。第 5 章主要介绍基于协作推荐的答案抽取。

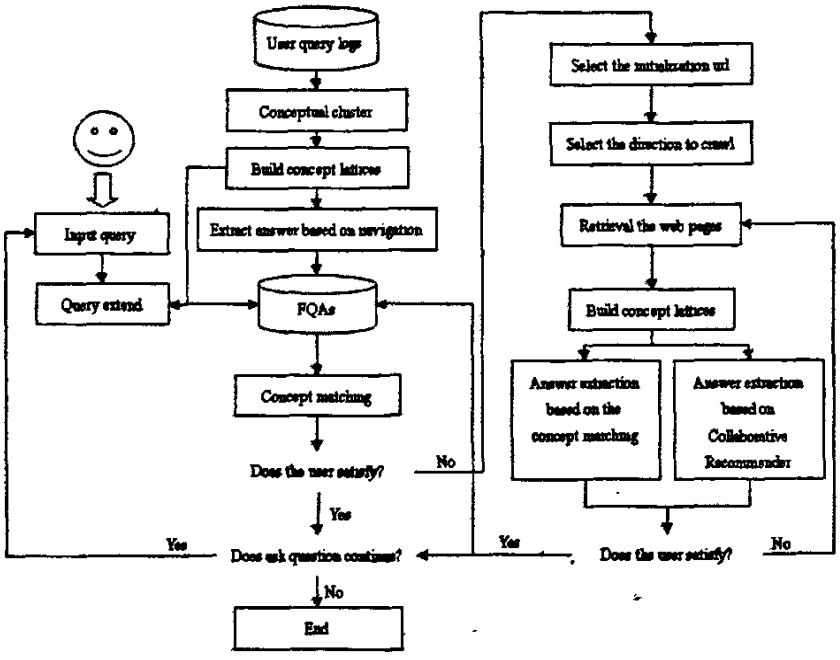

Figure 2.1 The whole structure of the system

图 2.1　系统的总体结构图

基于图 2.1 的系统，本章首先主要对概念匹配的答案抽取进行研究。本章

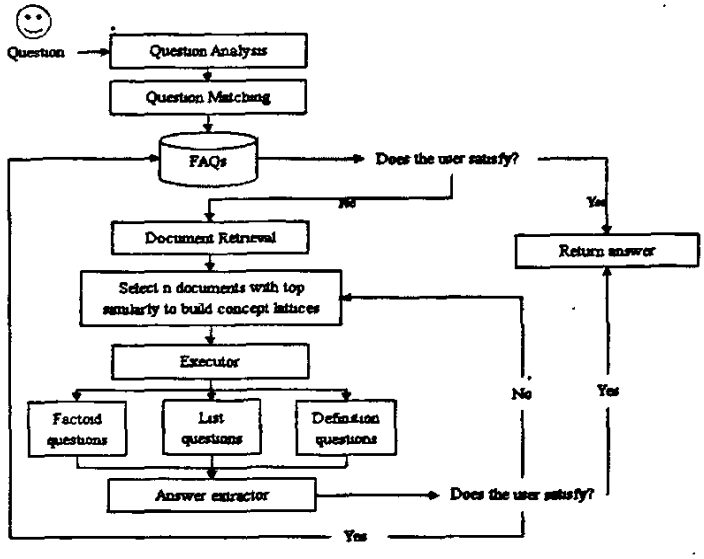

Figure 2.2 The structure of the paper research on

图 2.2 本章的研究结构图

的研究结构图，如图 2.2。首先，用户用自然语言输入一个问题，问题分析模块将首先处理这个问题，去掉停用词（stopwords），将查询项归类。这个模块的输出为问题表达式，它将被用到 FAQs 中的问题匹配和从 Web 中获取信息。本章使用问题的表达式和向量空间模型余弦值通过获取子模块得到相关的文档。下一步，本章使用获取的前 n 个最相关的文档形成形式背景，从而构建概念格。在执行子模块，本章分类问题，对不同的问题类型使用不用的识别方法来得到答案实体集。最后，在答案抽取模块中使用概念匹配来抽取答案。

这一章中，主要讨论对于不同的问题类型：仿真陈述型，列表型和定义型，使用的不同答案抽取方法。下面，本章将分别讨论这几种类型的问题的抽取方法。

2.2 答案抽取

一旦获取了初始的文档集，本文的问答系统将试图从这些文档中抽取答案。首先，本文选择相似度最高的文档集合构建概念格。本章自动化的选择名词短语作为文档的描述来建立基于 FCA 的信息获取框架。接下来，使用执行器识别问题的实体，将问题分为不同的问题类型，选择适合的答案抽取方案。最后，本章对于不同类型的问题，使用不同的抽取器来抽取答案。

2.2.1 构建概念格

(1) 信息组织模型

通过一个或者多个搜索引擎（如：google，Yahoo!, Netscape）获取一个有序文档列表。具体方法见第 4 章。本章组织这些搜索结果来组建一个概念格。

格的产生是基于一个形式背景 K=（G, M, I），这里 G={url_1, url_2,...,url_n}表示获取的 url 集，M={$keyword_1$, $keyword_2$,..., $keyword_m$ }表示相关这些 url 的关键字集，I 表示 G 和 M 的关系。

(2) 短语选择

对于名词短语的处理，本章主要包含两步：短语抽取，短语选择。

第一步： 短语抽取

短语抽取是为了发现所有可能的与文档相关且能描述文档的侯选短语，他

们与文档的域相关。这些短语将依次与术语表中的短语匹配。

第二步：选择策略

选择策略是为了选择能最好描述获取文档的短语。本章使用查询特殊平衡策略(Query Specific Balanced Strategy)来选择至少包含一个查询项的短语作为短语组成，他们都有最高的文档频率值作为文档的描述[37]。

2.2.2 问题概念节点

在本文中，主要围绕下面几种类型的问题进行研究。本文将问题划分为不同的类型，从而使用不同的处理策略和答案匹配样式：

[1] 仿真陈述型问题：How far is it from Earth to M? Is Bin Laden still alive?

[2] 列表问题：List names of cell phone manufact arsurers.

[3] 定义型问题：Who is Vlad the Impaler?

有了对问题类型的划分，本文从中找到问题的主题和问题的焦点，从而构建问题概念结点。问题的主题是指关于问题的对象或者事件。问题的焦点是指由问题找到的相关性质和实体。

例如：**In what state$_F$ is the Grand Canyon$_T$?**

What is the population$_F$ of Bulgaria$_T$?

What color$_F$ is a pomegranate$_T$?

这里，**state$_F$** 是指 **state** 是问题的焦点，**Grand Canyon$_T$** 是指 **Grand Canyon** 是问题的主题；**population$_F$** 是指 **population** 是问题的焦点，**Bulgaria$_T$** 是指 **Bulgaria** 是问题的主题；同理，**color$_F$** 是指 **color** 是问题的焦点，**pomegranate$_T$** 是指 **pomegranate** 是问题的主题。另外，本文还需确定问题的相应的答案类型。

例如：PERSON: who...?

LOCATION: where...?

DATE: when...?

NUMBER: How many...?

...

因此，问题格通过问题的主题，问题的焦点，问题的类型综合形成。本章用问题的主题作为对象集合，用问题的焦点和问题的类型作为属性集合。这样可以确保答案的精确率。

例如：

问题：how many **people**$_F$ in **china**$_T$?

其中，**people** 是指该问题的焦点，**china** 是指该问题的主题，**how many** 是指相关答案的类型。根据问题概念格的规定，相关的问题概念节点为：{China}→{people, number}.

2.2.3 使用概念匹配来抽取答案

与信息获取（IR）系统相似，本章的抽取策略需要对每个问题-答案对获取状态值（Retrieval Status Value）计算相似度，从而表示对于一个问题，答案的相似性和适当性。本章使用两个概念的局部匹配，将两个概念共同的对象集合作为新概念的对象集（外延），将两个概念共同拥有的属性作为新概念的属性集（内涵）。下面的举例说明了一个问题和文档之间的节点匹配情况。

形式背景如表 2.1，这个背景表示输入一个问题后，获取的与问题相关的文档集合。这个形式背景相关的概念格如图 2.3。这里，为了方便，将获取的文档集合{1, 2, 3, 4}表示成 1234，而文档集合的描述{a,b,c}表示成 abc。图 2.4 表示问题格。注：本文这里忽略问题格的其它节点。

表 2.1 形式背景

Table 2.1 The Formal Context

	a	b	c	d	e	f	g
1	×			×			×
2	×			×			×
3	×			×		×	
4	×			×		×	
5			×		×	×	
6			×		×	×	
7		×			×	×	
8		×			×	×	
9	×				×	×	

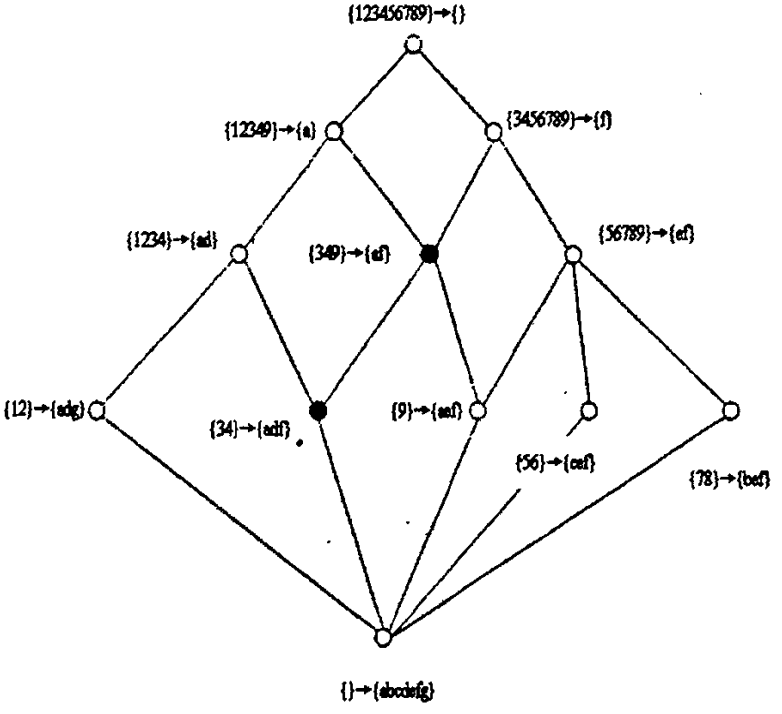

Figure 2.3 The Concept Lattice of Formal Context

图 2.3 该形式背景对应的概念格

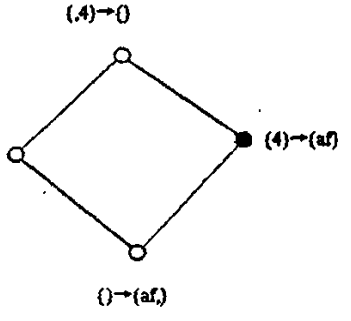

Figure 2.4 The Question Lattice

图 2.4 问题格

(1) 仿真陈述型问题

仿真陈述型问题，代表问答系统的一个方面，它的目的就是为人类提供使用自然语言访问信息的直接信息。

利用前面的研究，本文为仿真陈述型问题使用单个的对象-属性对（单元概念）来匹配。如果问题和文档概念相匹配的单元概念越多，本文认为这两个概

念越相似，从而获取答案。本文使用统计的方法来为答案识别最高的评分，这样也可以避免“噪声”（“噪声”即是指一些例外，比如，不相关的点击获取的文档，等等）。

图 2.5 举例说明使用形式概念分析对仿真陈述型问题的抽取策略。这里存在两个匹配，粗体表示关键字。根据本文的抽取答案的策略，这里将选择后者作为答案，因为它相关的单元概念比第一个多。

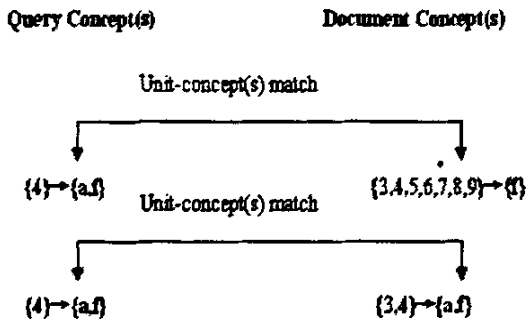

Figure 2.5 The Extracting Strategy of Factoid Question

图 2.5 仿真陈述型问题的抽取策略

(2) 列表型问题

对于回答列表型问题，本文同样使用单个的对象-属性对来匹配。但是本文

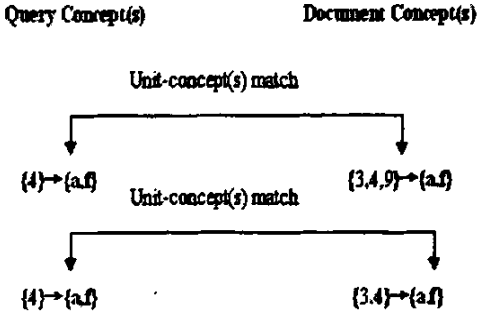

Figure 2.6 The Extracting Strategy of List Question

图 2.6 列表型问题的抽取策略

为列表型问题选择所有相匹配的概念。

图 2.6 举例说明了使用形式概念分析为列表型问题答案的抽取策略。在本文的策略中，该列表型问题的答案将选择第一个和第二个匹配，因为它们都是完全匹配。

(3) 定义型问题

在 TREC-2003 问答系统 track 中，包括了一个定义型问题的子工作，比如："Who is Aaron Copland?", "What is a golden parachute?", " What is Bausch&Lomb?"这一类的问题就必须通过一系列相关的信息金块（nuggets）来组成答案。解决定义型问题的主要方法涉及：信息抽取(information extraction)，多文档摘要(multi-document summarization)，答案融合(answer fusion)。

在本章中，对于回答定义型的问题，本文不希望忽略两个关键字节点间的任何可能的相关信息。这种单个的对象或者属性匹配被称为关键字匹配，并且可以为每一对关键字赋予重要性权重。对于定义型的问题，本文使用单元匹配和关键字匹配方法。本文选择整个相匹配的单元概念和关键字概念作为答案。

图 2.7 举例说明了定义型问题的抽取策略。

概念的部分匹配需要一种机制来定义部分匹配的相似度，并为单个的对象-属性对（单元概念）赋值。

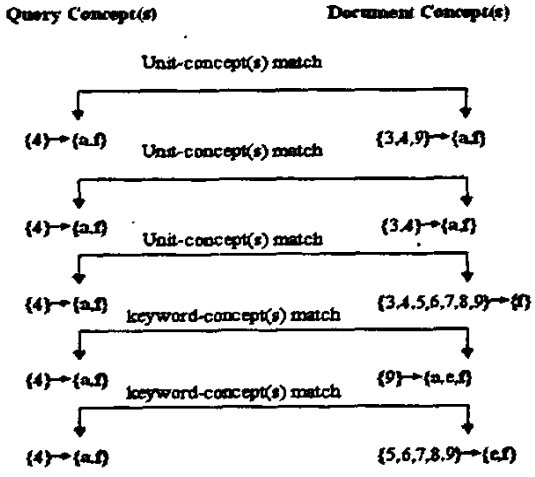

Figure 2.7 The Extracting Strategy of Definition Question

图 2.7 定义型问题的抽取策略

这里，本文使用基于关键字的相似度函数，其公式定义如下[39]：

$$similarity_{keyword}(p,q)=\frac{KN(p,q)}{Max(kn(p),kn(q))} \qquad (1)$$

其中，kn(.)表示概念节点中关键字的个数；KN(p,q)表示共同的关键字个数；p 表示问题概念格中一个概念节点；q 表示答案概念格中的一个节点。

如果这些项被赋予权重，可以使用余弦相似度来计算，具体公式如下[38]：

$$similarity_{w\text{-}keyword}(p,q)=\frac{\sum_{i=1}^{k} cw_i(p)\times cw_i(q)}{\sqrt{\sum_{i=1}^{m} w_i^2(p)}\times\sqrt{\sum_{i=1}^{n} u_i^2(q)}} \qquad (2)$$

其中，$cw_i(p)$和$cw_i(q)$分别表示节点 p 和 q 第 i 个共同关键字的权重，$w_i(p)$和$w_i(q)$分别表示节点 p 和节点 q 第 i 个关键字的权重。在这里，本文使用 tf*idf 计算关键字的权重。

2.3 Myaskanswer 与其它问答系统的区别

Myaskanswer 是一种基于多策略的问答系统，它是利用形式概念分析的方法，建立在语义基础上的问答系统。对于问答系统的三个核心问题，如何在问题理解阶段充分理解用户的提问意图，如何在信息检索模块中把相关的文档找出来，如何在答案抽取模块中准确地把答案从相关文档中抽取出来，都作了相应的研究。在问题分析阶段，本文构建问题格，并利用哈尔滨工业大学信息检索研究室《同义词词林》对问题格中的对象，属性做相应的扩展；在信息检索阶段，本文主要利用 Google API 找出相关的文档，构建文档格；在答案抽取阶段，本文引入了智能导航，概念匹配，协作推荐等技术，从而更准确的抽取答案。另外，本文还引入了常问问题集，把用户经常提问的问题和答案保存起来。对于用户输入的问题，首先在常问问题集中查找，如果能够找到相应的问题，就可以直接将相应的答案返回给用户，而不需要经过问题理解，信息检索，答案抽取等相关的复杂的处理过程，这样不仅可以提高效率，而且也能提高精度。

2.4 本章总结

本章使用了 FCA 来处理问答系统的答案抽取。这是本文的前期工作。在抽取处理中，本章首先在 FAQs 中寻找问题，如果该问题相应的答案不能满足用户的需要，本文再通过搜索引擎从网上获取相关的文档，从而使用返回的最相关的前 n 个文档建立概念格。最后，利用概念匹配的在格中抽取答案。对于不同的问题，本文使用了不同的抽取策略。实验证明通过这种混合的策略和多种资源，答案抽取系统取的了更好的性能。

第3章 智能化的构建FAQs

基于常问问题集的问答系统主要是根据用户的问题从FAQ库中获取答案。系统离线的在Web服务器日志和Web页面内容中挖掘大量的FAQ页面来建立数据库，从而从数据库中获取合适的问题答案对（Q/A）来响应用户的问题。为了构建本文的FAQs支持系统，本文必须解决下面的相关问题：如何从Web日志中取出FAQ网页；如何从收集到的这些网页中获取Q/A对；怎么通过获取到的Q/A响应用户的提问。本文提出了一个新的结合形式概念分析的概念化聚类用户日志方法，从而建立关于FAQ的形式背景。本文首先改进DBSCAN[40]算法，利用该算法来聚类Web日志，从而利用这些类来建立形式背景。接着使用这些聚类来构建概念格。最后，本文提出使用导航从获取的Q/A对中抽取答案。其中，它的性能评估主要是基于两个方面：聚类和回答。

3.1 FAQs介绍

3.1.1 FAQs基本介绍

问答系统被作为一种积极的研究主题已经很多年了。问答系统成为搜索引擎的新一代，它的工作就是从大量的文档收集中为由自然语言提出的问题获取适当的答案。目前，在问答系统中，主要有三个研究领域：基于常问问题集的，基于百科全书的，和基于开放域的。基于常问问题集的问答系统主要是做问题之间的匹配，它将用户提交的问题与本文数据库中的问题相匹配，然后通过系统返回相应的答案给用户。另外，基于常问问题集的问答系统也可以作为问答系统的一部分来研究。如果用户的问题与已有的其它用户的问题相一致，本文就可以立刻为用户返回答案。因此，FAQs的优点是不仅避免了复杂的形成答案的过程（主要包括：问题的分析，文档获取，和答案抽取等等），而且为问答系统增加了效率。

3.1.2 已有的FAQs与本文的FAQs的差异

事实上，现有的 FAQs 主要都是手工核对。换句话说，正确的答案已经由

人类的编辑准备好或者核对好。从而，这样可以确保，由系统提供的答案是相关的和正确的。但是，这样做不仅太复杂而且不利于管理和维护。从而，需要提出一种自动化的技术来帮助编辑识别 FAQs。聚焦爬行由于不能确定或者是限制在某个域，所以并不适宜于 FAQs。本文提出该研究的目的就是聚类相似查询/问题，从而发现 FAQs。为了构建本文的 FAQs，一个相关的问题必须解决：系统如何判断两个查询/问题相似？一个典型的处理该问题的方法为信息获取，即是根据关键字来计算查询/问题相似度的方法。然而，由于段的长度和相关语义问题，基于关键字的相似度计算是非常不准确的。

本章，本文提出了一个新的基于 FCA 的概念化聚类用户日志的方法，FCA 是一种数学的理论，可以应用于数据分析，信息获取，和知识发现等领域[27], [34]。本文聚类的目的是聚合相似的问题/查询在一起，这对发现用户之间共同的兴趣非常重要，从而发现 FAQs。由于日志非常大，而且每天都在变化，本文的方法能够基于聚类算法 DSBCAN 获得一个好的性能。本文改进 DSBCAN 来满足本文特殊的需要。该算法在后面的章节中详细介绍。接着，本文利用这些聚类来构建形式背景。本文试图根据问题/查询的内容本身和文档的点击来处理相似度问题。接着，本文使用形式背景构建概念格。为了加强问题/查询与文档之间的联系。本文假设，在查询词和文档之间存在一种关系，方法是基于下面的原则：（1）如果两个查询导致了相同的文档点击，那么它们是相似或者相关的，结合它们作为内涵；（2）如果文档集合导致了相同的查询选择，那么这些文档是相关的，结合它们作为外延。这两个原则结合传统的基于查询内容的方法，从而构建更准确的概念格。

概念格是结构化的，格中的概念形成了一种偏序。这个偏序关系能够被用来导航。这里，本文使用导航来从 FAQs 中抽取答案。

3.2 概念化聚类的相关研究

概念化聚类[41]是一种先进的数据挖掘技术，它可以用来构建一种概念层次。概念层次是通过概念化的类构建的。基于传统的概念化的聚类技术（比如：COBWED[41]和 AutoClass[42]）能够聚类特殊的数据类型（名词性的和数字性的）。由于其大部分的结构都像树，所以它不能支持多样的遗传化表示。CLASSIT[43]和 ECOBWEB[44]都改进 COBWEB，从而处理数据的数字性属

性。SBAC[45]是一种概念化的聚类技术，它可以处理数字性和名词性的数据。

概念格被用来表示数据之间固有的概念化的层次关系。概念格是数学理论形式概念分析的核心工具。FCA 定义形式背景来表示对象和属性之间的关系，将他们解释为相应的概念格。而概念格是概念化聚类非常适合的表示方法，例如：TOSCANA[46]和 INCOSHAM[47]。TOSCANA 是一种形式概念分析工具，它允许通过“等级”或者“主题”导航。它分解 facets 为组件，从而允许特殊的设计 facets 域，而不是统一的 facets。

当前共同的研究领域为数据库中的知识发现，一般包括：AI 和数据仓库（database warehouse）。一个概念化的聚类系统接受一个对象的描述集合（事件，事实，观察……），然后为他们产生一个分类框架。

3.3 概念化的聚类用户的日志

目前，很多的搜索引擎都积累了大量的查询日志，从这些日志中，本文能够知道用户输入了那些查询，他们选择了那些文档来阅读。这些查询的日志提供了有价值的信息。据调查，用户的信息需求范围随着时间是相当稳定的。更为特殊的是，只有 20%的信息需求随着时间的变化而变化[48]。因此，用户的查询能够通过用户过去的查询日志来预测（80%是稳定的/可靠的）。本文利用用户的日志来构建 FAQs，从而可以立刻回答大部分用户常问的问题。

本文的 Web 日志的结构为：(User-Id, Url, Date, Start-time, End-time, Keywords, Interest-Rank)。从已有的大量的用户日志数据中，本文抽取相关的查询会话。一个查询会话的定义如下：session :=<keyword> [url]*。即是说一个查询会话对应一个关键字和与关键字相关的用户点击。本文采用基于关键字的相似度来聚类这些 Urls。此外，本文也可以使用基于相同 url 的不同关键字来聚类关键字。

图 3.1 表示所提出的基于 FCA 的概念聚类方法。它由以下几步组成：概念化的聚类；形式概念分析，和层次关系的产生。

考虑用户日志的特点是稀疏且相当大的，如果背景信息非常庞大，那么构建概念格的过程是相当昂贵的花费。本文使用概念化的聚类作为预处理，从而构建一个分类的框架。接着，本文为每个聚类构建概念格。最后，根据概念格来产生层次关系。从而利用这个层次关系来构建 FAQs。

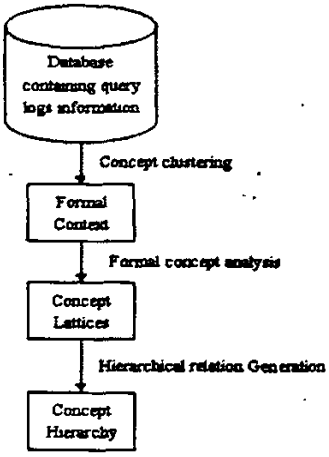

Figure 3.1 The proposed FCA-based approach to conceptual clustering query logs

图 3.1 基于 FCA 的概念化聚类查询日志的方法

3.3.1 概念聚类预览

考虑本文的聚类目标。首先，查询日志一般都非常的庞大，因此本文的算法应该在可接受的时间和空间条件下，有能力处理大量的数据集；其次，日志数据是每天都在变化的，本文的算法必须是增量型的；最后，日志信息是非常稀疏的，本文的算法应该有能力过滤低频率的查询，这里本文称低频率的查询为“噪声”。另外，本文的算法还应该有能力为本文的聚类命名，从而可以快速的找到本文期望的信息。这将意味着，这个名字将源自于该聚类的共同特征体现，但是由于稀疏的日志数据，他们存在很少的共同描述。

根据前面的分析，基于密度的聚类方法 DBSCAN 能够满足本文大部分的需求。DBSCAN 有能力处理大量的数据。它在处理空间数据时，存在很多的优点，比如：快速，有效过滤噪声，发现任意形状的聚类等等。DBSCAN 不需要输入具体的聚类数量。DBSCAN 的输入包括：最小点数-MinPts，距离极限-Eps 和未分类的点集。

定义 3.1 （点的 Eps 邻域）一个点 p 的 Eps 邻域定义记为 $N_{Eps}(p)$，定义为：$NEps(p) = \{q \in D \mid dist(p,q) \le Eps\}$，其中，q 表示属于类 D 中的点，dist（p，q）表示点 p，点 q 之间的距离。

对于某一聚类中的每个对象，在给定半径 Eps 的邻域内数据对象个数必须大于某个给定值 MinPts，即满足条件的 q 的个数大于 MinPts。Eps 是定位聚类

的密度的因素之一。MinPts 和 Eps 为已知的值，他们用来定位核心点。核心点的密度需要大于已知的 MinPts。从而 MinPts 能够将非常小的聚类作为噪声。

算法的中心思想是，对于某一个聚类中的每个对象，在给定半径 Eps 的邻域内数据对象个数必须大于某个给定值 MinPts。也就是说，领域密度必须超过某一阈值。DBSCAN 算法的聚类过程是基于如下事实的，即一个聚类可以由其中的任何核心点唯一确定。核心点就是指空间中某一点的密度如果大于某一给定阈值 MinPts。

但是传统的 DBSCAN 不能为聚类命名，且所形成聚类不能直接应用于 FCA。所以本文改进 DBSCAN，以适合本文特殊的需要。本文结合 FCA 来完成它。

3.3.2 概念聚类算法

本文采用 DBSCAN 作为查询化聚类的中心算法。另外，集合 FCA 来改进 DBSCAN，从而满足本文的需要。为了获得核心对象，本文必须使用由用户预先输入的 Eps 和 MinPts。如果在领域中的对象不小于 MinPts，那么该对象就称为核心对象。从而能够找到关于参数 Eps 和 MinPts 的聚类。这里，本文认为聚类的过程就是一个反复的区域查询处理。算法 1 中，本文给出与传统算法不同的描述。

```
Algorithm 1: conceptual clustering the user logs based on the FCA
INPUT: SetofKeywords, SetofUrls, Eps, MinPts    //SetOfKeywords is unclassified
01 clusterId:=nextId(Noise);
02 For i=1 to Setofkeywords.size Do
03    Point:=Setofkeywords.get(i);
04    if Point.clid=unclassified Then
05       if ExpandCluster(Setofkeywords, point, clusterId, Eps, Minpts)
06       Then
07          clusterId:=nextId(clusterId);
08          clusterName[i]:=point;
09       endif
```

```
10    endif
11 endFor
12 For k=1 to clusterId do
13    begin
14       AdjMatrix [] arcs;        //create adiacency matrix
15       for (m=0; m<keywords.size; m++)
16          for (n=0; n<urls.size; n++)
17            arcs[k][m][n]=(keyword[m], url[n]);   //storage data
18             Endfor
19          endfor
20       k=k+1;
21    end
```

OUTPUT: The planar table of formal context.

不同于传统的 DBSCAN，本文的方法包括命名核心对象，从而形成相应的形式背景。由于形式背景是基于二维表的，所以本文需要选择适合的基本数据结构来提高算法的有效性。另外，本文考虑构建概念格的过程中，时间和空间开销都大，从而本文使用一个聚类来构建一个概念格。因此，本文能够快速，方便的构建 FAQs。综上，本文选择邻接矩阵作为本文基本的数据结构，它是一种简单的数据结构。由于邻接矩阵是由二维表的形式存储，所以，本文能够利用它直接的作为本文的形式背景，从而构建概念格。

作为全局变量的 Eps 和 MinPts 是根据实验得出的。本文的 DBSCAN 中使用的相关函数为[40]中提出的 ExpandCluster。调用 SetofKeywords.regionQuery (Point, Eps)返回关键字集中相关点的 Eps 邻域。因此，关键的问题集中在相似度函数的定义上。本文通过关键字本身和它们相关的 urls 点击来计算相似度。

定义 3.2 相似度函数的定义如下：

$$similarity(p,q) = \alpha \frac{KN(p,q)}{Max(kn(p),kn(q))} + \beta \frac{RD(p,q)}{Max(rd(p),rd(q))} \quad (1)$$

这里，kn(.)表示关键字的个数，KN(p,q)表示两个关键字拥有相同关键字的个数，rd(.)表示对于一个关键字所拥有的点击 urls，RD(p,q)表示点击相同 url 的点击数。其中，参数 a 和 ß 由用户的需求设定。

3.3.3 层次关系的产生

概念化的聚类利用一个对象集合的描述（事件，事实，观察，…），产生一个分类表。它可以预测所有的或者重要的属性，可以被理解或者利用来产生聚类，从而产生层次，层次为一个识别处理过程或结构化的描述过程。总之，它可以解决如何在无监督的情况下，组建基于描述的对象。

基于本章的两个原则，结合传统的基于查询/问题内容的方法，构建初始的概念格。这个构建概念格的处理过程可以被简单的描述：令 Q= {1, 2,..., m}为查询集，D= {1, 2,..., n}为文档集合。数据库为二元关系$\sigma \subseteq Q \times D$。如果一个查询 q 导致了文档点击 d，本文就记为$(q,d) \in \sigma$。

例如：表 3.1 为初始的形式背景，里面的数字表示点击次数。虽然点击次数是离散的，但是，它的取值多，本文可以将其看作连续的。所以，本文需要首先将它的点击次数离散化。

表 3.1 初始形式背景

Table 3.1 The early formal context

	a	b	c	d
1	30	35	2	30
2	10	15	8	0
3	30	35	3	38
4	40	50	5	40

定义 3.3 关键字集合 D: {d1, d2, d3, ..., dn}，$dj \in D, j = 1,2,...n$，网页集合 V: {v1, v2,..., vm}，$vi \in V, i = 1,2,...m$，p(vi, dj) 表示关键字 dj 出现在网页的 vi 的频率，

$$\Delta dj = [\vee p(vi,dj) - \wedge p(vi,dj)]/k \quad \text{i,j=1,2,....} \tag{2}$$

$$b(vi,dj) = \lambda[(p(vi,dj) - \wedge p(vi,dj))/\Delta dj] + 1$$

$$\text{if k = n, b(vi,dj)} \in \{1,2,...,n\} \tag{3}$$

λ 表示取整。

表 3.2 表示 k=4 是由初始形式背景离散处理后的形式背景。这里，关键字

集 D= {a, b, c, d}，和网页集 V= {1, 2, 3, 4}。对于原则 1，查询（a，b）导致了相同的文档点击{1，2，3，4}，本文认为 a 和 b 是相似的，从而构建概念节点{1, 2, 3, 4} → {a, b}。对于原则 2，文档集合{1，3}被相同的查询{a，b，c，d}选择，本文认为文档项{1，3}是相关的，从而构建概念节点{1, 3} → {a, b, c, d}。

表 3.2 形式背景

Table 3.2 The formal context

	a	b	c	d
1	3	3	1	4
2	1	1	4	1
3	3	3	1	4
4	4	4	3	4

每个概念化的聚类为一个从概念格中抽取的子格。FCA 通过两个概念的子-超概念关系来模仿特殊性和一般性。

从前面的章节知，概念化的聚类集合中构建一个概念层次，必须找到它们的层次关系。从 FCA 理论中，任何有限的格为一个概念格。即如果一个概念格上确界和下确界都存在，那么它为一个完全格。基于这种分析，层次关系就很明显了。

定义 3.4 对于形式概念(X, Y)，本文有定义：

$$\uparrow: p(Vi) \to \{(dj)r \mid dj \in p(D)\}, X\uparrow = \{(dj)r \mid dj \in D, \forall vi \in X, b(vi,dj) = r\} \quad (4)$$

$$\downarrow: \{(dj)r \mid dj \in p(D)\} \to p(V), Y\downarrow = \{vi \mid vi \in V, \forall (dj)r \in Y, b(vi,dj) = r\}$$

$$\text{Thereinto, } X\uparrow = Y, Y\downarrow = X. \quad (5)$$

定义 3.5 当 H1=(X1, Y1), H2=(X2, Y2),如果 $H1 \le H2 \Leftrightarrow X2 \subset X1 \Leftrightarrow Y1 \subset Y2$，

$$\text{LUB: } \vee(Xi, Yi) = ((\cup Xi)\uparrow\downarrow, \cap Yi) \quad i=1...n \quad (6)$$

$$\text{GLB: } \wedge(Xi, Yi) = (\cap Xi, (\cup Yi)\downarrow\uparrow) \quad i=1...n \quad (7)$$

本文使用算法[69]来构建概念格。

图 3.2 表示形式背景（表 3.3）构建的概念格。

表 3.3 形式背景

Table 3.3 The formal context

	A	B	C	D
1	A4	B2	C1	D1
2	A1	B2	C4	D4
3	A4	B1	C1	D2
4	A1	B4	C1	D1

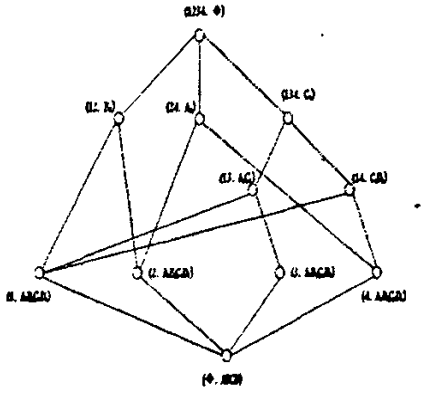

Figure 3.2 The concept lattice of the formal context in Table 3.3

图 3.2 关于形式背景表 3.3 的概念格

3.4 从 FAQs 中抽取答案

本文能够使用概念格来导航。概念格本身的特点就是结构化，格中的概念形成一个偏序。这个偏序能够被用来导航。另外，使用 FAQs 的目的是直接响应用户提问，这里本文需要一个可视化和交互式的技术。这一节，本文提出使用 FCA 来进行导航，从而实现 FAQs 的应用。主要分为 3 步进行：

第一，从包含查询/问题的节点开始进行导航。

第二，用户在格中选择一个子概念来进行导航，通过添加项使他的答案更加的特殊化，或者在格中通过选择一个超概念，通过减少项来使他的答案更加的一般化。通过计算平均精度（mean average precision）来添加项。平均精度主要用在排序获取摘要值。单个查询的平均精

度意思是每次获取的相关信息的精度值。运行的值是指单个平均精度成绩。它包含了召回率和精确度两个方面，对整个的排序很敏感。

第三，通过添加项，基于一个高的平均精度，到达最终的概念。另一方面，从 FCA 来看，可以用多个实例在形式背景中存储问题-答案对。多重的遗传被模型化为属性集合之间的偏序，它们可以由形式背景的二维关系来反映：如果一个问题与一个查询相关，那么它与偏序集中向上的所有查询都相关。

图 3.3 举例说明了导航。导航以用户问的问题作为起始点。首先，建立问题的概念节点，比如：{d,e,f,g}。请看实心的节点。用户使用 f 作为导航的方向。通过提高平均精度。通过添加属性项，获得一个高的平均精度值达到最终的概念节点。

本文使用不同的精度极限值 e 来衡量不同问题的答案。e 是一个用户给定的常量极限。

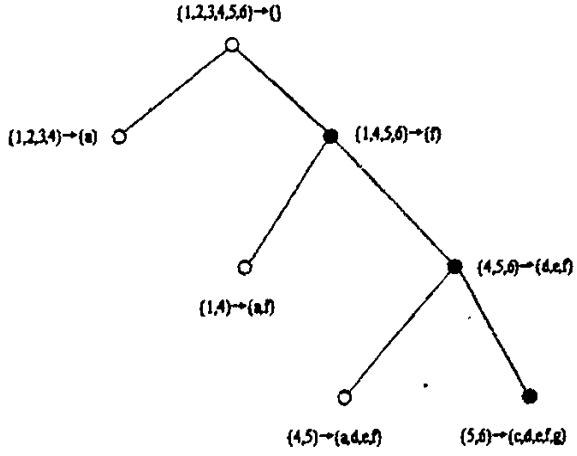

Figure 3.3 An example of navigation

图 3.3 导航的例子

3.5 总结

本章中，问答系统使用了概念化的聚类用户的日志来构建 FAQs。这个是本文问答系统的一个部分。本文使用用户的日志构建形式背景。本文试图根据它们的内容信息和文档点击来聚类相似的问题/查询。本文假设查询/问题和点击

文档之间存在相关性。本文的方法基于两个原则：（1）如果两个查询导致了相同的文档点击，就被认为他们是相似的或者相关的，从而合并他们作为内涵；（2)如果一个文档集合被频繁的由相同的查询选取，就认为这些文档是相关的，从而合并他们作为外延。这两个原则与传统的查询内容相结合，从而构建初始的概念格。由于日志信息的特点，所以本文使用基于 DSBCAN 聚类算法来获取更好的性能。由于概念格的特点，在形成相关 FAQs 的概念格中，本文使用导航来为问题抽取答案。

第4章 网页集的获取

近年来，在搜索引擎的研究领域中，元搜索引擎是研究是一个热点。元搜索引擎能为用户提供一个统一的访问接口。它方便而有效的为普通用户调用多个搜索引擎，而且可以在返回的结果中识别有用的文档。本文中，介绍了一个基于形式概念分析的个性化元搜索引擎，MySearch。它获取用户的信息，通过重排结果，提供一个实时的响应。重排是通过使用用户的使用日志和源搜索引擎返回的结果共同组建的概念格实现的。最后，改进的重排通过 MySearch 返回给用户。实验结果（见 6.3）表明，本文的方法为搜索结果与用户需求之间的满意度提供了一个积极的改进。

4.1 介绍

近年来，随着信息爆炸式的增加，搜索引擎成为数据挖掘和信息获取的关键技术。WWW 的迅猛增长为搜索引擎提出了空前的挑战。一般目的的搜索引擎[71]已经取得了很大的成功。它们是从庞大的信息资源中获取信息的有效工具。例如，Google，作为一个目前最流行的搜索引擎，它通过对 20 多亿网页进行整理，不仅为全世界的用户提供适需的搜索结果，而且搜索时间不到半秒。但是，随着 Web 技术的流行，基于 Web 上的信息需求以某种速率逐渐递增，从而搜索引擎很难与之同步。另外，任何搜索引擎的设计，均有其特定的数据库索引范围，独特的功能和使用方法，以及预期的用户群指向。一种搜索引擎不可能满足所有的检索需求。用户所需要的信息往往分布在多个搜索引擎数据库中。元搜索引擎能够为用户提供一个统一的访问接口。它方便而有效的为普通用户调用多个搜索引擎，而且可以在返回的结果中识别有用的文档。

元搜索引擎是一种调用其他独立搜索引擎的引擎，它被用来对多个独立搜索引擎整合，调用，控制和优化利用。又被成为“搜索引擎之母”。这里，“元”的意思是“总的”，“超越”。由于元搜索引擎被考虑为各源搜索引擎之上的接口，所以它能够增加覆盖率。但是，用户对通过一个查询返回的大量不相关的结果仍然感到困惑。另一方面，怎么选择数据库去识别本地的搜索引擎（源搜索引擎），对于一个给出的查询，哪些可能包含有用的文档。考虑到这些情况，一个

又希望（有前途）的解决方法是个性化。本章使用潜在在搜索引擎中的历史信息，用户的登陆信息，和用户的行为（点击）来动态的分配查询给适合的搜索引擎。接着，这些源引擎能够集中爬行某些特殊的主题。当爬行的页面与目标主题相关时，这样不仅可以减少获取的数据大小，而且可以提高精确度。

本文提出一种新的个性化的元搜索引擎 MySearch，它提取用户的兴趣，通过形式概念分析重排结果集合，为用户提供实时的响应。首先，MySearch 通过用户登陆信息构建初始查询。其次，MySearch 为用户的兴趣域和查询关键字提供一种映射。此外，利用源引擎返回的结果集合和用户的使用日志共同构建形式背景。随着用户行为的增加，本文的形式背景能够逐渐改善（增量型的）。接下来，本文构建概念格，通过计算权重，重排结果集，最后通过 MySearch 返回给用户。

4.2 相关的研究工作

目前，相关元搜索引擎和集成搜索引擎的研究很多[72][73][74]。元搜索引擎各具的特色，功能各有侧重。完全理想的元搜索引擎还没有出现。例如：有些元搜索引擎支持多语种[75]，特别是中文[76]。有些元搜索引擎能够实现语法转换能力，支持指定的字段检索，能够充分发挥各个独立搜索引擎的高级检索功能[77]。部分元搜索引擎存在源搜索引擎列表，用户能够自主的选择和调用源搜索引擎[78][79]。大部分的元搜索引擎仅支持调用 Alta Vista, Excite, GoTo.com, Yahoo!, Infoseek, Lycos 等常用的搜索引擎。一些大型搜索引擎如 NorthernLight、HotBot 等被排除在外。另外一个方面，元搜索引擎只能返回很少的“相关度”较高的结果，大量可能有价值的源搜索引擎的检索结果被忽视。[80]介绍了一种新的元搜索引擎 WenFusion。它基于用户的参数选择，在某一特定类别中学习潜在搜索引擎的行为（元搜索）。利用 OWA 算子的方法为结果列表排序，并且应用最优化的算子作为权重函数来进行处理。[81] 比较 3 个医药搜索引擎，它需要更多或更少的用户努力来形成查询 q 和评估结果。他们实现最终用户向 23 个用户学习的模式，这 23 个用户被一个已知的模式询问如何找到信息。他们使用一种平衡内主题（balanced within-subjects）的设计和报导，从最终用户的观点上，支持工具的效力，效率和可用性。

4.3 MySearch 的框架结构

当前的搜索引擎主要是基于关键字的，不能满足用户的信息需求。对于一个简单的关键字查询，典型的情况是返回数百万文档，这种情况下，用户很难找到自己需要的信息。考虑到这些，本文的搜索处理包含用户个性化的兴趣和主题，从而增加返回的精确率。另外一个方面，本文同样可以使用这些个性化的信息来帮助过滤结果。因此，捕获用户个性化的信息问题引起了关注。这里，利用用户的登陆信息和查询日志来获取用户的兴趣和主题。从而建立用户的兴趣库。

图 4.1 讨论了本文详细的框架结构。首先，用户通过 MySearch 的界面发送搜索查询。MySearch 通过 Yahoo 的分类目录选择用户的兴趣来限制查询范围。接着，MySearch 发送查询给足够多的搜索引擎，例如：Google，Yahoo，Lycos，Netscape 等。第三，源搜索引擎通过 MySearch 返回相关的结果。MySearch 存储这些响应，去除重复，返回一个结果列表。第四，MySearch 结合用户的使用日志，结合 FCA 的方法来排序结果。最后，MySearch 为用户展现排序后的结果，并且根据用户的行为更新兴趣库。

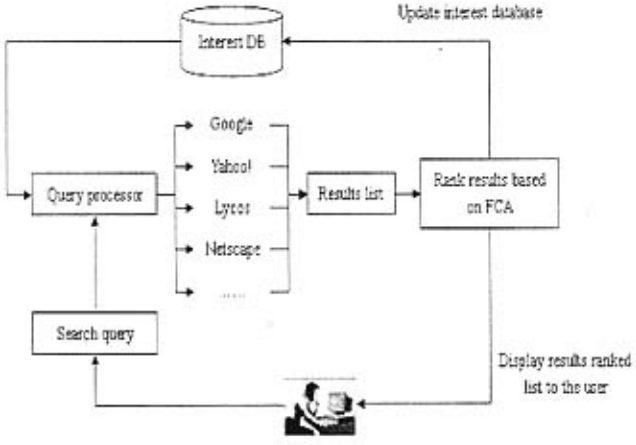

Figure 4.1 The framework of MySearch

图 4.1 MySearch 的框架

4.3.1 建立用户的初始查询

用户的知识库包含登陆信息和使用日志。本文的兴趣库通过每一次排序经常的更新。登陆信息的结构为：(userid, username, passage, sex, career, interest)。本文使用登陆信息来构建初始的查询。用户的兴趣通过 Yahoo 可以被划分为 14 类，比如：政治，经济，文化，娱乐等等。接着，一个特殊域的查询通过 MySearch 被提交给源引擎。另一个方面，查询词对应的 URL 通过兴趣库返回。本文的兴趣库结构被表示成 IDB (keywords, urls)。

4.3.2 基于 FCA，排序结果

本文的元搜索引擎的主要目的是整合由源引擎返回的结果，结合用户的兴趣来增加精确率。返回给用户的排序列表必须是用户最感兴趣的网页。本文使用用户的使用日志和源引擎返回的结果来构建兴趣形式背景。接着，通过这些形式背景来构建兴趣格。最后，本文从兴趣格中抽取兴趣信息，通过 MySearch 返回排序结果给用户。同时，本文更新兴趣库。

（1） 形成兴趣形式背景

使用日志的结构为：(userid, url, date, start-time, end-time, keywords, interest-rank)。通过源引擎返回的结果结果表示为：(id, keyword, title, url, snippet)。对于 titles，spippets 的预处理包括，切词，分词，去掉停用词。接着，titles，spippets 由一组单词集合组成。本文使用这些单词和使用日志中的关键字形成关键字集合。同时，本文利用使用日志和搜索引擎（SE）中的 URL，加上兴趣库中的返回的 URL，形成 URL 集合。

根据前面章节的分析，形式背景是基于二维表的。一个形式背景为 K:=(G, M, I)，这里 G 为 URL 集合，M 为关键字集合。包含关系 $(g,m)\in I$。I 为兴趣排序集合，且值属于[0，1]。表 4.1 举例说明。这里，G= (url1, url2, url3, url4), M= (keyword1, keyword2, keyword3, keyword4)，I 为兴趣排序度。

令 I (urli, keywordj) 为兴趣排序度，它可以通过以下几个因素计算：

1. SE 中的关键字有最高的权重。本文使用参数 σ 来衡量。对于 urli, 如果 keywordj 属于 SE 中的关键字集合，那么 I_1 (urli, keywordj) $=\sigma$ ；否则 I_1 (urli, keywordj) =0.

表 4.1 形式背景举例

Table 4.1 The example of formal context

	keyword1	keyword2	keyword3	keyword4
url1	0.50	0.20	0.05	0.25
url2	0.08	0.20	0.30	0.42
url3	0.46	0.08	0.11	0.35
url4	0.15	0.55	0.10	0.20

2. SE 中的 title 有比 Spippet 高的权重。本文分别用 ξ 和 η 来衡量他们的重要性。对于 urli，如果 keywordj 属于 title (snippet)集合，那么 I_2 (urli, keywordj) = $\xi(\eta)$；否则 I_2 (urli, keywordj) =0.
3. 对于相同的用户，URL 的点击次数。用户点击 URL 的次数越多，兴趣排序越高。本文使用参数 α 来衡量一次点击。例如：一个用户点击 URL1 的次数是 5，那么兴趣排序度为 5α。对于 keywordj，如果点击 urli 的次数为 n, n 为整数那么 I_3 (urli, keywordj) =nα；否则 I_3 (urli, keywordj) =0.
4. 用户浏览 URL 的时间。本文同样给一个参数 β 来衡量时间。用户浏览 URL 的时间越长，兴趣排序值越高。对于 keywordj, 如果用户浏览 urli 的时间为 m sec，那么 I_4 (urli, keywordj) =mβ；否则 I_4 (urli, keywordj) =0.
5. 对于相同的关键字，存在 url1→url2→url3，这个排序暗示，用户首先对网页 1 感兴趣，然后是网页 2，最后才是网页 3。网页 1 的兴趣排序应该比网页 2 高，网页 2 的兴趣排序应该比网页 3 高。本文给出一个衰减因子 γ。对于 keywordj，存在 urla→urlb→urlc, 那么 I(urlb, keywordj)= γ I(urla, keywordj); I(urlc, keywordj)= γ I (urlb, keywordj).

从上面的讨论，本文能够获取一个兴趣排序来构建形式背景。

定义 4.1 兴趣度 I (urli, keywordj) 的值为四个项的叠加。即：I(urli, keywordj)= I_1 +I_2+I_3 +I_4. 第 5 项被用来改进本文的形式背景。

其中，各参数的值为实验中取的的经验值，且都属于[0, 1]，这里 $\sigma > \xi > \eta$。

（2） 构建兴趣格

通过前面的形式背景来构建兴趣格，兴趣排序度应该首先被离散化。

定义 4.2 关键字集合{d1, d2, d3, ..., dn}，$dj \in D, j = 1,2,...n$，网页集合 V: {v1, v2,..., vm}，$vi \in V, i = 1,2,...m$，p(vi, dj)表示关键字 dj 出现在网页 vi 的频率。

$$\Delta dj = [\vee p(vi,dj) - \wedge p(vi,dj)]/k \qquad \text{i,j=1,2,....}$$

$$b(vi,dj) = \lambda[(p(vi,dj) - \wedge p(vi,dj))/\Delta dj] + 1 \quad \text{if k = n, b(vi,dj)} \in \{1,2,...,n\}$$

λ 表示取整。例如：表 4.2 表示离散后的形式背景。

定义 4.3 对于形式概念(X, Y)，本文有以下定义：

$$\uparrow: p(Vi) \to \{(dj)r \mid dj \in p(D)\}, X\uparrow = \{(dj)r \mid dj \in D, \forall vi \in X, b(vi,dj) = r\}$$

$$\downarrow: \{(dj)r \mid dj \in p(D)\} \to p(V), Y\downarrow = \{vi \mid vi \in V, \forall (dj)r \in Y, b(vi,dj) = r\}$$

其中，$X\uparrow = Y, Y\downarrow = X$.

表 4.2 离散后的形式背景

Table 4.2 The dispersed formal context

	keyword1	keyword2	keyword3	keyword4
url1	4	2	1	1
url2	1	2	4	4
url3	4	1	1	2
url4	1	4	1	1

定义 4.4 当 H1=(X1, Y1)，H2=(X2, Y2)，若 $H1 \le H2 \Leftrightarrow X2 \subset X1 \Leftrightarrow Y1 \subset Y2$.

LUB: $\vee(Xi,Yi) = ((\cup Xi)\uparrow\downarrow, \cap Yi)$ i=1, …n

GLB: $\wedge(Xi,Yi) = (\cap Xi, (\cup Yi)\downarrow\uparrow)$ i=1, …n

本文使用[82]中的算法来构建概念格。由表 4.2 中的形式背景构建的概念格如图 4.2 所示。为了方便，本文记{url1, url2, url3, url4} 为 1234, {Keyword1, Keyword2, Keyword3, Keyword4} 为 ABCD.

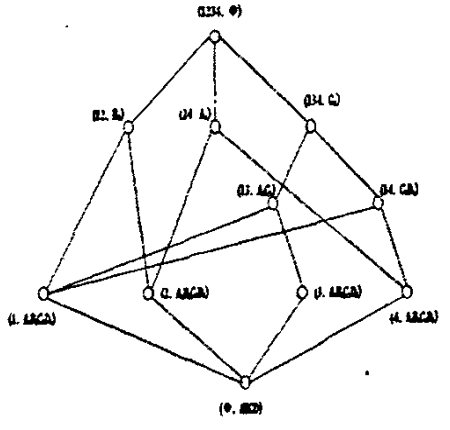

Figure 4.2 The concept lattice of the formal context in Table 4.2

图 4.2 表 4.2 中形式背景对应的概念格

（3） 通过 MySearch 返回重排序列给用户，更新兴趣库

由于的概念格中概念格节点间存在一个序列。本文利用推荐算法来实现重排。算法的描述如算法 1 和算法 2。

```
Algorithm 1: Our recommend algorithm
Input: L (Node, Edge)   //concept lattice
Output: The Re-rank
01 ChooseEarlyNode(L)
02 Re-rank←EarlyNode.object
03 while (SuperNode !=LUB(L)) do   //SuperNode isn't the max node
04 begin
05    SuperNode=FindSuperNode(EarlyNode)
                    // according to the set of Edge to Find SuperNode
06    Objects= SuperNode.Objects
07    for i=1 to Objects.length do
08    begin
09       Get(Objecti)
10       if Objecti !=EarlyNode.Objects then
11       begin
12          degree[i]=degree(Nodei)
13          next(Object)
14       endif
15    endfor
16    recommend (Object with max (degree))
17    Re-rank←Object
18    EarlyNode←Object
19    return 02
20 endwhile
21 Re-rank←the other objects
22 return Re-rank
Algorithm 2: ChooseEarlyNode(L) function
```

```
01 Node←Number(L.object)=1
02 EarlyNode=Get (Node)
03    for int i=1 to Node.length-1 do
04    begin
05       if EarlyNode.degree<Node.degree then
06       EarlyNode=Node
07    endfor
08 return EarlyNode
```

根据前面介绍的概念格的构建，MySearch 通过本文的推荐算法返回重排序列给用户。本文的推荐算法由 4 步组成。第一，从只有一个对象的节点出发，选择有最大值的节点作为初始节点，并且放入重排队列中。第二，根据概念格中边的集合，找出初始节点相应的超节点。这里，本文为这些超节点（除初始节点外）找出相应的对象集合，其中，通过相应的值的大小做推荐处理。本文选取有最大值的对象进行推荐，并且将这些对象放入我们的重排序列。另外，使用这些节点作为初始节点。最后，循环这种处理直到超节点是概念格中最大的节点。例如：图.4. 2，初始节点为(2, $A_1B_2C_4D_4$)， 它的超节点为：(12, B_1), (24, A_1)，推荐排序为 1，4。这里，本文比较 1 和 4 的值，有 degree (1)>degree (4)，所以，令(12, B_1)作为初始节点。重复这个操作，得到重排序列为 2, 1, 4, 3；也就是 url2,url1,url4,url3。鉴于兴趣库由关键词和 URL 集合组成。本文可以使用概念格直接更新兴趣库。

4. 4 总结与未来的研究

这篇文章，本文介绍了一种新的基于 FCA 的个性化的元搜索引擎，MySearch。它获取用户的信息，通过重排结果，提供一个实时的响应。重排是通过使用用户的使用日志和源搜索引擎返回的结果共同组建的概念格实现的。最后，改进的重排通过 MySearch 返回给用户。实验结果（见 5. 3）表明，本文的方法为搜索结果与用户需求之间的满意度提供了一个积极的改进。

第5章 定义型问题的答案抽取

TREC-2001 中，在真实的用户日志文件中发现关于问题类型的分布情况，有 27%的问题属于定义型的问题。定义型的问题是一个未涉及的大的研究领域。本章提出了一个新的方法来回答定义型的问题。本文使用形式概念分析来研究协作推荐，从而回答定义型的问题。推荐是计算机系统设计来帮助用户在一个大面积的可选择领域中找到更适合的项[49]。在协作推荐系统中，本文应用文档和问题之间的关系作推荐。FCA 组建文档和查询为概念，通过概念格来排序。另外，本文对推荐算法作了相应的研究。

5.1 协作推荐

推荐是计算机系统设计来帮助用户在一个大面积的可选择领域中找到更适合的项。概念上的划分推荐为两个方面：内容过滤和协作过滤。内容过滤方法是指基于文档本身的特征单独地推荐。例如：给出由相同作者写的所有文档，或者给出与前面文档相似的文档[50]。协作过滤[51]是评定当前用户所需要的文档与过去研究索引（Research Index）中的文档相似度。一般的用户相似度衡量包括皮尔森相关系数（Pearson Correlation Coefficient）[51]，均方差（mean squared error）[52]，向量相似度（vector similarity）[53]。最近的研究工作包括应用统计型的机器学习技术，比如：贝叶斯网络（Bayesian networks）[53]，依赖网络（dependency network）[54]，奇异值分解（singular value decomposition）[55]，潜在类模型（latent class models）[56][57]。当前的协作过滤系统应用于一个相当受限的交互式模型。这些系统智能化的根据用户初始登陆阶段的信息得出相关的信息，用户的这种推荐是属于间接的。2003 年，[58]提出了一种智能化的信息选取技术，他们从用户的相关行为中选取这些推荐信息。

很多的推荐算法是上下文相关的。这就是说，推荐的队列不依赖于用户当前浏览的上下文和最新的动作。[59]使用了一个极大熵方法在用户的当前浏览流中产生推荐，这种方法适合数据稀疏，高维和动态情形环境中。另外，[60]利用形式概念分析来作协作推荐。它们应用 FCA，为用户和协作推荐系统项之间构建二维关系。FCA 组建用户和项为概念，通过概念格来排序。

5.2 形式概念分析

1982 年，形式概念分析是由 Rudolf Wille 等人提出[27]。它是数据分析，知识表示和信息管理的方法之一。相关基本概念见 1.3。这里本文提出一个新的概念：入口级概念（Entry-level concepts）。

一个实体 d 的入口级概念是指：d 是一个外延的成员，但是它不属于 d 的任意子概念节点的外延成员。类似的，特征/属性 q 的入口级概念是指：q 是一个内涵的成员，但是它不属于任意超概念节点的内涵成员。

5.3 基于 FCA 的协作推荐

5.3.1 应用 FCA 来协作推荐

在本文的协作推荐中，考虑 n 个文档，D={d: 1...n}，m 个查询，Q={q: 1...m}，其中，参数 n，m 表示使用一个 $n \times m$ 的等级矩阵 $r_{d,q}$。本文使用穷尽推荐算法（EX-CR）来建立协作推荐[61]。这里，本文在这些等级矩阵中应用概念格来快速搜索邻节点。基于 FCA 的协作推荐包括四步：

第一：根据查询 q 随机的选取一个概念节点 N。

第二：选择 N 的最近的邻居节点。考虑到概念格的结构特点，本文只对它的子概念节点有兴趣。

第三：预测：选择子概念，如果它包含入口概念。计算文档之间的相似度，其中，一篇文档包含入口概念，另一篇没有包含入口概念。相似度的计算公式如下：

$$similarity(d1,d2) = \frac{KN(d1,d2)}{Max(kn(d1),kn(d2))} \qquad (1)$$

其中，kn（）表示文档中关键字的个数，KN（d1，d2）表示它们共同的关键字的个数，这里，d1，d2 表示文档。

第四：推荐：如果文档的相似度大于 σ，那么本文为 N 推荐入口概念。σ 是由用户给的最小相似度。通过对每个查询 q 找出预测等级，其中在 N 中没有被定值，但它至少被一个邻接点定值。根据相似度的大小来排序，从高到低的顺序做推荐。

5.3.2 回答定义型的问题

定义型的问题就是为给出一个定义型的问题，产生一个与之相关的各个“句子”的总结。定义型的句子有一个特点，它是以各种各样模式出现，需要一个灵活的模式匹配从而获取更高的召回率。一个典型的定义型的问答系统回答一个问题时，首先获取与目标相关的语言的结构（比如：句子，相关的字句），接着总结这些单元为可读的定义。为了识别这些结构，当前的系统使用定义样式来识别定义。[62]使用机器学习来分类和排序侯选的答案，而且只使用四个一般的属性来分类问题。[63]使用一个最大熵模型排序侯选答案，它使用了一个包含 8500 个样式的属性集。[64]集中研究定义型问题，目标是产生一个一致性的多句子定义。这个方法的核心思想是利用一个部件，该部件使用机器学习的方法来识别侯选句子。[65]结合并且扩展人工的词汇样式（manually crafted lexical patterns）和 WordNet。他们使用支持向量机（SVM）来识别和排序单个的 250 个特征片段，从而获取答案。这里，本文使用协作推荐来获取定义型的句子，从而返回定义型问题的答案。本文系统结构如图 4.1 所示。

首先，由用户使用自然语言提出一个定义型的问题。然后，处理这个问题，找出问题的焦点和问题的主题，从而构建问题格。本文首先使用概念匹配从 FAQs 中获取定义型的句子。接着，本文处理这些句子，去掉冗余的项，合并答案。最后，本文判断答案是否满足用户的需求，如果用户对答案满意，那么本文返回答案。如果用户不满意答案，那么本文的系统将使用查询表达式和向量空间模型余弦值来获取相关的文档。接着，选择最高相似度的文档来构建概念格。本文使用基于 FCA 的协作推荐获取定义型的句子，去掉冗余，然后合并答案。本文的系统重复的运行，直到用户对答案满意。

冗余的消除是通过为那些单词，表达式和关系赋信任度。冗余在多源知识整合中提出了一个主要的挑战。这个问题对于本文的数据存储有着特殊的意义。为了处理这个问题，本文应用一个简单的启发式规则来消除重复信息：如果两个“金块”拥有超过 60%的相同关键字，本文就将其中之一删掉。

最后，答案合并部件为一个给定的问题决定答案“金块”的数量。本文开始时，使用一个较低的极限值，接着为每个答案线形的增大极限值。这个机制可以在一个合理的范围内限制答案的数量。考虑获取了 n 个“金块”，本文计算

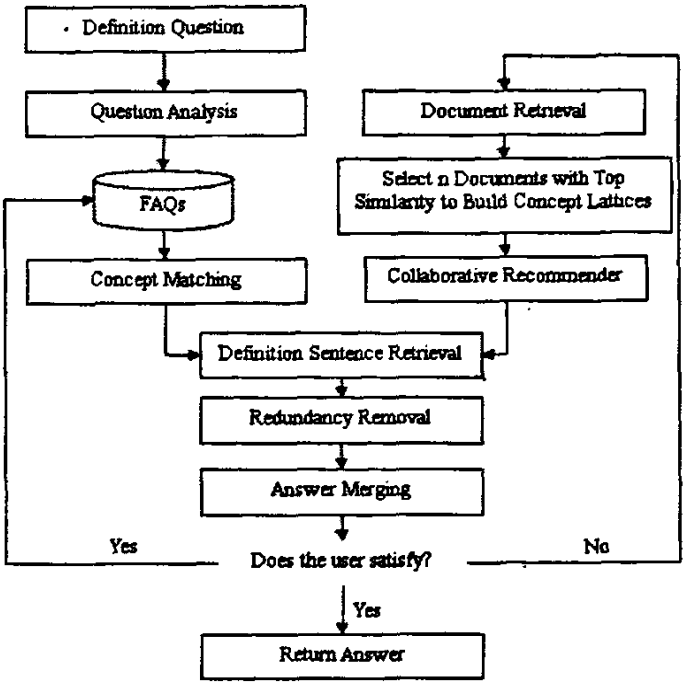

Figure 5.1 The system architecture of our definition question

图 5.1 定义型问题的系统结构

最后的“金块”数量为：

$$\begin{cases} n & \text{if } n=10 \\ n+(n-10)^{1/2} & \text{if } n>10 \end{cases}$$

5.4 协作推荐算法

考虑本文的算法将会在格中循环，本文的形式背景不能太大。否则，构建概念格的计算消费将会太昂贵，而且这个花费是没有必要的。另一方面，如果本文的形式背景太小或者太特殊，那么在很多情况下，搜索引擎都不能找到足够的相关文档，而本文的协作推荐也就没有意义了。此外，本文必须要避免“噪声”的干扰，比如不正确的信息。这里，本文选择相似度最高的前 20 篇文档来构建概念格。本文现在开始描述基于 FCA 的协作推荐算法。考虑本文算法中的以下方面。如果入口级概念存在于两个概念中，他们有相同的属性集。本文相

信这个对于本文的预测是没有用的。如果入口级概念是本文希望预测的概念的超概念，本文相信它对本文的预测仍然没有作用。本文具体算法见算法 1。

Algorithm 1 Using FCA to Collaborative Recommender

```
Input: L(Node, Edge)                    //Concept Lattices
       Q, s                             //the questions; s is the minimum similarity
01 InitQueue (&F)                       //initialization the queue (F)
02 the Queue of FIFO, F←(L.node)
03 While F≠Φ  Do
04 Begin
05   If F≠Φ  then
06   begin
07     QueueTraverse(F,visit(Q,F))      //deal with each element in F
08     f=max(sim(Q,F))
09     Return f
10   end
11   ClearQueue(F)
12   F←FindSubNodes(f)                  //require subnodes of f
13   While F≠Φ  Do
14   Begin
15     GetHead(F,e)
16     while Findentrylevelconcept(e)≠Φ  do   //find the entry level concept
17     begin
18       for i=1 to n do
19         if y=visit(xi,f)>σ  then     //σ is the minimum similarity
20           return S=xi
21     end
22   end
23 End
24 visit (p,q)                          //calculate the similarly
25 begin
```

26 sim(p, q)=KN(p,q)/Max(kn(p), kn(q)) //kn() is the number of the keywords; KN(q,p) //is the number of common key words.

27 end

Output: Definition Sentence S

5.5 评估

至今，NIST 有两个定义型问题的形式化评估，分别在 TREC2003 合 TREC2004 中。定义型问题的答案有一个无序集[document-id, answer string]对组成，这里的字符被预测来提供关于实体定义的相关信息。为了评估系统的反馈，NIST 从所有的系统中组建答案串，去掉他们之间的关系，为人类的评估员提供它们。评估员人工分类每一个“金块”为重要的（vital）或者是好的（okay）。重要的金块表示对于一个好的定义来说必须具备的；而好的金块贡献出有价值的信息但是并不是必须的[66]。当这个答案的重要的/好的金块被建立，评估者将手工的评价每次运行。

最后的每个答案的 F 得分如下。这里，r 表示返回的重要金块，a 表示返回的好的金块，R 表示答案中的重要金块，1 是整个答案串中好的金块。

$$\mathrm{Re}call(\beta) = r/R$$

$$Allowance(\alpha) = 100 \times (r + a)$$

$$\mathrm{Pr}ecision(\gamma) = 1 \text{ if } l<a; \quad 1-(l-a)/l \text{ otherwise}$$

$$\text{Finally, the } F(\delta) = (\delta^2+1) \times \gamma \times \beta / \delta^2 \times \gamma \times \beta$$

(In TREC 2003, δ =5; In TRCE 2004, δ =3.)

另外，[67]应用粗糙集自动化的评价定义型的问题，将这个工作看做是文档摘要的另外一种技术。[68]提出 POURPRE，它是基于 n-维共现（n-gram co-occurrences）的，但是他被修改来适合于问答系统独特的特征。它也是一种自动化的定义型问题的评估技术。

定义型问题的答案选取还可以采用下面的资源：WordNet 注释，从 biography.com 中收集的 14414 个传记；Mike Fleischman 描述了 966557 个重要人物的适当描述。

5.6 总结

本章介绍了一种新的方法来回答定义型的问题。本文利用基于形式概念分析的协作推荐来回答定义型的问题。推荐是计算机系统设计来帮助用户在一个大面积的可选择领域中找到更适合的项。在协作推荐系统中，本文应用文档和问题之间的关系作推荐。FCA 组建文档和查询为概念，通过概念格来排序。

第 6 章 实验

本章介绍基于校园网问答系统的的具体实现。本文的开发平台为 Microsoft Windows XP Professional 版本 2002 Service Pack 2；Intel(R) Pentium(R) M processor 1.70GHz 984 MHz, 512MB 的内存。开发工具为 Visual Studio 2003, SQL Server2000。

6.1 基于概念匹配的答案抽取实验

TREC 会议每年都会提供一个测试集，让参加 TREC 的研究人员评价自己的问答系统。本文采用了 TREC2004 测试集中的一部分对本文的问答系统做初步评估。实验中，本文选取每次返回文档（网页）数为 10；从而对这些网页做切词，分词，统计词频等预处理后，构建文档格；接着，本文使用已有的问题格与文档格相匹配。TREC 允许对每个问题给出 5 个答案。如果问答系统返回的第一个答案是对的，那么这个问题就得 5 分，如果问答系统返回的第二个是对的，得 4 分，如果问答系统返回的第三个问题是对得，得 3 分，依次类推。这里，本文认为，只要跟正确答案有 60%相同，就是对的。把每个问题所得得分加起来就可以得到问答系统所得总分。总分越高，该系统得到准确率越高。依照该方法，本文的实验结果如表 6.1 下：

表 6.1 基于概念匹配的答案抽取实验结果

Table 6.1 the results of experiment of answer extract based on concept matching

The type of question	Factoid	List	Definition
The number of questions	100	50	50
The total of scores	1864	736	678
Average scores	3.728	2.944	2.712
The percent of precision	74.56%	58.88%	54.24%
The average percent of precision	62.56%		

6.2 概念聚类日志与 FAQs 评估

由于本文的工作需求，本文使用 C#和 SQL Server2000 来实现算法。Web 日志信息如图 6.1。本文使用日志数据来评估改进后的 DBSCAN。本文将它与最初的 DBSCAN 的性能进行比较，如图 6.2。为了方便，实验中，本文将输入参数 MinPts 等于 10，而 Eps 等于 2。这里，X-轴表示用来聚类的数量，Y-轴表示已聚类的百分比。

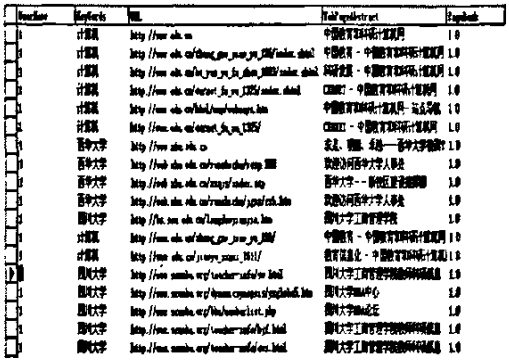

Figure 6.1 The part of web logs

图 6.1 部分 Web 日志

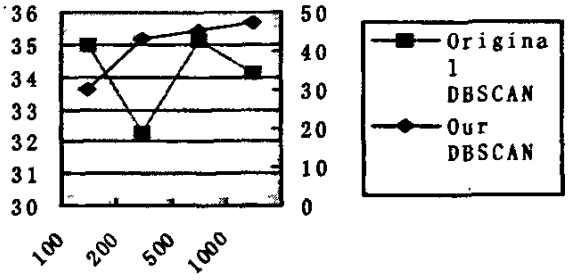

Figure 6.2 Compare with original DBSCAN

图 6.2 与初始 DBSCAN 比较

另一方面，本文讨论结合参数α和β。本文测试 3 种不同的相似度集合方式：(1) α=0.5 and β=0.5; (2) α=0.8 and β=0.2; (3) α=0.2 and β=0.8。图 6.3 表示对于这三种不同设置的聚类查询比例的结果。X-轴表示相似度极限，Y-轴表示聚类查询的比例。

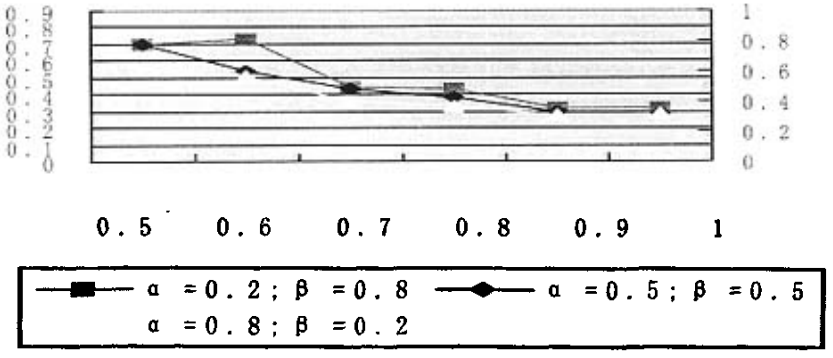

Figure 6.3 the result of 3 different combinations with the similarity

图 6.3 三种不同设置的聚类查询比例结果

为了评估 FAQs，本文需要适合的信息获取的评估策略。为了响应一个问题，本文的系统搜索与它相关的文档收集并且返回一个答案排序列表。实验中，本文构建数据集来评估 FAQs。本文发现本文的方法比 Google 和 AskJeeves 能更有效的构建 FAQs。本文利用前 10 个 urls 来进行评估。如果一个返回的 URL 存在于本文的数据集，本文给它一个衡量值 1。本文使用 100 个查询来做实验。那么总分为 100。表 6.2 给出本文的实验结果。

表 6.2 实验结果

Table 6.2 The results of our experiment

	Our FAQs	Google	AskJeeves
The total	100	100	100
score	65	54	36
Average score	0.65	0.54	0.36

实验中，本文目前只是证实本文方法在本文的数据集合上的有效性。在以后的工作中，本文将使用真正的搜索引擎日志来进行实验，另外，将本文的 FAQs 集成到本文的问答系统中，从而提高效率。

6.3 基于元搜索引擎信息获取

为了测试本文提出的基于 FCA 的元搜索引擎 MySearch，本文在计算机类中选取了 100 个样例查询词。MySearch 的性能通过获取的结果集的相似度来衡量。总的相似度主要包括两个方面。一个方面是平均点击率。它表示最高排序列表中的兴趣结果集的平均点击数。如果这个平均点击率很小，本文相信用户能够在排序列表中的前面部分找出有用的结果。另一个方面是返回结果集的内容相似度。这个相似度为根据日志数据找出的两个部分。为了计算结果集合的相似度，本文给出下面的公式：

$$relevancy\ (i) = \alpha * \frac{C}{R} + \beta * \frac{t}{T} \qquad (1)$$

这里，i 表示第 i 个关键字，C 是点击的 Url 数量；R 是返回的 Url 数量。T 是点击的 Url 浏览的时间。T 是关于该关键字浏览相应网页的总共时间。实验中，使用 $\alpha = \beta = 1/2$。图 6.4 表示 MySearch 与其它搜索引擎关于平均点击率的比较。

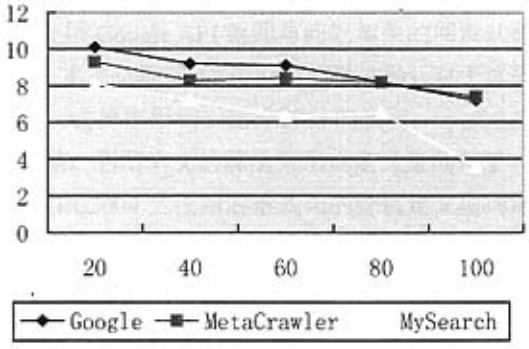

Figure 6.4 The average click rate of MySearch comparing to the other search engines

图 6.4 MySearch 与其它搜索引擎关于平均点击率的比较

图 6.5 给出 MySearch 与其它搜索引擎关于相似度的比较。实验中，取 $\delta = 1.5$；$\xi = 0.4$；$\eta = 0.3$；$\alpha = 0.1$；$\beta = 0.01$；和 $\gamma = 0.8$。本文认为该取值是较合理的。

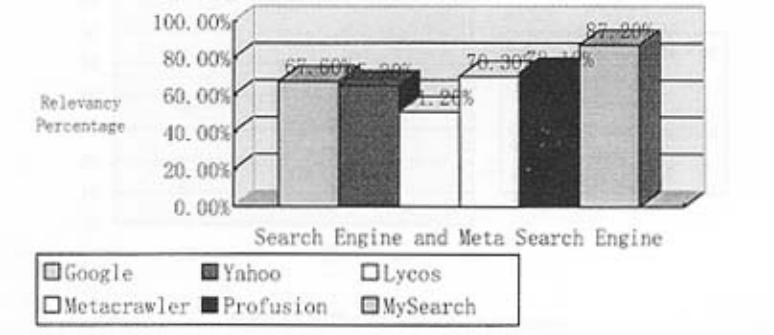

Figure 6.5 The Relevancy of the Results of MySearch Comparing to the Other Search Engines and Meta Search Engine

图 6.5 MySearch 与其它搜索引擎关于相似度的比较

6.4 利用协作推荐回答定义型问题

实验中，本文使用 Google API 返回与问题相关的网页从而提高本文问答系统的准确率。首先，本文选择前 10 个网页来做实验。对于这些网页的预处理主要包括切词，分词，去掉停用词。这个部分的实现，本文主要使用 ICTCLAS 切词分词软件来实现。利用本文的算法来形成定义型的句子。最后，本文使用在 TREC2003 和 TREC2004 上使用的形式化的评估定义型问题的方法来评估本文的答案。本文的实验结果证明本文的算法比使用一般的协作推荐算法[59]，[60]有效。实验结果如图 6.6。X-轴表示定义型问题的数量，Y-轴表示答案的正确率。这里的正确率是指答案正确的问题与所有的问题数的百分比。这里，只要答案中包括与本文数据集正确答案有 60%相同的单词，本文就认为该答案是正确的。

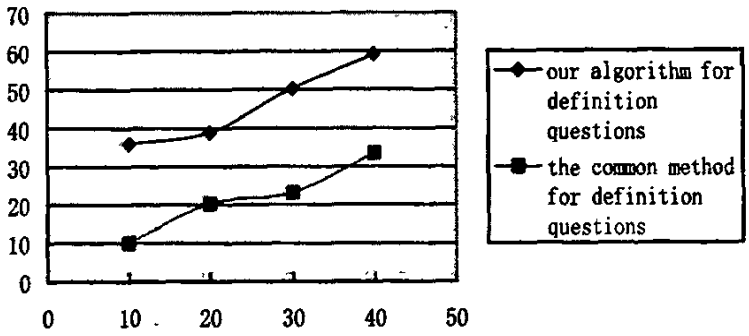

Figure 6.6 The Results of Experiment

图 6.6 实验结果

该实验只是对征对定义型问题提出了一个协作推荐算法的实现与验证。初步验证，该算法效果较好，值得进一步研究。在以后的工作中，本文计划将进一步实验于仿真陈述型，列表型问题。另外，在答案的准确度问题上，本文的方法有待改进。本文希望利用 WordNet 来对答案进行优化。

总结与展望

Myaskanswer 作为一个以 Web 为信息源的自然语言问答系统，使用了 FCA 来处理问答系统的答案抽取。在抽取处理中，本文首先在 FAQs 中寻找问题，如果该问题相应的答案不能满足用户的需要，本文再通过搜索引擎从网上获取相关的文档，从而使用返回的最相关的前 n 个文档建立概念格。最后，利用概念匹配在格中抽取答案。对于不同的问题，本文使用了不同的抽取策略。

本文的问答系统使用了概念化的聚类用户的日志来构建 FAQs。这个是本文问答系统的一个部分。本文使用用户的日志构建形式背景。本文试图根据它们的内容信息和文档点击来聚类相似的问题/查询。本文假设查询/问题和点击文档之间存在相关性。本文的方法基于两个原则：（1）如果两个查询导致了相同的文档点击，就被认为他们是相似的或者相关的，从而合并他们作为内涵；（2）如果一个文档集合被频繁的由相同的查询选取，就认为这些文档是相关的，从而合并他们作为外延。这两个原则与传统的查询内容相结合，从而构建初始的概念格。由于日志信息的特点，所以本文使用基于 DBSCAN 聚类算法来获取更好的性能。在 FAQs 中，本文使用导航来为问题抽取答案。

在信息获取方面，本文介绍了一种新的基于 FCA 的个性化的元搜索引擎，MySearch。它获取用户的信息，通过重排结果，提供一个实时的响应。重排是通过使用用户的使用日志和源搜索引擎返回的结果共同组建的概念格实现的。最后，改进的重排通过 MySearch 返回给用户。

另外，介绍了一种新的方法来回答定义型的问题。本文利用基于形式概念分析的协作推荐来回答定义型的问题。推荐是计算机系统设计来帮助用户在一个大面积的可选择领域中找到更适合的项。在协作推荐系统中，本文应用文档和问题之间的关系作推荐。FCA 组建文档和查询为概念，通过概念格来排序。

通过实验，这种混合的策略和多种资源，使本文的答案抽取系统获取了初步的成效。

在以后的研究中，本文还需要对其他部分做深入研究，需要引入其他技术，如，自然语言处理技术，XML 等等。从而可以获取准确率更高的答案。另外，在系统的实现方式上，本文可以考虑与其他硬件相结合，应用在移动通讯设备上，做到利用语音实现提问与回答。

参考文献

[1] Search Engine. Online at http://www.google.com, 2006.

[2] Web Search Engine. Online at http://www.baidu.com, 2006.

[3] Web Search Engine. Online at http://www.yahoo.com, 2006.

[4] Web Search Engine.Online at http://www.zhongsou.com, 2006.

[5] E. Brill, S. Dumais and M. Banko, An analysis of the AskMSR question-answering system. In Proceedings of 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002).

[6] Ellen M. Voorhees. Overview of the TREC 2002 question answering track. In *Proceedings of the Eleventh Text Retrieval Conference*. 2003.

[7] Voorhees, E. 2003. Overview of the TREC 2003 question answering track. TREC.

[8] Ravichandran, D., and Hovy, E. 2002. Learning surface text patterns for a question answering system. ACL.

[9] Moldovan. D, Clark. D, Harabagiu. S, and Maiorano. S, "Cogex: A logic prover for question answering", ACL, 2003.

[10] Xu. J, Licuanan. A, and Weischedel. R, "Answering definitional questions", Trec2003 qa, 2003.

[11] Azari. D, Horvitz. E, Dumais. S.T, and E. Brill, "Web-based question answering: A decision making perspective", UAI, 2003.

[12] Chu-Carroll. J, Czuba. K, Prager. J, and Ittycheriah. A, "In Question answering, Two heads are better than one" , Proceeding of HLT-NAACL, 2003,pp24-31.

[13] Burger. J, Ferro. L, Greiff. W, Henderson. J, Light. M, and Mardis. S, "Mitre's qanda at trec-11", TREC, 2002.

[14] Miller. G, "Wordnet: A lexical database for English", Communications of the ACM, Vol 38, No. 11, pp. 39-41, 1995.

[15] Brill. E, Lin. J, Banko. M, Dumais. S, and Ng A, "Data-intensive question answering", Proceedings of the Tenth Text REtrieval Conference (TREC 2001), 2001.

[16] Kwok. C, Etzioni. O, and Weld. D, "Scaling question answering to the Web", Proceedings of the 10th World Wide Web Conference (WWW'10), 2001,pp150-161.

[17] Clarke. C, Cormack. G, and Lyman. T, "Exploiting redundancy in question answering", Proceedings of SIGIR'2001, 2001,pp358-365.

[18] Clarke. C, Cormack. G, Lynam. T, Li. C.M, and McLearn G, "Web reinforced question answering (MultiText experiments for TREC 2001)", Proceedings of the Tenth Text REtrieval Conference (TREC 2001), 2001.

[19] Voorhees, E.: Overview of the TREC-2001 Question Answering Track. In Proceedings of the TREC-10 (2001) pp42–51

[20] Ravichandran, D., Ittycheriah, A., and Roukos, S.: Automatic Derivation of Surface Text Patterns for a Maximum Entropy Based Question Answering System. In Proceedings of HLT-NAACL, Edmonton, Canada (2003)pp85–87

[21] Echihabi. A, and Marcu. D, "A noisy-channel approach to question answering", ACL, 2003.

[22] Lita, L. V., and Carbonell, J, Instance-based question answering: A data-driven approach. EMNLP. 2004.

[23] Chu-Carroll. J, Czuba. K, Prager. J, and Ittycheriah. A, "In Question answering, Two heads are better than one" , Proceeding of HLT-NAACL, 2003,pp24-31.

[24] Burger. J, Ferro. L, Greiff. W, Henderson. J, Light. M, and Mardis. S, "Mitre's qanda at trec-11", TREC, 2002.

[25] Harabagiu. S, Moldovan. D, Pasca. M, Mihalcea. R, Surdeanu. M, Bunescu. R, Girju. R, Rus. V, and Morarescu. P, "Falcon: Boosting knowledge for answering engines", TREC, 2001.

[26] Ganter. B, and Wille. R, "Formal Concept Analysis. Mathematical Foundations", Springer-Verlag Berlin Heidelberg, 1999.

[27] Wille. R, "Restructuring lattice theory: an approach based on hierarchies of concepts", Rival I.(Ed): Ordered Sets, Reidel, Dordrecht, Boston, 1982, pp445-470,.

[28] Godin, R., and Missaoui, R.: An incremental concept formation approach for learning from databases. TCS, Vol. 133, No.2 (1994) 387–419

[29] Carpineto, C., Romano, G.: GALOIS: An order theoretic approach to conceptual clustering. In Proc. ICML 1993, Morgan Kaufmann Prublishers (1993) 33–40

[30] Neuss C, Kent R E. Conceptual Analysis of Resource Meta-Information [EB/OL]. http://www.igd.fhg.de~neuss.1999.

[31] Eklund P W, Martin P. WWW indexation and document navigation using conceptual structures [A]. 2nd IEEE Conference on Intelligent Information Processing Systems (ICIPS'98)[C]. IEEE Press, 1998.pp217-221.

[32] Lengnink, K.. Ahnlichkeit als Distanz in Begriffsverbanden. In G. Stumme,& R. Wille (Eds.), Begriffliche Wissensverarbeitung: Methoden und Anwendungen. 2000 , Berlin-Heidelberg: Springer.

[33] Carpineto, C. and Romano, G. (2000). Order-theoretical ranking, Journal of the American Society for Information Science 51(7): pp587-601.

[34] Wille R. Restructuring mathematical logic: an approach based on Peirce's pragmatism. In: Urshi A, Agliano P(eds.): Logic and Algebra. Marcel Dekker, New York 1996. pp.267-281.

[35] Richards D, Compton P. Combining formal concept analysis and ripple down rules to support reuse, Software Engineering Knowledge Engineering SEKE'97, Springer Verlag, Madrid, 18-20th June 1997.

[36] Cole R, Stumme G. CEM- A conceptual email manager. In: Proc.7th Internatioal Conference on Conceptual Structures (ICCS'2000), Springer Verlag.

[37] Juan. M, Anselmo. P, Julio. G, and Felisa. V, "Automatic selection of Noun Phrases as Document Descriptors in an FCA-Based Information Retrieval System", ICFCA, LNCS 3403, 2005,pp49-63.

[38] SALTON. G, and MCGILL. M.J, "Introduction to Modern Information Retrieval", McGraw-Hill New York, NY, 1983.

[39] Porter. M, "An algorithm for suffix stripping", Program.14. 3, 1980, pp. 130–137

[40] M. Ester, H. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," *In Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*, 1996, pp. 226–231.

[41] D.H. Fisher, "Knowledge acquisition via incremental conceptual clustering," *Machine Learning*, Vol. 2, 1987, pp139-172,.

[42] P. Cheeseman, J. Kelly, M. Self, J. Stutz, W. Taylor, and D. Freeman, "AutoClass: A bayesian classification system," *In Proceedings of the Fifth International Workshop on Machine Learning Morgan Kauffmann*, San Mateo, CA, 1988, pp54-64.

[43] J.H. Gennari, P. Langley, "Fisher: Models of incremental concept formation," J.Carbonell, Editor, *Machine.Learning: Paradigms and Methods*, the Netherlands: MIT Press, 1990, pp11-62,.

[44] Y. Reich, S.J. Fenves, "The formation and use of abstract concepts in design," In D.H. Fisher and M.J. Pazzani, Editors. *Concept Formation: Knowledge and Experience in Unsupervised Learning*, Morgan Kaufmann, 1991,pp323-353.

[45] C. Li, G. Biswas, "Conceptual clustering with numeric and nominal mixed data - A New Similarity Based System," *IEEE Transactions on Knowledge and Data Engineering*, 1996.

[46] F. Vogt, R. Wille, "TOSCANA: a graphical tool for analyzing and exploring data," In R.Tamassia and I. G. Tollis, editors. *GraphDrawing' 94*. Heidelberg, pp. 226-233, 1995.

[47] T.B. Ho, "Incremental conceptual clustering in the cramework of galois lattice," In H. Lu, H. Motoda, H. Luu, editors, *KDD: Techniques and Applications*, *World Scientific*, 1997, pp49-64,.

[48] X. Wensi, C. Abdur, S. Kush, and P.Greg, "Query log analysis," *Lecture notes in computer science*, 2002, pp1-5.

[49] Voorhees, E.M.: Evaluating Answers to Definition Questions. In Proceedings of HLTNAACL, Edmonton, Canada (2003) pp109–111

[50] Cosley, D., Lawrence, S., and Pennock, D.: An open framework for practical testing of recommender systems using Researchlindex. In International Conference on Very Large Databases (VLDB) (2002)

[51] Resnick, P., Iacovou, N., Suchak, M., Bergstorm, P., and Riedl, J.: GroupLens: An Open Architecture for Collaborative Filtering of Netnews. In Proceedings of ACM 1994 Conference on Computer Supported Cooperative Work, Chapel Hill, North Carolina (1994) pp175–186

[52] Shardanand, U., and Maes, P.: Social information filtering: Algorithms for automating "word of mouth". In Proceedings of ACM CHI'95 Conference on Human Factors in Computing Systems. Vol. 1 (1995) ,pp210–217

[53] Breese, J., Heckerman, D., and Kadie, C.: Empirical analysis of predictive algorithms for collaborative filtering. In Proceedings of UAI San Francisco, CA: Morgan Kaufmann Publishers ,1998,pp 43–52

[54] Heckerman, D., Chickering, D., Meek, C., Rounthwaite, R., and Kadie, C.: Dependency

networks for density estimation, collaborative filtering, and data visualization. Journal of Machine Learning Research. Vol. 1 (2000) 49—75

[55] Sarwar, B., Karypis, G., Konstan, J., and Riedl, J.: Analysis of recommender algorithms for e-commerce. In Proceedings of the 2nd ACM Conference on Electronic Commerce (2000) pp158–167

[56] Hofmann, T., and Puzicha, J.: Latent class models for collaborative filtering. In Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (1999) 688–693

[57] Popescul, A., Ungar, L., Pennock, D., and Lawrence, S.: Probabilistic models for unified collaborative and content-based recommendation in sparse-data environments. In Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence (2001) 437–444

[58] Giuseppe, C., Jocelyn, S., David, P.: Towards more Conversational and Collaborative Recommender Systems. In Proceedings of the International Conference of Intelligent User Interfaces, Miami, Florida, USA (2003) 12-18

[59] Dmitry, Y.P, David, M.P.: A Maximum Entropy Approach To Collaborative Filtering in Dynamic, Spares, High-Dimensional Domains. In Proceeding of the 16th Annual Conference on Neural Information Processing Systems (NIPS) (2002)

[60] Patrick, B.R., Derek, B.: Collaborative Recommending using Formal Concept Analysis. Knowledge-Based Systems. Elsevier B.V. (2006) 1-7

[61] Herlocker, J.L.: Understanding and Improving Automated Collaborative Filtering Systems. Ph.D. thesis, University of Minnesota (2000)

[62] Ng,H.T., Kwan, J.L.P., and Xia, Y: Question Answering Using a Large Text Database: A Machine Learning Approach. In Proceedings of EMNLP 2001, Pittsburgh, PA (2001) 67–73

[63] Ravichandran, D., Ittycheriah, A., and Roukos, S.: Automatic Derivation of Surface Text Patterns for a Maximum Entropy Based Question Answering System. In Proceedings of HLT-NAACL, Edmonton, Canada (2003) pp85–87

[64] Blair-Goldensohn, S., McKeown, K.R., and Schlaikjer, A.H.: A Hybrid Approach for Answering Definitional Questions. Technical Report CUCS-006-03, Columbia University (2003)

[65] Spyridoula, M., Ion, A.: Learning to Identify Single-Snippet Answers to Definition

Questions. In processdings of the 20th international conference on Computational Linguistics (COLING). Geneva, Switzerland (2004) 1360-1366

[66] Wesley, H., Boris, K., and Jimmy L.: Answering definition questions with multiple knowledge sources. In Proceeding of HLT/NAAC (2004)

[67] Jinxi, X., Ralph, W., and Ana, L.: Evaluation of an extraction-based approach to answering definition questions. In Proceeding of SIGIR (2004)

[68] Jimmmy, L., Dina, D.F.: Automatically Evaluating Answers to Definition Questions. In proceedings of the 2005 Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP), Vancouver, Canada (2005).

[69] Yajun D.: Research and implement the intelligent activity of search engine, the dissertation of doctor in Southwest Jiaotong University, 2005, pp47-50.

[70] Bordat, J. P.: Calcul pratique du treillis de galois d'une correspondance. Mathématiques et Sciences Humaines, 1986 (96):pp31-47.

[71] S. Brin and L. Page. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 1998,30(1–7):pp107–117.

[72] D. Dreilinger, A. Howe, Experiments with selecting search engines using metaserach, ACM Transactions on Information Systems 15 (3) (1997) .pp.195–222.

[73] M. Meng, K. Liu, C. Yu, W. Wu, N. Rishe, Estimating the usefulness of search engines, in: Proceedings of the IEEE International Conference on Data Engineering, Sydney, Australia, 1999, pp146–153.

[74] K. Liu, C. Yu, W. Meng, W. Wu, N. Rishe, A statistical method for estimating the usefulness of text databases, IEEE Transaction on Knowledge and Data Engineering , 2002, 14 (6):pp1422–1437.

[75] SavvySearch Web site, http://savvy.cs.colostate.edu, 2000.

[76] Ithaki Web site, http://www.ithaki.net/dir.html, 2006.

[77] ProFusion, Web site, http://www.profusion.com, 2006.

[78] Mamma Web site, http://www.mamma.com, 2006.

[79] MetaCrawler Web site, http://www.metacrawler.com, 2006.

[80] Amir H.K, Behzad M, Majid K, Maryam P, and Caro L, "Aggregation of web search engines based on user's preferences in webfusion", knowledge-based Systems, 2006, doi: 10.1016/j. knosys. 2006.08.001.

[81] Gondy L, Jennifer X, Wingyan C, Shauna E, and Hsinchun C, "An end user evaluation of query formulation and results review tools in three medical meta-search engines", International Journal of Medical Informatics, 2006, doi:10.1016/j.ijmedinf.2006.08.001.

[82] Yajun D, "Research and implement the intelligent activity of search engine", the dissertation of doctor in Southwest Jiaotong University, 2005, pp: 47-50.

[83]张卫丰，徐宝文，周晓宇等. Web 搜索引擎综述. 计算机科学. 2001，28（09）：pp24-29.

[84]杜亚军，邱小平，徐扬. 中文搜索引擎智能的探讨. 计算机应用研究. 2004, 21(4)：pp29-31.

致 谢

本文在导师杜亚军教授的悉心指导下完成的。三年来，作者学会了很多，不管在学习上还是生活上，三年中的每一天，作者都成长着，进步着。在此，作者首先感谢导师在学习上和生活上给予了无微不至的关怀。因为有了导师的悉心指导，严格要求，作者才在学习上和生活中有了如此大的进步。而且导师做事谨慎，对任何事情，无论大小，都非常认真，还有导师的治学态度，对工作的认真负责，等等使作者受益终身。同时也很感谢裴峥教授，彭宏教授在学术讨论中的指导和建议。另外，感谢董占兵，海宇峰，项磊同学的帮助，在与他们的交流过程中，作者吸取了很多的思想和启发。同时，也感谢2004级所有同学及朋友在生活上，学习上给作者的帮助。最后，感谢作者的家人在生活上给予我的帮助。

感谢王可亮对我的理解和支持。

最后，郑重感谢这3年来一起参加作者所在的研究团队每周一次学术讨论的所有师兄、师弟、师姐、师妹们，作者在与他们的讨论中受益匪浅。