分类号 TP391 学号 07069013

UDC 密级 公开

工学博士学位论文

面向搜索引擎的自然语言处理关键技术研究

博士生姓名 李莎莎

学科专业 计算机科学与技术

研究方向 智能信息处理

指导教师 李舟军教授

国防科学技术大学研究生院

二〇一一年十月

Research on Search Engine Oriented Natural Language Processing Technology

Candidate：Shasha Li

Supervisor：Prof. Li Zhoujun

A dissertation

Submitted in partial fulfillment of the requirements

for the degree of Doctor of Engineering
in Computer Science and Technology

Graduate School of National University of Defense Technology

Changsha，Hunan，P.R.China

October，2011

独创性声明

本人声明所呈交的学位论文是我本人在导师指导下进行的研究工作及取得的研究成果。尽我所知，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和撰写过的研究成果，也不包含为获得国防科学技术大学或其它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。

学位论文题目：面向搜索引擎的自然语言处理关键技术研究

学位论文作者签名：_____ 日期：　　年　　月　　日

学位论文版权使用授权书

本人完全了解国防科学技术大学有关保留、使用学位论文的规定。本人授权国防科学技术大学可以保留并向国家有关部门或机构送交论文的复印件和电子文档，允许论文被查阅和借阅；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。

(保密学位论文在解密后适用本授权书。)

学位论文题目：面向搜索引擎的自然语言处理关键技术研究

学位论文作者签名：_____ 日期：　　年　　月　　日

作者指导教师签名：_____ 日期：　　年　　月　　日

目录

摘要 i

Abstract v

第一章 绪论 1

1.1 研究背景 1

1.2 研究意义 3

1.3 本文的主要贡献 5

1.4 本文的组织结构 8

第二章 相关研究工作 11

2.1 引言 11

2.1.1 查询推荐 11

2.1.2 查询意图分类 11

2.1.3 查询理解 11

2.1.4 自动问答 12

2.2 国内外研究现状 12

2.2.1 查询推荐研究现状 12

2.2.2 查询意图分类的研究现状 13

2.2.3 查询理解的研究现状 14

2.2.4 自动问答技术的研究现状 14

2.3 现有工作的盲区与不足 14

第三章 基于比较关系的查询推荐 17

3.1 引言 17

3.2 相关工作 18

3.2.1 概述 18

3.2.2 Jindal&Liu 2006 19

3.3 弱监督的可比较实体挖掘方法 21

3.3.1 挖掘指示性抽取模式 22

3.3.2 比较对象抽取 25

3.4 比较对象排序 26

3.4.1 基于可比较性的排序方法 27

3.4.2 基于图的排序方法 27

3.5 实验 28
3.5.1 实验设置 29
3.5.2 实验结果 31
3.6 本章小节 39
第四章 基于比较关系的用户查询意图识别 41
4.1 引言 41
4.2 相关工作 42
4.2.1 查询意图分类 42
4.2.2 查询消歧 43
4.3 基于比较关系的用户意图识别方法 43
4.3.1 别名识别 44
4.3.2 比较对象聚类 46
4.3.3 聚类的语义标记 47
4.3.4 比较意图排序 48
4.4 实验及系统介绍 48
4.4.1 实验设置及实验结果 48
4.4.2 系统介绍 50
4.5 本章小节 51
第五章 面向开放域的无监督的查询理解 53
5.1 引言 53
5.2 相关工作 54
5.3 语义词典的建立 55
5.4 基于查询模式的词义消歧 55
5.4.1 非监督的查询模式挖掘 56
5.4.2 查询记录中的查询词词义消歧 59
5.5 实验 61
5.5.1 数据 61
5.5.2 实验结果 62
5.5.3 应用实例 62
5.6 本章小节 63
第六章 基于词的依赖层次的面向问题的答案摘要 65
6.1 引言 65
6.2 相关工作 66

6.3 基于词的依赖层次的文本摘要 67
6.3.1 过滤通用词 69
6.3.2 过滤问题背景词 69
6.3.3 答案话题相关词汇识别 70
6.3.4 标准化及参数设定 71
6.4 实验和评测 72
6.4.1 数据集 72
6.4.2 基准算法 73
6.4.3 评测指标 73
6.4.4 实验 74
6.5 本章小节 77
第七章 结论与展望 79
7.1 本文总结 79
7.2 研究展望 80
致谢 83
参考文献 87
作者在学期间取得的学术成果 93
作者在学期间参与的科研工作 95

表目录

表 3.1 候选序列模式 21
表 3.2 词组识别的启发性规则 23
表 3.3 不同类型比较对象的分布 29
表 3.4 弱监督自举算法与 J&L 方法的性能比较 31
表 3.5 特殊化模式及泛化模式对系统性能的影响 32
表 3.6 不同抽取模式选择策略的性能比较 32
表 3.7 不同初始模式对系统性能的影响 32
表 3.8 不同比较对象对个数对系统能能影响 33
表 3.9 查询的可比对象列表实例 36
表 3.10 Google 相关查询推荐实例 36
表 3.11 基于可比性和基于图的排序方法结果比较 37
表 3.11 基于可比性和基于图的排序方法结果比较(续) 38
表 3.12 参数对基于图的排序方法的影响 38
表 4.1 别名关系列表 45
表 5.1 查询模式实例 56
表 5.2 模板示例 56
表 5.3 基于查询模板的查询理解方法与基准方法的性能比较 62
表 5.4 查询理解实例 63

图目录

图 3.1 自举算法概览 22
图 3.2 查询“Obama”的可比较对象关系图 28
图 3.3 随着可比对象 E 在排序结果中序的变化查询“E vs Q”和“Q vs E”检索出的网页的个数。 35
图 4.1 基于比较关系的用户比较意图识别系统框架 44
图 4.2 别名聚类 46
图 4.3 比较关系图 47
图 4.4 针对用户比较行为的比较搜索系统 50
图 6.1 资询问题实例 68
图 6.2 词的层次结构 69
图 6.3 摘要生成的答案与“best answer”的召回率的比较 74
图 6.4 答案召回率比较 75
图 6.5 答案准确率比较 75
图 6.6 答案精确度比较 76
图 6.7 三种方法#N 答案召回率的比较 76

摘要

随着网络技术的飞速发展，互联网上的信息资源越来越庞大，用户越来越难以准确快捷地获取所需信息，从而产生了所谓的信息焦虑。互联网搜索引擎以其基于关键词匹配的信息检索机制为用户在瞬间搜寻出所需的相关信息，成为缓解人们信息焦虑最为有效的工具。目前，搜索已成为互联网上的一种日常活动，并带来了巨大的商机。然而，面对网络上越来越多样化的信息，基于关键字查询的搜索引擎存在和凸显出如下主要的缺点：难以构造出准确表达用户需求的查询请求，返回结果冗余甚至无用信息多，查询主观性信息的性能低下。为了最大程度地满足用户需求，面向人性化、智能化、个性化的第三代搜索引擎技术应运而生，并得到了广泛关注。

近年来，随着搜索由基于关键字层面向基于知识层面转化，面向搜索引擎的自然语言处理技术逐渐成为新的研究热点，其研究的主要关注点在于：更好地理解用户的查询意图，更准确地推荐相关查询请求，返回更相关的查询结果，更好地筛选和组织查询结果等，力求提供更智能化、更人性化的人机交互，辅助用户更加便捷地找到想要的信息。本文在相关的查询推荐、查询意图识别、查询理解、重用问题答案库时的答案摘要等方面进行了深入而细致的研究。本文工作的主要贡献和创新点总结如下：

1. 研究基于比较关系的查询推荐，并提出一种弱监督的从比较性问题中挖掘比较对象的方法

通常的查询推荐（Query Suggetion）都是推荐与用户初始查询请求相关的查询请求，例如，在用户搜索“ipod touch”时，搜索引擎推荐“ipod touch break prison”。然而，在不同的搜索场景下，用户所需要的相关查询是不同的。例如，在购买商品的产品搜索场景下，当用户搜索“你 Nikon d200”时，用户往往想要了解该产品的相关信息，并将其与同类产品作比较，从而做出购买决策。这时，推荐“Cannon 300d”并提供两者之间的对比信息，将有利于用户尽快做出购买决策。当然，“nokia d200 lens”也是有用的推荐，然而相比之下，类似“Cannon 300d”的查询推荐，往往需要用户具有一定的知识储备，对同类产品有一定了解，这也往往是用户最为缺乏的知识，用户对此类推荐的需求更为迫切。由此可见，将用户初始查询请求的相关查询请求按其与初始查询请求的语义关系进行分类，根据搜索场景的不同给予不同类别的查询推荐，将使得搜索引擎更加个性化、智能化。由于比较备选方案是人们的日常决策行为中至关重要的一个步骤，本文针对用户比较行为这一搜索场景，提出基于用户查询请求的可比较对象的查询推荐。

由于用户比较行为的主观性和复杂性，判断两个对象的可比性是困难的。幸

运的是，互联网上有网络用户产生的大量意图比较两个或多个对象的比较性问题，这些比较性问题提供了人们想比较什么的证据，例如“Which to buy, iPod or iPhone?”。在本文中，比较性问题中用于进行比较的对象被称为比较对象，例如上述例句中的“iPod”和“iPhone”。

为了从比较性问题中挖掘比较对象，本文首先需要识别一个问题是否是比较性问题。根据本文的定义，一个比较性问题必须是一个意在比较至少两个对象的问题。值得注意的是，包含两个对象的问题如果不包含比较的意图就不是比较性问题。但我们观察到：如果一个问题包含两个可比较对象，则该问题极有可能是比较性问题。本文利用这一点设计了一个弱监督的自举（bootstrapping）方法，可以在识别比较性问题的同时，抽取比较对象。

据我们所知，这是第一个意图通过推荐好的比较对象以支持网络用户的比较行为的研究。也是第一次提出使用网上的比较性问题作为媒介来反映用户真实关心的比较对象。本文提出的弱监督方法的 F1-measure 在比较性问题识别上达到 82.5%，在比较对象抽取上达到 83.3%，在从比较性问题识别到比较对象抽取的整体系统上达到 76.8%。

2. 利用对象之间的比较关系，提出一种基于图聚类的用户意图识别方法，并建立针对用户比较行为的比较搜索系统。

基于关键字查询的搜索引擎中，用户往往只能使用有限数量的词汇来抽象和概括他们的需求。在用户将其需求抽象成有限的查询关键字的过程中，部分有用信息被丢失，从而导致用户查询意图不慎清晰。目前，搜索引擎搜索的结果通常是满足用户各种查询意图的文档的合集，用户需要阅读大量相关文档才能获得自己确实所需的信息。因此，在对用户的查询请求进行搜索之前，尽可能确定用户的查询意图，并进行面向用户查询意图的搜索将有利于更准确地找到用户想要的信息。

如上所述，通常一个由一个或多个查询关键字组成的用户查询请求可能出于多种不同的意图。举个最经典的例子，当用户查询“apple”时，可能是查询一种水果，也可能是查询一种电子产品的品牌。而“apple”作为电子产品品牌时，用户可能想要查询“apple”的产品，也可能想要查询“apple”的网点分布。如果用户想要购买“apple”的产品，比如用户输入“apple itouch”，有可能是想搜索产品介绍，有可能是想看不同网站的价格比较，还有可能是想看该产品与其它同类产品的比较。即使我们确定用户查询的意图是想要将“apple itouch”与其它产品相比较，用户还可能想比较产品的不同方面。比如，从产品升级的角度，用户可能想将其与“ipod classic”，“iphone”相比较；从娱乐功能性的角度，用户可能想将其与“psp”相比较等。可见，确切地理解用户的意图并不是件简单的任务。

本文主要关注用户的比较行为，针对用户的查询请求，利用与其可比较的对象之间的比较关系，提出一种基于图聚类的用户意图识别方法，每个可能的用户意图由一组用户查询请求的可比较对象表示。利用信息抽取的方法，本文给识别出的每个用户意图赋予一个语义标签。实验证明，本文提出的用户查询意图识别算法的准确率达到92.7%以上。另外，本文还建立了一个基于用户比较意图识别的比较搜索系统，该系统在识别用户查询请求的不同比较意图的同时，还提供了不同的比较意图下，用户查询请求与相应的可比较对象的比较信息。

3. 研究面向开放域的查询理解，针对由多个查询关键词组成的复杂查询请求，提出一种无监督的基于查询模板的查询理解方法

在搜索引擎中，用户的查询请求除了单个实体或对象（如 Obama）以外，还存在大量的由多个查询关键词组成的复杂查询请求，例如“flight from Beijing to New York”。这些查询多是面向任务的，并要求得到精确的答案（如“0:00, Nov.19th, 2010”）。在现有的搜索引擎中，用户通常要进行如下操作：首先，检索相关网页或在线数据库；然后，通过逐一阅读相关网页或向相关的在线数据库提交数据库检索请求以寻找所需信息。为了简化这一繁琐的过程，研究者们提出了结构化检索。而查询理解是结构化检索的至关重要的一步。具体来说，本文所指的查询理解包括识别和消歧复杂查询中的查询关键字两个步骤。例如，给定查询“harry potter showtime in beijing”，本文先识别查询词“harry potter”，“showtime”，“beijing”（即查询关键字识别）；然后分辨每个查询词的含义并分别标记，这里“harry potter”被标记为“movie name”，“beijing”被标记为“city”，而“showtime”是电影的一个属性（即查询关键字词义消歧）。

本文主要关注如何在开放域的环境下理解用户的复杂查询请求：首先，利用现有技术自动创建一个语义词典；然后，尝试使用自动创建的语义词典进行开放域查询理解，即查询关键词识别与消歧。在本文的问题设定中，我们致力于解决以下两个问题：

1) 自动创建的语义词典无论在语义标签还是词或词组实例中都包含很多噪音，这些噪音将严重降低查询理解的性能。
2) 在开放域环境中，大量的语义标签是必需的。这使得以前用于处理有限数目的语义标签的基于序列标注技术的查询理解方法不再适用。

为了解决以上问题，本文提出一种基于查询模板的查询理解方法。该方法先利用一个无监督的互增强的算法挖掘查询模式；然后基于挖掘到的查询模式和语义词典进行查询关键字识别及消歧。据我们所知，本文的研究是第一个利用自动创建的语义词典进行查询理解的尝试。

4. 研究社区问答服务的知识重用中答案完备性问题，提出了一种基于词的依赖层

次的面向问题的答案摘要方法。

传统的搜索引擎在面对一些复杂问题的查询时，其查询结果往往差强人意，例如“how to recover my doc file?”或“what is the best smart phone?”。这些复杂问题通常需要人的经验或意见的参与，答案因人而异或因情况而异，没有唯一正确答案。而目前，社区问答服务的出现，为解决这一问题提供了新的资源。如何重用社区问答服务中积累的问题答案知识提高复杂问题查询的满意度成为该研究的一个研究热点。但是，目前的研究主要集中在评测社区问答服务积累的答案的准确性上，对于答案的完备性没有涉及。事实上，由于复杂问题的正确答案往往并不唯一，提供汇总了不同情况下不同个人的回答的完备答案对提高搜索满意度也是至关重要的。

本文尝试对一种特殊的问题——调研问题（survey questions）——进行答案摘要，以提高社区问答服务中可重用问题的答案的完备性。调研问题是指请求回答问题的用户推荐针对某种需求的最佳选择的问题。显然，对调研问题而言，答案的完整性至关重要，因为一方面不同的用户可能对不同的建议感兴趣，另一方面某个答案被推荐的次数也反映了该答案的值得推荐指数。

据我们所知，本文最先指出在社区问答服务的知识重用中答案完备性的重要性。同时，这也是第一个关注调研问题的研究，调研问题作为面向意见的问题中一种有趣的类型，问题的完整性对其而言至关重要。除此之外，本文推荐使用面向问题的答案摘要方式产生完整的答案。本文提出一种有效的建立词与词之间语义依赖性的层次结构的方法，并利用建立的层次结构进行面向问题的答案摘要，从而基于社区服务中用户提供的所有已有答案生成一个完整简洁的答案。

关键词：智能化搜索引擎；自然语言处理；查询推荐；查询意图识别；查询理解；答案摘要；信息抽取

Abstract

With the rapid development of Internet, information on the internet is ever-growing. It's becoming more and more difficult for Internet users to obtain required information accurately and quickly, which results in so-called information anxiety. Web search engines provide keywords matching based information retrieval mechanism for the users to assist them to get what they want instantly and have become the most efficient tools to get people rid of the information anxiety. Currently, web search is becoming a daily activity on Internet and has brought huge business opportunities. However, faced with the increasing variety of information on the network, the weakness of keyword-based search engine is becoming apparent, such as the difficulty of constructing a query accurately expressing user's information needs, the redundancy or useless of returned results and the low performance on retrieving subjective information. To meet users' needs to the best, the third generation of search engine, which is human-oriented, intelligent, personalized, has been widely studied.

In recent years, with the transferring from keyword-based searching to knowledge-based searching, natural language processing has become an emerging technique and a new hotspot. Natural language processing techniques in search engines mainly focus on query understanding, query reformulation, search result organization and etc. It strives to provide a more intelligent and more humanized human-computer interaction to assist users get required information more convenient. In this dissertation, we'll investigate on query suggestion, query intent identification, query semantic structure understanding and answer summarization in the reuse of Q&A achieves. The key contributions and innovations can be briefly summarized as follows:

1. Research on comparative relation based query suggestion and proposal of weakly-supervised method for comparator mining from comparative questions

Usually, query suggestion recommends queries relevant to user's original query. For example, search engine recommends "ipod touch break prison" to users when they launch query "ipod touch". However, in different search scenarios, users prefer different relevant queries. For example, in the scenarios of purchasing, when users launch a query like "Nikon d200", they usually want to know information about the product and compare it with comparable products to make a purchase decision. In this case, suggesting queries like "Cannon 300d" and providing corresponding comparison information are quite helpful for users to make a purchase decision quickly. Compared with "nokia d200 lens" which is also a useful suggestion, query suggestion "Cannon 300d" requiring users holding relevant knowledge is usually what users want to know. So, it will be meaningful to improve the performance on information retrieval and make the search engine more intelligence and personalized when we classify relevant query

suggestions by their semantic relations with user original query and provide different kinds of suggestion in different scenarios. Considering that comparing candidates is an essential step in users' decision making behaviors, we focus on the comparison search scenarios and investigate query suggestion based on comparison relations.

In general, it is difficult to decide if two entities are comparable or not due to the subjectivity and complexity of comparison. Fortunately, plenty of comparative questions which intend to explicitly compare two or more entities are posted online. Those comparative questions provide evidences for what people want to compare, e.g. "Which to buy, iPod or iPhone?". We call entities which are targets of comparison in comparative questions as comparators, such as "iPod" and "iPhone" in above example.

To mine comparators from comparative questions, we first have to detect whether a question is comparative or not. According to our definition, a comparative question has to be a question with intent to compare at least two entities. Please note that a question containing at least two entities is not a comparative question if it does not have comparison intent. However, we observe that a question is very likely to be a comparative question if it contains at least two potentially comparable entities. We leverage this insight and develop a weakly supervised bootstrapping method to identify comparative questions and extract comparators simultaneously.

To our best knowledge, this is thefirst attempt to specially address the problem on finding good comparators to support users'comparison activity. We are also the first to propose using comparative questions posted online that reflect what users truly care about as the medium from which we mine comparable entities. Our weakly supervised method achieves 82.5% F1-measure in comparative question identification, 83.3% in comparator extraction, and 76.8% in end-to-end comparative question identification and comparator extraction.

2. Proposal of a graph clustering based user intent detection methods by utilizing comparison relations and construction of a comparison behavior oriented comparison information retrieval system.

In keyword-based search engines, people are asked to utilize queries consisting of limited keywords to describe their information needs. Due to the information loss during the abstraction process from user needs to keywords, the search intent expressed in a query may be not clear. Currently, search engines usually return a mixed set containing documents relevant to various query intent. Users need to browse a large number of documents to find what exactly meet their search intents. So, determining user's search intent and performing intent-oriented information search will help users to acquire information more accurately and quickly.

As discussed above, there may be multiple user intents behind a query. For example, query "apple" may search for a kind of fruit or an electronic brand. When "apple" means an electronic brand, user who launches query "apple" may intents to learn

products of apples or know the location of apple stores. If a user want to purchase an apple product, for example, the user launch a query “ipod touch”, he may want to know relevant product information, or compare prices on different web sites, or compare the product with other products. And even when we’re sure a user want to compare the queried “ipod touch” with other products, users may want to compare products from different aspects. For example, in terms of product updates, people may want to compare “ipod touch” with “ipod classic” and in terms of entertainment, people may want to compare “ipod touch” with “psp”. All in a word, it is not a trivial task to understand user’s intent clearly.

In this dissertation, we focus on users’ comparison behaviors and proposed a graph clustering based user intent detection methods by utilizing comparison relations. User’s query intent is expressed by a set of comparators to the original query. A semantic label is assigned to the detected query intent utilizing an information extraction method. Experiments show that the accuracy of intent detection comes up to 92.7%. In addition, we build a user comparison intent detection system which provides different comparators and corresponding comparison information for the given query under different comparison intent.

3. Research on query understanding in open domain and proposal of multi-term queries oriented pattern-based methods of query understanding.

Besides entity queries, there are amounts of complexity queries consisting of multiple query terms, e.g., “flight from Beijing to New York”. To determining intents for this kind of queries, we need to recognize and disambiguating each query term. Especially, search engines have crawled a lot of structured data which is less ambiguous in nature. When search against structured data, it is beneficial to covert keyword queries into SQL-like queries, for which query term recognizing and disambiguation is essential. We refer to the process of recognizing and disambiguating query terms as query understanding. For example, given a query “*harry potter showtime in beijing*”, we firstly need to recognize “*harry potter*”, “*showtime*” and “*beijing*” as query terms, and then it is necessary to disambiguate the semantics of terms with relevant labels, e.g., “*harry potter*” as “*movie name*”, “*beijing*” as “*city*” and “*showtime*” is an attribute term for a movie.

In this dissertation, we focus on query understanding for multi-term queries in open domain. We firstly construct a semantic dictionary with existing methods; and then examine open domain query understanding (namely query term recognition and disambiguation) via the dictionary. In particular, we focus on addressing the two problems followed by our problem setting.

(1) Automatically constructed lexicons would contain much noisy in both labels and term instances. Such noisy can seriously deteriorate query understanding performances.

(2) The vast amount of labels is necessary in open domain environment and makes it hard to apply the previous query understanding approaches based on sequential labeling techniques, which are originally developed to deal with limited amount of term labels.

To resolve such a problem, we propose a pattern-based method to recognize a term and disambiguate its labels. In our approach, we firstly construct semantic lexicons by applying one developed method to extract hyponymy relations. Then, we propose a mutual reinforcement algorithm to mine context patterns. Based on the mined context patterns and semantic lexicons, we perform term recognition and disambiguation. To our knowledge, our study is the first attempt to try to understand open-domain queries utilizing automatically mined lexicons.

4. Research on answer completeness in the process of reusing Q&A resources collected by Community Question Answer (cQA) services and proposal of question oriented answer summarization based on hierarchical structure of semantically dependency among terms.

Traditional search engines don't work as well as expected on complex question queries, e.g., "how to recover my doc file", "what is the best smart phone?" and etc. These complex questions usually related to personal experiences or opinion and have different answers from different individuals. Fortunately, the appearances of cQA services provide large knowledge resources for such kind of questions. How to reuse Q&A archives in cQA services to improve satisfaction on complex question queries has become an attractive research field. However, current researches mainly focus on assessing whether answers in cQA are accuracy enough to be reused, and ignore the completeness of answers. In fact, since the answers of complex questions are not unique, the completeness of answers is also a critical factor for enhancing satisfaction of information retrieval.

In this paper, we try to do answer summarization for a particular type of questions: survey questions, which ask for recommendations on best choices. Obviously, the completeness of the answer is crucial because different users may be interested in different choice suggestions.

To our best knowledge, it's the first research pointing out the importance of answer completeness in cQA knowledge reuse. We are also the first to focus on survey question which is an interesting type of opinion questions and completeness of whose answers are potentially important for better reuse. Additionally, we recommend generating complete answers by question-oriented answer summarization. We propose an efficient algorithm to build hierarchical structure of semantically dependency among terms and perform question-oriented summarization via the structure to generate a complete answer based on existing answers from users in cQA services. The performance is promising.

Key words：Intelligence Search Engine，Natural Language Processing，Query Suggestion，Query Intent Detection，Query Understanding，Answer Summarization，Information Extraction

第一章 绪论

1.1 研究背景

目前，对大多数人来说，在 Web 上搜索信息是计算机的基本应用之一，已成为一项必不可少的日常活动。而搜索引擎（Search Engine）[1]作为互联网搜索的有效工具其相关研究在国内外获得了广泛关注。国际上，众多知名大学(如麻省理工大学、南加州大学分校、斯坦福大学等)和一流的研究机构(如微软研究院、Google 研究院、IBM 研究院、Yahoo 研究院等)都有专门的研究小组在进行相关研究。在国内，中国科学院、清华大学、北京大学、哈尔滨工业大学、国防科技大学、中国科技大学、上海交通大学、北京航空航天大学等科研机构都投入了大量的研究人员和经费，并取得了许多重大研究成果，使得我国在本领域的研究水平能够与国际接轨，并在某些子领域处于国际领先地位。搜索引擎技术之所以受到研究人员如此热烈的关注，与其强烈的应用需求和由此带来的巨大商业价值密切相关。

(1) 搜索引擎技术在人们的生活中占据了重要地位

目前社会正经历着一场信息革命，互联网的发展是这场革命的推动力。它架起了人们信息交流的桥梁。同时，随着网络技术的飞速发展，互联网上的信息资源越来越庞大，并且以几何级数持续增长。一方面，互联网包含了从技术资料、商业信息到新闻报道、娱乐信息等多种类别和形式的信息，为人们提供了一个极具价值的信息源。另一方面，互联网是一个具有开放性、动态性和异构性的全球分布式网络，资源分布分散，且没有统一的管理和结构，使得人们很难准确快捷地从中获取所需要的信息。面对信息的海洋，人们觉得力不从心，往往花费了很多时间却所获甚少，从而产生了所谓的信息焦虑。在互联网日益深入我们日常生活的今天，海量的存储和科学的搜索是人们信息行为中两样最重要的能力。互联网搜索引擎以其基于关键词匹配的信息检索机制为用户在瞬间搜寻出所需的相关信息，成为缓解人们信息焦虑最为有效的工具。

(2) 搜索引擎广泛的应用前景预示了其巨大商业价值

互联网搜索引擎发展至今虽只有十余年的历史，却在商界掀起了巨大波澜。著名因特网网站排名公司 Alexa 互联网[2]提供的数据显示，搜索引擎公司谷歌于2009年1月5日首次在Alexa日流量排名中占据第一，成为世界头号网站。comScore 在 2009 年 9 月发布的全球十大因特网资产也显示谷歌排名第二，排在第九、第十位的也均是搜索引擎公司，分别为百度和 Lycos[3]。而一些著名的门户网站，例如 Yahoo，新浪等也都拥有自己的搜索引擎服务。

目前，搜索引擎服务已经成为中国互联网用户的有力争夺者。艾瑞市场咨询

有限公司在其 2010-2011 中国搜索引擎用户行为研究报告中指出[4]：2010 年中国用户使用网络服务的情况中，有 12 项网络应用服务的使用比例均超过 50%，用户渗透率较高。其中，使用搜索的用户比例达到了 63.1%，表明搜索已经成为中国网络用户经常使用的网络应用服务之一。具体来说，中国用户的网络活动受到时间影响，呈现明显的差异性。在工作日期间，搜索信息仅次于收发电子邮件，其比例达到了 50.4%，属用户最常使用的网络应用；而在非工作日期间，搜索信息的用户占比排在了网上聊天、浏览购物网站、查看新闻以及网络游戏之后，为 33.6%。由此可见，搜索引擎服务在互联网上有着广阔的市场前景。

(3) 搜索引擎技术面临新的机遇和挑战

尽管现有的互联网搜索引擎已获得了巨大成功，并成为人们日常信息获取中不可或缺的工具，但仍存在很多不足之处。首先，目前搜索引擎基于关键字搜索的查询机制仍然对用户的查询技巧有很大考验。用户需要对想要搜索的信息有一定了解，才能够抽象出足以描述所需信息的查询关键字；即使是具备相同含义的查询关键字，由于现有搜索引擎主要是基于文字匹配而不涉及语义理解，搜索结果往往不同，选择的关键字的歧义性越大，搜索结果的准确率越低。其次，目前搜索结果返回大量相关文档，虽然一定程度降低了用户检索信息的难度，但是翻阅这些文档造成的时间耗费也相当可观；再次，现有搜索引擎主要是针对客观信息的查询，而日常生活中，主观性意见或个人经验的查询相当普遍。因此，人们已经逐渐不满足于传统的基于关键字的查询，对搜索引擎的人性化、智能化、个性化都提出了更多的要求。与此同时，各大搜索引擎也致力于从人机界面的人性化，搜索结果的准确性、简洁性、个性化等各个方面提高搜索用户的满意度。

(4) 搜索引擎技术的发展历程

第一代搜索引擎的特征是目录搜索，代表产品是 Yahoo。采用的办法是首先建立一套图书文献分类标准，然后将文献按照分类标准手工或者计算机辅助地进行分类，这样用户就可以按照这个分类进行文献的检索了。第一代搜索引擎的弊端在于：一方面，这个分类体系是由文献的管理者人为给出来的，普通用户并不一定清楚，或者说普通用户并不一定理解，这样普通用户就有可能找不到想要的信息；另一方面，手工分类的成本太大、效率太低，不能适应快速增长的互联网信息资源管理的需要。

第二代搜索引擎是基于关键词匹配的全文索引搜索引擎，它创新性地提出了页面重要性分析的 PageRank 技术[5]和超链分析技术[6]等，将最重要的页面优先呈现给用户。代表产品就是 Google。事实上，早于 1993 年底，Yahoo 出现之前，NASA 就已经提出 Repository-Based Software Engineering (RBSE) spider 系统[7]，该系统是是第一个索引 Html 文件正文的搜索引擎，也是第一个在搜索结果排列中采用关键

字串匹配程度的概念。相对于 Yahoo 而言，字符串匹配的模型计算机可以自动完成，无需人工干预，这使得大规模的搜索成为可能。但是该系统搜索结果的相关性差，查准率低[8]。Google[9]根据用户提交的查询关键字，通过链接分析对出现这些关键字的页面进行排序，并按照得分的高低顺序呈现给用户。返回的页面的质量和信息查准率大大提高。Google 也因此获得了蓬勃的发展。而我们所熟知的 FAST[10]，Openfind[11]，北大天网[12]，百度[13]等都是第二代搜索引擎中的佼佼者。

第三代搜索引擎的发展方向经过多年的探索和市场历练愈发清晰。1996 年推出的 Ask Jeeves[14]被设计成回答用户提问的搜索引擎，有超过 1000 万的大型问题库，支持自然语言提问搜索，适合搜索常识性的问题答案。搜索时，它首先给出的是数据库中可能存在的答案，然后才是网站链接。这种检索方法符合人们日常查找信息的思维方式。用户无需学习布尔检索式和掌握 and、or、not 的用法，抛开了有关关键词和词组的种种限制，也不需要牢记繁琐的检索规则，只要像平时提问一样。目前，除 Yahoo 搜索集团和 Google 外，Ask Jeeves 成为硕果仅存的，拥有自主技术的独立一线全文搜索引擎[107]。WolframAlpha[15]是一个计算答案，获取知识的搜索引擎。这个 2009 年 5 月由开发计算数学应用软件的沃尔夫勒姆研究公司推出的新一代搜索引擎，其真正的创新之处，在于能够马上理解问题，并给出答案。近日，腾讯搜搜在第 34 届国际计算机协会信息检索大会（SIGIR）上更是提出了“Task Engine”的第三代搜索引擎概念[16]。其描绘出如下的用户体验场景：某位用户在最近一段时间经常搜索三亚酒店、机票、天气等信息，任务搜索就能据此推断出，该用户搜索的意图是去三亚旅游。这时，任务搜索就能结合到三亚旅游这个任务的具体实施过程，将出行方式、订票方式、目的地天气、旅馆、景点、注意事项、当地习俗等深度挖掘的用户需求，都放在搜索结果中，并清晰地展现给用户。综上所述，人性化、智能化、个性化成为第三代搜索引擎的主流方向。

1.2 研究意义

搜索引擎技术的研究在国内外都获得了广泛关注。二十世纪九十年代，基于军事和反恐情报处理的需要，美国国防部高级研究计划署（DARPA）提出了 TIPSTER 文本处理计划，信息检索方面的重要评测会议——文本检索会议[17]（Text REtrieval Conference，简称 TREC）就是该计划的重要组成部分。而互联网搜索引擎作为信息检索的主要发展方向也成为该评测会议评测内容的重中之重。另外，国际上有多个专门的高级别会议聚焦于搜索引擎技术的研究，如 SIGIR(ACM Special Interest Group on Information Retrieval) 和 ECIR(European Conference on Information Retrieval)；一些网络应用、数据挖掘和自然语言处理方面的顶级会议

如 WWW、KDD、ACL、COLING、CIKM 等每年都收录了大量的搜索引擎技术相关的论文。在国际国内也有多个顶级研究小组在进行搜索引擎相关的研究工作。

虽然在这些会议和研究小组的推动下，搜索引擎技术有了长足的发展，但仍面临诸多挑战。中国互联网网络信息中心（CNNIC）的报告称[18]，用户认为在互联网上查询信息时遇到的最大问题是重复信息太多（44.6%）、信息太陈旧更新缓慢（27.5%）、得到的有用信息太少（10.7%）、信息查找不方便（10.2%）。其中第一项和第三项主要是由于搜索引擎对用户查询和文档内容没有充分的理解，对搜索得到相关文档没有恰当处理和组织。而第四项主要由于，一方面基于关键字的搜索与用户的日常习惯并不相符，用户更习惯自然语言的方式提问；另一方面，用户可能会选择不同的查询关键字组合表达相同或类似的信息需求，但所得到的结果却不尽相同，因此用户需要变换不同的可能关键字以得到所需信息的全貌。因此，支持自然语言搜索方式，充分理解用户查询及相关文档，最大限度简化搜索结果，对于提高目前搜索引擎的用户满意度将有很大帮助。自然语言处理技术作为解决以上问题的核心技术，成为搜索引擎的新兴技术和研究热点之一。

自然语言处理(Natural Language Processing，简称 NLP)[19]是人工智能和语言学领域的一种分支学科，是一门研究如何让计算机“懂”人类语言的学科。包括自动分词、词性标注、句法解析、短语识别、词义消歧、信息抽取、自动摘要、机器翻译、语言生成等多个方面的研究。自然语言处理技术在搜索引擎中的应用能够将搜索从目前基于关键词层面提高到基于知识层面，对知识有一定的理解与处理能力，并提供友好的人机界面，从而使网络交流更加人性化，使信息查询变得更加方便、快速和准确。但是要建立真正的基于自然语言处理技术的智能搜索引擎，还存在很多的技术难点。首先，面对互联网上的海量数据，很多传统的自然语言处理技术，例如句法解析，由于时间耗费的问题并不适用；其次，在 Web 2.0 时代，越来越多的网络内容来自普通网络用户，语言的不规范性使得传统的成熟的自然语言处理技术（如分词、词性标注）性能下降；再次，网络内容的丰富也导致了语言表达方式的多样性，使得收集包含足够样例的训练集越发困难。因此，面向互联网搜索引擎的自然语言处理技术也成为一个新的研究领域。其研究意义总结如下：

- 通过利用自然语言处理技术获取与用户查询请求语义相关的查询请求，可以进行查询推荐，通过与用户的交互，逐步明确用户的查询意图。特别是，可以根据用户的查询意图，描绘用户的查询情境，为用户可能的查询情境提供相关知识，从而简化用户交互过程并提高搜索引擎的智能性。
- 通过利用自然语言处理技术对查询的理解，可以将用户的复杂查询请求转化为类似数据库查询，进而用于自动检索网络上来自不同网站不同接口的相关数据

库，直接获取查询的准确信息，可避免手动逐一检索每个数据库。

- 通过利用自然语言处理技术对搜索引擎返回的相关文档进行抽取、摘要，可大大减少检索结果的阅读量，帮助用户快速获得相关信息。尤其针对现有搜索引擎难以处理的主观性问题，通过对问题的分析及对网络上用户生成问题-答案资源的理解和整理，可以帮助用户了解广大网络用户的不同想法。

1.3 本文的主要贡献

本文在基于比较关系的查询推荐、基于比较关系的用户查询意图理解、多查询关键词查询的语义理解、复杂问题的答案摘要等方面进行了系统深入的研究，本文的主要贡献和创新点总结如下：

1. 研究基于比较关系的查询推荐，并提出一种弱监督的从比较性问题中挖掘比较对象的方法

通常的查询推荐（Query Suggetion）都是推荐与用户初始查询请求相关的查询请求，例如，在用户搜索“ipod touch”时，搜索引擎推荐“ipod touch break prison”。然而，在不同的搜索场景下，用户所需要的相关查询是不同的。例如，在购买商品的产品搜索场景下，当用户搜索“你 Nikon d200”时，用户往往想要了解该产品的相关信息，并将其与同类产品作比较，从而做出购买决策。这时，推荐“Cannon 300d”并提供两者之间的对比信息，将有利于用户尽快做出购买决策。当然，“nokia d200 lens”也是有用的推荐，然而相比之下，类似“Cannon 300d”的查询推荐，往往需要用户具有一定的知识储备，对同类产品有一定了解，这也往往是用户最为缺乏的知识，用户对此类推荐的需求更为迫切。由此可见，将用户初始查询请求的相关查询请求按其与初始查询请求的语义关系进行分类，根据搜索场景的不同给予不同类别的查询推荐，将使得搜索引擎更加个性化、智能化。由于比较备选方案是人们的日常决策行为中至关重要的一个步骤，本文针对用户比较行为这一搜索场景，提出基于用户查询请求的可比较对象的查询推荐。

由于用户比较行为的主观性和复杂性，判断两个对象的可比性是困难的。幸运的是，互联网上有网络用户产生的大量意图比较两个或多个对象的比较性问题，这些比较性问题提供了人们想比较什么的证据，例如“Which to buy, iPod or iPhone?”。在本文中，比较性问题中用于进行比较的对象被称为比较对象，例如上述例句中的“iPod”和“iPhone”。

为了从比较性问题中挖掘比较对象，本文首先需要识别一个问题是否是比较性问题。根据本文的定义，一个比较性问题必须是一个意在比较至少两个对象的问题。值得注意的是，包含两个对象的问题如果不包含比较的意图就不是比较性问题。但我们观察到：如果一个问题包含两个可比较对象，则该问题极有可能是比

较性问题。本文利用这一点设计了一个弱监督的自举（bootstrapping）方法，可以在识别比较性问题的同时，抽取比较对象。

据我们所知，这是第一个意图通过推荐好的比较对象以支持网络用户的比较行为的研究。也是第一次提出使用网上的比较性问题作为媒介来反映用户真实关心的比较对象。本文提出的弱监督方法的 F1-measure 在比较性问题识别上达到 82.5%，在比较对象抽取上达到 83.3%，在从比较性问题识别到比较对象抽取的整体系统上达到 76.8%。

2. 利用对象之间的比较关系，提出一种基于图聚类的用户意图识别方法，并建立针对用户比较行为的比较搜索系统。

基于关键字查询的搜索引擎中，用户往往只能使用有限数量的词汇来抽象和概括他们的需求。在用户将其需求抽象成有限的查询关键字的过程中，部分有用信息被丢失，从而导致用户查询意图不慎清晰。目前，搜索引擎搜索的结果通常是满足用户各种查询意图的文档的合集，用户需要阅读大量相关文档才能获得自己确实所需的信息。因此，在对用户的查询请求进行搜索之前，尽可能确定用户的查询意图，并进行面向用户查询意图的搜索将有利于更准确地找到用户想要的信息。

如上所述，通常一个由一个或多个查询关键字组成的用户查询请求可能出于多种不同的意图。举个最经典的例子，当用户查询“apple”时，可能是查询一种水果，也可能是查询一种电子产品的品牌。而“apple”作为电子产品品牌时，用户可能想要查询“apple”的产品，也可能想要查询“apple”的网点分布。如果用户想要购买“apple”的产品，比如用户输入“apple itouch”，有可能是想搜索产品介绍，有可能是想看不同网站的价格比较，还有可能是想看该产品与其它同类产品的比较。即使我们确定用户查询的意图是想要将“apple itouch”与其它产品相比较，用户还可能想比较产品的不同方面。比如，从产品升级的角度，用户可能想将其与“ipod classic”，“iphone”相比较；从娱乐功能性的角度，用户可能想将其与“psp”相比较等。可见，确切地理解用户的意图并不是件简单的任务。

本文主要关注用户的比较行为，针对用户的查询请求，利用与其可比较的对象之间的比较关系，提出一种基于图聚类的用户意图识别方法，每个可能的用户意图由一组用户查询请求的可比较对象表示。利用信息抽取的方法，本文给识别出的每个用户意图赋予一个语义标签。实验证明，本文提出的用户查询意图识别算法的准确率达到 92.7%以上。另外，本文还建立了一个基于用户比较意图识别的比较搜索系统，该系统在识别用户查询请求的不同比较意图的同时，还提供了不同的比较意图下，用户查询请求与相应的可比较对象的比较信息。

3. 研究面向开放域的查询理解，针对由多个查询关键词组成的复杂查询请求，提

出一种无监督的基于查询模板的查询理解方法

在搜索引擎中，用户的查询请求除了单个实体或对象（如 Obama）以外，还存在大量的由多个查询关键词组成的复杂查询请求，例如“flight from Beijing to New York”。这些查询多是面向任务的，并要求得到精确的答案（如“0:00, Nov.19th, 2010”）。在现有的搜索引擎中，用户通常要进行如下操作：首先，检索相关网页或在线数据库；然后，通过逐一阅读相关网页或向相关的在线数据库提交数据库检索请求以寻找所需信息。为了简化这一繁琐的过程，研究者们提出了结构化检索。而查询理解是结构化检索的至关重要的一步。具体来说，本文所指的查询理解包括识别和消歧复杂查询中的查询关键字两个步骤。例如，给定查询“harry potter showtime in beijing”，本文先识别查询词“harry potter”，“showtime”，“beijing”（即查询关键字识别）；然后分辨每个查询词的含义并分别标记，这里“harry potter”被标记为“movie name”，“beijing”被标记为“city”，而“showtime”是电影的一个属性（即查询关键字词义消歧）。

本文主要关注如何在开放域的环境下理解用户的复杂查询请求：首先，利用现有技术自动创建一个语义词典；然后，尝试使用自动创建的语义词典进行开放域查询理解，即查询关键词识别与消歧。在本文的问题设定中，我们致力于解决以下两个问题：

1) 自动创建的语义词典无论在语义标签还是词或词组实例中都包含很多噪音，这些噪音将严重降低查询理解的性能。
2) 在开放域环境中，大量的语义标签是必需的。这使得以前用于处理有限数目的语义标签的基于序列标注技术的查询理解方法不再适用。

为了解决以上问题，本文提出一种基于查询模板的查询理解方法。该方法先利用一个无监督的互增强的算法挖掘查询模式；然后基于挖掘到的查询模式和语义词典进行查询关键字识别及消歧。据我们所知，本文的研究是第一个利用自动创建的语义词典进行查询理解的尝试。

4. 研究社区问答服务的知识重用中答案完备性问题，提出了一种基于词的依赖层次的面向问题的答案摘要方法。

传统的搜索引擎在面对一些复杂问题的查询时，其查询结果往往差强人意，例如“how to recover my doc file?”或“what is the best smart phone?”。这些复杂问题通常需要人的经验或意见的参与，答案因人而异或因情况而异，没有唯一正确答案。而目前，社区问答服务的出现，为解决这一问题提供了新的资源。如何重用社区问答服务中积累的问题答案知识提高复杂问题查询的满意度成为该研究的一个研究热点。但是，目前的研究主要集中在评测社区问答服务积累的答案的准确性上，对于答案的完备性没有涉及。事实上，由于复杂问题的正确答案往往

并不唯一，提供汇总了不同情况下不同个人的回答的完备答案对提高搜索满意度也是至关重要的。

本文尝试对一种特殊的问题——调研问题（survey questions）——进行答案摘要，以提高社区问答服务中可重用问题的答案的完备性。调研问题是指请求回答问题的用户推荐针对某种需求的最佳选择的问题。显然，对调研问题而言，答案的完整性至关重要，因为一方面不同的用户可能对不同的建议感兴趣，另一方面某个答案被推荐的次数也反映了该答案的值得推荐指数。

据我们所知，本文最先指出在社区问答服务的知识重用中答案完备性的重要性。同时，这也是第一个关注调研问题的研究，调研问题作为面向意见的问题中一种有趣的类型，问题的完整性对其而言至关重要。除此之外，本文推荐使用面向问题的答案摘要方式产生完整的答案。本文提出一种有效的建立词与词之间语义依赖性的层次结构的方法，并利用建立的层次结构进行面向问题的答案摘要，从而基于社区服务中用户提供的所有已有答案生成一个完整简洁的答案。

1.4 本文的组织结构

本文共分为七章，第一章为绪论，第二章介绍相关工作，第三、四、五和六章中分别介绍了本文在面向搜索引擎的自然语言处理技术研究中的四个成果，本文最后一章是总结和展望。各章节的主要内容安排如下：

第一章为绪论，首先介绍了课题的研究背景，指出搜索引擎技术获得广泛关注和应用的同时还面临着诸多挑战，并概要介绍了搜索引擎技术的发展历程。然后讨论了研究面向第三代搜索引擎的自然语言处理技术的重要性和意义。随后总结了本文研究的主要贡献。本章最后是论文的组织结构。

第二章综述了面向第三代搜索引擎的自然语言处理技术的相关研究。在广泛阅读国际国内学术会议和学术期刊论文的基础上，分析了面向第三代搜索引擎的自然语言处理技术研究的重点和已有工作，并在最后指出现有工作的盲区和不足。

第三章研究了基于比较关系的查询推荐。首先阐明基于比较关系的查询推荐与传统的查询推荐的不同之处；然后提出一种弱监督的利用网络用户产生的比较性问题挖掘可比较对象的方法；最后对每个用户查询请求，根据挖掘出的可比较关系，对与之可比较的对象排序。实验表明，我们的比较关系抽取方法的 f1 值达到 76.8%，远远高于基准方法。

第四章研究了基于比较关系的用户比较意图识别。首先提出一种基于图聚类的方法识别用户查询请求的比较意图；然后利用信息抽取的方法为识别出用户意图赋予一个语义标签；最后介绍基于比较意图识别的比较搜索系统。

第五章研究了面向开放域的多查询关键词的复杂查询理解。首先，将该问题

转化为两个子问题——查询关键字识别及利用自动创建的语义词典进行语义消歧；然后，为解决自动获取的语义词典不完整、噪音严重、语义标签规模大等问题，提出一种非监督的基于查询模式的查询理解方法。实验显示查询关键字识别并消歧的准确率达到 73.8%。

第六章研究了在重用社区问答服务中积累的问题-答案资源过程中答案的摘要。首先分析了社区问答系统中积累的问题-答案对是否可以重用、最佳答案是否唯一、答案质量是否可以提高等问题；然后指出在重用社区问答服务中积累的问题-答案对时答案完备性的重要性；然后，以一种特殊的问题——调研问题为例，提出一种基于词汇层次关系的答案摘要方法。实验结果表明，摘要的结果在答案准确性及召回率都有所提高。

第七章总结了本文的主要工作，并对下一步的工作进行了展望。

第二章 相关研究工作

本章第一节简要介绍目前第三代搜索引擎便捷化、智能化、个性化的方面的几个主要尝试。后面两节重点介绍已有研究工作，并在最后一节总结现有工作的盲区和不足，指出本文的研究方向。

2.1 引言

前一章已经指出，第三代搜索引擎的主要方向是人性化、智能化、个性化，提供更加友好的人机界面，更准确了解甚至预测用户需求，提供更加准确简洁的信息。在向这个方向努力的过程中，国内外的研究者们主要在以下几个方面做出了尝试。

2.1.1 查询推荐

在很多时候，用户之所以进行搜索是由于用户对所要搜索的信息知之甚少，甚至毫无概念，这时用户很难构造好的查询；另外，有时用户想要查询几个相关话题，或是作为对初始查询的深入了解，或是比较几个不同的话题。为了建立用户更好的交互体验，一方面帮助用户构造更好的查询，另一方面预测用户的下一步查询，很多搜索引擎推出了查询推荐（Query Suggestion）服务，即向搜索引擎用户推荐与其初始查询相关的查询。

2.1.2 查询意图分类

随着互联网的发展，Web 上的内容越来越多，充满了各种各样的信息。这些信息以传统网页、格式化文档、图片、视频、讨论区等不同的形式存在。而在基于关键字查询的搜索引擎中，用户往往只能使用有限的查询关键字来抽象和概括他们的需求。在用户将其需求抽象成有限的查询关键字的过程中，部分有用信息被丢失，从而导致用户查询意图不慎清晰。查询分类作为分析用户的查询意图的手段之一，得到了广泛的研究。从功能和目的上来说，查询分类是为用户的查询Q，确定一个有序而且按照某种相关度递减的类别的列表C_{i1}，C_{i2}，…，C_{in}使得Q可能属于这些类别，而这些类别来自于一个预先已经定义的含有 N 个类别的集合$\{C_1$，C_2，…，$C_n\}$[20]。

2.1.3 查询理解

目前，互联网上涌现了大量的在线数据库，用户通过在其对应的网站上填写

表单向这些数据库提交查询请求，数据库通过网站返回结果给用户（例如，携程网）。然而，用户所需的信息（例如机票信息）往往存在于不同的在线数据库，因此，用户需要从不同的网站提交类似的表单来查询不同的数据库，以期获得完整的信息。为了方便用户查询，搜索引擎技术中产生了面向深层 Web 搜索（Deep Web Search）[21]的概念，即通过搜索与用户查询请求相关的网站并将基于关键字的用户查询请求转换成满足每个相关网站需求的数据库查询命令，为用户提供搜索引擎统一接口访问在线数据库。深层 Web 搜索中至关重要的一步即查询理解（Query Understanding）。查询理解即是识别查询语义结构的过程。

2.1.4 自动问答

自动问答（Query Answering, QA），是指让计算机为用户以自然语言形式提出的问题找到一个明确的答案，广义上讲，自动问答是信息检索的高级形态。与目前流行的搜索引擎（如 Google、百度、Yahoo！等）相比，自动问答系统有以下优势：一方面自然语言的问题的人机交互界面更加自然和人性化；另一方面，简洁的答案可以使用户一目了然地获得想要的信息，而不需要翻阅大量包含冗余甚至无用信息的文档。因此，互联网自动问答系统更加受到互联网用户的欢迎。

2.2 国内外研究现状

2.2.1 查询推荐研究现状

对查询推荐的研究可以追溯到上世纪 90 年代，根据其所依赖的数据的不同可分为三类: 基于文档的方法，基于查询日志的方法及基于点击日志的方法。

基于文档的方法主要通过从查询返回的相关文档或人工编辑的语料(如词典)中找出与用户查询请求相关的词或短语, 然后利用这些相关词或短语构建查询推荐。例如，White 等人[22]将包含查询词的文档中的高频非停用词作为与用户查询请求相关的查询关键词推荐给用户；Xu 和 Croft[23]提出计算包含查询词的文档中每个词与用户查询请求的关系紧密程度的 LCA 算法；Shieh 等人[24]利用社会化标注的 wikipedia 查找与用户查询请求相关的词或短语；Xu 等人[25]则利用人工编辑的词典 Open Directory Project 获取与用户查询请求相关的词。

基于查询日志的方法主要依靠分析搜索引擎的查询日志，寻找曾与用户查询请求在出现时间上紧密关联的查询请求，并以此向用户提供查询推荐。这类方法假设用户在搜索过程中为了同一个检索目标所做的一系列检索行为构成一个 Session。我们可以想象，在用户输入 Session 中的第一个查询时得到的结果不满意，持续通过改变查询关键字或其组合方式，直至找到满意的查询。基于这一思想，

Cucerzan 和 White[26]提出一套规则判别 Session 中的最终结果网页，进而向提出 Session 中某一个查询请求的用户推荐能直接返回最终结果网页的查询。Fonseca 等人[27]则考虑同一 Session 中的查询是相关的，并利用关联规则挖掘算法衡量查询之间的相关性。

基于查询点击日志（click log）的方法主要依靠每次查询时用户点击的 URL 之间的关联度衡量查询请求之间的关系紧密程度。王继民和彭波[28]提出一种基于不同查询请求之间共有点击 URL 数的查询推荐方法。但是由于一次查询平均只有几次点击，而网络上很多不同的网页都有相似内容，因此，很多相似查询没有相同点击。为解决这一稀疏问题，Antonellis 等人[29]提出利用点击的网页的相似性和查询的相似性之间的关系，不断迭代利用点击的网页的相似性估计查询的相似性，再利用查询的相似性估计点击的网页的相似性的过程，得到精确的查询请求间的相似性，并利用该相似性进行查询推荐。

2.2.2 查询意图分类的研究现状

在查询意图分类领域，目前没有标准的分类体系，不过后续的关于查询意图的分类研究工作都受到了 Broder 等人提出的分类标准的影响。Broder 等人[30]在用户调查和对查询日志进行手工分类的基础上将查询意图分为导航类(Navigational)、信息类(informational)和事物类（transactional）。Rose 等人[31]在 Broder 的基础上，提出了更细致的层次结构，最终得到 11 个小类别。Baeza-Yates 等人[32]虽然认同将查询分为三类，但他们在 Broder 等人提出的分类标准的基础上，对具体的类别和定义有较大的修改，他们将查询分为信息类、非信息类（Not Informational）及歧异类。

与所有的分类任务一样，查询意图的分类可分为特征提取、训练集和测试集的构造、分类器的训练及应用几个步骤。在具体问题中，不同分类问题的分类特征的提取尤为重要。在查询意图分类中，由于查询请求本身的信息量少，从中提取特征显得尤为困难。Kang 等人[33]构造了两个文档集，第一个是包含话题类页面的，第二个是包含主页类页面的，据此计算查询请求在这两个文档集合里分布的差异、互信息的差异、作为锚文本的使用率以及词性信息：最后一个特征只依赖于查询本身，而前三个特征借助了外部信息。Beitzel 等人[34]利用标注以后的查询请求，使用分类器和选择性属性发掘查询请求内在的特征，同时还使用了对某段时间比较流行的查询请求进行精确匹配的方法（使用查询请求自身的字符串作为特征）。Jansen 等人[35]使用各类（导航、事务、信息）查询请求的一组启发式特征来区分查询请求，比如他们总结出含有“ways to”，“how to”等词汇的查询请求极有可能属于事务类查询。Lee 等人[36]使用了查询请求在查询日志中点击的 url

分布及平均点击次数。Dou Shen 等人[20]使用了提交查询请求以后由搜索引擎返回的检索结果里文档的标题、文档片段以及结果文档的全文作为特征来训练分类器。

2.2.3 查询理解的研究现状

查询理解是随着近年深层 Web 研究的兴起而逐渐受到关注的，仍属于一个新兴领域，其相关工作并不多。目前的研究工作通常将查询理解看作序列标注的问题，面向特定领域，根据有限的语义标签及自定义的词表，将用户的查询请求标注为语义标签的序列[37-41]。例如，Li 等人[37]针对商业领域的查询，提供了给定语义标签如何利用查询点击及数据库信息获取词表的方法，并且提出一种基于条件随机场（CRF）的半监督模型进行用户查询请求的语义结构理解。Agarwal 等人[39]则提出一种基于模板的非监督模型。他们利用领域词表，根据查询日志中的查询记录生成可能模板，然后利用查询点击、用户查询、查询模板三者之间的关联，通过对模板的评测选出可靠性最强的模板。

2.2.4 自动问答技术的研究现状

事实上，对自动问答系统的研究源于上世纪 60 年代，但早期的自动问答系统多是从人工编写的知识集中寻找答案，类似于专家系统。由于知识集编写的困难，这些问答系统大多局限于某个特定领域，问答的质量也依赖于知识集编写的质量，因此并没有得到广泛的推广。

随着 90 年代搜索引擎的发展，从海量的互联网信息中获取知识成为可能，而基于搜索引擎的自动问答系统[42-45]的出现使得自动问答脱离了领域的限制，再次吸引了人们的注意。这一时期的基于搜索引擎的自动问答系统通常由三部分组成：问题到查询的转化；相关文档或相关文段的检索；答案抽取。然而这种方法只适用于简单问题的回答，对于复杂问题，用户满意度仍然很低。

随着 Web 2.0 时代的到来，互联网用户越来越多地在网上分享自己的感受、经验，尤其是社区问答服务（例如 Yahoo! Answer，百度知道等）的出现，为复杂问题的自动回答提供了新的资源，使其有了新的解决方案。社区问答服务中积累了大量的问题-答案知识，给定一个新的用户问题，我们只需要检索相关问题，返回这些问题的答案即可。基于已有问题-答案库的自动问答系统的研究主要集中在以下两个方面：1）如何衡量已有问题-答案对中答案的质量[46,47]；2）给定一个新的问题如何找到已有的相似问题[48,49]。

2.3 现有工作的盲区与不足

尽管第三代搜索引擎技术的研究，尤其在以上几个方面，得到了国内外学术界和工业界的广泛关注和投入，它仍然是一个尚未完全成熟的研究领域。还有诸多问题亟待解决:

- 对查询请求的语义理解还处于初级阶段。例如，查询推荐通常只推荐与用户查询请求在统计意义上相关的查询请求，并不考虑推荐的查询请求为何相关。事实上，被推荐的相关查询请求与初始用户查询请求是有着不同的语义关系的。以查询“Obama”为例，与其相关的查询包括历届总统，竞选对手，家人等等。在竞选期间，人们可能很关注“Obama”与其竞选对手的比较；当其竞选成功或离任时，人们可能更关注其在任期间与历任总统的表现的比较；另外，Obama总统的粉丝可能也关注他的家人。因此，在不同的应用场景下，用户需要的是对与初始查询请求有不同语义关系的相关查询请求推荐。
- 很多应用限定在特定域中，对人工标注依赖较大，面向开放域的技术还有待开发。例如，在目前的用户查询意图分析中，系统通常根据通过大量人工标注获得的分类系统对用户查询请求的意图进行分类。这种用户查询意图分析手段存在以下不足：一方面，用户并不了解预定义的查询意图分类系统，在与系统交互以确认自身查询意图的过程中将产生使用上的不便；另一方面，系统不能根据具体情况对用户查询请求进行更细化的意图分类。再例如，在目前查询理解中，通常都是用有限的标签描述特定领域，再根据标签建立词表完成查询的语义结构分析。然而，采用有限标签描述特定领域是很困难的，系统迁移到其它领域的代价也较高。
- 自动问答系统中，尤其是在基于问题-答案对重用的自动问答系统中，研究者们在考虑答案质量时，主要关注于答案的准确性，忽略了答案的完备性。然而，互联网上的问题往往没有唯一正确答案，尤其对主观性问题而言，由于其没有标准答案，任何一个答案都是有价值的。因此，产生满足用户不同需求的完备性答案是必要的。

本课题将针对上述问题展开系统而深入的研究，力争在基于语义关系的查询推荐、面向开放域的查询理解、可重用问题的答案完备性研究等方面取得重要进展。

第三章 基于比较关系的查询推荐

3.1 引言

查询推荐通常只推荐与用户查询请求相关的查询请求，并不考虑推荐的查询请求为何相关。事实上，被推荐的相关查询请求与初始查询请求是有着不同的语义关系的。在不同的应用场景下，用户需要的是与初始查询请求有不同语义关系的相关查询请求。以查询“Obama”为例，与其相关的查询请求包括历届总统，竞选对手，家人，生平简历等等。在竞选期间，人们可能很关注“Obama”及其竞选对手；当其竞选成功或离任时，人们可能更关注其在任期间与历任总统的表现的比较；另外，Obama 总统的粉丝可能也关注他的家人、生平简历等。再比如，在产品搜索场景下，用户搜索“iphone 4”时，目前的搜索引擎查询推荐系统很大可能会推荐“iphone 4 shell”。然而，用户可能面临的是购买决策的问题，更需要的是推荐与“iphone 4”可比较的查询对象，并获取“iphone 4”与该可比较对象的比较信息，从而尽快做出购买决策。

事实上，比较备选方案是人们的日常决策行为中至关重要的一个步骤。例如，如果某人对某种产品或某种服务感兴趣，比如数码相机或某种医疗，在做购买决策之前，他会想要知道可选的数码相机或医疗服务有哪些并比较这些备选数码相机或医疗服务的优缺点。这种比较活动在我们的日常生活中非常普遍，但需要很丰富的知识。例如，Comsumer Report 和 PC Magazine 等杂志及 CNet.com 等在线媒体力求提供经过编辑的比较内容和调查来满足这一需求。譬如以上讨论的例子中，尽管“iphone 4 shell”也是有用的推荐，然而类似“HTC Incredible S”、“Samsung Galaxy S”之类的查询请求，通常是需要用户对目前比较流行的高端智能手机有所了解才能产生，用户需求更为迫切。因此，这种针对用户查询请求的可比较对象的推荐在实际应用中有着广泛的应用前景。在本文中，我们致力于为用户的查询请求，找到一组与其可比的对象作为查询推荐。

总的来说，由于人们可能出于不同原因比较两个对象，因此很难决定两个对象是否可比。例如，“Ford”和“BMW”可能作为汽车制造商或不同的产品市场是可比的，但是人们却很少比较“Ford Focus”和“BMW 328i”。当对象有不同的功能的时候，事情会变得更加复杂，例如，人们可能把“iPhone”和“PSP”当作便携式游戏机相互比较，也可能将“iPhone”和“Nokia N95”当作手机进行比较。幸运的是，互联网上存在网络用户产生的大量的比较性问题，这些比较性问题提供了人们想比较什么的证据，例如“Which to buy, iPod or iPhone?”，我们称该例句中的“iPod”和“iPhone”为比较对象。在本文中，我们给比较性问题和比

较对象如下定义：

- **比较性问题：**指意图比较两个或多个对象的问题，问题中必须显式提到被比较的对象。
- **比较对象：**指比较性问题中被比较的目标。

根据定义，如下的问题 Q1 和 Q2 不是比较性问题，而问题 Q3 是比较性问题，“iPod Touch”和“Zune HD”是比较对象。

Q1:“Which one is better?”

Q2:“Is Lumix GH-1 the best camera?”

Q3:“What’s the difference between iPod Touch and Zune HD?”

本研究工作的目标是从比较性问题中挖掘比较对象，进而对一个用户输入的查询对象，提供一组排序的可比较对象。为了从比较性问题中挖掘比较对象，我们首先需要识别一个问题是否是比较性问题。根据我们的定义，一个比较性问题必须是一个意在比较至少两个对象的问题。值得注意的是，包含两个对象的问题如果不包含比较的意图就不是比较性问题。但是，我们观察到：一个包含两个可比较对象的问题有极大可能是比较性问题。我们利用这一点设计了一个弱监督的自举方法（bootstrapping），可以在识别比较性问题的同时抽取比较对象。

据我们所知，这是第一个意图为用户查询请求推荐好的比较对象以支持网络用户的比较行为的研究。也是第一次提出使用互联网上的比较性问题作为媒介来反映用户真实关心的比较对象。本文提出的弱监督方法的 F1 值在比较性问题识别上达到 82.5%，在比较对象抽取上达到 83.3%，在从比较性问题识别到比较对象抽取的整体系统上达到 76.8%。这一结果明显优于 Jindal & Liu [50,51]提出的目前与本文工作最相关的方法。

本章的余下部分组织如下：3.2 节讨论了已有相关工作；3.3 节描述了用于比较对象挖掘的弱监督方法；3.4 节讨论了给定一个用户查询请求如何排序其可比较对象；3.5 节报告了评测结果；3.6 节为本章小结。

3.2 相关工作

3.2.1 概述

从发现一个对象的相关对象的角度讲，本文的工作与推荐对象给用户的推荐系统相似。推荐系统主要依赖于对象与对象之间的相似度或在用户日志数据中的统计关联度[52]。例如，Amazon[53]基于客户的购买历史、相似客户的购买记录及产品之间的相似度推荐产品给客户。但是，推荐某个事物不等于发现一个可比较的事物。在 Amazon 的例子中，推荐的目的是通过建议与已买或已浏览的商品相似或

相关的商品引诱客户将更多的商品加入他的购物车中。但是在比较行为中，我们要帮助用户找到与已选商品相当甚至可代替已选商品的备选商品。

例如, 如果用户对“iPod”感兴趣，推荐系统推荐“iPod speaker”和“iPod batteries”给用户是合理的，但是我们不会将这两种商品与“iPod”比较。而从用户发表的比较性问题中发现的“iPhone”和“PSP”跟“iPod”从音乐播放器的角度是可比的。而“iPhone”和“PSP”从相似度的角度是很少被推荐系统推荐给选择了“iPod”的用户的。尽管三者都可以作为音乐播放器，但是“iPhone”主要是被看作一种手机，而“PSP”主要是被看作一种便携式游戏设备。三者从音乐播放器角度有相似的功能但是又是不同的，因此需要比较。显然，比较对象的挖掘与商品推荐相关但并不相同。

本文的比较对象挖掘工作与信息抽取中的对象和关系抽取相关[54-58]。最相关的工作是 Jindal 和 Liu[50,51]的关于挖掘比较性句子和比较关系的工作。他们的方法使用从已标注的新闻和评论数据集中学习的类序列规则（class sequential rules, CSR）[50]和标记序列规则（label sequential rules, LSR）[50]分别识别比较性句子和抽取比较关系。相同的技术也可以应用于比较性问题的识别和比较对象的抽取。他们的方法虽然可以获得高准确度但由于标注数据的有限导致了规则有限从而引起了低召回率的问题[51]。然而，保证高召回率对本文设定的应用场景是至关重要的。为了解决这一问题，本文开发了一个充分利用未标注的数据的弱监督的自举（bootstrapping）模式识别方法。

自举方法（bootstrapping）在已有的信息抽取研究中已经显示出其有效性[56, 57, 59,60]。本文的工作从使用自举技术（bootstrapping）抽取有特定语义关系的实体的方法论角度上来讲与之相似。但是，本文的任务与之不同，在本文的任务中，我们不但要抽取对象（比较对象抽取）还要确定该对象来自比较性问题（比较性问题识别），而后者在传统的信息抽取任务中是不需要的。

3.2.2 Jindal&Liu 2006

在本小节中，本文对 Jindal&Liu[50,51]的比较关系挖掘算法做了简要回顾。该方法代表了这一领域最新状态并将被用作基准方法。我们首先介绍 CSR 和 LSR 规则的定义，然后描述他们的比较关系挖掘算法。读者可以参考 J&L 的文章以了解算法细节。

3.2.2.1CSR 和 LSR

CSR 是一个分类规则。它将一个序列规则 $S(s_1 s_2 \ldots s_n)$影射到一个类 C。在本文设定的问题中，类 C 可以是比较性问题或非比较性问题。给定一个有类信息的

序列集合，每个 CSR 与两个参数相关：支持度和置信度。支持度是指集合中包含子序列 S 的序列所占的百分比;置信度是指包含子序列 S 的序列中被标记为类别 C 的序列所占的百分比。这些参数对评价一个 CSR 是否可靠很重要。

LSR 是一个标注规则。它通过将输入序列的第i项s_i替换成定义好的标记l_i从而将输入序列影射为标记好的序列。这里输入序列中的第i项s_i被称为“锚”。当输入序列中的“锚”相对应的标签l_i是我们所感兴趣的标签（例如本文设定的问题中的“比较对象”标签）时，“锚”，即s_i，被抽取为l_i的一个实例。LSR 规则也是从一个标注数据集中挖掘得来。每个 LSR 也有两个参数：支持度和置信度。LSR 中这两个参数的定义与 CSR 相似。

3.2.2.2 监督的比较关系挖掘算法

J&L 将比较性句子识别作为分类问题，将比较关系抽取作为信息抽取问题来对待。他们先创建一个包含诸如 beat, exceed, outperform 等 83 个关键词的集合。这些关键词对一个句子是否是比较性句子有一定的提示作用。他们被用作支点来产生词性序列数据。一个赋有类信息（如比较性句子或非比较性句子）的手动标注的数据集被用于创建序列数据，并从中挖掘 CSR 规则。然后他们使用挖掘到的 CSR 作为特征，训练朴素贝叶斯分类器，并用训练好的分类器识别比较性句子。

给定一个比较性句子集合，J&L 手动标注每个句子中两个被比较的对象为$ES1 和$ES2, 两个对象被比较的特征为$FT。J&L 限制被比较的对象及其被比较的特征只能是名词或代词。为了区分不是比较对象或被比较特征的名词和代词，他们增加了一个标签$NEF，即非比较对象和被比较特征。以上四个标签被用作支点，与一些代表位置的特殊符号一起生成用于训练的序列数据。这些代表位置的特殊符号包括：l_i（从支点标签位置左数第i个位置，$1 \le i \le 4$），r_j（从支点标签位置右数第j个位置，$1 \le j \le 4$），#start(句子的开始位置)，#end(句子的结束位置)。保留只包含一个标签且支持度大于 1%的序列并根据这些序列创建 LSR。当利用学到的 LSR 抽取感兴趣的信息时，有较高置信度的 LSR 被优先选择。

J&L 在他们的实验中验证了方法的有效性，但是该方法存在以下缺点：

- J&L 的方法的性能很大程度上依赖于识别比较性句子的关键词集合的创建。在 J&L 的方法中，这些关键词是手动创建的且没有提供选择关键词的方法指导。另外，关键词集合的完整性也很难保证。
- 用户可以用各种方法表达比较性句子或比较性问题。为了提高算法的召回率，一个大规模的标注数据集是必须的。但是，要获得大规模的标注数据集，代价是昂贵的。

- J&L[50]中给出的 CSR 和 LSR 的实例中的模式序列大多由词性标记和关键字组成。这些规则应用于比较性句子识别和比较关系抽取时，有高准确度低召回率的特点。他们将这一结果归咎于词性标注的错误。但是，本文猜测它们规则可能过于特殊化，过匹配了他们小规模的训练数据集（包含大约 2,600 个句子）。本文意图提高算法召回率，避免过匹配，并允许模式序列中包含有分辨力的关键词以维持准确率。

下一节介绍一种弱监督的基于模板的方法来解决以上问题。

3.3 弱监督的可比较实体挖掘方法

本文提出的弱监督学习方法是一个类似 J&L 的方法的基于模板的方法，但是它在很多方面与 J&L 的方法不同：本文提出的方法致力于学习能够同时用于识别比较性问题并抽取比较对象的序列模式，而不是分别使用 LSR 和 CSR 完成这两个任务。

在本文提出的方法中，序列模式被定义为一个序列$S(s_1 s_2 \dots s_i \dots s_n)$，其中$s_i$可以是一个词，一个词性标记，一个代表比较对象（$C）、句子开头（#start）或句子结尾（#end）的符号。如果一个序列模式可以用来可靠地识别比较性问题并抽取比较对象，则这个序列模式可称为指示性抽取模式（indicative extraction pattern, 简称 IEP）。本文将在下一节中形式化定义一个模式的可靠性。如果一个问题匹配某个 IEP,它将被分类为比较性问题，且问题中与比较对象（$C）槽相对应的词的序列被抽取为比较对象实例。从同一个比较性问题中抽取出的两个比较对象之间存在比较关系，本文称之为“比较对象对”。当一个问题与多个 IEP 匹配时，本文选择最长的 IEP。综上，本文提出的方法是基于一个自动挖掘的 IEP 集合，而不是一个手动创建的指示性关键词列表。接下来的章节将描述如何利用一个大规模的未标注数据集，采用需要最小监督的自举算法，自动获取 IEP。3.5 小节的实验确认了本文提出的弱监督方法在维持高准确率的情况下可以获得高召回率。

表 3.1 候选序列模式

<#start which city is better, $C or $C ? #end>
<, $C or $C ? #end>
<#start $C/NN or $C/NN ? #end>
<which NN is better, $C or $C ?>
<which city is JJR, $C or $C ?>
<which NN is JJR, $C or $C ?>
……

本文中序列模式的定义受到 Ravichandran 和 Hovy 的工作的启发[45]。给定示例问题“which city is better, NYC or Paris?”，表 3.1 给出了其能匹配的一些序列模式。本文允许出现形如“<, $C/NN or $C/NN ? #end>”的模式，即利用词性标签或词性标签序列对抽取出的比较对象的形式进行限制。此例的序列模式限制了抽取出的比较对象（$C）只能是名词或名词词组（NN）。

3.3.1 挖掘指示性抽取模式

本文提出的弱监督指示性抽取模式（IEP）挖掘方法基于以下两个关键假设：

- 如果一个序列模式可以被用于抽取多个可靠的比较对象对，则这个序列模式很可能是一个指示性抽取模式(IEP)。
- 如果一个比较对象对可以通过一个指示性抽取模式（IEP）抽取出来，这个比较对象对是可靠的。

基于这两个假设，本文设计了如图 3.1 所示的自举算法。自举过程始于一个自定义的 IEP。利用该 IEP，算法抽取一个初始的比较对象对集合。对每个比较对象对，从问题集合中检索所有包含该比较对象对的问题并假定这些问题是比较性问题。根据这些假定的比较性问题和其中包含的比较对象对，算法根据 3.3.1.1 小节描述的模式生成方法生成所有可能的序列模式并通过在 3.3.1.2 小节定义的模式可靠性来评测生成的序列模式。被评测为可靠的模式作为 IEP 被加入到 IEP 库。

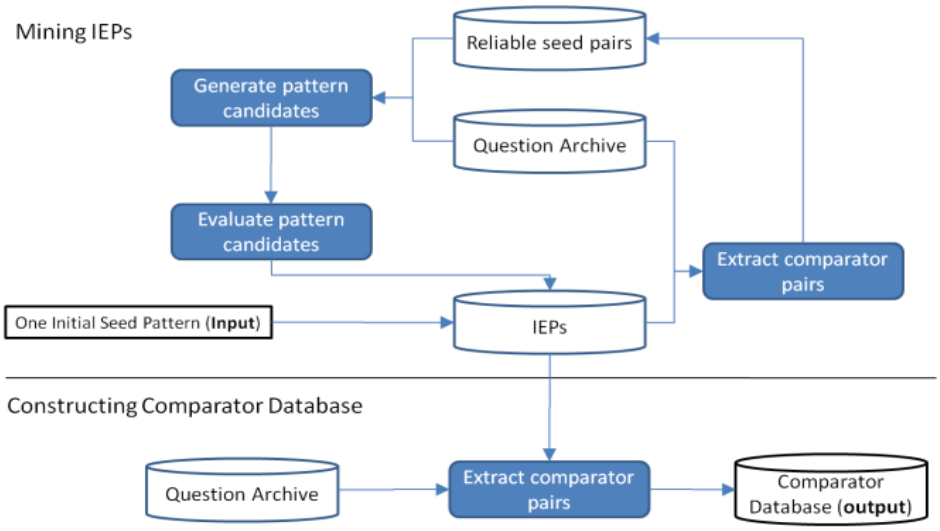

图 3.1 自举算法概览

然后，用新挖掘的 IEP 从问题集合中抽取新的比较对象对，并将其加入到一

个可靠比较对象对集合，在下一次迭代中，它们将被用作模式学习的新的种子。为了从后续的迭代中可以有效地发现新的模式，从中可以抽取已有的可靠比较对象对的问题被从问题集合中删除。以上过程反复迭代，直至没有新的模式生成。算法 3.1 给出了以上算法的伪码:

算法 3.1 弱监督的自举模型

Input: CP, G

Initialize solution: $Q \leftarrow \{\}$, $P \leftarrow \{\}$, $P_{new} \leftarrow \{\}$, $CP_{new} \leftarrow CP$

Repeat
  $P \leftarrow P + P_{new}$
  $Q_{new} \leftarrow ComparativeQuestionIdentify(\ CP_{new})$
  $Q \leftarrow Q\ +\ Q_{new}$
  for $q_i \in G$ **do**
    if $IsMatchExistingPatterns(P, q_i)$ **then**
      $Q \leftarrow Q - q_i$
    end if
  end for
  $P_{new} \leftarrow MineGoodPatterns(Q)$
  $CP_{new} \leftarrow \{\ \ \}$
  for $q_i \in G$ **do**
    $cp \leftarrow ExtractComparableComparators(P, qi)$
    if $cp \neq NULL$ **and** $cp \notin CP$ **then**
    $CP_{new} \leftarrow\ \ CP_{new} + \{cp\}$
    end if
  end for
until $P_{new} = \{\}$
return P

3.3.1.1 模式生成

本文改进了文[45]中介绍的表层文本模式（surface text pattern）挖掘方法来生成序列模式。对任何给定的比较性问题及其包含的比较对象，问题中的比较对象被符号$C 替换。符号#start 和#end 被附加在问题中每个句子的开头和结尾的位置。为了降低序列数据的多样性并挖掘潜在的模式，本文还应用了如表 3.2 所示的一组启发式规则识别问题中的词组，并将词组所对应的词性标注序列替换为相应词组标记。

表 3.2 词组识别的启发性规则

NP*->NP
NN*->NN
NN+NNS->NNS
NP+NPS->NPS
“More”+ADJ->JJR
“Most”+ADJ->JJS
……

算法最终生成的序列模式包括以下三种：

- 词汇模式：词汇模式指的是只包含单词和符号（$C, #start 和#end）的序列化模式。这种模式由后缀树方法生成[61,62,63]且有两个限制：每个模式包含$C 符号的个数大于 1 且每个模式在序列数据中出现的频率大于某个经验阈值β。
- 泛化模式：在某些情况下，词汇模式由于要求问题必须包含某个词汇序列才能与之匹配，可能过于特殊化。因此，我们可以通过将词汇模式中的部分词或词组替换成它们的词性标记来泛化词汇模式。一个除了符号$C 外包含 N 个词或词组的词汇模式最多可生成$2^n - 1$个泛化模式。
- 特殊化模式：在某些情况下，不对被抽取出的比较对象做任何限制的模式可能过于泛化。例如，尽管问题“ipod or zune?”是一个比较性问题，但其对应的模式“<#start $C or $C？>”过于泛化，与该模式匹配的问题中包括很多非比较性问题（如“true or false?”）。因此，本文通过对该模式中比较对象进行限制对该模式进行特殊化，“<#start $C/NN or $C/NN?>”即为特殊化后的模式。

注意，泛化模式产生自词汇模式，特殊化模式产生自泛化模式和词汇模式。最终的候选模式集合是词汇模式、泛化模式和特殊化模式的合集。

3.3.1.2 模式评测

根据算法的第一个假设，候选模式p_i在第k次迭代时的可靠性$R^k(p_i)$定义如下：

$$R^k(p_i) = \frac{\sum_{\forall cp_j \in CP^{k-1}} N_Q(p_i \rightarrow cp_j)}{N_Q(p_i \rightarrow *)} \qquad (3.1)$$

其中，CP^{k-1}指在第$(k-1)$次迭代时已经积累的已知可靠比较对象对集合。$N_Q(x)$指满足条件x的问题的个数。条件$p_i \rightarrow cp_j$代表可以通过应用模式p_i从中抽取cp_j；条件$p_i \rightarrow *$代表可以匹配模式p_i。

但是，公式 (3.1)会遭遇到对可靠比较对象对知识不足的问题。例如，在自举过程的早期，我们只能确定极少可靠比较对象对。在这种情况下，公式（3.1）的值会被低估，从而降低公式(3.1)将 IEP 从非可靠的序列模式中识别出来的有效性。我们将在第k次迭代中的候选序列模式集合表示为$\hat{P}^k$，将可以被$\hat{P}^k$中的序列模式抽取出来但不存在于目前的可靠比较对象对集合中的比较对象对表示为$\widehat{cp}_i$。$\widehat{cp}_i$的支持度定义如下：

$$S(\widehat{cp}_i) = N_Q\left(\hat{P}^k \rightarrow \widehat{cp}_i\right) \qquad (3.2)$$

其中，条件$\hat{P}^k \rightarrow \widehat{cp}_i$代表应用$\hat{P}^k$中的某个序列模式可以从中抽取$\widehat{cp}_i$。直观地讲，如果从多个问题可以通过$\hat{P}^k$中某个模式从中抽取$\widehat{cp}_i$,则$\widehat{cp}_i$在下一次迭代中很有可能被抽取为可靠的比较对象对。基于这一直观假设，支持度 S 大于阈值α的比较对象对$\widehat{cp}_i$被当作潜在可靠比较对象对。本文定义使用潜在可靠比较对象对估计的前瞻序列模式可靠性$\hat{R}(p_i)$如下:

$$\hat{R}^k(p_i) = \frac{\sum_{\forall \widehat{cp}_i \in \widehat{CP}^k_{rel}} N_Q(p_i \rightarrow \widehat{cp}_i)}{N_Q(p_i \rightarrow *)} \qquad (3.3)$$

其中，$\widehat{CP}^k_{rel}$表示基于$\hat{P}^k$的一个潜在可靠比较对象对集合。结合公式（3.1）和公式（3.3），本文最终定义模式的可靠性$R(p_i)^k_{final}$如下:

$$R(p_i)^k_{final} = \lambda \cdot R^k(p_i) + (1-\lambda) \cdot \hat{R}^k(p_i) \qquad (3.4)$$

使用公式(3.4)，算法可以评测所有的候选序列模式并选择可靠性大于阈值γ的模式为指示性抽取模式（IEP）。在评测过程中，所有的参数都是经验值。3.4 节将详细解释如何确定这些阈值。

3.3.2 比较对象抽取

通过应用学习到的 IEP，我们可以很容易地识别比较性问题并收集从问题库的问题中抽取出的比较对象对。给定一个问题和一个 IEP, 比较对象抽取的过程细节如下所示：

- 生成问题对应的序列。如果给定的 IEP 是一个词汇模式，算法只需要对问题进行分词处理，问题对应的序列即分词序列。否则，如果给定的 IEP 是一个泛化模式或特殊化模式，即该 IEP 中包含词性标记，则词性标注和词组识别过程是必须的。在这种情况下，问题对应的序列是词块序列。以问题“which is better phone iphone or nokia n95?”为例，如果我们对该问题应用词汇模式“which is better $C or $C?”，算法生成序列“which| is| better |phone| iphone| or| nokia| n95| ?”。如果我们对该问题应用泛化模式“which is better NN $C or $C?”，则算法生成序列“Which/WP| is/VBS| better/JJR| phone/NN| iphone/NP| or/CC| nokia n95/NP|?/?”
- 检测问题对应的序列是否与给定的 IEP 相匹配。当问题与给定的 IEP 相匹配时，分别抽取出序列模式中两个$C 对应的单词或单词序列作为比较对象对。如果 IEP 是一个特殊化模式，抽取出的比较对象对需要满足该 IEP 对其词性序列给

出的限制。

需要指出的是，大约 67%的比较性问题可以匹配到多个 IEP，且对大约 11%的比较性问题，应用不同 IEP 会抽取出不同的比较对象对。以问题“which is better phone iphone or nokia n95 ?”为例，如果我们应用 IEP“which is better $C or $C?”，将抽取到错误的比较对象“phone iphone”和“nokia n95”。但是，如果我们应用 IEP“which is better NN $C or $C?”，将会抽取到正确的比较对象对“iphone”和“nokia n95”。因此，设计一种恰当的策略用以决定选择哪一个 IEP 从指定问题中抽取比较对象对是必要的。对这一问题，有两个因素是重要的：

- 一个 IEP 在识别比较性问题并抽取比较对象对时的可靠性。以问题“What do you prefer? Coke or Pepsi?And the reason?”为例,IEP“? $C or $C ?”和“NN or $C? and $C ?”都可以被应用到该问题。但是，前者比后者更可靠。在该例子中，前者应用到该问题时抽取到的比较对象对（“Coke” and “Pepsi”) 比较可靠.
- IEP 与问题的匹配度。以问题“which is better, iphone or my touch or HTC G2?”为例，IEP“, $C or $C or NN?”和“, $C or $C ?”都可以被应用于该问题。如果我们应用模式“, $C or $C or NN?”，问题中的 4 个词被准确匹配，即“,”，“or” (第一个)，“or” (第二个)以及“?”。但是，如果我们使用“, $C or $C ?”，只有三个词被准确匹配，即“,”，“or” (第一个)以及“?”。这意味着模式“, $C or $C ?”更适合上述例句。

根据以上的观察，本文尝试了以下策略：

- **随机策略：**给定一个问题，从可以应用于该问题的 IEP 中随机选择一个用于从问题中抽取比较对象对。
- **最大长度策略：**给定一个问题，从可以应用于该问题的 IEP 中选择最长的用于从问题中抽取比较对象对。根据以上的讨论，序列模式越长，问题中可以被匹配的词就越多，这意味着该序列模式与问题的匹配度越高。
- **最大可靠性策略：**给定一个问题，从可以应用于该问题的 IEP 中选择可靠度最高的用于从问题中抽取比较对象对。

3.4 比较对象排序

接下来的问题就是给定一个用户查询请求，如何给出并排序与用户查询请求可比较的对象。本文尝试了以下排序模型。

r3.4.1 基于可比较性的排序方法

直观来说，如果一个对象被较频繁地与输入的用户查询请求相比较，则意味着人们对比较该对象与查询请求对象更有兴趣。基于这一观察结果，本文定义了一种简单的排序函数$R_{freq}(c;e)$，这个函数根据可比较对象c在问题库Q中与用户的查询请求对象e比较的频率对e的可比较对象进行排序：

$$R_{freq}(c;e) = N(Q_{c,e}) \qquad (3.6)$$

其中，$Q_{c,e}$是一个从中可以将c和e抽取为比较对象对的问题的集合。本文将这种排序方法称为基于比较频率的排序方法。

进一步考虑，本文发现，虽然不同的问题形式都可以表达用户的比较意图，但是所表达的比较意图的强度并不相同，这也意味着其所比较的对象之间的可比性是不同的。因此，本文根据在比较对象挖掘阶段用于抽取比较对象对的模式的可靠性来衡量两个对象之间的可比性，并定义基于可比较对象间的可比性的排序函数如公式（3.7）所示，

$$R_{rel}(c;e) = \sum_{q \in Q_{c,e}} R(p_{q,c,e}) \qquad (3.7)$$

其中，$p_{q,c,e}$在比较对象挖掘阶段被选择的从问题q中抽取比较对象对c和e的模式。这种排序方法本文称之为基于可比性的排序方法。

3.4.2 基于图的排序方法

尽管以上定义的可比性对用户查询请求的可比较对象的排序是行之有效的，但是当用户输入的查询请求对象在问题集中极少被作为比较对象时，可比性对其可比较对象的排序能力将大大降低。在这种情况下，我们需要考虑比较对象的代表性（Representability）。当一个比较对象在用户感兴趣的领域中经常被用作基准被比较时，我们称该比较对象是有代表性的。例如，当一个用户想要购买一个智能手机且考虑购买“Nokia n82”时，“Nokia N95”往往是他想要比较的对象。这是因为“Nokia N95”是一个很著名的高端智能手机且经常被用作比较基准帮助用户更加了解感兴趣的智能手机的性能。

一个考虑比较对象的代表性的可行方案是采用基于图的方法，例如 PageRank。当一个比较对象跟输入的查询请求对象的其它多个可比对象可比，这个比较对象将被认为是一个在排序中有价值的比较对象。基于这一想法，本文尝试使用 PageRank 算法为用户查询请求对象提供排序的可比较对象列表。这种排序方式将

兼顾可比较对象的代表性及其与用户查询请求对象的可比性。

本文将用户查询请求对象的可比较对象之间的比较关系图表示为：$G = (V, E)$，V是节点集合，由与用户查询请求对象的可比较对象组成；E是边的集合，如果在抽取出的比较对象对中，v_i和v_j所代表的比较对象之间是可比的，则v_i和v_j之间存在一条边e_{ij}。图 3.2 所示为针对用户输入“Obama”所建立的图。

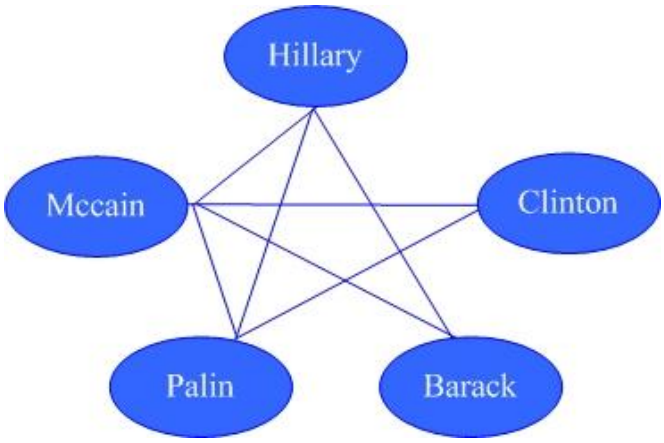

图 3.2 查询“Obama”的可比较对象关系图

本文通过基于图G的 PageRank 算法对G中的比较对象排序。其中，转移概率$P(v_i|v_j)$定义如下：

$$P(v_i|v_j) = \frac{Count(v_i, v_j)}{Count(v_j, *)} \tag{3.8}$$

$Count(v_i, v_j)$是比较对象对v_i和v_j从问题库 Q 中抽取出来的频率。对一个用户输入的比较对象e,节点v_i的PageRank 分值定义如下[64,65]:

$$S^{(k)}(v_i) = \lambda \cdot S^{(0)}(v_i) + (1-\lambda) \cdot \sum P(v_i|v_j) \cdot S^{(k-1)}(v_j) \tag{3.9}$$

其中，节点v_i按以下方法被初始化，

$$S^{(0)}(v_i) = \frac{Count(e, v_j)}{Count(e, *)} \tag{3.10}$$

所有节点的分值按公式(3.10)循环迭代直至收敛。

3.5 实验

3.5.1 实验设置

3.5.1.1 数据源

所有实验在一个从Yahoo! Answers收集的包含约60M个问题的问题库上进行。Yahoo! Answers 的问题由问题标题、问题描述及问题答案三部分组成，本文只利用了问题标题部分。原因在于：一方面，通常问题标题可以清楚得表达提问者的主要意图，如果提问者有比较某两个对象的意图，通过分析问题标题即可获得；另一方面，问题标题的形式简单，更容易获取模式。

3.5.1.2 比较对象抽取的评测数据

本文分别创建了两个数据集进行评测。首先，我们从 Yahoo! Answers 的 26 个顶层目录中的每个目录分别采样 200 个问题，获得 5,200 个问题。然后，两个标注者分别将采样得到的每一个问题手工标注成比较性问题(comparative)，非比较性问题(non-comparative)或未知类型(unknown)。其中，139(2.67%)个问题被分类为比较性问题，4,934(94.88%)个问题被分类为非比较性问题，127(2.44%)个问题被分类为未知类型。该数据集被称为 SET-A。

由于 SET-A 中只有 139 个比较性问题，对于测试而言数目过少，因此，本文创建了另一个包含较多比较性问题的数据集。首先，我们手动创建了一个包含 53 个例如“or”，“prefer”等对识别比较性问题有指示性的关键词的集合。在 SET-A 中，97.4%的比较性问题至少包含该关键词集合的一个关键词。然后，从每个 Yahoo! Answers 的顶层目录中分别采样另外 100 个至少包含一个关键词的问题并以与标注 SET-A 同样的方法标注这些问题。这个数据集被称为 SET-B。它包含 853 个比较性问题和 1,747 个非比较性问题。为了评测算法抽取比较对象对的性能，我们对 SET-B 中的比较性问题所包含的比较对象进行标注。比较对象可以是名词/名词词组，动词/动词词组，代词等。不同词性的比较对象分布如表 3.3 所示。

表 3.3 不同类型比较对象的分布

比较对象类型	数目	百分比
名词或名词词组	1,471	84.78%
代词	3	0.17%
动词或动词词组	78	4.50%
形容词或形容词词组	60	3.46%
以上皆不是	123	7.09%
总数	1,735	100%

对比较性问题识别的实验，本文使用标注的 SET-A 和 SET-B。对比较对象抽取的实验，本文只使用 SET-B。问题库中余下的未标注的问题集合（被称为 SET-R）被用于训练本文提出的弱监督方法。

本文认真地实现了 J&L 的方法作为评测的基准算法。我们先从标注的问题中学习 CSR，然后将这些 CSR 作为特征建立一个识别比较性问题的统计分类器。我们检验了 J&L 的实验中所尝试的支持向量机(SVM)和朴素贝叶斯(Naïve Bayes)模型的分类性能。另外，我们从 SET-B 中学习 LSR 进行比较对象抽取。

在弱监督方法的实验中，算法先将自定义的 IEP“<#start NN/$C vs/cc NN/$C ?/. #end>”应用到所有的 SET-R 的问题中并收集到 12,194 个比较对象对，以此启动自举过程。弱监督方法有四个需要确定的经验参数，即 α, β, γ, 和 λ。算法先从包含抽取到的初始比较对象对的问题中挖掘候选模式，利用这些候选模式抽取一个新的候选比较对象对集合（包含 59,410 对）。我们从这些新的候选比较对象对中随机选择 100 对，并手动将其分类为可靠的或不可靠的比较对象对。然后，α 被设置为在不损害召回率的情况下使得可靠比较对象对识别准确较高的候选比较对象对频率阈值。在本文的实验中，α 设为 3。相似的，用于候选模式评估的参数 β 和 γ 分别被设为 10 和 0.8。对公式(3.3)中的插值参数 λ，算法假设两个可靠性分值同等重要，简单地设为 0.5。

本文使用准确率、召回率和 F1 值作为比较性问题识别和比较对象抽取的评测指标。所有的结果都使用 5 层交叉检验。注意，J&L 的方法需要训练数据，但是本文提出的算法只需要使用未标注数据(SET-R)并通过弱监督的方法获得恰当的参数设置。两种方法在同样的 5 层交叉检验的数据划分上进行，所有的评测值通过在 5 个划分上平均获得。

本文使用微软亚洲研究院自然语言计算组开发的 NLC-PosTagger 对问题进行词性标注。它采用修改的 Penn Treebank 词性集作为输出，例如，NNS（名词复数），NN（名词），NP（名词词组），NPS（名词词组复数），VBZ（动词第三人称单数，一般现在时），JJ（形容词），RB（副词），等等。

3.5.1.3 比较对象排序的评测数据

通过将挖掘到的 IEP 应用到问题库，本文自动识别出 679,909 个比较性问题，并利用最大长度策略从中抽取出 328,364 个比较对象对。对每个比较对象，平均可以抽取 3 个与之可比较的对象。特别是，一些受欢迎的比较对象，例如“windows”，“apple”等，与其可比较的比较对象的数目可以达到 500。基于这些比较对象对，本文尝试了不同的可比较对象排序算法。

基于抽取出的每个比较对象的可比较对象的数目，本文建立三种不同的用户输入集合，即频繁对象集合（FREQ-QUERY），中等对象集合（MID-QUERY）及稀疏对象集合（RARE-QUERY）。FREQ-QUERY 由 200 个可比较对象数目大于 50 的比较对象组成；MID-QUERY 由 200 个可比较对象数目大于 20 但小于 30 的比较对象组成；RARE-QUERY 由 200 个可比较对象数目小于 10 的比较对象组成。每个集合中的比较对象都是随机选取。

3.5.2 实验结果

3.5.2.1 比较性问题识别和比较对象抽取

表 3.4 列出了本文的实验结果。在该表中“比较性问题识别”列中列出的是比较性问题识别的性能。“比较对象抽取”列中列出的是给定一个比较性问题，正确抽取其中包含的比较对象的性能。“All”代表的是自动识别比较性问题并从识别出的比较性问题中抽取比较对象的系统整体性能。注意，这里 J&L 的方法的在本文的数据集及问题设定下的性能与在论文[50]、[51]的设定下的性能是可比的，这说明本文较好地重现了其方法。表中带*号的值较 J&L 显著提高（t-test，$p<0.001$）。

从准确率来看，在比较性问题识别上 J&L 的方法与弱监督自举算法是可比的；而在比较对象抽取上，弱监督自举算法则优于 J&L 的方法 6%。但是，J&L 的方法的召回率远远低于弱监督自举算法。从召回率来看，在比较性问题识别和比较对象抽取上弱监督自举算法分别优于 J&L 的方法 35%和 22%。分析发现，J&L 的方法的低召回率主要是由学习到的 CSR 在测试集上覆盖率低引起的。

表 3.4 弱监督自举算法与 J&L 方法的性能比较

	比较性问题识别			比较对象抽取		All		
	J&L		弱监督自举算法	J&L	弱监督自举算法	J&L		弱监督自举算法
	SVM	NB				SVM	NB	
召回率	0.601	0.537	0.817*	0.621	0.760	0.373	0.363	0.760*
准确率	0.847	0.851*	0.833	0.861	0.916	0.729	0.703	0.776*
F1 值	0.704	0.659	0.825*	0.722	0.833	0.493	0.479	0.798*

从系统的整体性能来看，弱监督自举算法显著优于 J&L 的方法。弱监督自举算法在 F1 值上提高了约 55%。这一结果主要得益于：弱监督自举算法用一个模式同时识别比较性问题和抽取比较对象；而 J&L 的方法使用两种模式，即 CSR 和 LSR，其性能由于错误的扩散明显降低。J&L 的方法的 F1 值在“All”中分别较“比

较性问题识别”和“比较对象抽取”中差 30%和 32% ，而弱监督自举算法在“All”中性能只有少量降低（大约 7-8%）。

另外，本文还分析了模式泛化和模式特殊化的有效性。表 3.5 显示了这一结果。尽管本文进行模式泛化和模式特殊化的方法都很简单，但是这两个过程对性能的提高都产生了显著影响。结果表明了学习多样化的序列模式对捕捉比较性问题多样性表达的重要性。在学习到的 6,127 个 IEP 中，5,930 个模式是泛化模式，171 个模式是特殊化模式，只有 26 个模式是非泛化/特殊化的模式。

表 3.5 特殊化模式及泛化模式对系统性能的影响

	召回率	准确率	F1 值
词汇模式	0.689	0.449	0.544
+特殊化模式	0.731	0.602	0.665
+泛化模式	0.760	0.776	0.768

表 3.6 展示了不同的比较对象抽取策略的性能。结果表明，最大可靠性策略与随机策略的性能相似，最大长度策略性能稍优。这是因为通常 IEP 拥有相近的可靠性分值，很难利用可靠性区分并选择恰当的 IEP 抽取给定比较性问题中的比较对象。

表 3.6 不同抽取模式选择策略的性能比较

	召回率	准确率	F1 值
随机策略	0.744	0.891	0.810
最大长度策略	0.760	0.916	0.833
最大可靠性策略	0.747	0.891	0.813

为了测试弱监督自举算法对初始种子 IEP 的鲁棒性，本文比较了使用两个不同的种子 IEP 时的性能。结果如表 3.7 所示。尽管不同的 IEP 从问题库中抽取的种子比较对象对数目相差很大，弱监督自举算法的性能依旧是稳定的。结果表明，弱监督自举算法对 IEP 的选择并不敏感。

表 3.7 不同初始模式对系统性能的影响

初始模式	种子比较对象对个数	F1 值
<#start nn/$C vs/cc nn/$C ?/. #end>	12,194	0.768
<#start which/wdt is/vb better/jjr , nn/$C or/cc nn/$C ?/. #end>	1,478	0.760

表 3.8 同样显示了弱监督自举算法的鲁棒性。在表 3.8 中，“全部”行列出的是种子 IEP 抽取出的所有比较对象对都被用于接下来的迭代过程时算法的性能；“部分”行列出的是只从其中采样 1000 个比较对象对时算法的性能。结果显示，性能并没有发生太大变化，这又一次说明算法对种子比较对象对的个数并不敏感。

表 3.8 不同比较对象对个数对系统能的影响

种子比较对象对个数	召回率	准确率	F1 值
全部（12,194）	0.760	0.774	0.768
部分（1000）	0.724	0.763	0.743

除此之外，本文对弱监督自举算法没有能够正确抽取比较对象对的情况进行了错误分析：

- 8%的未识别出的比较性问题是由于其包含的比较对象的形式稀有。例如，模式“#start $C or $C ? #end”并不足够准确。当我们限制模式，只有“#start NN/$C or NN/$C ? #end”这一个特殊化模式产生，在这种情况下，“Saving a dog or breaking an entering ?”就不能被识别。
- 3.3%的未识别出的比较性问题是由错误的词性标注引起的。例如，如果“survey: ps2 VS. wii?”被词性标注为“VB : NN CC NN .”，则它能很容易地被模式“: NN/$C CC NN/$C ?”识别出来，但是它被标注为“VB: CD NN NN .”，这使得它被分类为非比较性问题。
- 88.7%的未识别的比较性问题由于比较对象的上下文的模式稀有。这种情况又可以分为以下三类：
  1) 比较对象的上下文中包含稀有字（24.6%）。例如，问题“What's your preference air cooled or water cooled 911's?”很难识别，因为由于“preference”出现的频率很低以至于包含该词的模式“what's your preference $C or $C ?”很难被挖掘。而模式“what's your NN $C or $C ?”又不够可靠。
  2) 比较对象的上下文中的词是频繁的，但是组合模式是稀有的（61.9%）。当问题中包含描述性句子时，这种情况尤其严重。例如，对问题“I have the option of going to either Japan or Argentina; both are a month long in January. Where should I go?”很难找到一个具有普遍性的模式用以识别该比较性问题。
  3) 非规范拼写（2%）。社区问答服务中用户生成的问题表述更加自由。一方面存在很多如“thx”(“thanks”)，“u”（“you”）等缩写，这些缩写不易识别导致了词性标注的错误。另一方面，很多比较性问题识别的关键

词发生拼写错误，例如“diffrent”,“wich”等。包含这些偶然的错误拼写错误的问题很难匹配到一个已挖掘出的 IEP。

3.5.2.2cQA 问题和其它数据源比较对象抽取和排序结果的比较

挖掘到的比较对象对对用户比较意图的覆盖率无疑是评测的一个重要因素。通过从社区问答服务抽取比较对象对与从其它资源中抽取的比较对象对的比较，本文定性地分析了其覆盖率。本文的研究将从社区问答服务中抽取的比较对象对与从查询日志中抽取的进行了比较。

根据我们的观察，“vs”是查询日志中比较意图的一个强有力的线索。本文从某个商业搜索引擎的跨时 6 个月，大约 13M 条频率大于 10 的查询记录中筛选形如“X vs Y”的查询记录，然后从这些查询记录中抽取“X”和“Y”对应的字符串。如果“X”和“Y”的长度都大于 3，本文把“X”和“Y”当作一个比较对象对。从查询日志中获得的比较对象对总数是 4,200。接着，本文计算了分别来自查询日志和社区问答服务的问题的比较对象对的重叠率。

结果表明：在考虑频率时,从社区问答服务的问题中抽取的比较对象对覆盖了 64%的从查询记录中抽取的比较对象对；但是查询日志中抽取出的比较对象对只覆盖了从社区问答服务的问题中抽取的比较对象对的 0.6%。对于在查询日志中频率大于 100 的查询请求，从社区问答服务的问题中抽取的比较对象对的覆盖率达到 76%。考虑到用“X vs Y”模式从查询日志中抽取出的比较对象对有很多噪音，以上覆盖率的数目足以说明：本文构建的比较关系对数据库可以覆盖大部分查询日志中抽取出的比较对象对，社区问答服务的问题对于比较对象的抽取是个丰富的资源。

可比较对象排序目的是给定一个用户查询请求对象，将与其可比的可比较对象按某种准则排序并呈现给用户。由于比较的主观性，决定一个可比较对象对一个用户查询请求是否应该优先推荐并不是件容易的事。因此，本文的研究只是计算了可比较对象在社区问答服务的问题中和在 Web 网页中排序结果的相关性，从而证明本文给出的可比较对象排序并不偏置于社区问答服务的问题。本文用“vs”将两个可比较对象“X”和“Y”连接在一起形成查询请求，并将查询在搜索中返回结果的数目作为“X”和“Y”在网页中被比较的频率。细节如下所述。

首先，对一个输入对象 Q，本文用 3.4 节所述的方法将与其可比较的对象 E 排序。然后，我们构造形如“Q vs E”和“E vs Q”的查询（注意 Q 和 E 的字符串需要用引号），将构造的查询输入搜索引擎并获取返回结果的数目作为 Q 与 E 在网页中被比较的次数。如果很多人对比较 Q 和 E 感兴趣，相应地将会有很多网页包含“Q vs E”和“E vs Q”。

图 3.3 显示了给定用户输入的查询请求对象，其可比较对象 E 在社区问答服务的问题中的排序和按其在网页中被比较的频率排序的相关度。如图所示，随着 E 的排序位置降低，其在网页中与 Q 比较的频率逐渐降低。尤其是，排在前面区域的 E，包含其与 Q 比较信息的网页数目远远大于其它区域。图 3.3 证明了社区问答服务中捕捉到的用户比较兴趣点与网页中的紧密相关。

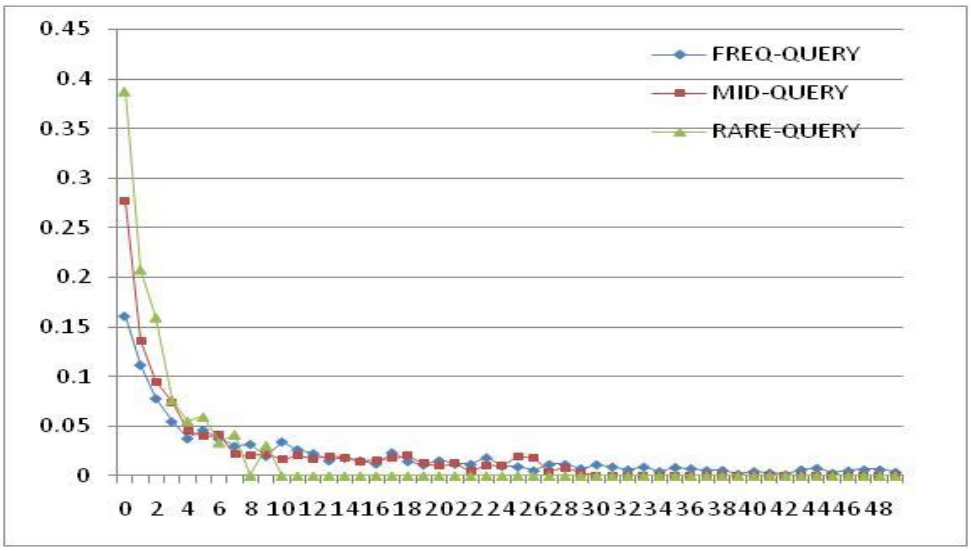

图 3.3 随着可比对象 E 在排序结果中序的变化查询“E vs Q”和“Q vs E”检索出的网页的个数。

3.5.2.3 比较对象抽取和排序实例

表 3.9 列出了在问题集 Q 中对一个用户查询请求对象，如“Channel”，“Gap”，它经常被比较的前 10 个对象。如表所示，本文提出的比较对象挖掘算法成功地发现了现实中用户查询请求对象的可比较对象。例如，对查询请求“Channel”，大多数结果都是高端时尚品牌，如“Dior”，“Louis Vuitton”等；但是“Gap”的可比较对象通常包含相似的较为平价的服装品牌，例如“Old Navy”，“Banana Republic”等。对篮球明星“Kobe”，大多数排在前面的可比较对象也是一些著名运动员。同样的，对公司“Cannon”也有一些有趣的结果。“Cannon”以其数码相机、打印机等产品出名，因此它可以与以这些产品闻名的不同公司相比，比如，作为打印机的生产商，它可以与“HP”，“Lexmark”或“Xerox”相比较；而作为数码相机的生产商，它也可以与“Nikon”，“Sony”或“Kodak”相比较。除了这些作为品牌或公司的比较泛化的比较对象，算法也挖掘到一些针对特定事物的可比较对象。例如，算法推荐“Nikon d40i”，“Canon rebel xti”，“Canon rebel

xt”，“Nikon d3000”，“Pentax k100d”，“Canon eos 1000d”作为特定的数码相机产品“Nikon 40d”的可比较对象。

表 3.9 查询的可比对象列表实例

	Chanel	**Gap**	**Nikon d40**	**Kobe**	**Canon**
1	Dior	Old Navy	Nikon d40i	Lebron	Nikon
2	Louis Vuitton	American Eagle	Canon rebel xti	Jordan	Sony
3	Coach	Banana Republic	Canon rebel xt	MJ	Kodak
4	Gucci	Guess by Marciano	Nikon d3000	Shaq	Panasonic
5	Prada	ACP Ammunition	Pentax k100d	Wade	Casio
6	Lancome	Old Navy brand	Canon eos 1000d	T-mac	Olympus
7	Versace	Hollister	-	Lebron James	Hp
8	LV	Aeropostal	-	Nash	Lexmark
9	Mac	American Eagle outfitters	-	KG	Pentax
10	Dooney	Guess	-	Bonds	Xerox

表 3.10 给出了本文提出的比较对象挖掘与项目推荐的区别，斜体部分为两者重叠的部分。如表所示，“Google related searches”推荐为用户输入的查询提供两种相关查询推荐：（1）与用户输入的查询的子话题相关的查询（例如，当用户输入查询“Channel”时推荐“Channel handbag”）；（2）其可比较项（例如，当用户输入查询“Channel”时推荐“Dior”）。这一结果进一步确认了我们所声明的比较对象挖掘与查询/项推荐是相关的但是并不相同。

表 3.10 Google 相关查询推荐实例

Chanel	**Gap**	**Nikon d40**	**Kobe**	**Canon**
Chanel handbag	Gap coupons	Nikon d60	Kobe Bryant stats	Canon t2i
Chanel sunglass	Gap outlet	Nikon d40x	Lakers Kobe	Canon printers
Chanel earrings	Gap card	Nikon d40 review	Kobe espn	Canon printer drivers
Chanel watches	Gap careers	Ritz camera	Kobe Dallas Mavericks	Canon downloads
Chanel shoes	Gap casting call	Nikon d80	Kobe NBA	Canon copiers
Chanel jewelry	Gap adventures	Nikon d50	Kobe 2009	Canon scanner
Chanel clothing	*Old navy*	Nikon d40 kit	Kobe san Antonio	Canon lenses

表 3.11 基于可比性和基于图的排序方法结果比较

	Obama		**iphone**		**Xbox 360**	
	Comparability	PageRank	Comparability	PageRank	Comparability	PageRank
1	Mccain	Mccain	ipod touch	ipod touch	Playstation 3	Playstation 3
2	Clinton	Clinton	Blackberry	Blackberry	Wii	Wii
3	Hillary	Hillary	itouch	itouch	Ps3	Ps3
4	Palin	Hilary	Blackberry storm	Blackberry storm	Nintendo Wii	Nintendo Wii
5	Hilary	Palin	Voyager	Voyager	ipod touch	Psp
6	Bush	Bush	Sidekick lx	Sidekick lx	Psp	Xbox
7	Biden	Biden	Sidekick	Sidekick	Xbox	ipod touch
8	Hillary Clinton	Hillary Clinton	Blackberry Curve	Blackberry Curve	Xbox 360 elite	Xbox 360 elite
9	Osama	Osama	ipod	ipod	pc	pc
10	John Mccain	John Mccain	Instinct	Blackberry Pearl	Playstation 2	Playstation 2
	BMW 328i		**Nokia N75**		**Nikon D200**	
	Comparability	PageRank	Comparability	PageRank	Comparability	PageRank
1	Toyota Avalon	**Cadillac Cts**	LG Shine	LG Shine	Nikon D80	**Canon 30D**
2	BMW 335i	Toyota Avalon	Samsung Sync	Blackberry Pearl	Canon 40D	Nikon D80
3	Mercedes c300 sport	Integra Gsr	Black Berry Curve	Samsung Sync	**Canon 30D**	Canon 40D
4	Audi A3	Acura TL	Motorola k1 Krzr	**Sony Ericsson w580i walkman**	Canon EOS 40D	Canon EOS 40D
5	Honda Accord 08	Honda Accord 08	Samsung D807	Nokia 6555	Canon EOS 30D	Nikon D40x
6	Acura TL	Audi A3	ipod nano	Samsung D807	Canon EOS 5D	Canon EOS 400D
7	Integra Gsr	Lexus 350	Nokia 6555	ipod nano	Nikon D40x	camera
8	**Cadillac Cts**	BMW 335i	Motorola Razr2	Motorola Razr2	Canon EOS Rebel xsi	Canon EOS 30D

表 3.11 基于可比性和基于图的排序方法结果比较(续)

	Obama		**iphone**		**Xbox 360**	
	Comparability	PageRank	Comparability	PageRank	Comparability	PageRank
9	Mercedes c300	Mercedes c300	Motorola A1200 Ming	Motorola A1200 Ming	Sigma SD14	Canon EOS 5D
10	Lexus 350	Mercedes c300 sport	**Sony Ericsson w580i walkman**	Motorola k1 Krzr	Canon EOS 400D	Canon EOS Rebel xsi

表 3.11 比较了基于可比性和基于图的可比较对象排序方法。对某些与不同的可比较对象比较的频率相差较大的用户查询请求对象，如“Obama”，“iphone”和“xbox 360”，两种方法的排序结果差别并不大。这是由于对这种用户输入，在用基于图的排序方法为与其可比较的对象排序时，比较频率起了很重要的作用。但是，对那些与其可比较对象比较频率相似的用户查询请求对象，例如“BMW 328i”，“Nokia N75”和“Nikon D200”，基于可比性和基于图的排序方法的排序结果的区别变得很明显。例如，“Cadillac Cts”与“BMW 328i”的可比较对象中多个对象是可比的，因此，在基于图的排序算法中，尽管“Cadillac Cts”与“BMW 328i”比较的频率并不比“Toyota Avalon”高，但是被排序较前。

表 3.12 参数对基于图的排序方法的影响

λ	0	0.2	0.4	0.6	0.8	1
1	Mccain	Mccain	Mccain	Mccain	Mccain	Mccain
2	Clinton	Clinton	Clinton	Clinton	Clinton	Clinton
3	Hillary	Hillary	Hillary	Hillary	Hillary	Hillary
4	Hilary	Hilary	Hilary	Hilary	Hilary	Palin
5	Palin	Palin	Palin	Palin	Palin	Hilary
6	Bush	Bush	Bush	Bush	Bush	Bush
7	Biden	Biden	Biden	Biden	Biden	Biden
8	Hillary Clinton	Hillary Clinton	Hillary Clinton	Hillary Clinton	Hillary Clinton	Hillary Clinton
9	Osama	Osama	Osama	Osama	Osama	Osama
10	John Mccain	John Mccain	John Mccain	John Mccain	John Mccain	John Mccain

表 3.12 所示的是对用户查询请求对象“obama”，参数λ对于基于图的排序算法的排序结果的影响。λ等于 0 时，只考虑比较对象的代表性。当λ等于 1 时，只考虑比较对象与“Obama”的可比性。我们可以看到，随着λ的变化，由于“Hilary”更经常与其它比较对象比较，只有“Hilary”和“Palin”的顺序发生的变化。排序结果如此一致主要是因为，当一个比较对象经常与用户输入相比较时，它通常也是一个好的比较基准，在相应的比较领域有一定的代表性。

尽管以上给出的例子中，可比较对象的排序结果合理地反映了用户比较的兴趣点，但是仍然存在一些问题：

- 结果中的冗余，例如，“Hillary”和“Hillary”，必须通过自动的方法去除。这些冗余主要是由比较对象的别名或词的错误拼写引起的。另外，去除与给定的输入对象的可比较对象中的噪音，例如“camera”，也是一大挑战。
- 某些情况下，对一个对象而言，其比较关系是多维的。例如，对用户输入“London”，与之可比较的比较对象可以是英国与其可相提并论的城市，如“Manchester”；也可以是其它著名的旅游地，如“Paris”。在这种情况下，有必要区别不同比较意图下的比较对象。

3.6 本章小节

本章首先提出针对具体的应用场景，查询用户需要与用户的查询请求有不同语义关系的查询推荐。然后，以用户多选择决策场景的为例，提出基于比较关系的查询推荐。本文利用社区问答服务中用户提出的问题库，基于一个弱监督的方法识别比较性问题的同时从中抽取存在比较关系的对象，即比较对象对。本章依赖一个直观的想法：好的识别比较性问题的模式应该能抽取好的比较对象对；好的比较对象对通常出现在明显的比较性问题中。利用这一想法，本章自举抽取和识别过程。通过利用大量未标注的数据和需要少量监督决定四个参数的自举过程，本章挖掘出 328,364 个比较对象对及 6,869 个抽取模式。

实验结果表明，本章提出的方法对比较性问题识别和比较对象抽取都是行之有效的，在获得高召回率的同时也保证了高准确率。本章给出的例子表明挖掘出的比较对象对确实反映了用户的比较兴趣点。

本章中抽取的比较对象可以是解决方案、服务、产品、旅游地点、教育机构、人物等，可以被用于商业搜索或产品推荐系统。例如，在用户做出购买决定之前，自动建议用户在比较行为中可以比较的对象将对用户起到很大的辅助作用。另外，本章的结果还可以为想要准确定位其竞争者的公司提供有用信息。

在未来的工作中，我们将提高模式选择策略的准确性，并挖掘稀有的模式。如何识别比较对象的别名关系（例如“LV”和“Louis Vuitton”）以及如何区分对象的歧异（例如，Paris 在“Paris vs. London”中是一个地名，而在“Paris vs Nicole”中是名人）都将是有意义的研究方向。另外，摘要比较给定比较对象对的问题的答案也是待研究的方向之一。

第四章 基于比较关系的用户查询意图识别

4.1 引言

基于关键字查询的搜索引擎中，用户往往只能使用有限数量的词汇来抽象和概括他们的需求。在用户将其需求抽象成有限的查询关键字的过程中，部分有用信息被丢失，从而导致用户查询意图不慎清晰。目前，搜索引擎搜索的结果通常是满足用户各种查询意图的文档的合集，用户需要阅读大量相关文档才能获得自己确实所需的信息。因此，在对用户的查询请求进行搜索之前，尽可能确定用户的查询意图，并进行面向用户查询意图的搜索将有利于更准确地找到用户想要的信息。

如上所述，通常一个由一个或多个查询关键字组成的用户查询请求可能出于多种不同的意图。举个最经典的例子，当用户查询“apple”时，可能是查询一种水果，也可能是查询一种电子产品的品牌。而“apple”作为电子产品品牌时，用户可能想要查询“apple”的产品，也可能想要查询“apple”的网点分布。如果用户想要购买“apple”的产品，比如用户输入“apple itouch”，有可能是想搜索产品介绍，有可能是想看不同网站的价格比较，还有可能是想看该产品与其它同类产品的比较。即使我们确定用户查询的意图是想要将“apple itouch”与其它产品相比较，用户还可能想比较产品的不同方面。比如，从产品升级的角度，用户可能想将其与“ipod classic”，“iphone”相比较；从娱乐功能性的角度，用户可能想将其与“psp”相比较等。可见，确切地理解用户的意图并不是件简单的任务。

已有的关于用户查询意图的研究多关注于用户查询意图的分类[30-35]。这些研究基于一些人工定义的用户意图分类体系，利用用户的查询点击记录、不同的查询意图对应的相关文档的词汇分布及查询请求中的关键词信息对用户查询进行分类。这些研究在确定用户查询所需信息类型方面有很大帮助，但是距离我们以上所述用户查询意图识别的目标还有很大的距离。一方面，建立完善的适用于所有查询的分类体系并不是件容易的事，到目前为止查询意图分类领域都没有标准的分类体系；另一方面，查询意图分类并不能识别一些与具体的查询请求相关的查询意图，例如前面例子提到的“apple”的歧义造成的不同查询意图及“ipod touch”从不同的角度与不同产品比较时的不同查询意图，这些意图无法事先定义导致其不能通过分类识别。

根据以上分析，本文以用户的比较行为为例，研究了面向开放域的非监督的用户比较意图识别方法。直观地讲，在用户的比较行为中，在用户不同的查询意图下，其查询请求将与不同的对象是可比较的。例如，当用户查询“apple”的意

图是查询一种水果时，该查询与“orange”，“grape”等是可比的；而当用户查询“apple”的意图是查询一种电子产品品牌时，该查询与“IBM”,“HP”是可比的。因此，识别用户查询请求的不同的可比较对象集合，可以识别用户的不同查询意图，从而通过交互帮助用户进一步确认自己的查询意图，以便提高搜索的准确率。

本文利用用户查询请求的可比较对象间的可比较关系，提出一种基于图聚类的用户比较意图识别方法。每个可能的用户比较意图由一组用户查询请求的可比较对象表示，且通过信息抽取的方法被赋予一个语义标签。实验证明，本文提出的用户比较意图识别方法的准确率达到 92.7%。另外，本文还针对用户的比较行为建立了一个比较搜索系统，该系统在识别用户查询请求的不同比较意图的同时还提供了不同的比较意图下，不同可比较对象与用户查询请求的比较信息。

本章的组织结构如下：第 4.2 节介绍了相关工作；第 4.3 节描述了基于比较关系的用户意图识别方法及用户意图标签方法；第 4.4 节给出了实验结果并介绍了一个基于比较意图识别的比较搜索系统；最后在第 4.5 节总结本章。

4.2 相关工作

本章的工作主要与两个方面的研究相关：查询意图分类和查询消歧。本节将概述这两方面研究的相关工作并比较其与本章的用户查询意图识别任务的不同之处。

4.2.1 查询意图分类

查询分类即是为用户的查询从预先定义的分类体系中选择可能的类别。在查询意图分类领域，目前没有标准的分类体系，不过后续的关于查询意图的分类研究工作都受到了 Broder 等人提出的分类标准的影响。Broder 等人[30]在用户调查和对查询日志进行手工分类的基础上将查询意图分为导航类(Navigational)、信息类(informational)和事物类（transactional）。Rose 等人[31]在 Broder 的基础上，提出了更细致的层次结构，最终得到 11 个小类别。Baeza-Yates 等人[32]虽然认同将查询分为三类，但他们在 Broder 等提出的分类标准地基础上，对具体的类别和定义有较大的修改，他们将查询分为信息类、非信息类（Not Informational）及歧异类。

与所有的分类任务一样，查询意图的分类可分为特征提取、训练集和测试集的构造、分类器的训练及应用几个步骤。在具体问题中，不同分类问题的特征的提取尤为重要。查询意图分类由于查询本身的信息量少，使得从中提取特征显得尤为困难。Kang 等人[33]构造了两个文档集，第一个是包含话题类页面的，第二个是包含主页类页面的，据此计算查询在这两个文档集合里分布的差异、互信息的

差异、作为锚文本的使用率以及词性信息：最后一个特征只依赖于查询本身，而前三个特征借助了外部信息。Beitzel 等人[34]利用标注以后的查询，使用分类器和选择性属性发掘查询内在的特征，同时还使用了对某段时间比较流行的查询进行精确匹配的方法（使用查询自身的字符串作为特征）。Jansen 等人[35]使用各类（导航、事务、信息）查询的一组启发式特征来区分查询，比如他们总结出含有“ways to”，“how to”等词汇的查询极有可能属于信息类查询。Lee 等人[36]使用了查询的历史中被点击的 url 分布、查询的平均点击次数。Dou Shen 等人[20]使用了提交查询以后由搜索引擎返回的检索结果里文档的标题、搜索引擎返回的检索结果中的文档片段以及结果文档的全文作为特征来训练分类器。

4.2.2 查询消歧

查询消歧的主要目的是为用户查询请求中包含的词找到恰当的同义词或词义相关词并将其用于检索，从而提高检索的召回率和准确率，是查询扩展的关键研究之一，获得了广泛的关注[66-74]。这些研究虽然目的与本文的研究相似，都是旨在更好地理解用户查询，但与本文的研究存在本质的差别。其一，从任务来看，本文的研究并不止于消歧查询词的语义，例如，对于查询请求“apple itouch”，其表示一个电子产品，并不存在歧异。而在本文设定的任务中，我们仍需要进一步猜测用户可能的比较角度。其二，技术角度来讲，本文并不对查询请求中的每个词进行词义消歧，而是将用户的查询请求看作一个整体，通过对相关查询推荐进行聚类实现不同查询意图的识别。事实上，通过应用查询消歧提高检索性能的尝试并没有达到预期中的效果。Krovets 和 Croft[75]研究了查询请求中的查询词及其在相关文档中的词义的匹配关系，并指出：查询请求中词与词的搭配和共现关系实际上已经实现了部分词义消歧。另外，Rutger 大学的研究人员曾作了一系列关于检索系统的人机交互实验[75-78]，这些研究显示，相对于完全透明的查询扩展方式，用户更喜欢他们可以加以选择、控制的查询推荐方式。

4.3 基于比较关系的用户意图识别方法

本节将介绍在用户的比较行为中，如何利用上一章挖掘的比较关系及基于比较关系的用户推荐，识别用户比较意图。系统结构如图 4.1 所示。首先，给定一个用户查询请求，系统先对该查询请求的可比较对象进行别名识别；接着，将不包含冗余的可比较对象集合进行聚类，每个聚类代表一个可能的用户意图。这里，“Comparator Database”存储了按本文上章所述方法从 Yahoo! Answers 的问题中抽取的比较关系，从这些比较关系中，可以获取用户查询请求的可比较对象及这些可比较对象之间的比较关系；然后，系统利用 Web 信息给可比较对象聚类产生

的每个类进行标记；最后对识别出的用户意图按其出现频率进行排序。当用户选择某个查询意图下的某一比较对象时，“Comparison Information Retrieval”模块将通过检索收集的 Yahoo! Answers 问题库，返回被选择的比较对象与用户查询的相关信息。

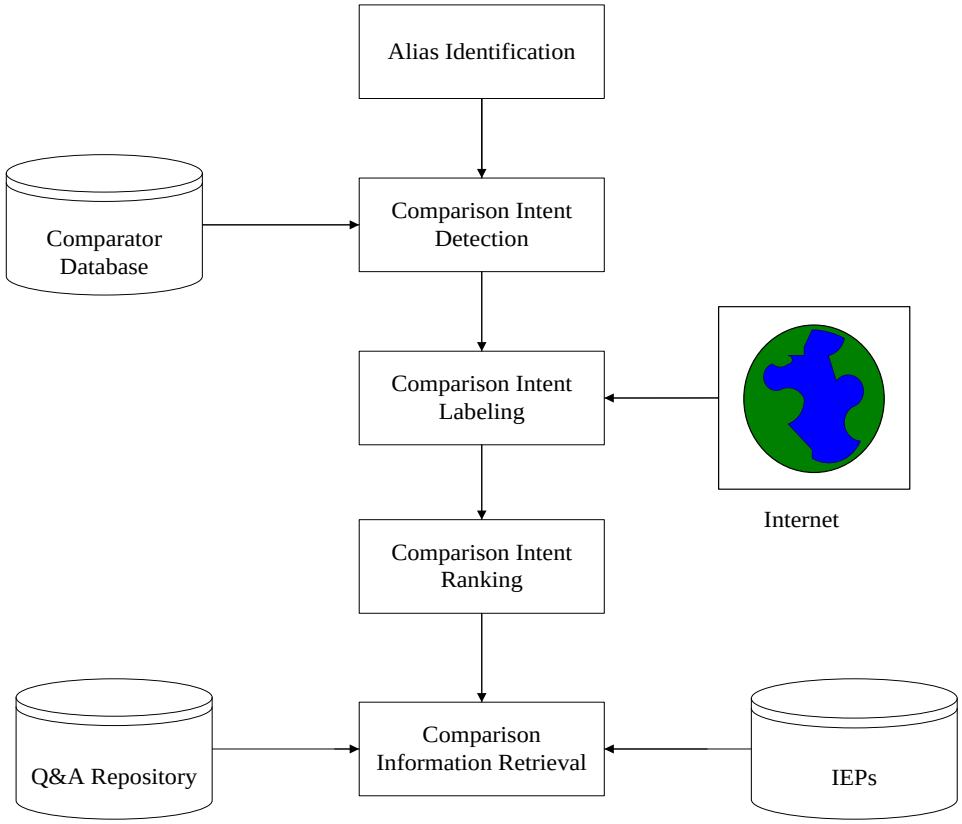

图 4.1 基于比较关系的用户比较意图识别系统框架

4.3.1 别名识别

由于 Yahoo! Answers 的问题都是由网络用户生成，存在多样性和不规范性的特点，存在大量缩写、略写、甚至拼写错误的情况，导致挖掘出的用户查询请求的同一个比较对象以多种形式存在，例如“Hilary”,“Hillary”，“Clinton Hilary”及“Clinton Hillary”。虽然在开放域中，以上几个字符串可能代表的不是一个对象，但是在给定其比较对象是“Obama”的情况下，这几个字符串只代表总统侯选人之一“Clinton Hillary”。而这些同一对象的不同形式也造成了可比较对象集合的冗余。因此，给定一个用户查询，有必要先对查询的可比较对象进行别名识别。

本文认为，在与给定用户查询请求可比的情况下，相似的字符串是互为别名的。我们利用以下规则判断两个字符串 A 和 B 是否是相似字符串：

- 字符串 A 是字符串 B 的子串；
- 字符串 B 是字符串 A 的子串；
- 字符串 A 和字符串 B 的编辑距离足够小。本文采用 Levenshtein 距离[79]计算字符串 A 和字符串 B 的编辑距离。Levenshtein 距离被定义为将一个字符串转变为另一个字符串所需要的最少操作数目。这里的操作可以是插入、替换或删除。当两个字符串的编辑距离满足以下条件时，我们说字符串 A 和字符串 B 的编辑距离足够小。

$$\frac{D(A,B)}{min(Len(A),Len(B))} > \theta \qquad (4.1)$$

其中，$D(A,B)$是字符串 A 和字符串 B 的 Levenshtein 距离；$Len(*)$是字符串*的长度；θ是一个指定的阈值，本文中θ设为 0.8。

然而根据以上方法确定的别名的准确率并不高，尤其是前两种基于子串关系的别名识别。例如，“Clinton”是“Clinton Hilary”的子串，但是众所周知，Clinton”和“Clinton Hilary”并不是同一个对象。因此，本文添加以下的约束，过滤上述规则识别的“伪别名”。

- 当两个字符串之间存在比较关系时，两个字符串不能互为别名。

以上条件显而易见，当两个字符串之间存在比较关系时，两个字符串必然指代两个不同的对象。接下来的任务是根据识别出的别名关系将指代同一比较对象的别名聚类。表 4.1 给出了我们识别出的一组别名关系，

表 4.1 别名关系列表

(“Hilary”，“Clinton Hilary”)
(“Hilary”，“Hillary”)
(“Clinton Hilary”，“Clinton Hillary”)
(“Clinton Hillary”，“Hillary”)
(“Clinton”，“Clinton Hilary”)

针对以上关系，本文采用如图 4.2 所示的图聚类算法[80]。首先，算法将“Hilary”加入聚类当中；然后依次查找与其存在别名关系的字符串，若该字符串与当前聚类中超过百分比α（这里α设为 0.5）的字符串都存在别名关系，且不与聚类中任何字符串存在比较关系，则该字符串应当加入“Hilary”所在聚类。表 4.1 所示的别名关系下，“Clinton Hilary”、“Hillary”被依次加入聚类。对每一个新加入的字

符串重复上述操作，直至没有新的字符串可加入该聚类。因此，形成聚类{“Clinton Hillary”，“Hilary”，“Clinton Hilary”，“Hillary”}。而“Clinton”因与“Clinton Hilary”存在可比较关系而被拒绝。对尚未加入任何已有聚类的可比较对象重复以上操作，直至所有可比较对象均被加入某个别名聚类为止。

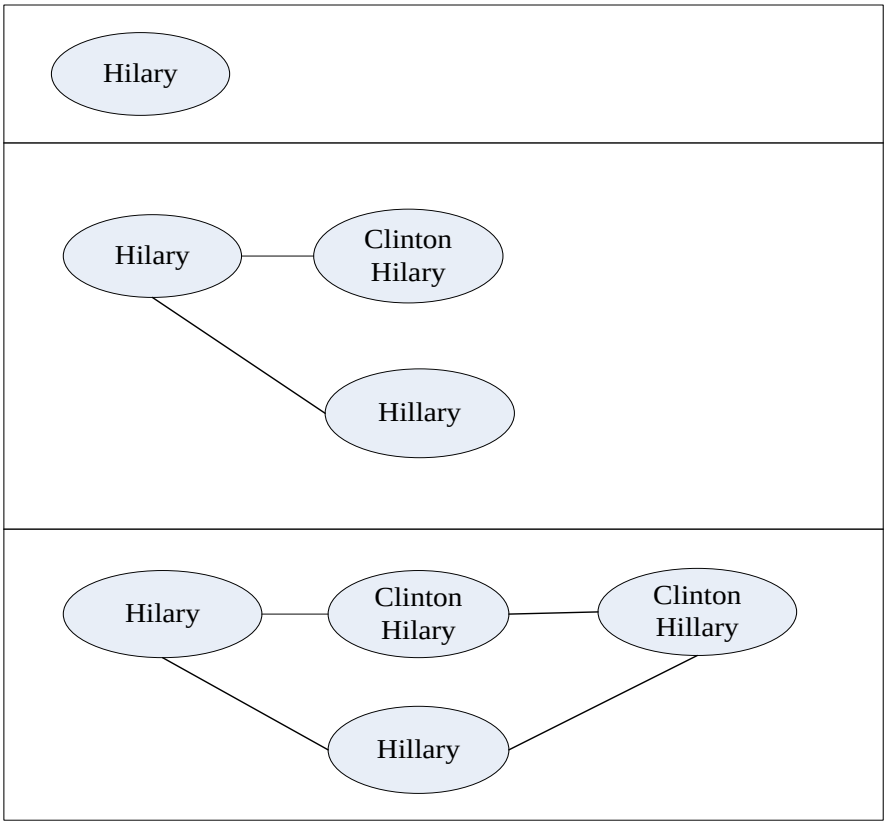

图 4.2 别名聚类

4.3.2 比较对象聚类

给定一个用户查询请求，本文把 4.3.1 节中识别的别名聚类看作一个可比较对象。本小节将介绍如何聚类这些可比较对象，识别用户可能的比较意图。首先，给定一个用户查询请求q及其可比较对象集合$C = \{c_1, c_2, \ldots\ldots, c_n\}$，建立比较关系图$G = (V, E)$。$V = \{v_0, v_1, \ldots\ldots, v_n\}$是由$q$和其可比较对象组成的节点集合。$E$是节点间边的集合，若节点$v_i$的任意一个别名与$v_j$的任意一个别名存在比较关系，则节点之间存在边$e_{ij}$，即$E = \{e_{ij} =< v_i, v_j > | v_i, v_j \in V \wedge Comparable(v_i, v_j)\}$。图 4.3 给出了查询“apple”的比较关系图实例。

从图中可以看出，当“apple”作为一种水果时，会被拿来与“orange”、“grape”、“pear”相比较；而当“apple”作为一个品牌或公司时，会被拿来与“Dell”、“hp”、

“Sony”、“Compaq”相比较。这两个比较意图中下的比较对象呈现比较明显的聚类，即不同比较意图下的可比较对象彼此之间的可比性小于同一意图下的可比较对象彼此之间的可比性。

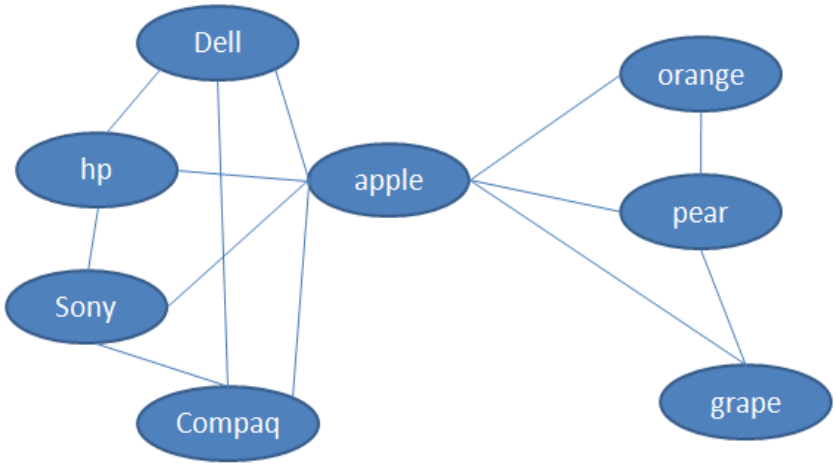

图 4.3 比较关系图

基于以上观察，本文提出利用基于连接度的图聚类的用户查询识别算法[80]。算法先将给定查询请求的可比较对象按上章所述的方法排序。对每个可比较对象，若该对象与某个已有聚类的连接度大于某阈值β，则将该对象加入该已有聚类；否则，为该对象创建一个新的聚类。算法的细节如算法 4.1 的伪码所示。当设定β为一个连接时，该算法退化为寻找比较关系图中的连通图。

算法 4.1 比较对象聚类

Input: $G =< V, E >$

1. **Foreach** comparator $c_i \in V - \{q\}$
2. **Foreach** cluster $I_j \in I$;
3. **If** $connectivity(c_i, I_j) > \beta$
4. Add c_i to I_j;
5. **Else** create new cluster $I_t = \{c_i\}$ and add I_t to I ;
6. **End Foreach**
7. **End Foreach**
8. **Return** $I = \{I_1\{c_{11}, c_{12}, \ldots\ldots, c_{1n_1}\}, \ldots\ldots, I_k\{c_{k1}, c_{k2}, \ldots\ldots, c_{kn_k}\}\}$

4.3.3 聚类的语义标记

直观地讲，每个对象有不同的语义标签，例如“apple”可以是水果，也可以是品牌。另外，通常两个可比较对象拥有共同的语义标签，且共同的语义标签是较为确定的。例如，“ipod touch”的可能的语义标签有“portable player”、“web-enabled

device”、“touch device”等，分别注重“ipod touch”的不同功能面或技术特点，但当其与“psp”相比较时，其拥有的共同的语义标签只有“portable player”。注意，虽然两个可比较对象通常拥有共同的语义标签，但是反过来并不成立，即并不是所有有同样的语义标签的对象都具有可比性。

基于以上观察，对每个用户可能的查询意图（即比较对象聚类）$I_i = \{c_{i1}, c_{i2}, \ldots\ldots, c_{in_i}\}$，本文采用形如“NP, such as q, c_{ij}”的模式抽取的可能标签。其中，q为用户查询请求，c_{ij}为在q的查询意图I_i下的一个可比较对象。首先，算法将形如“, such as q, c_{ij}”及“, such as c_{ij}, q”的查询分别输入搜索引擎，抓取前100个返回结果的相关文档片段（snippet）。注意，这里的查询“, such as q, c_{ij}”或“, such as c_{ij}, q”必须用引号，表示该词序列必须完整出现在搜索结果中。然后，利用对应的模式“NP, such as q, c_{ij}”或模式“NP, such as c_{ij}, q”，从收集的相关文档片段中抽取字符串“such as q, c_{ij}”或“such as c_{ij}, q”前面的名词或名词词组作为可比较对象对（c_{ij}, q）的可能的语义标签。可比较对象对（c_{ij}, q）的每个可能语义标签l_{ik}都被赋予一个被抽取出的频率值，作为该语义标签对（c_{ij}, q）的权重值，记为$w_{ik}(c_{ij}, q)$。用户的查询请求q在查询意图I_i下与每个可比较对象c_{ij}可能的共同标签的合集即为查询意图I_i的可能标签集。算法利用公式4.2选择其中出现最频繁的标签作为该用户查询意图的标签。

$$l_i = argmax \sum_j w_{ik}(c_{ij}, q) \tag{4.2}$$

4.3.4 比较意图排序

在该模块中，算法将用户的比较意图按照其出现的次数进行排序。用户比较意图出现的次数主要取决于其包含的可比较对象出现的次数，因此，用户的比较意图函数如下所示，

$$S(I_i) = \sum_i \sum_j N(Q_{c_{ij},q}) \tag{4.3}$$

其中，$N(Q_{c_{ij},q})$代表包含比较对象对（c_{ij}, q）的问题的个数。每个用户查询意图中的可比较对象仍旧按上章所述方法排序。

4.4 实验及系统介绍

4.4.1 实验设置及实验结果

数据集：本章实验从上章挖掘出的比较关系中随机选择 200 个比较对象作为用户查询请求。为保证其可比较对象拥有别名及不同的比较意图，以便评测别名识别算法及比较意图识别算法的性能，我们要求被选择的比较对象至少有 5 个可比较对象。对每个用户查询请求，标注者被要求识别其不同可比较对象的别名集合。在标注者标注的别名集合基础上，利用 4.3.2 所述算法对每个用户查询的可比较对象进行聚类，并利用 4.3.3 节所述方法给每个聚类赋予语义标签。测试集的标注者被要求标注每个识别出的用户比较意图是否正确，对正确比较意图，其中每个可比较对象的聚类结果是否准确。

评测指标：对于别名识别，本文采用别名关系识别的准确率、召回率、F1 值及聚类中常用的 Rand Index 指标[81]作为评测指标。对于查询意图识别，由于人们比较行为的复杂性，即使是人类专家也很难建立用户完备的查询意图识别评测标准。这里，本文只能针对识别出的用户比较意图，根据经验，评测其识别的准确性。

实验结果：系统别名关系识别的准确率为 86.3%，准确度为 94.03%，F1 值为 0.9, Rand Index 值达到 98.4%。本文分析了其中错误的原因：82.5%的未识别出的别名关系是由于错误拼写与正确拼写差距较大，例如，尽管人可以判断“torjon”是“trojan”的错误拼写，但是其编辑距离较大；17.5%的未识别出的别名关系是由于算法并未涉及缩写关系造成的，例如，“USA”，“the United States”及“America”，这一类问题可以在后续的算法完善过程中通过引入已有的缩写列表得到改善；另外，错误别名关系识别主要是由于子串关系识别的不准确性及挖掘的比较关系的局限性引起的，例如，“circle square”和“square”是两个不同的对象，但是它们之间存在子串关系，且两者之间并没有比较关系。

系统查询意图识别的准确性达到 92.7%，主要的错误来源于抽取的语义标签的不完整性。例如，从句子“So we can see the software is compatible with all generation of models such as iPod touch, iPod nano, iPod classic, iPad as well as iPhone”中抽取比较对象对(“ipod touch”，“ipod nano”)的语义标签，只能抽取到“models”。从该句子出现的上下文，我们可以猜测该“models”应该是指“ipod models”，然而，单独抽取出“models”作为语义标签时，其语义信息变得不甚完整。

系统识别正确的查询意图下，聚类比较对象的准确率约为 83.6%。主要是由于，给定一个用户查询请求，当其与某个可比较对象相比时，并不是只有一个比较目的。例如“Paris”作为一个城市时，与城市“London”是可比的，但是，当其作为名人时，同样与名人“London”是可比的。再比如，“iphone”与“ipod touch”作为可移动设备(“portable device”)或触摸屏设备（“touch device”）都是可比的。这使得有时两个比较意图下的比较对象并不是完全没有交叉的。

这里，需要解释的是，由于用户有时倾向于将同一类产品同某一个常用的大家

所熟悉的比较标准相比较，因此比较对象之间的比较关系图多呈现为以比较标准为中心的星状分布，其它比较对象之间的关联度不如我们所期待的大。再加上挖掘的比较关系的局限性，导致系统识别出的用户意图冗余性较高，即系统识别出很多小的聚类。在未来的工作中，一方面可以期待挖掘更多更丰富的比较关系；另一方面，利用聚类标签之间的相似性合并某些标签也是一个可能的解决方案。

4.4.2 系统介绍

本文实现了如下图所示的针对比较行为的比较搜索系统。当用户输入查询请求，网页左边栏将列出相应的查询推荐。其中上半区为搜索引擎Bing(www.bing.com)的相关搜索推荐，下半区为本文提出的基于比较关系的查询推荐。系统根据 4.3 节所给出的比较意图识别及排序方法，列出用户可能的比较意图。当点击其中一个比较意图，系统将给出用户查询请求在该比较意图下所有相关的可比较对象。选择其中一个可比较对象，网页的正文区将给出搜索结果。这里，系统列出两种搜索结果。网页正文区的上半部分是将用户查询请求和用户选择的比较对象用“VS”连接作为新的查询，用该新的查询在搜索引擎 Bing 中搜索获得的结果。网页正文的下半区，则是根据用户查询请求和用户选择的比较对象，利用在可比较关系挖掘过程中产生的模式，从我们收集的 Yahoo！Answers 问题库中获得的相关信息。

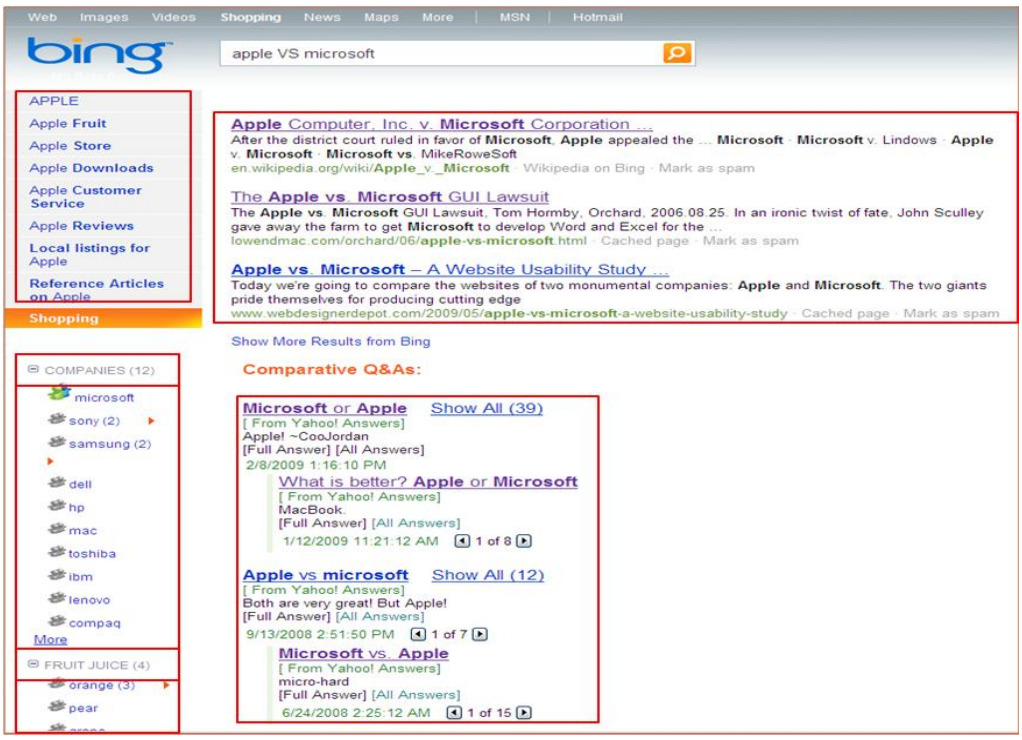

图 4.4 针对用户比较行为的比较搜索系统

图 4.4 给出的是用户查询请求“apple”的应用实例。从中我们可以看出，Bing 的相关查询推荐给出了“Apple Store”，“Apple Downloads”，“Apple Customer Service”等候选项，其中并不包含与用户查询请求可比较的对象推荐。可见，Bing 的相关查询推荐并不能支持用户的比较行为。

在比较对象推荐部分，系统识别出了常用的用户比较意图。图中列出了两种比较意图，“company”和“fruit juice”：前者表示用户意在从公司产品、发展前景、公司文化、口碑等各方面将“Apple”与其它公司相比较；后者表示用户意在比较从口感、营养价值等方面将“apple”与其它口味的果汁比较。事实上，系统还识别出一种用户的比较意图“body shape”，通常这种有这种比较意图的用户多是女性，意在更加了解自己的特点，选择恰当的健身、美体方式或选择恰当的时装。相比之下，这种比较意图较少，也比较难被发现。由于空间所限，图上并没有显示出来。在每个比较意图下，系统列出了排序后的可比较对象。例如，在“company”的比较意图下，“apple”的可比较对象包括“microsoft”，“sony”，“samsuang”等国际著名科技公司。“sony”后缀的数字“（2）”表示，系统从挖掘出的比较对象对中识别出“sony”的两个别名。

从搜索结果来看，虽然搜索引擎 Bing 的搜索结果包括“Apple VS Microsoft”，但是，前两个结果是关于两个公司间的诉讼案，后一个结果是关于两个公司网站可用性的比较。与比较两个公司的意图并不相符。这一结果说明，通过基于关键字的查询构造比较意图并不是件显而易见的事。而利用模式从我们收集的 Yahoo! Answers 问题库中搜索到的信息则大多是用户需要的比较信息。

4.5 本章小节

基于关键字查询的搜索引擎中，用户往往只能使用有限数量的词汇来抽象和概括他们的需求。在用户将其需求抽象成有限的查询关键字的过程中，部分有用信息被丢失，从而导致用户查询意图不慎清晰。以用户的比较行为为例，用户的查询请求对象会有多个侧面，要了解其不同的侧面需要跟不同的对象比较。例如，当用户查询“Paris”时，可能是要比较旅游地点，也可能是想比较各国首都，还有可能“Paris”是名人而不是地点。直观地讲，用户的同一种比较意图下的可比对象之间比较的几率也比较大，而不同比较意图下的可比对象之间比较几率较小。基于这一观察，针对每个用户查询请求对象，本文提出一种基于其可比较对象之间的比较关系的比较意图识别算法。算法识别用户查询的不同比较意图，从而通过交互帮助用户进一步确认自己的查询意图。每个用户查询意图由一组用户查询请求的可比较对象表示，且通过信息抽取的方法被赋予一个语义标签。实验证明，本章的用户查询意图识别方法的准确率达到 92.7%。另外，本文还针对用户的比较

行为建立了一个比较搜索系统，该系统在识别用户查询请求的不同比较意图的同时还提供了不同的比较意图下，不同可比较对象与用户查询请求的比较信息。

第五章 面向开放域的无监督的查询理解

5.1 引言

近年来，用户的查询请求除了单个实体或对象（如“Obama”）以外，还存在大量由多个搜索关键字组成的复杂查询，例如“harry potter showtime beijing”。这些查询多是面向任务的，并要求得到精确的答案（如“0:00, Nov.19th, 2010”）。在目前的搜索引擎中，用户通常要进行如下操作：首先检索相关网页或在线数据库；然后逐一阅读相关网页寻找所需信息或向相关在线数据库提交数据库检索请求，寻找所需信息。为了简化这一繁琐过程，研究者们提出了结构化检索。众所周知，一些商业搜索引擎（例如 Microsoft Bing, Google）已经爬取并索引了许多可用于结构化检索的结构化数据。这些数据与纯文本数据相比，意义较为清晰。结构化检索先对基于关键字的查询请求进行语义理解，然后将其转化为适合各个在线数据库的检索请求，最终从在线数据库中检索出结果并直接将结果返回给搜索用户。

对基于关键字的用户查询请求进行语义理解是结构化检索中的一个关键步骤，简称**查询理解**。它实际上是一个识别用户查询请求中的查询关键字并给其赋予一个语义标签从而实现语义消歧的过程。例如，为了从电影信息数据库中给查询请求“harry potter showtime in beijing”找到答案，我们首先需要识别查询关键字“harry potter”，“showtime”和“beijing”，这些查询关键字可能是数据库中的实体或属性；然后需要分辨每个查询关键字的含义并给其赋予一个语义标签，“harry potter”被标记为“movie name”，“beijing”被标记为“city”，而“showtime”是电影的一个属性。

查询理解不是一个简单的工作。例如，词组“harry potter”，在不同的查询请求中，它可以表示一个电影的名字，一本书的名字，或者电影中的一个角色。在简单查询“harry potter”时，任何一种含义都是正确的；但是，在以上包含多个查询关键字的复杂查询请求中，每个查询关键字的意义都是比较确定的，很容易判断其中“harry potter”是指一部电影。问题是，尽管在包含多个查询关键字的复杂查询请求中，每个查询关键字的含义比较确定，但是可用于帮助计算机确定其含义的上下文信息却是极少的，这给查询理解带来的很大困难。

许多之前的研究工作[37-41]将查询理解看作一种序列标注任务。虽然在他们的实验中，这些方法在查询理解的准确率方面表现出良好的性能；但是，由于它们只作用于指定域，依赖一个手动创建的领域标签集，而在开放域的环境下手动创建这些资源是不太可行的，因此这些方法不能被应用于面向开放域的结构化搜索。

本文主要关注如何在开放域的环境下理解用户的复杂查询请求：首先，利用现有技术自动创建一个语义词典；然后，尝试使用自动创建的语义词典进行开放域查询理解，即查询关键词识别与消歧。在本文的问题设定中，我们致力于解决以下两个问题：

1) 自动创建的语义词典无论在语义标签还是词或词组实例中都包含很多噪音，这些噪音将严重降低查询理解的性能。
2) 在开放域环境中，大量的语义标签是必需的。这使得以前用于处理有限数目的语义标签的基于序列标注技术的查询理解方法不再适用。

为了解决以上问题，本文提出一种基于查询模板的方法以识别并消歧查询请求中的查询关键字。该方法受词义消歧技术的启发。词义消歧与查询理解问题相似，需要从大量可能的语义标签中为指定的词选择恰当的语义标签进行消歧。本文先利用一个无监督的互增强的算法挖掘查询模式；然后基于挖掘到的查询模式和语义词典进行查询关键字识别及消歧。据我们所知，本文的研究是第一个利用自动创建的语义词典进行查询理解的尝试。据我们所知，本文的研究是第一个利用自动建立的语义词典进行面向开放域的查询理解的尝试。

本章的余下部分按以下方式组织：5.2 节讨论了已有相关工作；5.3 节介绍了大规模语义词典的自动创建；5.4 节描述了非监督的互增强的查询模式挖掘算法以及如何利用挖掘到的查询模式实现用户查询理解；5.5 节介绍了实验的设置，并针对实验结果进行了讨论；最后在 5.6 节对本章做出小节。

5.2 相关工作

从任务的角度看，本文的研究与很多试图进行结构化查询的研究相关。这些研究工作通常将查询理解看作序列标注的问题，面向特定领域，根据有限的语义标签及自定义的词表，将用户查询请求标注为语义标签的序列[37-41]。例如，Li 等人[37]针对商业领域的查询，提出了根据给定的语义标签利用查询点击及数据库信息获取词表的方法，并且提出一种基于条件随机场（CRF）的半监督模型进行语义结构理解。Agarwal 等人[39]则提出一种基于模板的非监督模型。他们利用词表根据查询日志中的查询记录生成可能模板，然后利用查询点击、用户查询、查询模板三者之间的关联，通过对模板的评测选出可靠性最强的模板。

从技术的角度看，本文的工作与语义消歧技术相关，目标是分割用户查询，识别其中的查询关键字并为每个查询关键字选择最恰当的语义标签。文本中的语义消歧可以分成两个流派。其一是根据文本的上下文或者篇章信息[82-84]，这类算法基

于两个假设：一个词出现在不同篇章中的相同上下文搭配中时，该词的含义保持一致；一个词出现在同一篇章的不同上下文搭配中时，该词的含义保持一致。基于这两个假设，可以挖掘某个词以某种词义出现时不同的上下文搭配。由于查询理解不存在篇章关系，这种方法并不适用于查询理解中的词义消歧。另外，对查询请求中可能出现的每个查询关键字学习其不同语义的上下文搭配并不适用于开放域的情况。文本中的另一种语义消歧方法是借助一个预定义的词典，利用词典中的词义定义与词语上下文信息的相似度度量选择词义[84,85]。但是创建一个良定义的包含用户查询中所有可能出现的关键词的词典是不可能的。

另外，在查询扩展的研究中存在大量关于用户查询词的词义消歧的研究[66,67]。但是这些研究的主要目的是为查询词找到恰当的同义词或词义相关词用于检索，从而提高检索的有效性，与本文通过查询关键字的词义消歧为其找到合适的语义标签并不相同。

5.3 语义词典的建立

在本文中，语义词典中的每条记录由一个词及其一组可能的语义标签组成。每个标签代表一个语义类，在每条记录中其被赋予一个相对于该条记录中的词的权重。语义词典的自动创建业已受到广泛的关注[86-90]。本文利用一个已有的算法从大规模的开放域语料集中抽取上下位关系（is-a），上位词可作为下位词的语义标签。具体来说，本文利用一组人工设计的表层文本模式（例如，“NP such as NP”，“NP like NP”，“NP is a NP”）来挖掘上下位关系。每个上下位关系被赋予一个频率值，这个频率值暗示了抽取该上下位关系时支持句子的个数[91]，可用作上位词作为下位词语义标签时的权重。

在这里，本文也同时计算了词汇间的分布相似度。本文采用大小为 2 的文本窗口作为上下文信息，通过计算词汇与其上下文词汇之间的互信息作为特征值建立词汇的特征向量，最终确定词与词之间的 consine 相似度。文献[92]指出，这种计算词汇相似度的方法优于其它方法。

5.4 基于查询模式的词义消歧

之前相关的研究已表明，词汇出现的上下文信息对其词义的消歧有一定的提示作用[66-69]。本文的算法采用查询模式的形式表示查询中词汇的上下文信息。

本文的查询模式由两部分组成：模板和语义分布。每个查询模式中的模板定义为一个序列$S(s_1 s_2 \ldots\ldots s_n)$，其中$s_i$可以是一个词，也可以是一个表示槽的“$[KEYWORD]$”标记。每个模板中只能有一个槽。每个模板被赋予一个语义的概率

分布。这个概率分布给出的是，与该查询模板匹配的查询中，与槽的位置对应的词或词组的后验语义概率分布。表 5.1 给出一个查询模式的例子。在该例子中，当某个查询（例如“Paris best performance”）与该查询模式中的模板“[*KEYWORD*] best performance”相匹配时，槽“[*KEYWORD*]”所对应的词“Paris”的语义为“artist”（演员）的概率为 0.72，“celebrity”（名人）的概率为 0.16，以此类推。尽管语义标签“artist”、“celebrity”、“performer”之间存在语义重叠，它们的区别也是明显的，例如，“Obama”是“celebrity”却不是“artist”。

表 5.1 查询模式实例

模板	语义概率分布
[*KEYWORD*] best performance	Artist:0.72;celebrity:0.16; performer:0.04;NA:0.04; actress:0.04

当一个查询与一个查询模板匹配，通过考虑与槽“$[KEYWORD]$”位置相对应的词或词组的先验语义概率分布及其在查询模式中的后验语义概率分布，可以确定该词或词组的语义。当一个查询可以匹配多个查询模式时，通过查询模式及其识别出的词或词组在数据集中的频率，可以确定一个优化的查询模式集合。以下的章节将描述如何获取查询模式及如何利用查询模式进行查询理解。

5.4.1 非监督的查询模式挖掘

本文选用一个大规模的查询日志作为训练集，并通过将每个查询记录中在语义词典中存在的词或词组替换成槽“$[KEYWORD]$”生成查询模式的模板。以查询记录“Harry Potter showtime in Beijing”为例，在自动创建的语义词典中包含记录“Harry Potter”,“showtime”和“Beijing”，因此，算法生成表 5.2 所示的三个模板：

表 5.2 模板示例

[KEYWORD] showtime in Beijing
Harry Potter [KEYWORD] in Beijing
Harry Potter showtime in [KEYWORD]

接下来的关键问题是如何估计每个查询模式中的语义分布。本文提出一种非监督的互增强算法，该算法通过不断迭代语义概率分布估计和词义消歧的过程，最终获得较为准确的查询模式语义概率分布。算法基于以下两个假设：

1) 当一个查询请求与某个查询模式的模板完全匹配时，槽“[*KEYWORD*]”对应

的词或词组更有可能被消歧为与模板相关度较高的语义

2) 当一个查询请求与某个查询模式的模板完全匹配时，槽“[*KEYWORD*]”对应的词或词组都倾向于某个语义时，在该查询模式的语义分布中，此语义拥有较高的概率。

算法的整个过程始于利用语义词典中词或词组的语义先验概率分布对查询记录进行词义消歧。然后，算法将词义消歧的结果作为训练集进行查询模式的语义概率估计。接着，词义消歧通过查询模式的语义概率分布进一步精化。后续两个步骤反复迭代，直至查询模式的语义概率分布趋于稳定。算法具体过程如算法 5.1 所示。

算法 5.1 非监督查询模板挖掘算法

Input: $Q, D = \{< t, l, Pr(l|t) > | t \in T, l \in L\},\ P$

Output: $\{Pr(l|p) | l \in L, p \in P >\}$

1. **Foreach** pattern $p \in P$
2. **Foreach** term t appearing with p
3. $\hat{t} \leftarrow Enrich Term(p, t, Q)$
4. **Foreach** label $l \in L$
5. $Pr(l|\hat{t}) \leftarrow EstimateEnrichedTermLabelPro c(D, \hat{t})$
6. **End foreach**
7. **End foreach**
8. **End foreach**
9. **Repeat**
10. **Foreach** pattern $p \in P$ and $l \in L$
11. $Pr(l|p) \leftarrow EstimatePatternLabelsPro(D)$
12. **End foreach**
13. **For each** query $q \in Q$
14. **Foreach** term $\hat{t}$ in q
15. $p \leftarrow GeneratePattern(q, \hat{t})$
16. $\hat{l} \leftarrow TermDisambiguation(p, \{Pr(l|p) | l \in L\}, D)$
17. **End foreach**
18. **End foreach**
19. $Pr(l|\hat{t}) \leftarrow ReestimateEnrichedTermSensesPro(Q)$
20. **End repeat**
21. **return** $\{Pr(l|p) | l \in L, p \in P >\}$

从上述过程可以看出，查询模式的挖掘有三个关键的步骤：(1)词义的初始化；(2)查询模式的语义概率分布估计；(3)基于查询模式的词义消歧。后续的章节将详细解释这些步骤。

5.4.1.1 词义的初始化

直观地想，对查询中出现的每个词，我们可以设置其先验概率最高的词义为初始词义。但是，这种不考虑词出现的语境的初始化方法的准确率不能保证。例如，当“apple ”出现在形如“[*KEYWORD*] juice”的查询中时，“apple”的语义显然为“fruit”，但是，在“apple”的先验语义概率分布中，“apple”最有可能的语义为“brand”。

因此，当按词的先验语义概率分布对其消歧时必然导致消歧错误，而这种错误在后续的迭代过程中也将导致查询模式语义概率分布的错误估计。在自动创建的语义词典噪音严重的情况下，这种错误将更加明显。

虽然我们不期待完全正确的词义初始化结果，但无疑好的词义初始化结果将引导算法得到更为准确的查询模式的语义概率分布。因此，尽可能提高词义初始化的准确度是必要的。幸运的是，从查询日志中可以收集到很多曾出现在同一个上下文中的词，例如，“apple”和“orange”都曾出现在“[*KEYWORD*] juice”中。基于出现在同样的上下文语境中的词通常有同样的词义的这一假设，本文尝试利用与待消歧的词出现在同一上下文语境的词更好地对其进行词义初始化。由于“apple”和“orange”都出现在形如“[*KEYWORD*] juice”的上下文语境中，这两个词在该语境中更倾向于被解释为它们共有的词义“fruit”，而不是“brand”或“color”。

具体来说，给定一个词或词组t和查询模板p，算法首先发现一个与t同样出现在上下文语境p中且与t在分布上相似度最大的词t'，然后用t'来限定此处t的语义只能是t与t'所共有的语义之一。本文将此处的t称为被t'限定的t，表示为$\hat{t}$。$\hat{t}$的语义的初始语义概率分布用以下方式计算：

$$Pr(l \mid \hat{t})^{(0)} = \frac{Pr(l|t)+Pr(l \mid t')}{2} \tag{5.1}$$

$$Pr(l|t) = \frac{freq(l,t)}{\sum_{l_i \in T} freq(l_i,t)} \tag{5.2}$$

这里，$Pr(l|t)$是语义标签l对t的先验概率；$freq(l,t)$是在自动建立的语义词典时上下位关系(l,t) 的概率。

词义“NA”表示在自动建立语义词典时遗漏的语义。“NA”的概率用如下的方式估计：

$$Pr("NA" \mid \hat{t})^{(0)} = 1 - \sum_{l_i \in L(t) \cap L(t')} Pr(l_i \mid \hat{t})^{(0)} \tag{5.3}$$

出现在查询上下文语境p中的t的词义被初始化为$\hat{t}$的语义概率分布中概率最高的词义。如果大量出现在某个查询上下文语境中的词被消歧为“NA”，则代表该上下文不能暗示任何已知语义标签。也就是说，出现在该上下文中的词可能是任何语义，因此没有必要对出现在该上下文中的词进行消歧。例如，如果“apple”出现在上下文“what is $[KEYWORD]$”中，“brand”和“fruit”都有可能是“apple”的语义。

5.4.1.2 词义概率重估

基于词义初始化或词义消歧的结果，被t'限定的t的词义概率分布被以如下方法重新估计。

$$Pr(l|\hat{t})^{(n)} = \frac{Count(l,\hat{t})}{Count(\hat{t})} \quad (5.4)$$

其中，$Count(l,\hat{t})$是指$\hat{t}$被消歧为l的次数；$Pr(l|\hat{t})^{(n)}$是在第n次迭代后所估计的$\hat{t}$的词义l的先验概率。基于第二个假设，查询模式的语义概率分布可以以如下的方法被重新估计，

$$Pr(l|p)^{(n)} = \sum_{t_i \in T} Pr(l|\hat{t_i})^{(n-1)} Pr(t_i|p) \quad (5.5)$$

其中，$Pr(l|p)^{(n)}$是指第n次迭代后，查询模式p的语义分布中语义l的后验概率分布。$Pr\ (t_i|p)$的计算方法如下，

$$Pr(t|p) = \frac{freq(t)}{freq(p)} \quad (5.6)$$

5.4.1.3 词义重消歧

正如以上所讨论的，“限定的词$\hat{t}$”的概念的引入对于查询中的词义消歧很有帮助。这一概念也被应用于词义的重新消歧。对查询日志中的每个查询记录q，算法识别该查询记录中的每个查询关键词t及其上下文查询模板p，并找到合适的t'将其转变成“限定的t”，即$\hat{t}$。基于第一个假设，算法通过使用以下方法从$\hat{t}$可能的语义中选择一个适当的语义，

$$\hat{l}^{(n)} = argmax(\frac{Pr(l_i|p)^{(n)}}{Pr(l_i)^{(n)}}) \quad (5.7)$$

$Pr(l_i)^{(n)}$是第n次迭代中语义l_i出现的概率。这个概率按以下的方法估计，

$$Pr(l)^{(n)} = \sum_{t_i \in T} Pr(l|\hat{t_i})^{(n)} Pr(\hat{t_i}) \quad (5.8)$$

5.4.2 查询记录中的查询词词义消歧

本文设计了一个简单的基于查询模板的查询理解模型，用以评估挖掘到的查询模式在查询关键字识别和消岐上的性能。

5.4.2.1 查询关键词识别

给定一个查询记录q，算法把任意的连续的词的序列作为候选的查询关键词。查询关键词识别的目标是从候选查询关键词集合中找到一个最有可能的查询关键字集合，该集合中的查询关键词互不冲突。这里所说的相互冲突是指两个查询关键词在查询记录中存在重叠的部分。例如，在查询“Harry Potter showtime in Beijing”，候选查询关键词“in Beijing”和“Beijing”就相互冲突。本文采取贪婪策略逐一从候选查询关键词集合中选择查询关键词。

直观来讲，如果给定的查询记录中候选的查询关键词及其上下文都在查询日志中频繁出现，则该候选的查询关键词很有可能是一个正确的查询词。基于这一点，本文借助以下公式，采用贪婪策略从给定的查询记录中逐个识别查询关键词，

$$S(t, p_{t,q}) = freq(t) * freq(p_{t,q}) \quad (5.9)$$

这里，t代表一个候选的查询词，$p_{t,q}$代表t在给定的查询记录q中的上下文查询模式。贪婪策略的具体算法如下所示。

算法 5.2 查询关键字选择算法

Input: $T = \{t_1, t_2, \ldots\ldots, t_n\}$

Output: $T' = \{t_1', t_2', \ldots\ldots, t_k'\}$

```
While T ≠ ∅
    t_i' = argmaxS(t, p_{t,q})
    T = T − {t_i'}
    T' = T' + {t_i'}
  For each candidate term t_i ∈ T
    If IsConflict(t_i, t_i')
      T = T − {t_i}
    End If
  End for
End while
return {Pr(l|p) | l ∈ L, p ∈ P >}
```

5.4.2.2 词义消歧

给定一个查询记录，其中每个查询关键词t的词义主要取决于以下两个因素：

- 查询关键词本身的先验语义概率分布。例如，对查询记录“windows breakdown”，“windows”对应的上下文查询模板是“[*KEYWORD*] breakdown”，该模板本身并不倾向于“窗户”或“操作系统”的语义，但是，“windows”本身更倾向于一种“操作系统”。
- 与查询关键词对应的上下文查询模式所决定的后验语义概率分布。例如，“safari”可以是一种浏览器或一种活动，但是在查询“safari release”中，“safari”

被确定为一种浏览器。

综上，本文采用以下公式进行查询关键词的词义消歧，

$$\hat{l} = argmaxPr(l|p,t) = argmaxPr(l|p)\,Pr(l|t) \qquad (5.10)$$

5.5 实验

5.5.1 数据

5.5.1.1 语义词典

本文从一个包含 500M 网页的数据集合中自动创建语义词典。建立的语义词典包含 150,000 个语义标签及 876,542 个词条。

5.5.1.2 查询日志

本文所使用的查询日志是来自一个商业搜索引擎的真实的查询日志。它包含 100M 条查询记录。

5.5.1.3 测试数据

本文从查询日志中随机挑选了 400 条查询记录。这 400 条查询记录与训练数据集中的查询记录没有重叠。标注者被要求识别查询记录中的查询关键词并基于创建的语义词典选择适当的语义标签。对在词典中没有合适的语义标签的查询关键词，其语义被赋为“NA”。每个查询记录先被两位标注者标注。对其中不一致的查询关键词识别或语义标签选择，本文邀请第三位标注者在两种结果中选择其一。总体来说，本文创建的测试集中包含 1025 个查询词，其中 877 个查询关键词可以在自动创建的语义词典中找到相应词条，且该词条的可能的语义标签中包含适合该查询关键词所出现的查询上下文语境的语义标签。

5.5.1.4 基准方法

给定一个查询记录，本文的基准方法采用与 3.1.2.1 小节相似的贪婪策略识别查询关键词。不同之处在于：在每次选择中，基准算法不考虑候选的查询关键词所在的上下文，只是选择语义词典中频率最高的候选查询关键词。同样，在为选择的查询关键词确定其语义标签时，基准算法选择其在语义词典中权重最大的语义标签。

以上方法是评测语义消歧技术时常用的基准方法。候选的查询关键词在自动创

建的语义词典中的频率按以下方法计算，

$$freq(t) = \sum_{l_i \in L} freq(l_i, t) \qquad (5.11)$$

这里，L是在自动创建的语义词典中t可能的语义标签的集合；$freq(l_i, t)$是在创建语义词典的过程中l_i被抽取为t的语义标签的频率。

5.5.2 实验结果

表 5.3 基于查询模板的查询理解方法与基准方法的性能比较

	基准算法			基于查询模板的方法		
	识别	消歧	All	识别	消歧	All
召回率	0.791	0.542	0.338	0.236	0.738*	0.172
准确率	0.668	0.542	0.286	0.948*	0.738*	0.690*
F1 值	0.724	0.542	0.309	0.378	0.738*	0.275

表 5.3 给出了本文的实验结果。在该表中，列“识别”是给定一组查询记录识别其中的查询关键词的性能；列“消歧”是给定一组查询记录中识别正确的查询关键词，为其选择正确的词义的性能。列“All”是给定一组查询记录识别其中的查询关键词并对识别出的查询关键词进行词义消歧的性能。

从准确度来讲，本文提出的算法远远优于基准算法。查询词关键词识别的准确率优于基准算法 1.41 倍，词义消歧的准确率优于基准方法 1.36 倍，而在整体的准确率优于基准算法 2.41 倍。但从召回率的角度来讲，基准算法更优。这是由于在基准算法中，如果候选的查询关键词是语义词典中的词条，查询关键词识别必然返回结果且返回一个为其选择的词义。但是在本文提出的方法中，由于实际数据的多样性，挖掘到的上下文模式的覆盖率必然有局限，这导致了实验结果的召回率低。但无论如何，从以上的实验中我们可以看到查询模式在提高查询理解准确率方面的潜力。而如何提高召回率是我们下一步将要研究的课题。

5.5.3 应用实例

表 5.4 显示了一些面向开放域的查询理解的有趣的结果。这些实例涉及很多领域，包括旅游、电影、食物、产品等。如表中所示，本文提出的方法可以根据词的上下文成功地识别查询关键字并对其消歧。例如，当“safari”与“plug-in”和“download”一起出现时，更有可能是一种著名的浏览器（“software”），但是，当“safari”和“south Africa”一起出现时，则更有可能是一种活动(“activity”)。同样的，在查询记录“apple and orange”中，“apple”和“orange”歧异的，但是当

它们同时出现时，它们都被消歧为水果（“apple”）。另外，基于查询模板的查询理解算法一个很大的优势在于，它能够区分在词汇上相似的查询中同一个词的不同含义。例如，查询记录“Paris Hilton in Paris”和“Hilton Paris in Paris”是词汇相似的，但是“Hilton”在两个查询记录中分别是指名人(“celebrity”)和连锁酒店（“chain”）。另外，本文提出的算法能够区分某些在使用上差别很小的语义。例如，品牌（“brand”）和公司(“company”)在很多情况下都是等价的。但是，在查询“apple stock price”中，只有公司（“company”）是恰当的语义标签。

表 5.4 查询理解实例

查询	查询关键字	语义标签
Paris Hilton in Paris	Paris Hilton	celebrity
Hilton Hotel in Paris	Hilton	chain
Harry Potter Cheats	Harry Potter	game
Download Harry Potter movie	Harry Potter	movie
Apple stock price	apple	company
Apple device	apple	brand
Apple and orange	apple	fruit
Apple and orange	orange	fruit
Orange color consume	orange	color
Safari in south Africa	Safari	activity
Safari plug-in download	Safari	software

5.6 本章小节

近年来，用户的查询请求除了单个实体或对象（如“Obama”）以外，还存在大量由多个搜索关键词组成的复杂查询请求，例如“harry potter showtime beijing”。这些查询请求多是面向任务的，并要求得到精确的答案（如“0:00, Nov.19th, 2010”）。针对这一应用需求，研究者们提出了结构化搜索。结构化搜索一个重要步骤即是对查询请求的语义理解，我们称之为查询理解。目前的查询理解主要是针对特定领域的，基于一个领域相关的手动创建的标签集。这类方法不适合开放域的查询理解。

据我们所知，本文的工作是第一个利用自动创建的语义词典进行面向开放域的查询理解的研究。本文基于一个大规模的语义词典设计了一个非监督的互增强的挖掘查询模式算法。为了解决语义词典的不完整性和噪音带来的问题，算法引入了“限定的词$\hat{t}$”及“NA”的概念。最终，本文通过一个简单的基于模板的查询理

解模型来评估挖掘的查询模式的有效性。实验结果表明，挖掘到的上下文模式对于词义消歧是准确的。在未来的工作中，我们将致力于提高挖掘的查询模式在查询理解中的召回率。

第六章 基于词的依赖层次的面向问题的答案摘要

6.1 引言

基于社区的问答服务，例如 Yahoo! Answers，百度知道等，已经成为引出用户生成内容（User Generated Content，简称 UGC）的一种很流行的方式。它通过用户生成的内容产生一种社区意识，并从而产生服务的凝聚度。中国互联网咨询公司“China IntelliConsulting”最近报导，“百度知道”现在已经成为百度访问量的主要来源，50%使用其搜索服务的用户会使用其社区问答服务。通常，在社区问答服务中，用户提出一个问题，其他社区用户可以回答该问题，提问者从所有的回复中选择最好的答案或者留待社区决定哪个答案最好。从 2005 年 12 月 Yahoo! Answers 发布以来，Yahoo! Answers 已经累积了 87M 问题。鉴于社区问答服务的受欢迎程度，如何重用社区问答服务中累积的问答数据也成为一个有趣的研究领域。

通常，社区问答服务中的问题数据由三部分组成：问题标题（问题的核心），问题描述（问题的背景信息），回复（包含来自不同回复者的候选答案）。最佳答案被从所有回复中标记出来。通过手动检查用户或社区选择的最佳答案（“Best Answers”），我们发现这些最佳答案的正确率很高，可以被重用。但是，本文主张在重用过程中，考虑答案正确率的同时答案的完备性也很重要。例如，用户提出一个问题：“Where to go in Beijing?”，有很多正确的答案，如“Summer Palace”，“Great Wall”，“Forbbiden City”等等。“Summer Palace”可能被提问的用户选为最佳答案。但是，对其他用户而言，这个最佳答案可能并不像它对提问的用户那么有用。其他用户或许因为已经去过“Summer Palace”而想得到其它有趣的答案。在这种情况下，提问的用户选出的最佳答案可能会被其他用户质疑。

本文尝试对一种特殊的问题，调研问题（survey questions），的答案进行摘要，从而获得可重用的较用户或社区选择的最佳答案更加完备的答案。调研问题是指请求回答问题的用户推荐针对某种需求的最佳选择的问题。显然，对调研问题而言，答案的完整性至关重要：一方面，不同的用户可能对不同的建议感兴趣；另一方面，某个答案被推荐的次数也反映了该答案值得被推荐的指数。调研问题不同于文本检索会议（Text REtrieval Conference，即 TREC）所提出的列表问题（list question）。列表问题同样要求系统提供不止一个答案，但是列表问题是面向事实的问题，而调研问题是面向意见的问题。例如，“what is the best brand name of Belgain chocolate?”是一个调研问题，答案中只需要列出曾经被推荐的 Belgain 巧克力品牌。

据我们所知，本文的研究首次指出在社区问答服务的知识重用中答案的完备性的重要性。同时，这也是第一个关注调研问题的研究。调研问题作为面向意见的问题中一种有趣的类型，对其而言，问题的完备性至关重要。除此之外，本文尝试使用面向问题的答案摘要方式产生完备的答案。本文提出一种有效的建立词与词之间语义依赖层次结构的方法，并利用建立的层次结构进行面向问题的答案摘要，从而基于社区服务中用户提供的所有已有答案生成一个完备简洁的答案。

本章的组织结构如下：6.2 节回顾了文本摘要的相关工作；6.3 节中详细介绍了如何建立词与词之间依赖关系的层次结构并利用建立的层次结构产生摘要；6.4 节描述了实验设置，展示并分析了实验结果；最后，6.5 节对本章做了小节。

6.2 相关工作

文本摘要是自动产生给定文本的压缩版本的过程。目前主要有两种类型的摘要方法：抽象式摘要（abstractive summarization）和抽取式摘要（extractive summarization）。在前一种类型中，文本中的信息被重写；但在后一种类型只是抽取原文本中的句子来组成摘要。由于抽象式摘要涉及到很多类如语义表达、推导、自然语言产生等挑战，关于抽象式摘要的研究很少。本文中主要关注抽取式摘要[93,94]。

抽取式摘要系统通常由以下几个部分组成：句子特征抽取和建模，句子打分及排序，句子选择。最近若干年的研究中，句子打分和排序方面有若干解决方案被讨论。You 等人[95]提出一种自动生成句子打分的训练语料的方法并利用回归模型组合句子的各种特征。Erkan 和 Redev[93]首先提出将 PageRank 的方法应用于文本摘要。在他们提出的方法中，文本被表示为无向图。图中每个节点代表一个句子，用一个向量模型表示。当节点所表示的句子之间的 cosine 相似度大于某个阈值时，对应节点之间存在一条边。然后，他们通过在生成的无向图上执行 PageRank 算法，对文本中的句子按其重要性进行排序，并截取给指定长度的文字作为文本的摘要。这种方法的性能优于传统的基于句子与篇章的相似性的基于中心的摘要算法（centroid-based summarization）[96]。Zha[97]提出将文本表示为一个词和句子组成的加权二部图并运用互增强理论同时抽取重要词和句子。Wei 等人[98]将该模型延伸成面向查询的增强链，并将其运用于面向查询的多文档摘要。

相对于句子打分及排序的研究，句子特征抽取及建模方面的研究并没有得到充分的重视。恰当的句子建模是好的句子打分和排序的基础。句子特征抽取和建模的目的是抓住句子中与摘要相关的最重要的信息。早期的对于抽取式摘要的研究主要是基于简单的启发式特征对句子建模，例如句子位置、基于句子中的词在整个文本中的频率的句子内容打分等。复杂一些的系统还会考虑句子之间的篇章

关系[99]。但是，识别句子之间的篇章关系的相关技术，例如指代消解和创建篇章结构，仍旧是自然语言处理领域的难点。Hovy 等人[100]提出一种基于基本元素（Basic Element，即 BE）的方法，该方法通过一个句法树抽取句子中词的依赖关系，然后利用词与词之间的依赖关系对句子进行建模。该方法均衡考虑了句法信息和词频信息。Daumé[101]采用一种基于贝叶斯的方法实现面向查询的文本摘要，该方法采用贝叶斯方法衡量句子中的词与查询的相关性，并用这种相关性作为句子特征对句子建模。另外，话题识别模型被用于获取文本中的话题特征，进而利用话题特征表示句子进行文本摘要。这种方法被证明对性能的提高大有裨益。Wang 等人[102]提出一种新的基于贝叶斯的句子话题识别模型，并将其用于文本摘要。该模型同时利用了词与文本及词与句子之间的关联。然而，尽管以上这些的文本摘要技术在其实验设定中都证明了其有效性，但是这些技术主要讨论单一话题，或单一话题的多个侧面。而本文所针对的资询问题的答案通常涵盖多个话题的多个侧面，每个侧面可能被某个话题所独有或在多个话题中共享。因此，与已有的文本摘要不同，话题与侧面的关系不再是一个树形结构，以上方法并不能直接应用于资询问题的答案摘要。

另外，大多数文本摘要的研究都是关注于通用的或者面向查询的抽取式摘要。通用的文本摘要的目的在于覆盖文本中的所有信息而不是针对用户特定的喜好。而面向查询的文本摘要只摘要用户查询所感兴趣的内容。值得注意的是，面向查询的文本摘要与面向问题的文本摘要存在本质的区别：查询可以作为所需要的信息的核心；而面向问题的文本摘要中，用户所需信息的核心是问题答案，而不是已知的问题。正如在之前的自动问答系统研究中讨论的一样，问题和答案之间存在一个词汇鸿沟（lexicon gap）。本文将关注于面向问题的文本摘要中句子的特征抽取及建模，致力于识别问题答案的核心信息，并利用问题答案中词与词之间依赖关系的层次结构识别答案中的多个话题和多个侧面。

O 6.3 基于词的依赖层次的文本摘要

这一节将详细描述我们的算法。如前所述，本文主要研究在面向问题的答案摘要中如何为句子抽取适当的特征并对其建模，并不关注句子的打分排序及选择过程，因此，本文采用了一种已有的经典的句子打分排序及选择算法以完成文本摘要任务。给定社区问答服务中一个问题及其所有回复，对其所有回复的具体摘要过程如下：

1) 将所有回复分割成句子的集合；
2) 抽取句子特征并将每个句子表示为特征向量；
3) 采用 Cosine 相似度[108]计算句子之间的相似度并用 K-means 方法[109]对句子进

行聚类；

4) 对聚类按其包含句子的多少从大到小排序；

5) 依次从每个聚类中选择最短的句子组成摘要。

Question: Where are San Francisco's best restaurants?
Answer1: Try the Carnelian Room - it has a great panoramic view of the city. I'vebeen there for New Year's Eve to see fireworks and it was very nice.
Answer2: If you want simple and classic, go to Gary Danko or Boulevard – thoughthey will break the bank. Cafe Jacqueline is known for their souffles, and as aromantic venue. Though it is a bit cramped, so go on a weeknight. If you likeIndian food, Maharani has an amazing Fantasy Room, which must be a great placeto propose – and the food is great there too.
Answer3: try aqua. very classy upscale popular place in sf. for views, try thecaprice in tiburon. its right on the water, and is amazing and very romantic, andnot too pricey
Answer4: The only thing that I can think of what best decribes what you arewanting is the Hornblower Yacht. You get the best views of the Bay Area, you canpropose on the bow of the ship underneath the Golden Gate Bridge, Sausalito.
Answer5: The above comments are sound. I would add the following: Gaylords...great Indian food (I proposed there in 1985 and got a "yes") The Van's on thePeninsula... GREAT views, reasonable prices... outside of the city.
Answer6: I really like Gary Danko. It's not cheap but has the best value for foodgiven the price that you're paying for.
Answer7: Micheal Mina at St. Francis Hotel... very romantic, high end, and excellentfood!..

图 6.1 资询问题实例

其中，第二步是本文研究的重点。传统的抽取式摘要主要是关注句子中的词在整个数据集中的频率(df)或其在给定文本中的相对频率（tfidf）。在面向查询的文本摘要中，文本中的词与查询的相关性也被考虑。但是在本文的问题中，这还远远不够。如图 6.1 所示的例子，回答问题的用户给出很多候选的餐厅。这些餐厅主要被从四方面特性描述：景观（view），食物(food)，浪漫氛围(romantic venue)及价格(price)。在提供的候选答案中，有些词，例如，“the”，“only”，它们与问题无关，本文称之为“通用词”；有些词，例如“restuarant”和“San Francisco”，它们与问题相关，但是与答案无关，本文称这种词为“问题背景词”；有些词，例如“view”,“food”,“romantic venue”及“price”，它们与答案密切相关，但是并没有指出答案，它们的出现依赖于推荐的餐厅的名字。如果一个候选答案没有提供餐厅的名字只是提到“good view”，这个答案是没有意义的。本文称这种词为“答案细节/侧面相关词汇”。最后，一个候选答案中餐厅的名字决定这个候选答案的核心内容，是必须出现在一个完备的答案摘要里的，本文称这种词为“答案话题相关词汇”。根据已有研究很容易区分出通用词和问题背景相关词汇。前者在所有的问题中拥有相似的概率分布。后者在给定问题的所有回复中有相似的概率分布。为了区分“答案细节/侧面相关词汇”和“答案话题相关词汇”，本文

根据词与词之间的依赖关系，建立一个词汇层次结构，如图 6.2 所示。

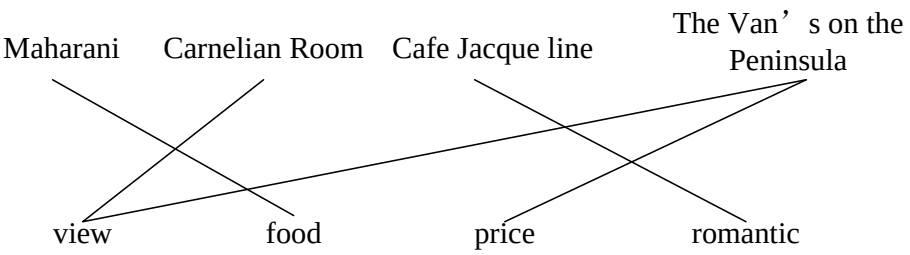

图 6.2 词的层次结构

在该层次结构中，答案话题相关词汇可以覆盖更多信息并被相关的答案细节/侧面相关词汇所描述，处于层次结构的上层。答案细节/侧面相关词汇描述了相关的答案话题相关词汇，且如果没有其相关的答案话题相关词汇是没有意义的。这些词汇处于层次结构的下层。在本文所述的问题中，我们不需要确切地识别一个答案细节/侧面相关词汇与那些答案话题相关词汇相关联，只需要区分处于不同层次的词即可。详细的过程请参看以下章节。

6.3.1 过滤通用词

如上讨论的，给定一个问题及其所有回复，其回复中的通用词在该问题（包括问题标题，问题描述、问题回复）中的分布与其在整个数据集中的概率分布相似。但是一个非通用词（或称问题相关词）出现在问题中的概率要大于出现在整个数据集中的概率。对问题回复中的每一个词，其问题相关性用以下方式测量[103]，

$$Q-Relevance(t_i,q_j)=TF(t_i,q_j)log\frac{N_q}{N_q(t_i)} \tag{6.1}$$

$TF(t_i,q_j)$是词t_i在问题q_j中出现的频率；N_q是数据集中的问题总数；$N_q(t_i)$是包含词t_i的问题数。当词t_i在问题q_j中的问题相关性小于某个阈值时，t_i被判定为通用词。

6.3.2 过滤问题背景词

给定一个问题及其回复，过滤了通用词之后，余下的词为问题相关词。这一小节中将进一步针对每个回复，从问题相关词中过滤问题背景词。根据以上的讨论，问题背景词在一个问题的各个回复中均匀分布。问题相关词中，除问题背景词以外的词汇本文称之为答案相关词。对问题q_j中的第 k 个回复a_{jk}，词t_i的答案相关性

可以用以下方法计算，

$$A-Relevance(t_i,a_{jk})=TF(t_i,a_{jk})log\frac{N_a(q_j)}{N_a(q_j,t_i)} \tag{6.2}$$

$TF(t_i,a_{jk})$是词t_i在问题q_j的第 k 个回复a_{jk}中的频率；$N_a(q_j)$是问题q_j的回复的个数；$N_a(q_j,t_i)$是问题q_j的回复中包含词t_i的回复的个数。当词t_i在问题q_j中是问题相关的，且在问题q_j的第 k 个回复a_{jk}中的答案相关性小于某个阈值时，t_i被判定为问题背景词。

6.3.3 答案话题相关词汇识别

这一小节针对给定问题的每一个回复，在过滤了通用词和问题背景词之后，区分剩余词汇中的答案话题相关词和答案细节/侧面相关词。如上讨论的，答案细节/侧面相关词汇的出现依赖于答案话题相关词汇的出现。这意味着答案话题相关词汇预测了答案细节/侧面相关词汇的出现。本文称一个词预测另外一些词出现的能力为“覆盖力”。一个词的覆盖力可以通过信息增益来获取。信息增益是一种用于衡量在已知某个特征的出现与否的情况下预测目标所需要的信息比特数的指标[104]。因此，一个词t_i预测另一个词t_j的出现的能力可以按以下方式定义，

$$\begin{aligned}IG(t_i,t_j)=&-P(t_i)logP(t_i)-P(\overline{t_i})logP(\overline{t_i})\\&+P(t_j)\left(P(t_i|t_j)logP(t_i|t_j)+P(\overline{t_i}|t_j)logP(\overline{t_i}|t_j)\right)\\&+P(\overline{t_j})\left(P(t_i|\overline{t_j})logP(t_i|\overline{t_j})+P(\overline{t_i}|\overline{t_j})logP(\overline{t_i}|\overline{t_j})\right)\end{aligned} \tag{6.3}$$

词t_i在问题q_j的第 k 个回复a_{jk}中的覆盖力定义如下：

$$S_{cov}(t_i,a_{jk})=\sum_{t_j\in a_{jk}}IG(t_i,t_j) \tag{6.4}$$

另外，为了降低冗余度，两个关联度很高的答案话题相关词没有必要同时选择。例如，“Micheal Mina”和“St. Francis Hotel”是两个关联度很高的答案话题相关词，“St. Francis Hotel”是“Micheal Mina”的位置。事实上，对摘要而言，同时包含“St. Francis Hotel”和“Micheal Mina”的句子并不见得比只包含“Micheal Mina”的句子重要。甚至，人们更偏向于只包含“Micheal Mina”的句子，因为两者都包含相同的信息，即“Micheal Mina”餐厅值得推荐，而后者的长度较短。因此，本文的算法中，当其中一个词被选择，另一个被过滤。具体来说，算法用如下的互

信息公式来衡量两个词之间的关联度[105]，

$$MI(t_i, t_j) = log\frac{P(t_i \wedge t_j)}{P(t_i)P(t_j)} \quad (6.5)$$

词t_i与在问题q_j的第 k 个回复a_{jk}中已选择的答案话题相关词集合$T^{(r-1)}$之间的关联度定义如下，

$$S_{rel}(t_i, a_{jk})^{(r)} = \sum_{t_j \in T^{(r-1)}} MI(t_i, t_j) \quad (6.6)$$

综上，本文选择的答案话题相关词应该更倾向于能够预测其它的词的分布，且与已选择的答案问题相关词关联性小的词。具体选择方法如下所示，

$$S_{top}(t_i, a_{jk})^{(r)} = 1 + \lambda S_{cov}(t_i, a_{jk})^{(r)} - (1-\lambda)S_{rel}(t_i, a_{jk})^{(r)} \quad (6.7)$$

其中，$S_{top}(t_i, a_{jk})^{(r)}$为在选择问题$q_j$的第 k 个回复$a_{jk}$的第$r$个答案话题相关词时的话题相关性。当一个词的答案话题相关性大于某个阈值时，算法认为该词是答案话题相关的。值得注意的是，未被选择的词并不都是答案细节相关的词，其中也包含与所选择的话题相关词冗余的词。事实上，答案细节相关词在对最终生成的摘要答案中的每个话题进一步进行多侧面摘要时是有很重要的意义的，但对生成一个简单的摘要答案而言，答案话题相关词才是至关重要的。

6.3.4 标准化及参数设定

在识别问题相关性词（即过滤通用词）时，算法需要设置一个过滤通用词的阈值。为了提高该阈值在不同问题中的通用度，本文使用以下方法对词的问题相关性进行标准化[106]，

$$N_{Q-relevance}(t_i, q_j) = \frac{Q-Relevance(t_i, q_j) - \mu_j}{\sigma_j} \quad (6.8)$$

$N_{Q-relevance}(t_i, q_j)$是标准化后词$t_i$在问题$q_j$中的问题相关性；$\mu_j$是问题$q_j$中所有词的问题相关性的平均值；$\sigma_j$是问题$q_j$中所有词的问题相关性的标准方差。我们本文可以用相似的方法，对词的答案相关性和答案话题相关性进行标准化。公式如下所示，

$$N_{A-relevance}(t_i, a_{jk}) = \frac{A-Relevance(t_i, a_{jk}) - \mu_{jk}^{A-rel}}{\sigma_{jk}^{A-rel}} \quad (6.9)$$

$$N_{top}(t_i, a_{jk}) = \frac{S_{top}(t_i, a_{jk}) - \mu_{jk}^{top}}{\sigma_{jk}^{top}} \quad (6.10)$$

$N_{A-relevance}(t_i, a_{jk})$和$N_{top}(t_i, a_{jk})$分别为标准化后词的答案相关性和答案话题相关性。$\mu_{jk}^{A-rel}$为问题$q_j$的第 k 个回复$a_{jk}$中所有问题相关词的答案相关性的平均值；$\sigma_{jk}^{A-rel}$为问题$q_j$的第 k 个回复$a_{jk}$中所有问题相关词的答案相关性的标准方差。$\mu_{jk}^{top}$为问题$q_j$的第 k 个回复$a_{jk}$中所有答案相关词的答案话题相关性的平均值；$\sigma_{jk}^{top}$为问题$q_j$的第 k 个回复$a_{jk}$中所有答案相关词的答案话题相关性的标准方差。最终，本文对每个标准化词的问题相关性，答案相关性及答案话题相关性选择的阈值分别为 0、0.2、0.1。

6.4 实验和评测

本节将介绍本文所使用的评价实验数据集，基准方法及评测指标，最后展示并分析实验结果。

6.4.1 数据集

由于这是首次对资询问题的候选答案进行摘要的研究，我们需要构建自己的评测数据集。本文从 Yahoo! Answers 的 25 个顶层目录（除了目录“Best of Answers”及“Yahoo! Product”）中分别随机地选择 200 个问题，共得到 5,000 个问题。考虑到有足够的回复才有摘要的必要，本文筛选出其中至少有 10 个回复的问题，共有 869 个。我们邀请了两个标注者标注这 869 个问题是否是资询问题。标注的一致性达到 87%。其中，被两个标注者都标注为调研问题的问题达到 97%。由于进一步的标注极其耗时耗力，本文只选择了其中 200 个问题进行进一步标注。

我们的实验目的是评测本文提出的算法是否可以产生能够覆盖更多候选答案的完备答案。因此，两位标注者被要求对以上选择的 200 个问题的回复进行标注，识别这些问题的回复中的“候选答案”及支持该候选答案的句子。以图 6.1 中的第一个回复为例，“Carnelian Room”是一个“候选答案”，而“Try the Carnelian Room”是一个支持此候选答案的句子，被称为“贡献句”。对一个被不同的标注者标注为不同的“候选答案”的句子，我们邀请了第三位标注者在已选的两个不同的“候选答案”中，决定其属于哪个“候选答案”的“贡献者”。这里，如果其中标注者 A 标注出的某个“候选答案”的一个“贡献句”未被标注者 B 标注为任何“候选答案”的“贡献句”，实验认为标注者 B 将该“贡献句”标注给一个名为“空答案”的虚拟候选答案。

6.4.2 基准算法

本文的实验使用了两个基准系统。其中，系统 System QR 采用与本文提出的方法相同的算法框架，不同之处在于，该系统对每个问题提取所有的问题相关词作为特征，每个特征的权重采用公式（6.1）计算得到。而 System Random 随机选择问题回复的第一个句子生成摘要。

6.4.3 评测指标

本文分别采用答案召回率、答案准确率、答案精确度三个指标分别对利用文本摘要方法生成的答案从答案的完备性、正确性和冗余性三个方面进行评测。在这里需要强调的是，在社区问答服务中，一个调研问题，回复者会提供多个候选答案。每个候选答案会有被多个回复者所支持，相应的有多个贡献句。本文认为，如果一个候选答案的任意一个贡献句出现在生成的摘要答案中，则意味着该候选答案出现在生成的摘要答案中。三个指标的具体计算方法如下：

答案召回率：该评测指标意在衡量生成的摘要答案覆盖了多少候选答案。

$$AnswerRecall = \frac{N_{correct}}{N_{candidate}} \qquad (6.11)$$

$N_{correct}$是给定一个问题，在其生成的摘要答案中出现的候选答案的个数；$N_{candidate}$是在社区服务中，给定一个问题，所有回复者提供的候选答案的个数。

答案准确率：这个评测指标是用来评测生成的摘要答案是否提供了准确的候选答案信息而不是将答案的细节信息作为候选答案提供给用户。

$$Answer\ Precision = \frac{N_{contributor}}{N_{return}} \qquad (6.12)$$

$N_{contributor}$是给定一个问题，在其生成的摘要答案的句子中，候选答案贡献句的个数；N_{return}是给定一个问题，在其生成的摘要答案中句子的个数。

答案精确度：这个评测指标是用来评测生成的摘要答案是否存在冗余出现的候选答案。

$$Answer\ Accuracy = \frac{N_{correct}}{N_{contributor}} \qquad (6.13)$$

$N_{correct}$是给定一个问题，在其生成的摘要答案中出现的候选答案的个数；$N_{contributor}$是给定一个问题，在其生成的摘要答案的句子中，候选答案贡献句的个数。

6.4.4 实验

本小节将展示并分析了实验结果。在本文的实验中，摘要的长度限制设为 100 个词。

6.4.4.1 比较生成的摘要答案与社区服务中选择的“Best Answer”

事实上，从正确性的角度来讲，大部分的社区服务中所选择的“Best Answer”可以被重用。但是，从完备性的角度来讲，这些“Best Answer”是可以被改进的。本小节的实验目的就是为了验证，从完整性的角度讲，“Best Answer”是可以被改进的，且本文所提出的摘要方法可以很好地提高答案的完备性。实验比较了生成的摘要答案和“Best Answer”的答案召回率。图 6.3 的结果表明，在相同的摘要长度内，本文提供的候选答案的数目明显优于社区服务所选择的“Best Answer”。

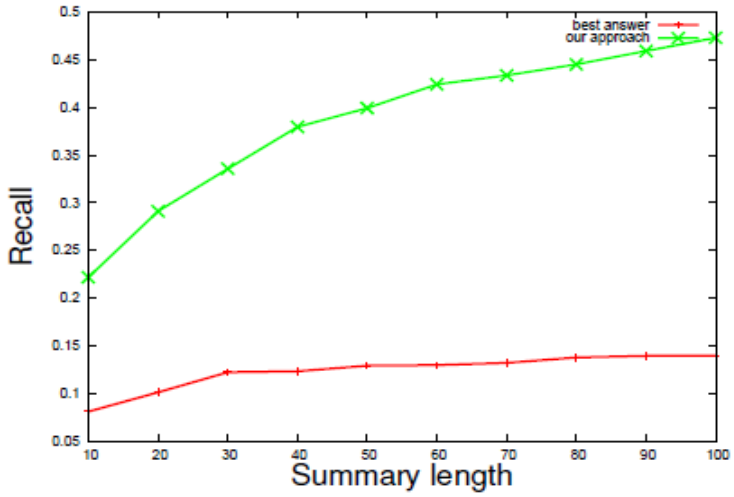

图 6.3 摘要生成的答案与“best answer”的召回率的比较

6.4.4.2 比较不同的摘要方法

如图 6.4 所示，在相同摘要长度的情况下，本文方法生成的摘要答案的答案召回率要明显优于两种基准方法。这意味着本文方法生成的摘要答案更加完备。需要注意的是系统 System QR 的召回率甚至要比系统 System Random 还低（此现象同样出现在答案准确率和答案精确度的比较中）。我们分析了 System QR 系统产生的摘要答案，发现其中包含大量关于候选答案的细节的信息，这些信息与问题标题中所包含的信息更加相关，而与问题所需的答案的相关性较低。但是，在 System Random 中，则没有这种偏向性。

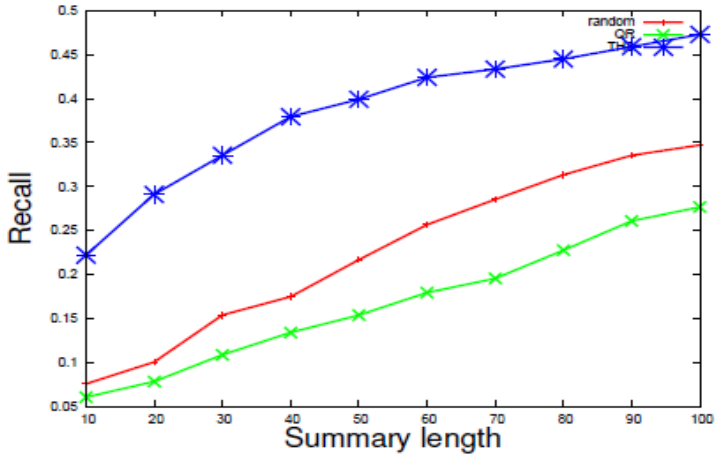

图 6.4 答案召回率比较

我们称包含候选答案的句子为“真贡献句（true contributor）”，而不包含候选答案的句子为“伪贡献句（pseudo contributor）”。从图 6.5 可以看出，本文方法生成的摘要的准确度随着摘要长度的增长逐渐下降。换言之，对本文方法生成的摘要答案而言，较短的摘要答案比较长的摘要答案更加准确。也就是说，对本文方法而言，在句子选择的过程中，真贡献句较伪贡献句通常要较早被选择。相对的，两个基准算法的准确度随着摘要长度的增长而增加。也就是说，对基准算法而言，在句子选择过程中，真贡献句较伪贡献句较早被选择。因此，本文方法在正确排序句子并准确选择句子方面优于基准算法。

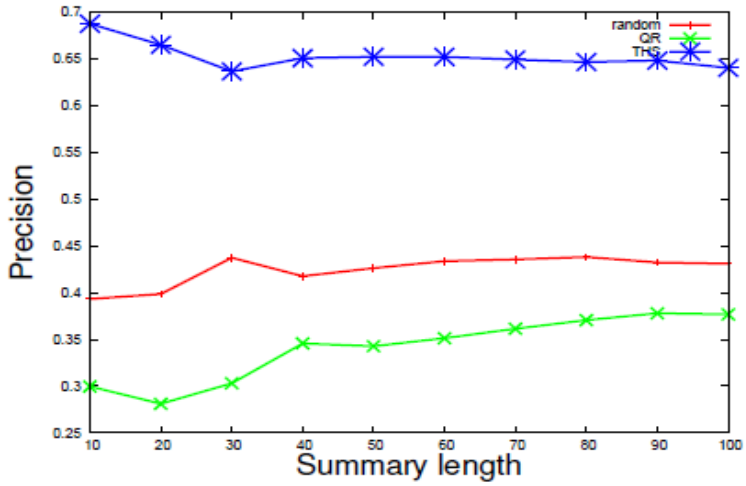

图 6.5 答案准确率比较

从图 6.6 可以看出，本文方法生成摘要的精确度优于基准算法。这意味着，本文方法可以更准确地识别同一个候选答案的不同贡献句，尽可能的避免生成的摘要答案中的冗余。这一结果表明，6.3.3 节识别的层次结构的上层词捕获了文本所

表达的更重要的信息。

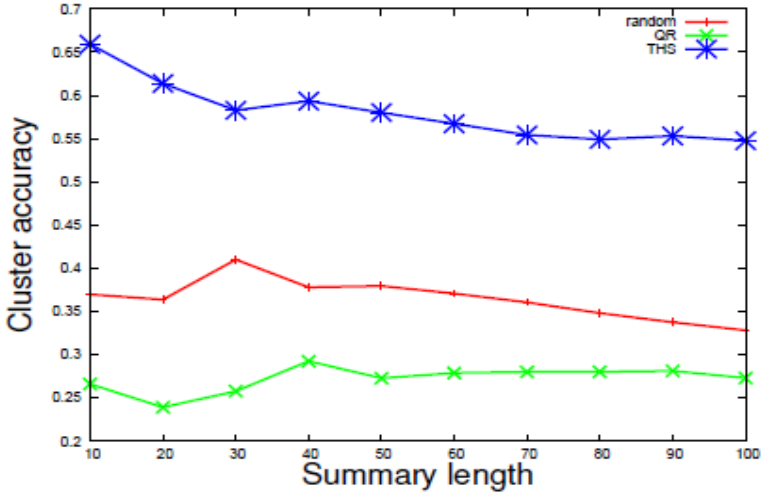

图 6.6 答案精确度比较

6.4.4.3 #N 答案召回率比较

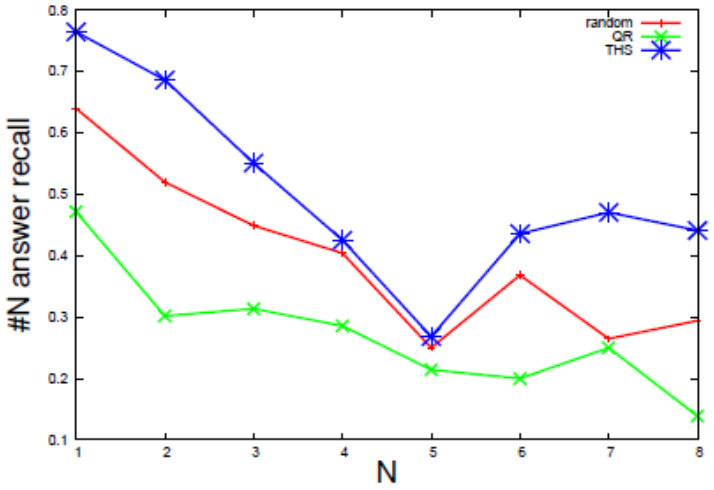

图 6.7 三种方法#N 答案召回率的比较

尽管每个候选答案都是正确答案，但是被更多人支持的候选答案应该比少数人支持的候选答案更有价值。包含更受欢迎的候选答案的摘要答案要优于包含较不受欢迎的候选答案的摘要答案。本小节实验将所有的候选答案按其支持者的个数进行排序。#N 答案召回率量度排在前 N 个的候选答案在生成的摘要答案中有多少被返回。这个实验的目的是评估生成的摘要答案是否偏向于优先包含被多数人支持的候选答案。#N 答案召回率的曲线如图 6.7 所示。在本文方法中，受欢迎的候选答案出现在生成的摘要答案中的可能性更大，且在同样的摘要长度的情况下，

本文方法生成的摘要答案包含更多的受欢迎的候选答案。

6.5 本章小节

随着社区问答服务的发展，一个大型的问题答案对形式的知识库被快速建立。如何重用这些问答知识成为一个有趣的研究领域。到目前为止，大多数研究者都专注于被重用的问题的答案正确性，很少注意答案的完备性。然而，在某些情况下，答案的完备性也很重要。本文关注一种特殊的问题——调研问题。调研问题是指请求回答问题的用户推荐针对某种需求的最佳选择的问题。它没有唯一正确答案，在社区问答服务中选择的“best answer”不足以让所有的用户满意。对这种问题来说答案的完备性至关重要。本文尝试使用面向问题的文本摘要技术对社区问答服务中所有回复者所提供的答案进行摘要，从而解决答案完备性的问题。实验 6.4.4.1 比较了生成的摘要答案和社区问答服务中所选择的“Best Answer”，实验结果表明，从答案的完备性角度看，通过摘要生成的答案可以包含更多的候选答案。

从技术角度上看，尽管目前备受关注的基于查询的文本摘要和面向问题的文本摘要都需要从待摘要的文本中摘要出指定话题的内容，他们还是存在本质的区别：查询提供了指定话题的关键字，而问题只提供了指定话题的背景信息，我们仍然需要从待摘要的回复中识别问题所需的话题。而对调研问题而言，如果把每个候选答案作为一个话题，答案话题特征识别则是一个多话题多侧面的问题。这使得答案话题特征识别成为调研问题答案摘要的一大挑战。本文提出一种新的基于答案相关词的层次结构的抽取式摘要方法。实验结果表明从完备性、正确性和低冗余性来讲，本文方法产生的摘要答案都要明显优于两个基准方法。

第七章 结论与展望

7.1 本文总结

自上世纪 90 年代互联网搜索引擎出现以来，互联网搜索已经成为人们日常生活必不可少的一部分。而随着互联网信息的日趋丰富，人们对搜索引擎的人性化、智能化、个性化等方面提出了更高层次的要求。将自然语言处理技术应用于搜索引擎，从而使搜索引擎从关键字匹配层面走向真正的知识层面，成为目前的迫切需求。本文从更友好的人机界面，更智能的搜索，更简洁准确的结果显示等角度出发，对搜索引擎的查询推荐、查询意图识别、查询理解、问题答案的重用等方面进行了深入研究。对本文各个研究点的工作总结如下：

(1) 本文提出针对具体的应用场景，用户需要基于不同语义关系的查询推荐，并以用户多选择决策场景为例，提出基于比较关系的查询推荐。本文利用社区问答服务中收集的问题，基于一个弱监督的方法在识别比较性问题的同时从中抽取存在比较关系的对象，即比较对象对。这里，比较的对象可以是解决方案、服务、产品、旅游地点、教育机构、人物等。本文依赖一个直观的想法：好的识别比较性问题的模式应该能抽取好的比较对象对；好的比较对象对通常出现在具有明显比较意图的比较性问题中。利用这一想法，本文自举抽取和识别过程。通过利用大量未标注的数据和需要少量监督决定四个参数的自举过程，本文挖掘出 328,364 个比较对象对及 6,869 个抽取模式。实验结果表明，弱监督自举算法对比较性问题识别和比较对象抽取都是行之有效的，在获得高召回率的同时也保证了高准确率。本文给出的实例表明，挖掘出的比较对象对确实反映了用户的比较兴趣点。本文提出的比较对象挖掘方法可以被用于商业搜索或产品推荐系统。例如，在用户做出购买决定之前，自动建议用户可以比较的对象将对用户起到很大的辅助作用。另外，本文的结果还可以为想要准确定位其竞争者的公司提供有用信息。

(2) 本文提出基于用户查询请求的可比较对象间的比较关系的图聚类算法以识别用户不同的比较意图。该算法基于一个直观的想法：用户查询请求在同一种比较意图下的可比较对象之间的可比性也比较大，而不同比较意图下的可比较对象之间的可比性较小。算法识别出的每一个用户可能的比较意图由一组用户查询的可比较对象表示，且通过信息抽取的方法被赋予一个语义标签。实验证明，本文的用户比较意图识别算法的准确率达到 92.7%。另外，本文还针对用户的比较行为建立了一个比较搜索系统，该系统在识别用户查询请求的不同比较意图的同时还提供了不同的比较意图下，不同可比较对象与用户查询请求的比较

信息。

(3) 本文提出一种基于查询模式的面向开放域的查询理解方法。首先，针对开放域中需要的语义标签难以预测的特点，本文利用已有信息抽取算法自动创建一个大规模的语义词典；然后，针对自动创建的语义词典语义标签数量庞大的特点，本文提出一种无监督的互增强的查询模式挖掘算法。为了解决语义词典不完整性和噪音带来的问题，算法引入了“限定的查询词t”及“NA”的概念。最后，本文建立了一个简单的查询词识别和消歧模型来测试挖掘出的查询模式的有效性。实验结果表明，挖掘到的查询模式对于查询理解是准确的。

(4) 本文针对调研问题，尝试采用面向问题的文本摘要技术，对社区问答服务中每个问题所有回复者所提供的答案进行摘要，从而解决问题答案库重用过程中答案的完备性问题。实验结果表明，从答案的完整性角度看，本文提出的方法所生成的答案较社区问答服务中所选择的“Best Answer”包含更多的候选答案。另外，从摘要技术的角度，本文发现，尽管目前备受关注的面向查询的文本摘要和面向问题的文本摘要都需要从待摘要的文本中摘要出指定话题的内容，它们还是存在本质的区别：查询提供了指定话题的关键字，而问题只提供了指定话题的背景信息。在面向问题的文本摘要中，我们仍然需要从待摘要的文本中识别问题所需话题的特征或关键字。而对调研问题而言，如果将每个候选答案作为一个话题的话，答案话题特征识别是一个多话题多侧面的问题，这使得答案话题特征识别成为调研问题答案摘要的一大挑战。本文提出一种新的基于答案相关词的层次结构的抽取式摘要方法。实验结果表明，从完备性、正确性和低冗余性来讲，本文提出的方法所产生的摘要答案都要明显优于两个基准方法。

7.2 研究展望

本文在从自然语言处理的角度提高搜索引擎性能方面取得了一些研究成果，但还有部分问题亟待后续深入研究。同时，由于互联网信息形式的不断丰富，信息数量的不断膨胀，互联网搜索也将是一个长远的研究主题。自然语言处理技术作为一门研究人机交互的语言，在帮助计算机理解搜索用户需求，并准确提供用户需要的信息方面将大有可为：

一、对本文工作的深入研究

(1) 本文提出在不同的应用场景下，用户需要与其初始查询请求有不同语义关系的查询推荐。而用户选择的查询推荐，也将帮助搜索引擎推测用户可能的应用场景，从而预测用户行为。研究如何根据与用户查询请求的语义关系的不同，对相关查询推荐聚类，从而识别不同的查询场景或意图，将有利于搜索的智能化。

(2) 本文提出利用查询模板进行面向开放域的多关键词的复杂查询理解。实验表明，

查询理解，即查询关键字的识别和消歧，准确性有大幅提高。但是只包含一个槽的表层查询模板对于查询理解的召回率很难保证。如何生成泛化的模板以提高查询理解的召回率是下一步需要研究的内容。

(3) 本文提出并尝试解决问题-答案库重用时的答案完备性问题。这一问题主要是针对网络用户个体差异性大，对不同的用户，同一问题的答案不同。然而，对具体的用户而言，完备答案必然存在冗余信息。在基于重用的问题-答案库进行搜索或问答时，同样需要考虑用户的个体差异性。尤其是在近年来兴起的社会化网络网站中（如 twitter, 新浪微博、facebook、人人网等），对给定的问题，如何利用用户兴趣、查询记录、与其他用户的关联、交互等个性化信息从重用的问题-答案库的完备答案中选择恰当的信息呈现给用户还有待研究。

二、挖掘新的研究点

(1) 自然语言中的一词多义及一义多词现象一直是阻碍搜索智能化的一大难题。理解词的不同含义，可以帮助计算机更好地理解查询；而识别同一含义下不同词，可以帮助计算机识别多种形式表达的相关信息。真正的智能搜索可以理解成一个“词汇->语义->词汇”的过程。然而，在目前的搜索引擎中，人们还不能真正理解词背后的语义。在一些社会化网络的搜索中，对理解词汇背后的语义的需求更加迫切。尤其是在 twitter、新浪微博等新兴网站中，网络信息多是短文本形式，相关信息之间词汇的重叠性很低，语义层的匹配变得尤为重要。

(2) 面向大规模数据的自然语言处理技术。当自然语言处理技术应用于海量信息搜索时，很多代价较高的深层分析不再适用，例如，句法解析。可替代这些深层分析的浅层分析成为热点。

(3) 无监督的自然语言处理技术。网络信息的多样化导致了构造恰当训练集的复杂性。尤其是在 Web2.0 时代，网络上出现的用户生成信息越来越多，这些信息往往不是很规范，不符合句法或语法信息，用词也天马行空，无论是构造语料库分析还是理解这些信息都变得需要高额代价。因此，无监督的方法受到越来越多的关注。

致谢

博士论文完成之际，我五年的博士生涯即将结束，也是我二十多年的求学之路的一个暂时终点。从此，我的人生将进入新的篇章。回首往昔，泪水与汗水挥洒无数，失败与成功如影随形。乐观的精神让我在物质和精神困难时都没有屈服，但每一步成长都离不开老师、朋友和亲人的关心与帮助，衷心感谢你们！

首先感谢我的导师李舟军教授！从大三开始，我就在您的关怀和指导下开始研究生涯。您严谨的研究态度、渊博的学识、儒雅的风度是我一生学习的榜样。感谢您对待学生宽容与耐性，使我能够从一次次的挫折中站起来，平静地思考自己的缺点和错误，不断完善自己。感谢您在人生观价值观上对我的影响，每当我面对困难想要放弃的时候，总是您的话鼓励我不断挑战自我，追求更高的目标。感谢您在生活上的关心与照顾，让我总是能够感到如沐春风般的温暖。

感谢和缅怀我的导师陈火旺院士！作为您的最后一位学生，能够有幸聆听您的教诲，是我求学生涯最大的幸福。在短暂的接触中，学生已经被您的学术造诣和学术精神所折服。至今仍然无法忘记那个万里冰封的日子，噩耗传来，天地悲恸！弟子无以为报，唯有化悲痛为力量，在学业和事业上更加努力，告慰您的在天之灵。

感谢微软亚洲研究院的 Chin-Yew Lin 研究员。在微软亚洲研究院的将近四年时间里，每一次的 one-one 交流，都受益匪浅。您对研究的热诚深深地感染了我，尤记得您所说的“如果自己都对自己所做的事情没有信心没有激情，谁还会对它有兴趣呢？”。您从不会强迫我做什么，而是通过一次次的质疑，引导我一步步地深入思考，走入研究的正确轨道。而您的敬业精神、对时间的有效管理、与人合作沟通的方式更是我学之不尽的财富。

感谢香港理工大学的李文捷老师。在香港理工大学的半年时间里，您已经成为我的良师益友。您亲切的笑容，平易近人的话语帮助我快速融入实验室的研究工作。您开放的思路也总是让我茅塞顿开，每次讨论都成为一种头脑的享受。您逐字逐句地修改我的报告，并提示我包括用词在内每个细节的把握，使我对研究体悟达到新的层次。

感谢教育和帮助过我的所有老师！宁洪老师、张春元老师、王保衡老师、王戟老师、殷建平老师、唐玉华老师、毛晓光老师、毛新军老师、王挺老师、陈跃新老师、谭庆平老师等等，一路过来，您们的诲人不倦让我在求学的路上收获了知识，同时也收获了无私奉献的博大胸怀！

感谢微软研究院帮助指导过我的所有研究员。感谢 Young-In Song 研究员指导了我包括编程风格、报告组织、写作技巧等多方面研究必须的技能，并总是在我

沮丧的时候给我鼓励；感谢曹云波研究员总能在需要的时候帮我解决模型建立中遇到的困难，并无私地与我分享其研究经验；感谢赖伟研究员作为潮人代表总是分享他所发现好用的工具，网络上新型应用，为我开拓新的研究思路。另外，感谢周明老师、Tetsuya Sakai 、李牧研究员、周晓华研究员、蒋龙研究员、Henry Li 研究员、孙剑涛研究员等等，在与你们共同的工作学习中，我感受到你们对生活的热情和乐观向上的精神。

感谢师门的所有兄弟姐妹们！因为有了你们，我们才组成了一个集体，在这个集体中一起奋斗！李梦君、颜跃进、巢文涵、孙云、张丽娟、文建、刘万伟、陈小明、陈立前、刑建英、陈石坤、张帆、贾仰理、张晓燕、魏登萍、张献、王杰生、叶云，从各位师兄师姐身上我学会了许多！李强、蔡衡、萧鹏、李晨、张甲、马晓婷、尹晓诗，跟你们这些师弟师妹一起讨论和游玩都是非常开心的事情！

感谢在微软亚洲研究院认识的朋友，宋鑫莹、焦斌星、魏晓娟、孟莎、刘璟、廖振、戴李灿、蒋敬田、孙韬、徐昊、潘兆泰、杨楠、李方涛、程宇、Makato、aikawa、Hajimi、JT Lee、陶李天、肖桐、胡辛遥、段惠中、周超、伏晶晶、邸燕凤、孙宏、明朝燕、Joty、Minwoo、王永强、鲁静、张兰、何音、崔磊、黄书剑、时雄一、孙凌，来自不同学校，甚至不同国家的朋友，能够彼此认识就是莫大的缘分，何况曾经一起爬过山、涉过水、打过球、唱过歌，你们的友谊使我宝贵的财富。

感谢在香港认识的好友！Dehong Gao、Renxian、Jian Xu, 跟你们同在一个实验室并能够得到如此多的帮助，是我非常荣幸的经历! Bo Liu、Changsheng Li，在初到香港的日子，感谢你们带我熟悉周围的生活环境、尝试美食、游历香港，使我的生活更加丰富多彩。感谢王飞、刘宇静、王勇、冯权友，作为校友兼室友，彼此之间的相互帮助，彼此扶持，使得在香港的时光充满了家庭般的温暖。

感谢 01 级本科、05 级硕士和 07 级博士同学。侯可佳、谭麟、刘波、张倩、肖林、范永亮、尹铨、陈凌鹤、李松涛、黄立波、林毅、金晶、段玉龙、陈涛、陈丽莉、曹丹、冯振乾、白冰、唐滔、马俊、张静、侯婕、马千里、刘丽霞等，有友情的我们一起走过了那些军训和考试的日子，这份同学情战友情将终身珍藏。

感谢培养我十年的科大！一个年轻人的梦想在您这里得以实现，祝福您早日成为世界一流大学，为我党我军培养更多的优秀人才！感谢为了培育我们而辛勤劳动的学校、学院和学员大队各级领导和参谋们。更要感谢的是，陪伴我们多少个日日夜夜的队干部们，陈队长、汪教导员、李教导员、刘队长、王队长、郑政委、鲍政委、吉队长、林政委，你们辛苦了！

感谢远在烟台的小学、初中和高中母校以及那里曾经教育过我的老师们！因为你们，我才有良好的基础和乐观的精神成就今天！

最后把我所有的感激都送给我的家人。你们无私伟大的爱，伴随我走遍天涯海角，使我永远不会感到孤独。感谢爱人余杰多年以来的照顾和体贴。从相知、相识、到相爱、再到结发，漫漫长路，我们一起经历过丰收的喜悦，也经历过失败的苦涩。在我迷茫和焦躁时，你总会用温柔的安慰和支持来助我渡过难关；在我骄傲和自满时，你总是用理性的分析和劝慰来让我沉着冷静！感谢爸爸妈妈，感谢你们对我的爱让我形成现在乐观向上的性格！感谢公公婆婆，感谢你们的疼爱与包容！感谢爷爷、奶奶、外公、外婆，您们从小疼爱的孙儿如今已经长大成人！

再次感谢所有帮助、支持、关心和鞭策过我的人。学无止境，我将一如既往的前进、前进！

参考文献

[1] Web Search Engine, http://en.wikipedia.org/wiki/Web_search_engine

[2] Alexa 互联网，http://cn.alexa.com

[3] 全球十大互联网资产排行榜，http://bbs.tiexue.net/post2_4128196_1.html

[4] 2010-2011 年中国搜索引擎用户行为研究报告，http://www.iresearch.com.cn/Report/1563.html

[5] PageRank，http://en.wikipedia.org/wiki/PageRank

[6] 超链分析，http://baike.baidu.com/view/613667.html

[7] RBSE spider, http://rbse.jsc.nasa.gov/eichmann/home.html

[8] David Eichmann. The RBSE Spider - Balancing Effective Search Against Web Load， In Proceedings of WWW'94, 1994.

[9] Google, http://www.google.com

[10] FAST, http://en.wikipedia.org/wiki/Fast_Search_%26_Transfer

[11] Openfind, http://www.openfind.com/china/index.php

[12] 北大天网，http://www.sowang.com/beidatianwang.htm

[13] 百度，http://www.baidu.com

[14] Ask Jeeves, http://www.ask.com/

[15] WolframAlpha , http://www.wolframalpha.com/

[16] http://www.3g4g5g.com/itxinwen/hulianwang/2011/0727/46707.html

[17] Text REtrieval Conference(TREC), http://trec.nist.gov/

[18] CNNIC，http://www.cnnic.net.cn/dtygg/dtgg/201101/t20110118_20250.html

[19] NLP, http://en.wikipedia.org/wiki/Natural_language_processing

[20] Dou Shen, Jian-Tao Sun, Qiang Yang, Zheng Chen. Building bridges for web queryclassification. In Proceedings of SIGIR'06, 131-138, 2006.

[21] Deep Web，http://en.wikipedia.org/wiki/Invisible_Web

[22] Ryen W. White and Gary Marchionini. Examing the effectiveness of real-time query expansion. Information Processing Management, 43(3) : 685-704, 2007

[23] Jinxi Xu and W. Bruce Croft. Query expansion using local and global document analysis. In Proceedings of SIGIR'96, 4-11, 1996.

[24] Jyh-Ren Shieh, Yung-Huan Hsieh, Ching-Yung Yeh, Ching-Yung Su, Ching-Yung Lin and Ja-Ling Wu. Building term suggestion relational graphs from collective intelligence. In Proceedings of WWW'09, 1091-1092, 2009.

[25] Yang Xu, Gareth J. F. Jones, Bin Wang. Query dependent pseudo relevance feedback based on Wikipedia. In Proceedings of SIGIR'09, 59-66, 2009.

[26] Silviu Cucerzan and Ryen W. White, Query suggestion based on user landing

pages. In Proceedings of SIGIR’07, 875-876, 2007.

[27] Bruno M. Fonseca, Paulo B. Golgher, Edleno S. de Moura, Nivio Ziviani. Discovering Search Engine Related Queries Using Association Rules. In Proceedings of LA-WEB’03, 66-74, 2003.

[28] 王继民, 彭波. 搜索引擎用户点击行为分析. 情报学报, 25(2)，2006 .

[29] Ioannis Antonellis, Ioannis Antonellis and Chi-Chao Chang. SimRank++: query rewriting through link analysis of the click graph. In Proceedings of VLDB’08, 408-421, 2008

[30] Andrei Broder. A taxonomy of web search. SIGIR Forum, 3-10, 2002.

[31] Daniel E. Rose and Danny Levinson. Understanding user goals in web search. In Proceedings of WWW '04, 13-19, 2004.

[32] Ricardo Baeza-Yates, Liliana Calderón-Benavides and Cristina González-Caro. The intention behind web queries. In Proceedings of SPIRE’06，9-109, 2006.

[33] In-Ho Kang and GilChang Kim. Query type classification for web documentRetrieval. In Proceedings of SIGIR '03, 64-71, 2003.

[34] Steven M. Beitzel, Eric C. Jensen and Ophir Frieder. Improving automatic query classificationvia semi-supervised learning. In Proceedings ofICDM '05, 42-49, 2005.

[35] Bernard J. Jansen, Danielle L. Booth, Bernard J. Jansen, Danielle L. Booth. Determining the user intentof web search engine queries. In Proceedings of WWW '07, 149-1150, 2007

[36] Uichin Lee, Zhenyu Liu, Junghoo Cho. Automatic identification of user goals inweb search. In Proceedings of WWW '05, 391-400, 2005.

[37] Xiao Li, Ye-Yi Wang, and Alex Acero. Extractingstructured information from user queries withsemi-supervised conditional random fields. In Proceedingsof SIGIR ’09, 572–579.2009.

[38] Ye-Yi Wang, Raphael Hoffmann, Xiao Li, and JakubSzymanski. Semi-supervised learning of semanticclasses for query understanding: from theweb and for the web. In Proceedings CIKM ’09, 37–46.2009.

[39] Ganesh Agarwal, Govind Kabra, and Kevin Chen-Chuan Chang. Towards rich query interpretation:walking back and forth for mining query templates.In Proceedings of WWW’10, 1–10. 2010

[40] Xiao Li. Understanding the semantic structureof noun phrase queries. In Proceedings of ACL ’10, 1337–1345.2010.

[41] Nikos Sarkas, Stelios Paparizos, and PanayiotisTsaparas. Structured annotations of webqueries. In Proceedings ofSIGMOD ’10,771–782. 2010.

[42] Ulf Hermjakob. Parsing and Question Classification for Question Answering. In Proceedings ofACL’01, 2001

[43] Zhiping Zheng. AnswerBus question answering system. In Proceedings of HLT'02, 399-404, 2002.

[44] Abraham Ittycheriah, Martin Franz, Weijing Zhu, Adwait Ratnaparkhi and Richard J. Mammone.. IBM's Statistical Question AnsweringSystem. In Proceedings of TREC-9, 2000.

[45] Deepak Ravichandran and Eduard Hovy. Learning surface text patterns for a Question Answering system. In Proceedings of ACL'02, 41-47. 2002.

[46] Jiwoon Jeon, W. Bruce Croft, and Joon Ho Lee. Finding semantically similar questions based on their answers. In Proc. of SIGIR'05, 617-618, 2005.

[47] Jiwoon Jeon, W. Bruce Croft, and Joon Ho Lee. Finding similarquestions in large question and answer archives. In Proc. of CIKM'05, 84-90, 2005

[48] Jiwoon Jeon, W. Bruce Croft, and Joon Ho LeeSoyeon Park. A Framework to predict the quality of answers with non-textualfeatures. In Proc. of SIGIR '06, 228-235, 2006.

[49] Xudong Tu, Xin-Jing Wang, Dan Feng, Lei Zhang. Ranking community answers via analogical reasoning.In Proceedings of WWW'09, 1227-1228, 2009.

[50] Nitin Jindal and Bing Liu. Identifying comparative sentences in text documents. In Proceedings of SIGIR '06, 244–251, 2006a.

[51] Nitin Jindal and Bing Liu. Mining comparative sentences and relations. In Proceedings of AAAI '06, 2006b.

[52] Greg Linden, Brent Smith and Jeremy York. Amazon.com Recommendations: Item-to-Item Collaborative Filtering. IEEE Internet Computing, 76-80, 2003.

[53] Amazon, http://www.amazon.com/

[54] Mary Elaine Califf and Raymond J. Mooney. Relational learning of pattern-match rules for information extraction. In Proceedings of AAAI'99 /IAAI'99, 1999

[55] Claire Cardie. Empirical methods in information extraction. AI magazine, 18:65–79, 1997.

[56] Ellen Riloff and Rosie Jones. Learning dictionaries for information extraction by multi-level bootstrapping. In Proceedings of AAAI '99 /IAAI '99, 474–479, 1999.

[57] Ellen Riloff. Automatically generating extraction patterns from untagged text. In Proceedings of the 13th National Conference on Artificial Intelligence, 1044–1049, 1996.

[58] Stephen Soderland. Learning information extraction rules for semi-structured and free text. Machine Learning, 34(1-3):233–272, 1999.

[59] Zornitsa Kozareva, Ellen Riloff, and Eduard Hovy. Semantic class learning from the web with hyponym pattern linkage graphs. In Proceedings of ACL-08: HLT, 1048–1056, 2008.

[60] Raymond J. Mooney and Razvan Bunescu. Mining knowledge from text using information extraction. ACM SIGKDD Exploration Newsletter, 7(1):3–10, 2005.

[61] Peter Weiner. Linear pattern matching algorithm. In Proceedings of the 14th Annual IEEE Symposium on Switching and Automata Theory, 1-11, 1973.

[62] Marina Barsky, Ulrike Stege, Alex Thomo, and Chris Upton. Suffix trees for very large genomic sequences. In Proceedings of CIKM '09,1417-1420, 2009.

[63] Dan Gusfield. Algorithms on strings, trees, and sequences: computer science and computational biology. Cambridge University Press, New York, NY, USA, 1997

[64] Lawrence Page, Sergey Brin, and Rajeev Motwani and TerryWinograd. The PagRank citation ranking: Bringing order to the web. Stanford Digital Libraries Working Paper, 1998.

[65] Glen Jeh and Jennifer Widom. Scaling personalized web search. In Proceedings of WWW ’03, 271–279, 2003.

[66] Christopher Stokoe, Michael P. Oakesand John Tait. Word sense disambiguation in information retrieval revisited. In Proceedings of SIGIR’03, 159-166, 2003.

[67] Shuang Liu, Clement Yu and Weiyi Meng. Word sense disambiguation in queries. In Proceedings of CIKM’05, 525-532, 2005.

[68] Ricardo Baeza-Yates and Berthier Ribeiro-Neto. ModernInformation Retrieval, Addison-Wesley, 1999.

[69] Julio Gonzalo, Felisa Verdejo, Irina Chugur and Juan M.Cigarran. Indexing with WordNet synsets can improve TextRetrieval. CoRR cmp-lg/9808002: (1998).

[70] Shuang Liu, Fang Liu, Clement Yu and Weiyi Meng: Aneffective approach to document retrieval via utilizing WordNetand recognizing phrases. In Proceedings of SIGIR’04, 266-272, 2004.

[71] Mark Sanderson. Word Sense Disambiguation andInformation Retrieval, In Proceedings of SIGIR’94, 142-151,1994.

[72] Christopher Stokoe, Michael P. Oakes and John Tait. Wordsense disambiguation in information retrieval revisited. In Proceedings of SIGIR,159-166,2003.

[73] Ellen M. Voorhees. Query Expansion Using Lexical-Semantic Relations. In Proceedings of SIGIR’94, 61-69, 1994.

[74] Ellen M. Voorhees. Using WordNet to Disambiguate WordSenses for Text Retrieval. In Proceedings of SIGIR’93, 171-180, 1993.

[75] Robert Krovetz and W. Bruce Croft. Lexical Ambiguity andInformation Retrieval.ACM Transactions onInformation Retrieval Systems, 10(2):115 –141, 1992.

[76] J. Koenemann. Relevance feedback: usage, usability, utility. Ph. D. Dissertation, Rutgers University, Department of Psychology. 1996.

[77] N. J. Belkin, C. Cool, J. Head, J. Jeng, D. Kelly,S. J. Lin, L. Lobash, S. Y. Park, P. Savage-Knepshield and C. Sikora. Relevance feedback versus Local Context

Analysis as term suggestion devices. In Proceedings of TREC8, 2000.

[78] N. J. Belkin. Helping people find what they don’tknow. Communications o f the ACM, 43(8) : 58-61, 2000.

[79] Levenshtein distancehttp://en.wikipedia.org/wiki/Levenshtein_distance

[80] Satu Elisa Schaeffer. Graph Clustering. Computer Science Review, I(2007)27-64, 2007

[81] Rand Index http://en.wikipedia.org/wiki/Rand_index

[82] David Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In Proceedingsof ACL ’95, 189–196.1995.

[83] Rada Mihalcea, Paul Tarau, and Elizabeth Figa. PageRank on semantic networks, with application to word sense disambiguation. In Proceedings of COLING ’04, 2004.

[84] Roberto Navigli. Word sense disambiguation: A survey. ACM Computer Survey, 41:10:1–10:69, 2009.

[85] Siddharth Patwardhan, Satanjeev Banerjee and Ted Pedersen. Using measures of semantic relatedness for word sense disambiguation. In Proceedings of CICLing’03, 2003.

[86] Marti A. Hearst . Automatic acquisition of hyponyms from large text corpora. In Proceedings of COLING’92, 539-545, 1992.

[87] Patrick Pantel and Deepak Ravichandran.Automatically Labeling Semantic Classes. In Proceedings of HLT-NAACL’04, 2004.

[88] Rion Snow, Daniel Jurafsky and Andrew Y. Ng. Learning Syntactic Patterns for Automatic Hypernym Discovery. In Proceedings of NIPS’04, 2004.

[89] Benjamin Van Durme and Marius Pasca. Finding cars, goddesses and enzymes: Parametrizable acquisition of labeled instances for open-domain information extraction. In Proceedings of AAAI’08, 2008.

[90] Partha Pratim Talukdar and Partha Pratim Talukdar. Experiments in Graph-based Semi-Supervised Learning Methods for Class-Instance Acquisition. In Proceedings of ACL’10, 1473—1481, 2010.

[91] Huibin Zhang, Mingjie Zhu, Shuming Shi and Jirong Wen. Employing Topic Models for Pattern-based Semantic Class Discovery. In Proceedings of ACL’09. 2009.

[92] Shuming Shi, Huibin Zhang, Xiaojie Yuan andJirong Wen. Corpus-based Semantic Class Mining: Distributional vs. PatternBased Approaches. In Proceedings of COLING’09. 2009.

[93] Günеş Erkan and Dragomir R. Radev. LexRank: Graph-based lexical centrality as salience in text summarization. Journal of Artificial Intelligence Research (JAIR),22:457-479, 2004.

[94] Dragomir R. Radev, Eduard Hovy, and Kathleen McKeown. Introduction to

thespecial issue on summarization. Computational Linguistics, 28(4):399-408, 2002.

[95] Ouyang You, Sujian Li, and Wenjie Li. Developing learning strategies for topic-basedsummarization. In Proceedings of CIKM'07, 79-86, 2007.

[96] Dragomir R. Radev, Hongyan Jing, Malgorzata Styś, and Daniel Tam. Centroid-based summarization of multiple documents. Information Processing Management,40:919-938, 2004.

[97] Hongyuan Zha. Generic summarization and keyphrase extraction using mutual reinforcementprinciple and sentenceclustering. In Proceedings of SIGIR '02, 113–120, 2002.

[98] Furu Wei, Wenjie Li, Qin Lu, and Yanxiang He. Query-sensitive mutual reinforcementchain and its application in query-oriented multi-document summarization. InProceedings of SIGIR'08, 283-290, 2008.

[99] Daniel Marcu. From discourse structures to text summaries. In Proceedings of ACL'97/EACL'97, 82-88, 1997.

[100] Eduard Hovy, Chin-Yew Linand Liang Zhou. A BE-based multi-document summarizationwith sentence compression. In Proceeding of ACL'05, 2005.

[101] Hal Daumé III and Daniel Marcu. Bayesian query-focused summarization. In Proceedings of ACL'06, 305-312, 2006.

[102] Dingding Wang, Shenghuo Zhu, Tao Li, and Yihong Gong. Multi-document summarizationusing sentence-based topic models. In Proceedings of the ACL-IJCNLP'09,297-300, 2009.

[103] Huajun Zeng, Qicai He, Zheng Chen, Wei-Ying Ma, and Jinwen Ma. Learning tocluster web search results. In Proceedings of SIGIR'04, 210-217, 2004.

[104] C.E.Shannon, A Mathematical Theory of Communication, Bell SystemTechnical Journal, 379–423 & 623–656, 1948.

[105] L. J. Cronbach, On the non-rational application of informationmeasures in psychology, in H Quastler, ed., Information Theory in Psychology: Problems and Methods, 14–30, 2004.

[106] Bolstad BM, Irizarry RA, Astrand M and Speed TP.A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. Bioinformatics, 185-93, 2003

[107] http://www.se-express.com/about/ask-jeeves.htm

[108] http://en.wikipedia.org/wiki/Cosine_similarity

[109] http://en.wikipedia.org/wiki/K-means_clustering

作者在学期间取得的学术成果

[1] Shasha Li, Chin-Yew Lin, Young-In Song and Zhoujun Li. Comparable Entity Mining from Comparative Questions. ACL'10 , 2010 (领域顶级国际会议)

[2] Shasha Li, Chin-Yew Lin, Young-In Song and Zhoujun Li. Comparable Entity Mining from Comparative Questions. IEEE Transactions on Knowledge and Data Engineering (SCI 源刊，领域顶级国际期刊）

[3] Shasha Li, Zhoujun Li. Question-Oriented Answer Summarization via Term Hierarchical Structure. Int'l Journal of Software Engineering and Knowledge Engineering.(SCI 源刊)

[4] Shasha Li, Zhoujun Li. Answer summarization via term hierarchical structure. FSKD'10, 2010. (EI 收录)

[5] 李莎莎, 陈火旺, 李舟军. 篇章中的消解问题与消解算法:研究综述.计算机科学. 2007, 34(7).

[6] Yuanjie Liu, Shasha Li, Yunbo Cao, Chin-Yew Lin, Dingyi Han, Yong Yu: Understanding and Summarizing Answers in Community-Based Question Answering Services. COLING'08, 2008. (领域顶级国际会议)

[7] Shasha Li, Young-in Song, Yu Cheng, Chin-Yew Lin, Zhoujun Li. Query parsing and interpretation for semantic search. Submitted to Information Retrieval（SCI 源刊）

[8] Shasha Li, Wenjie Li, Dehong Gao, Zhoujun Li. When will people publish a tweet? ——Information Propagation and Modification in Twitter. To be submitted to WWW'12.

作者在学期间参与的科研工作

[1] 数据挖掘中的若干关键技术研究(2006.01-2007.01), 国家自然科学基金项目, 重要成员。

[2] 基于面向话题的加权社会网络的个性化推荐及检索技术研究(2012.01-2014.12), 国家自然科学基金，重要成员

[3] 意见性内容的摘要和检索架构研究(2011.05-至今), 香港政府资助项目，核心成员。

[4] 社区问答系统构建及关键技术研究(2007.09-2010.06)，微软公司研究项目，核心成员。

[5] 知识综合与语义搜索(2010.06-2011.01)，微软公司研究项目，核心成员。