Internet 上的搜索引擎和元搜索引擎

Search Engines and Meta Search Engines on Internet

彭洪汇　林作铨

（北京大学信息科学系　北京 100871）

Abstract Search Engines are the widest used information searching tools on Internet. In this paper, an extensive and in-depth survey is provided for development of search engine and meta search engine. Current status of this kind of searching tools and relevant technologies are analyzed. Further research and development are discussed.

Keywords Search engine, Meta search engine, Information retrieve

1. 引言

Internet 自诞生以来不断成长，尤其是最近几年更是得到长足发展，功能不断扩展，信息容量呈爆炸性趋势增长，仅 Internet Archive[1]收集的 1996 年以来的 Web 内容就达到四十亿个页面，容量达到 40TB。据 Internet Domain Survey[2]统计，从 1996 年到 2001 年，Internet 上的主机数量从两千万增长到一亿四千万。Internet 作为一个信息平台在人们的生活和工作中发挥越来越重要的作用，人们越来越多地通过 Internet 获取信息。然而在信息极大丰富的同时，用户也面临着信息过载和资源迷向的问题。Internet 上的信息过于庞杂，而且具有不稳定和变动快的特点，没有也不可能有一个权威机构能对这些信息进行全面的整理和归类，因此往往让用户面对五花八门扑面而来的各种信息而无所适从，不知道如何去获取自己需要的内容。

人们上网获取信息的一种普遍方式是浏览。Internet 上的文档一般都是通过超链接互相联系起来的，借助 Internet 浏览器（如 Netscape Navigator 和 Internet Explorer），人们可以追随这些链接来浏览 Web 页面的内容。浏览方式局限于链接所联系的对象，能否找到结果不是必然的，比较适合目的不明确、时间要求不紧迫的情况。当需要查找一个具体的内容时，浏览方式就像是在一个没有书目目录的图书馆中查找一本书，效率很差，一般不能在短时间内获取到想要的信息，尤其是对 Internet 不太熟悉、缺乏上网经验的用户。因此，对于上网查找信息的用户来说，功能更先进效率更高的信息检索工具显得极为重要。

信息检索一直是计算机研究中的一个重要方向，Internet 的发展使网络环境下的信息检索更是成了一个新的研究热点。研究者已经做了大量工作，力图开发出高效的检索工具[3]，以方便用户获取信息。大约 1994 年前后，Web 上出现了搜索引擎，World Wide Web Worm（WWWW）[4]就是其中的一个。现在搜索引擎是上网用户最经常使用的网络服务之一，仅次于电子邮件。

随着 Web 内容的飞速发展，搜索引擎对 Web 建立索引的能力受到越来越严重的考验。事实上，以今天 Internet 的规模和发展速度，没有一个 Internet 索引系统能够适时地全面覆盖 Web 的文档。根据文[5]，在 1997 年 7 月，数据库容量极大的 AltaVista 也只能对整个 Web 文本数据的 8～40%建立索引；同时，不同的搜索引擎的检索结果的重复率仅仅在 10～30%之间。因而一个搜索引擎通常不能找到用户需要的所有信息，用户在进行检索时需要在多个搜索引擎之间进行切换，在多个检索结果列表之中挑选对自己有用的内容。出于这个原因，一个能集成不同搜索引擎检索结果的检索工具——元搜索引擎出现了。

最初的搜索引擎不少都是在学校里面开发出来，但这些系统很快就商业化了；而后来的搜索引擎有很多是公司里面进行开发的。这些搜索引擎所采用的技术和实现情况都被视为私有财产而秘而不宣。因此，很少有文章披露搜索引擎的详细设计和实现，这在很大程度上制约了搜索引擎技术的继承、交流和发展。本文对 Internet 上的搜索引擎和元搜索引擎做了充分调查，分析了目前主流搜索引擎的特征，介绍了相关技术，并指出了这类工具的研究方向和发展前景。

2. 关于信息获取

信息检索方面的研究至今已有四十年左右的历史了。最初的信息检索方法是布尔检索，用户的查询表示为关键词的布尔关系表达式，检索的标准是匹配的文档就返回，不匹配的放弃。后来，人们为信息检索建立了向量空间、概率方法等数学模型，并引入了基于频率的词权和文档相关性排序概念，使得信息检索质量大大提高。下面介绍本文要用到的信息检索方面的一些概念和技术。

2.1 查全率和查准率

查全率（Recall）和查准率（Precise）[6]是衡量一个信息获取系统的检索效率的主要指标。在对传统的信息获取系统进行评测时，通常是预先选定一个文档集合（比如经常使用的 Text Retrieval Conference[7]评测文档集）和一个检索条件集，这个文档集中的文档和这些检索条件之间的相关情况是已经知道的。然后使用信息获取系统根据这些检索条件对文档集中的文档进行检索，根据检索结果的情况来分析该系统的检索效率。

查全率是指信息检索系统检索到的相关文档占被检索文档集中所有相关文档的比重，可以用下面的公式计算：$R_{recall}=R/R_{all}$。其中 R 是搜索到的相关文档数，R_{all} 是被检索文档集中所有的相关文档数量。一个系统的查全率越高，则说明它发现相关文档的能力越强。

查准率是指信息检索系统检索到的所有文档中相关文档所占的比率，可用如下公式计算：$R_{precise}=R/D$。其中 D 是检索到的文档数。一个系统的准确率越高，则其检索到的信息噪音越低。

2.2 文档表示

在信息检索系统中，为了能对文档进行检索处理，首先必须将文档转化成可计算对象。文档表示就是指以一定的规则和描述方法来表示文档。文档表示模型有多种，包括布尔逻辑型、向量空间型、概率型和混合型。

2.3 文档向量空间模型

在上面提到的一些文档表示方法中，向量空间模型(VSM，Vector Space Model)[6]在实际应用中实现起来比较简单但效果很好。所谓向量空间模型，是指将文档表示成由一组规范化正交词条矢量所组成的向量空间中的一个点或矢量，通常用一个多元组表示。

对于任意给定的一篇文档 d，先去掉其中不包含特殊语义信息的一般词，比如虚词、介词等，然后对剩下的词做提取词干和同义近义消除处理，得到了一个该文档的词语列表 $(T_1, T_2, \cdots, T_m)$。对于其中的每一个词 $T_i(1 \leqslant i \leqslant m)$，取它的一种统计属性 w_i 作为其权值。这样，文档 d 就可以表示为：$(T_1, w_1, T_2, w_2, \cdots, T_m, w_m)$。如果把 T_i 看作 m 维向量空间中的一组基，w_i 作为坐标值，那么文档 d 就是这个向量空间中的一个矢量 $(w_1, w_2, \cdots, w_m)$。

w_i 的取法可以有多种。最简单的是 T_i 在 d 中的出现频率 $tf(T_i, d)$。如果考虑文档长度的影响，可以以文档 d 的长度 L(d)对这个频率进行标准化：$tf(T_i, d)/L(d)$。采用这个统计属性的出发点是：一个词在文档中出现的次数越多，那么它对体现文档的内容就越重要，可以作为该文档的重要特征。

另一方面，如果一个词在越多的文档里出现，就说明这个词用以区分各个文档的意义越小，越不适合作为该文档的特征词。把这一点综合到 w_i 的计算中，可以得到式(1)：

$$w_i = tf(T_i, d)/idf(T_i, D) \tag{1}$$

其中 D 是检索的文档集，$idf(T_i, D)$体现的是 D 中包含 T_i 的文档数量属性。这个计算方法就是 tf * idf(Term Frequency Times Inverse Document Frequncy，即词频乘以文档频率的倒数)[8]。$idf(T_i, D)$通常采用 $\log(N/n_i)$，其中 N 是 D 中所有的文档数，n_i 是 N 中包含 T_i 的文档数。

采用向量空间模型很容易计算文档的相似度或者相关性：在向量空间中，如果两个向量的夹角越小，那么它们就越相似。通常采用余弦函数作为相似度计算函数，亦即两个文档向量的相似程度是它们的夹角的余弦：

$$Sim(d_1, d_2) = \frac{\sum_{i=1}^{m}(w_{i,1} \times w_{i,2})}{\sqrt{\sum_{i=1}^{m} w_{i,1}^2 \times \sum_{i=1}^{m} w_{i,2}^2}} \tag{2}$$

2.4 文档分类

文档分类是指机器根据某种文档表示和分类算法，将内容相近或具有相同特征的文档放到同一类中的过程。文档分类通常采用机器学习方法。

3. 搜索引擎

3.1 实现原理

目前的搜索引擎一般以两种方式提供服务：基于关键词的文档检索和目录结构检阅。下面分别介绍这两种方式的实现原理。

3.1.1 *基于关键词的文档检索* Web 上的文档通过超链接互相联系起来，这种链接结构可以看作是一个图，每个文档就是这个图中的一个节点，而一个超链接则是一条有向边。基于这种有向图结构，我们可以用一个软件来遍历 Web，循着超链接去访问每个 Web 文档。人们已经开发了不少这种软件，就是所谓的爬虫软件(spider、crawler、bot 等)[9]。

搜索引擎首先利用爬虫软件按照某种遍历搜索算法来对 Web 进行遍历访问，并取回访问到的每个文档。这些文档以及它们的位置信息被交给搜索引擎的索引系统。

索引系统从文档中提取一些有代表意义的信息，将它们保存在数据库里。当用户输入关键字进行查询时，搜索引擎检索数据库以查找匹配的文档，将结果返回给用户。

3.1.2 *目录结构检阅* 不少搜索引擎在提供关键字搜索的同时还建立了一个文档目录结构，用户可以根据自己要查找的内容主题，按照目录结构逐步缩小范围。每个目录下都有一些以该目录为主题的站点，用户可以选择浏览这些站点。比如说用户想在 Yahoo 上查找与防火墙有关的信息，他可以先进入“Computers and Internet”目录，再依次进入“Software”、“System Utilities”、“Security and Encryption”、“Firewalls”子目录，在 Firewalls 子目录下会发现对他有价值的信息。此外，有些搜索引擎提供先选择一个目录然后在该目录内进行关键词检索的功能。

文档目录结构的建立有两种方法：

· 雇用具有专家知识的目录编辑人员，对已经采集回来的 Web 文档或站点内容以及由站点维护人员提交的描述进行分类。人工建立目录的优点是具有较高的精确度，经过确认的内容往往具有较高的价值。但是代价较高，同时可维护的内容容量和更新速度受到制约。为了解决这个问题，Open Directory Project[10]采用开放的目录管理，国内也有搜索引擎采取同样的做法。但这需要严格的管理。另外，人工维护不可能分类到很细的目录，在进行归类时会不可避免地带有一些主观色彩。

· 利用机器对文档进行自动分类。文档分类是信息处理方面的一个重要研究方向，目前采用得比较多的是机器学习方法。自动文档分类具有处理速度快的特点，能进行大批量处理和快速更新；但是在处理的精度上不及人工分类，而且自动分类所产生的类别不一定和人们常识中的概念相符合。

3.2 现有搜索引擎的主要缺点

在搜索引擎刚产生的时候，人们认为可以建立一个完整的索引，然后就可以容易地查找到任何东西。但是随着 Internet 的发展，人们意识到这一想法显然是错误的。

首先，Internet 的发展正在以加速度进行，内容急剧膨胀，不要说为其建立一个完整的索引，就是对其内容做一个准确的统计都是不可行的。因此现在用户不能期望搜索引擎返回和他的查询条件有关的所有结果了。也就是说，搜索引擎的查全率达不到理想的效果。

其次，人们发现，索引的完整性并不是影响搜索引擎检索结果的唯一因素：检索中经常是大量的无用的结果，真正有用的结果却被淹没在其中不容易发现。搜索引擎的索引和以前相比已经有了极大的增长，一般检索时都会返回大量的结果。但是人们察看和选择结果的能力与耐心没有得到相应的提高，通常还是只会注意最前面的部分。因此，搜索引擎的“精度”，尤其是检索结果排在前面的部分对于用户的有用性，是非常重要的，有时候相对于查全率来说显得更为突出[11]。

再次，搜索引擎在用户接口以及互动性方面存在缺陷。搜索引擎应该让用户能够准确地表达出其检索条件，否则可能检索到偏离用户意图的结果。但是在这方面，现在的搜索引擎显得支持不够，尤其对于缺乏经验的用户。简单的逻辑(甚至有些搜索引擎都不能支持)有时候不足以描述查询条件，也会导致查询结果有较大的噪音，而且对于部分用户来说操作有

困难。大部分搜索引擎没有采取手段获取用户的评价或意见，也不能做到为用户量体定做查询环境。

3.3 搜索引擎要处理的核心问题

在搜索引擎的设计和实现中，文档采集、索引建立、数据结构设计、排序函数、查询处理是几个需要仔细考虑的核心问题。

3.3.1 *文档采集和更新* 搜索引擎的数据库的质量直接影响到搜索引擎的检索结果。作为建立和更新数据库的必需过程，文档采集对搜索引擎来说至关重要。

在进行文档采集时，着重要考虑的是采集的覆盖面、采集对象的质量和采集的频率。搜索引擎覆盖的 Internet 文档越全面，它找到的有用文档就可能越多，有助于提高查全率；它搜集的文档质量越高，对用户的有用性就越大；它对文档的更新情况能越快地反应出来，检索出来的结果就越符合 Internet 上的真实情况，死链接就越少。文档采集和更新面对的难题是：

①Internet 信息容量的海量特征。现今 Internet 的文档内容极其丰富，没有一个机构能够对其数量做出全面准确的统计，单个搜索引擎肯定不可能采集到所有的 Web 文档。因此，搜索引擎只能视自己的能力和具体情况决定自己数据库的规模，必须在覆盖面和自己的系统效率之间做一个权衡。另外，在决定了数据库规模的情况下，如何选择比较重要和有意义的文档进行采集也是很重要的问题，这关系到搜索引擎检索结果的质量。

②Internet 的动态和不稳定性。经常上网的人常常会遇到这样的情况：平常浏览的一个页面在某一天突然消失了。Internet 是各种文档的聚散地，但是对各种文档页面的命名、出现、存在和消失完全没有规范，它们的拥有者和存放主机可能会因为各种原因而改变它们的存在状态，用户并不一定会得到通知。Web 文档的出现、消失、迁移、版本变动是搜索引擎进行文档采集的主要困难之一。为了保证索引数据库的适时性，搜索引擎必须以尽快的速度更新数据库，这会给系统带来沉重的负担。但是不管搜索引擎如何卖力，其检索结果总是在一定程度上是过时的。

③Web 结构的复杂性。Web 文档结构和形式多种多样，从简单文本到复杂文件形式到各种媒体文件不一而足，搜索引擎必须能识别有用类型而进行采集。这些 Web 文档之间相互联系的方式也是变化多样，最普通的是简单的 URL 超链接，处理起来比较容易，但是很多情况是通过脚本和表单来控制的，搜索引擎对这种情况显得无能为力。此外还有针对浏览器特性而设置的跳转和重定向等等，这都给搜索引擎的文档采集过程增加了复杂度。此外，Web 文档的作者会出现各种各样的错误，这些千奇百怪没有规律的错误经常使爬虫程序不能正常工作。

3.3.2 *索引方法* 搜索引擎采集回来的文档要被用来建立索引数据库。索引的建立方法对搜索引擎来说具有很大的影响，好的索引能提高搜索引擎系统运行的效率以及检索结果的质量。建立索引时主要考虑如下几点：

①搜索引擎为完成一次查询而检索数据库索引所花的代价。毫无疑问，搜索引擎是访问量最大的 Internet 系统之一，1994 年的 WWWW 平均每天处理 1500 个搜索请求，1997 年的 AltaVista 每天处理大约两千万个请求，而到了今天，主流搜索引擎的日访问量更是超过上亿个。如此繁重的任务除了要求硬件性能优越之外，还要求搜索引擎系统具有极高的处理效率，在极短的时间内完成单次查询处理。检索索引是查询处理过程的主要部分，因此，必须尽量优化索引以缩短处理时间。

②通过建立良好的索引来改善检索结果的质量。比如说，索引项有没有包含一定的语义信息，索引有没有包含索引项和文档记录的相关信息，这些都会对检索结果的质量产生很大的影响。

③存储索引的空间开销。1997 年规模最大的几个搜索引擎索引的文档已经达到了几百万到一亿个左右；当今的主要搜索引擎的规模已经达到了几亿个文档页面。要为如此之多的文档建立索引一定要考虑存储空间代价。

④更新一次索引所需的计算代价。如前所述，现在搜索引擎要处理的文档数量庞大，而且搜索引擎为了能反映 Web 上的真实情况必须以尽可能短的周期来更新文档。更新一次索引需要庞大的计算量，索引的设计应该有利于缩小这个计算开支。

3.3.3 *数据结构设计* 搜索引擎是典型的对海量数据进行操作的系统。它会一次性从 Web 上采集回百万乃至上亿数量级的文档，这些文档的存储和管理需要很大的代价。对这些文档进行处理以生成索引时所产生大量的中间数据。产生的索引被以极高的频率作检索操作。所有这些数据操作都提出了一个问题：数据该以何种形式被存储和操作才能获得最高的运行效率，以及需要最小的空间开销。

3.3.4 *排序函数* 人们在用搜索引擎进行检索的时候，经常需要从结果中仔细挑选可能会对自己有用的记录，因为很多都是只有那么一点点“形似”的文档；真正有用的文档也许在其中，但不是一眼能看出来。搜索引擎检索到的结果和检索关键词之间具有不同的相关度，这些文档的质量和重要性也有所差别。为了改善输出结果的质量，现在的搜索引擎都在不遗余力地开发好的排序算法，力图把最符合用户查询条件、最有参考价值的文档排在最前面，让用户能尽快找到需要的材料。好的排序函数有助于提高用户查找信息的效率。为了开发一个好的排序算法，必须要考虑和文档相关的所有信息，尽可能地利用文档的各种属性。在这方面，各个优秀的搜索引擎提出了很多不错的想法。

3.3.5 *查询处理* 搜索引擎所做的一切都是为了能为用户提供查询服务。那么，如何处理用户的查询请求，让用户操作简易、查询有效率，是搜索引擎要考虑的问题之一。一方面，搜索引擎要考虑为用户提供尽量友好的检索界面，让用户能尽可能准确地表达其搜索请求。除了采取必要的技术支持（如逻辑查询语法、自然语言查询提问）外，还得让用户操作起来方便。另一方面，搜索引擎要具有强大的查询处理能力。好的搜索引擎应该能处理各种逻辑查询、短语查询，能进行查询精化，以及根据用户的意见做出调整。

3.4 搜索引擎的关键组件

从上面的介绍中我们知道一个搜索引擎要为用户提供检索服务必须具有这几个功能模块：页面采集系统、索引系统、数据库、查询匹配系统。

3.4.1 *页面采集系统* 负责搜索引擎对 Web 文档的获取和更新。由于搜索引擎覆盖的文档数量极其庞大，采集过程必须是高效的，以较好的时效性来实现索引更新。

采集系统的核心是遍历算法。最简单的是宽度优先算法和深度优先算法。但这是不够的，为了提高效率，采集过程必须由一些智能特性，例如关于文档更新的频度的经验知识，关

于文档类型分布的知识。另外，我们要考虑如下原因：Web 不是一个静态的图，每时每刻都在变动之中，所以采集系统要具有一定的适应和调整能力。其次是出错处理：Web 上的链接会出现很多的异常，Web 文档的作者可能会犯各种各样的创作错误，每种情况都是不可预料的，采集系统必须被设计成足够健壮，不会因为这些潜在的错误而崩溃。

页面采集系统的工作情况一般是这样的：给出一集起始文档 URL，采集系统用 HTTP 协议同这些文档的主机交互，取回文档内容。接着对这些文档进行分析，提取出其中的超链接，然后再取回这这些链接所指向的文档。如此一直继续下去，直到完成预定的采集额。如果遇到以前已经采集过的文档，就检查一下是不是有版本更新，若有再取回。一般搜索引擎在采集时都是同时启用多个线程进行并行采集，URL 数量的增长速度比较快，因此需要有一个组件来管理这些 URL，在这些线程之间做任务的分配和协调。取回来的文档一般以压缩格式存储起来。

3.4.2 *索引系统* 对页面采集系统采集到的文档进行处理，以生成索引数据库。不同的搜索引擎通常采取了不同的索引策略。索引系统对搜索引擎的检索效果影响很大，是数据库的核心部件。为了改善索引对检索效果的作用，搜索引擎在建立索引时利用了文档的各种特征，提出了各种技术。

索引系统的基本处理过程如下：首先对文档内容进行分析，如分词处理，再进一步转化成文档表示形式，如向量表示。然后决定根据文档的什么元素建立索引，如单词或短语等；再参考文档的各种属性，根据搜索引擎自己的计算模型得出这些索引元素的评价（权值），把这个结果加入到索引数据库中。

3.4.3 *数据库* 是搜索引擎存放索引信息的地方。有些搜索引擎只是将文档索引保存在数据库里，文档在经过处理后则不予保留；有些搜索引擎则还保留原文以备用户需要。鉴于互联网信息的海量特征，我们可以想象搜索引擎的索引数据库容量是很大的；搜索引擎每天要处理巨大数量的搜索请求，因此搜索引擎的数据库必须要高效、快速、并发处理能力强。

3.4.4 *查询处理系统* 搜索引擎的查询匹配系统是另一个对检索效果有直接影响的地方。首先，一个搜索引擎能不能为用户提供比较方便的检索接口是很重要的，比如是否能处理逻辑查询语法，能否逐步精化查询。其次，检索到的结果以何种方式提供给用户也是很重要的：用户应该能最快找到最好的结果。匹配系统应该把匹配程度最好的结果放在前面。

用户通过查询接口（一般是 Web 浏览器显示的 HTML 页面）填写了检索请求后，提交给搜索引擎。查询处理系统首先将这些查询请求转化为其内在表示形式，然后与索引进行匹配，这时，在涉及到逻辑查询或短语查询时不同搜索引擎的处理方法是不同的。接着，查询处理系统对符合条件的结果进行一些调整，然后以一定的格式返回给查询者的浏览器。

3.5 搜索引擎技术

到目前为止，Internet 上的搜索引擎已比比皆是了。在搜索引擎的发展过程中，研究人员和开发者不断提出和应用新技术，IR（Information Retrieve，信息获取）和 IF（Information Filtering，信息过滤）方面的研究也不断为搜索引擎的发展注入新的活力。现在 Internet 上有不少搜索引擎具有不错的检索效果。下面介绍搜索引擎常采用的技术以及一些优秀搜索引擎的技术特色。

3.5.1 *采集策略* 对于一个索引系统来说，当然是覆盖面越全越好。但是，对于一个搜索引擎系统来说，它必须要考虑实际情况。显然现在没有一个索引系统能够为整个 Web 的内容建立索引，搜索引擎只能根据自身软硬件系统的处理能力确定其索引数据库的规模，在此前提下进行文档采集。这些限制条件主要包括：硬盘的存储容量——能为多少文档的存储、处理以及索引库提供空间；搜索引擎具备的网络带宽——可以以多大速度采集文档，进而决定采集周期和更新频率是否能达到理想水平；运行在硬件平台上搜索引擎各软件模块的效率——能进行多大规模的数据处理同时又要提供理想的访问处理能力。

在确定了其采集规模之后，搜索引擎在进行采集时需要有所取舍，以提高其索引文档的质量。关于这一点，Martin Coster[12]提出了一个编写爬虫程序时遵循的若干建议，避免采集无用的文档。站点管理员可以通过创建一个 Robot 文件（Robot.txt）来指导 Robot 采集其站点上的有用文档，这些建议往往是很有价值的。在文[13]中，讨论了如何选择 URL 进行采集。

WebCrawler[11]提出的一个基本策略是：采集的文档要来自尽可能多的站点。WebCrawler 采用一个经过修改的宽度优先遍历算法进行采集，保证每个站点至少有一个文档被建立索引。这是力求扩大覆盖面的方法。WebCrawler 的遍历算法如下：

每当位于一个新的站点（以前没访问过）上的文档被发现，该站点被加入到一个要进行采集的站点列表中。在继续文档采集之前，要从每个这些新发现的站点上取回一篇文档建立索引。所有的站点都被访问之后，采集过程继续在这些已经发现的站点中进行，直到又发现了新的站点，然后再重复上面的过程，直到消耗了预定的时间或采集回了预定数量的文档。

Google 提出的一个方法是通过代理服务器（Proxy）的缓存来建立索引，因为 Proxy 的缓存是用户因其信息需求而进行浏览的内容。

3.5.2 *索引技术* 不同的搜索引擎建立索引的方法各不相同。就索引项来说，一般是基于词的索引。在国内的某些搜索引擎上，曾出现过以字作索引的情况，结果是检索结果中经常出现令人啼笑皆非的项目：以词 BC 作检索词查询时，结果中会出现包含 ABCD 的文档，但是在此文档里，AB 是一词，CD 是一词，B 和 C 却不成词。当然这也是由于汉语特定的语法结构不易处理所造成的。以词作索引的时候，一般需要一个词典，记录不需要建立索引的词（无意义词）。此外，短语也被作为建立索引的对象[14]，因为短语相对于单个的词来说，包含了更为丰富的语义信息，易于分析关键词和文档的相关性。短语索引通常结合词的索引一起使用。建立短语索引的一个问题是索引数据库容量的极大膨胀。

就建立索引的内容来说，有的是对文档里出现的每个词建立索引，亦即全文索引；有的只是对标题建立索引，如 WWWW；有的利用了超链接文字，如 Jumpstation[15]。WebCrawler[11]的方法是采取向量空间模型，对内容和标题都进行索引。Google 建立的是全文索引。

只对链接文字或者标题中出现的词建立的索引具有较高的价值，这种索引检索出来的结果一般具有不错的可用性。但是，一篇文档的主题词肯定有相当一些是没有出现在标题里的，那么以这些关键词进行检索的时候，这篇本来很有用的文档就没有被检索出来。另外，标题是 HTML 文档的一个可选部分，有不少文档根本没有标题，这给索引造成了困难。根据

WebCrawler 的总结和调查，基于内容的索引（全文索引）是必要的。下面介绍一下基本的全文索引。

全文索引，就是对文档中出现的每个词（无意义的词除外）建立索引。全文索引的好处在于简单和易于实现，并且检索时不会错过每个包含检索关键词的文档。但是问题在于：每个文档中都有不少和文档主题没有关系的词，当以这些词作为关键词进行检索时，这篇文档就会出现在检索结果中，成为一条不相关或相关性不高的记录。大量的这种结果湮没了真正相关的东西，往往让用户不易从中挑选出真正有意义的东西，导致了整个搜索质量的下降，这正是当前搜索引擎的一个主要问题。

全文索引的一般实现方法是：首先使用一个分词工具对文档进行分解，生成一个词序列。分词工具必须能够处理文档作者可能犯下的各种 HTML 标记错误或语法错误。接着用一个无关词列表对这个词序列过滤，剔除其中无意义的词，然后用文档向量表示方法计算词的权值。然后建立反向索引便于查询。

在建立索引时，为了区分索引词与不同文档的相关度（在不同文档中的重要性），引入了索引词在文档中的权值。这是为了在查询时依据这个权值来进行排序。

在建立索引时可以考虑的一个技术是多级索引。通过建立多级索引，可以进一步提高精度和检索效率，当然同时要进一步增大索引占用空间。

人们意识到建立索引时要达到的期望是：对能体现文档主题的内容建立索引，并只对这些建立索引；索引要能很好地体现出相关性。

3.5.3 *相关性分析* 确定关键词和文档之间的相关度。鉴于这个相关度是对最后结果进行排序的重要依据，各个搜索引擎都不遗余力地开发各种技巧力求得到最好的效果，主要包括：充分利用文档本身的各种信息——所包含的术语词汇以及表现形式；利用超文本的信息来改善效果[16~19]，尤其是链接结构[20]和链接文字。

被普遍采用的是利用文档的内容来进行相关性分析。在生成索引的时候，如果没有为索引对象产生权值，那么就没有进行相关性分析，检索出来的结果也没有相关性分析，看不出来孰优孰劣。实际上，文档中有很多信息可以用来产生权值：

· 最基本的就是，利用索引对象在文档中出现频次的统计属性作为其相关性权值。如果考虑到文档长度的影响，可以对长度进行标准化[21]。比较好的方法是采取文档向量空间表示方法（见 2.3 节）中的权值计算方法。

· 索引对象出现的位置。在一些特殊位置上出现的内容可以作为文档的核心内容而赋予较高的权值，比如 HTML 标题，HTML META 标记等，文档的标题或小标题以及章节标题。

· 索引对象的显示形式。在文档中，有些内容的显示方式不同于周边的文字，例如用粗体、斜体，用不同的颜色。这些内容一般是作者有意突出的，因此可以赋予较高的权值。

链接文字是很有价值的信息，但是以前根本没有被利用，或是仅仅作为包含链接的文档中的内容进行处理。但是，实际上链接文字对于分析文档和关键字的相关性非常有用处：链接文字往往是对被链接的文档的内容的描述和概括，而且是人（链接文字的创作者）的理解和评价，可信度和参考价值比机器分析和处理方法高；链接有时指向的是无法进行采集的对象，如图像、数据库等，此时可以根据链接文字对这些不能采集的对象进行索引。将链接文字与它所指向的页面联系起来的技术最早被 WWWW[4]所采用，其目的是帮助检索非文本信息，以及通过较少的文档采集来获得较大的索引覆盖面。Google 开始利用链接文字来进行相关性分析和提高结果质量。

3.5.4 *重要性分析* Web 上的文档的作者差异很大，他们编写 Web 文档的背景互不相同，关于同一主题而创作出的文档质量也是参差不齐，在 Web 上的影响和对用户信息需求的参考价值有大小之分。因此在对文档进行区分时，除了考虑相关性之外，还引入了文档重要性的概念。

在分析文档的重要性时，主要有两种方法：一是 Google 所提出的 Web 文档链接结构[20]分析的方法，称为 PageRank。Web 文档之间的相互引用（链接）情况是很有价值的信息，但是在以前的搜索引擎中没有被意识到而加以利用。类似于文献的引用关系，如果一篇 Web 文档被越多的文档引用，那么它的 PageRank（重要性）就越高。并不是所有的引用都同等重要，被 PageRank 越高的文档引用，被引用的文档的 PageRank 得到的好处越多。PageRank 是用文档被引用的重要性情况的客观度量来替代人的主观重要性评价。PageRank 的具体描述是这样的[22]：

假设有 n 个页面 $T_1, T_2, \cdots, T_n$ 指向（引用）页面 A。参数 d 可以在 0 到 1 之间取值，通常取为 0.85。C(A)为 A 中指向外边的链接数量。那么 A 的 PageRank 定义如下：

$$PR(A)=(1-d)+d(PR(T_1)/C(T_1)+\cdots+PR(T_n)/C(T_n))$$

PageRank 在网页上形成一个概率分布，所有网页的 PageRank 的和是 1。

PageRank 可以采用简单的递归算法来计算。从上面的公式可以看出，如果一个页面被很多的页面引用（链接），或是被一些 PageRank 很高的页面链接，那么它就具有很高的 PageRank。这是可以理解的：如果被广泛引用，那么说明该页面得到普遍的认可或很有参考价值，应该具有较高的 PageRank；如果被比较重要的页面引用，它也应该具有较高的 PageRank——一个重要的页面（比如 Yahoo 或其他影响较大的网站）上出现的内容链接当然是经过仔细挑选的。

他们还为这一技术作了浏览行为解释：PageRank 可以理解为一个用户行为模型——一个没有明确目的、对 Web 文档进行随机浏览的一个用户。为他提供一个随机的页面，他就开始点击上面的链接并继续下去，但不点击已经访问过的 URL，如果厌倦了这一次浏览，就给他一个新的随机页面，从这个页面开始作同样的浏览。那么，这个用户访问到一个页面的概率是该页面的 PageRank。上面 PageRank 的计算公式中的 d 因子是该用户在每个页面感到厌倦而请求一个新的开始页面的概率。

PageRank 技术现在已经被应用到 AltaVista、Excite、Fast、NorthernLight 等著名搜索引擎中。

第二种分析文档重要性的方法是认为：一个网站或网页如果被越多的人选择浏览，那么它的重要性就越大[23]。根据以前大量的检索行为中，用户选择访问的网站页面以及他们在这些页面上逗留的时间来确定这些网站和页面的重要性。HotBot、Lycos、WebSideStor、Yep.com 等搜索引擎也采用了这种技术。这种方法还可以用来作网站和关键词的相关度分析。

这两种方法都具有明显的趋众性，不过这是合理的。实际

效果说明这两种做法都是不错的。

3.5.5 查询处理 用户提交了查询条件之后，搜索引擎要进行查询处理。一般来说，查询处理包括如下过程：查询条件处理和转化、匹配、排序、返回。

尽管一般用户在进行查询时给出的查询条件都比较简单（平均 1.5 个单词），但是在进行匹配之前还是有必要转换成搜索引擎的内部表示形式。如 WebCrawler 在查询匹配时采用的是向量空间模型，因而必须将其查询条件转化为向量表达方式。匹配过程就是选择满足查询条件的结果的过程，一般来说就是选取查询条件对应的索引项下的文档。排序的依据主要有两个：文档和查询条件的相关度，文档的质量（重要性）。Google 是结合查询词的权值（相关性）和 PageRank（重要性）对查询结果进行排序。有些搜索引擎是将这些信息直接包含在索引当中，排序时就按照这些信息；有些搜索引擎则是要进一步考虑查询条件中各个查询词的重要性，如 WebCrawler 的向量表示匹配方式需要分析各个查询词的权值，这样，查询条件和文档的相关性是实时计算出来的。部分搜索引擎在将结果返回给用户之前会对每条结果给出一个得分评价，而这个评价标准也是各不相同、互不兼容，但是都是用来反映结果的可用性。

3.6 如何评价一个搜索引擎

3.6.1 从用户的角度看搜索引擎 从用户角度评价搜索引擎，主要依据如下几个方面：

· 最重要的，是搜索引擎的检索效率。用户对搜索引擎检索效率的要求概括起来是：搜索信息准确，搜索速度快，信息量大。根据调查，目前用户对搜索引擎感到最不满意的地方是：搜索速度慢，死链接太多、重复信息或不相关信息较多。

· 搜索引擎的用户接口。一个好的搜索引擎用户界面概括起来应该是：简洁——干净的界面总是能让用户集中注意力，而广告和繁杂的周边内容易于让用户感到反感；易用——用户能够很容易看出如何表达自己的检索需求，如怎样使用逻辑查询方法，以及一些该引擎的特色使用技巧；可定制性——搜索引擎是否提供了足够的功能选项让用户来进行设置，以用户偏好的方式来进行检索。

· 搜索引擎的反应时间。据调查，一般用户都不是很有耐心的，在查找信息时如果超过了一定的时间，一般就会放弃或另寻他径。搜索引擎要进行频繁的请求处理，因此除了要准备充足的网络带宽之外，还需要优良的软硬件环境和高效的搜索引擎软件。

· 检索结果的显示方式。用户不希望在一大堆结果里花很长的时间进行挑选，因此必须要把最好的结果放到最前面。有些搜索引擎还对结果的重要性给出了评分。此外，应该有关于检索结果的简单摘要或包含关键字部分的摘录。有的搜索引擎还会提供将结果按照网站或类别归类，以便于用户察看。

3.6.2 衡量检索效率的指标 衡量一个信息检索系统的检索效率的指标通常是查全率和查准率。它们分别体现的是检索行为发现的相关文档数量占总相关文档数量的比例和检索结果中相关文档的比率。

在对传统的信息检索系统进行评价时，一般是在一个已知的具体文档集上进行的，这个文档集中文档的相关情况是已知的。因此可以按照前面的公式得到一个具体的数值。

但在网络环境下，是完全不一样的情形。网络文档集是一个完全没有控制的异类文档的集合，文档的内在内容和潜在信息都有很大区别[22]。网络文档集的庞杂和动态性使得 R_{all} 是不可知的，因此没有办法计算查全率。同时，一次检索活动经常会返回大量结果，由于都是网络上的文档，要得到 R 需要很大的代价，因此计算查准率基本上也不现实。

既然不能以这种方式来评价搜索引擎的检索效果，那么就应该有比较适合网络环境的替代方法。事实上，查全率反映找回的相关文档的比率，在用户方面的效果是用户信息需求得到满足的程度；查准率反映检索结果中噪音的程度，在用户方面的效果是：用户需要花多长时间（检阅多少条结果记录）才能从中得到一定数量的有用结果。我们可以这样认为：搜索引擎如果能在最短的时间里让用户的信息需求得到最大的满足，那么它就是最好的。因此，我们也可以从两个方面来衡量搜索引擎的检索效率：能在多大程度上满足用户的信息需求；达到这一点用户花了多少功夫。但是要得到这些信息，需要用户提供意见。

单从查全率上考虑，文[5]提出了一个相对的方法：元搜索引擎找到的结果的数量与检索结果最多的一个搜索引擎的检索结果数量相比。实际情况下得出的数值是 2～5 倍。

3.7 主要搜索引擎评析

目前 Internet 上的搜索引擎为数极多，在此仅对一些有代表性的搜索引擎做出一些分析和评价：

3.7.1 Google 是斯坦福大学的 Sergey Brin 和 Lawrence Page 小组研发的搜索引擎模型[22]，致力于高扩展性、强处理能力和提供高质量的检索结果。Google 强调利用文档包含的信息（主要是超链接结构和超链文字）来改善检索结果。它根据这一想法提出来的特征技术是 PageRank。

Google 利用的另一个超链接属性是超链文字，将超链文字同被连接的文档联系起来。这有两个好处：超链文字往往提供了对被连接文档的很好描述；对于图片、程序等不能进行文本处理的文档可以据此进行简单处理。这个想法最初是 WWWW[4]提出来的。

Google 还包括一些其他特点：利用文档中单词位置的亲近关系；对文档中的字体进行区别；保留原始文档。

正因为 Google 采用了这些比较先进的技术，它的检索结果具有较高的质量。现在 Google 已经成为一家很有影响的搜索引擎服务提供商，Yahoo! 采用的就是 Google 的搜索引擎。

Google（http://www.google.com）的一些主要使用特点包括：

Google 允许用户进行以下设定以供每次检索时使用：如界面指令语言、查询语言、每页显示结果数目、是否在新窗口中察看结果。

Google 的高级检索功能可以支持逻辑语法（与、或、非），可以进行短语和句子查询，可以指定检索词出现的位置，指定在什么网域（如 .org）内进行查询，搜索和某个网页类似的网页，对某个网页进行链接的网页。

Google 保留了原始页面以备用户需要，这对于对时间不敏感的检索条件往往很有作用，能产生检索结果的简单摘要。

3.7.2 百度（http://www.baidu.com）搜索引擎 是国内的搜索引擎服务提供商，同 Google 一样为多家网站提供搜索引擎服务。目前采用百度搜索引擎系统的有：硅谷动力（http://www.enet.com.cn），Chinaren（http://www.chinaren.com），搜狐（http://www.sohu.com），21cn（http://www.21cn.com），新浪（http://www.sina.com.cn），广州视窗（http://www.gznet.com），263 在线（http://www.263.net），赛迪网（http://itsearch.ccidnet.com），Tom（http:

//www.tom.com)，大千网(http://www.sznews.com)，清华大学(http://www.tsinghua.edu.cn)。百度公司的创始人李彦宏博士是ESP技术(Extra Search Precision，专门优化常用词检索的技术)的提出者，并对网页的排序问题提出网页质量和相关性相结合的方法。百度搜索引擎具有较高的技术含量，不错的检索效率，在PC Computing的搜索评测中在网页搜索方面领先。

百度搜索引擎采用了中文语言处理技术，可以较好地处理中文检索条件，如百度为Chinaren推出的"孙悟空搜索"可以用自然语言进行查询。百度建立索引是同时基于字和词的，有效地处理了汉语的独特性。

百度的spider程序覆盖面比较广，基本包括了世界上的华语地区。目前百度已经拥有了世界上最大的中文信息库，包括了4500多万个页面，而且在以每天超过20万页的速度增长。

百度对检索结果的排序是基于网页内容分析和链接分析相结合的。为了缩短系统反应时间百度采用了多线程检索方法。

3.7.3 Inktomi 是另一家搜索引擎服务提供商，它的客户包括：美国在线(http://www.aol.com)，微软网站(http://www.msn.com)，iwon(http://www.iwon.com)，Wired Ventures(http://www.hotbot.com)，Looksmart(http://www.looksmart.com)，Goto.com(http://www.goto.com)，NBCI(http://www.nbci.com)等著名网站。

Inktomi提供的搜索服务具有多元性：包括目录服务、多媒体检索、新闻检索。Inktomi拥有比较先进的内容采集系统。搜索引擎运行时，首先对超过10亿个URL进行处理，分析每个页面关键特征和对应的链接结构，创建一个WebMap数据库。

Inktomi的相关性确定技术综合利用了文档文本、来源、相关链接和一些其他特征，而且这个相关性模型不断地被调整以确保比较接近人的判断。同时有一个相关性监测模块以提供相关性反馈。

Inktomi支持逻辑查询语法，还能指定关键词出现在文档中的位置：标题、文档中、站点名称、URL当中等等。检索时能限制文档的最后更新日期，以及每页显示多少文档，按相关性还是日期或是标题进行排序。

3.7.4 InfoSeek 提供多种检索，其建立的索引包括：Web页面、Usenet新闻、新闻在线(News Wires)、邮件地址、公司、FAQs以及图像检索。InfoSeek支持两种逻辑语法：+保证查询词一定包含在结果里，而"-"则是查询词不出现在结果里。支持短语查询。

3.7.5 AltaVista 致力于建立最完整和具有时效性的索引，数据库规模庞大。其建立索引的内容主要包括Web文档以及Usenet新闻组。

AltaVista支持各种逻辑查询语法。+、-分别表示查询词必须出现和不出现在结果中，另外还可使用AND、OR、NOT、NEAR等查询关系词，还支持带权值检索关键词。可以进行短语检索，还可使用通配符*。

3.7.6 Excite 建立索引的内容包括Web页面，Usenet过去两周的新闻、评论以及分类广告。

Excite支持各种逻辑检索语法：AND、AND NOT、OR、+、-以及括号，重复的关键词则加强其重要性。在检索时能自动检索同义词以及相关词。Excite的检索页面支持用户的定制，用户可以设置自己喜欢或经常访问的内容。

3.7.7 Lycos 可以检索的内容包括：Web页面、分类目录、声音文件、图像。它对逻辑查询语法的支持比较弱。但是能自动检索诸如单词复数等变体形式。可以使用通配符(在单词的右边使用$)。

3.7.8 WebCrawler 始创于1994年，是第一个大规模建立索引进行全文检索的Web搜索引擎。它采用宽度优先算法对Web进行遍历以采集文档，然后采用向量空间模型对文档的标题和内容都建立索引。

WebCrawler支持各种逻辑查询语法，也能进行短语查询。WebCrawler还提供一个桌面用户软件，它可以在用户自己的机器上运行，按照用户的要求自动进行Web浏览，取回对用户有用的文档。

3.7.9 天网 此搜索引擎是北大计算机系研发的，能进行中英文检索，特点是检索处理效率高，索引更新速度快。但是检索的准确率欠佳。天网还提供了FTP检索服务。

3.7.10 悠游 此搜索引擎是第一个智能中文搜索引擎：结合人工智能技术，提供自然语言查询接口。收录的网站资源比较丰富，能进行类目搜索、网站检索、全文检索以及站内新闻搜索。悠游具有较好的查询结果查准率，能转换国标码和大五码，还提供个人书签服务。

3.7.11 搜索客 提供类目检索和全文检索功能，还能进行相关新闻检索。分类目录比较细致，但是收录站点不够丰富，查全率和查准率一般。它提供了一个好站目录，列出一些值得推荐的站点。

3.8 搜索引擎的发展趋势

搜索引擎自产生以来不断地进步和发展，最近几年Internet的高速发展更是大大刺激了基于网络的各种应用的发展。搜索引擎对提高用户访问量的巨大作用早已为各大网络公司所知晓，因此，在最近的几年里，搜索引擎得到了飞速发展。总体上看来，搜索引擎的发展呈现出如下趋势：

·搜索引擎开发的集中和专业化。当搜索引擎刚开始兴起的时候，由于它能为用户提供需求最广泛的信息检索服务，又有良好的发展前景，因此出现了不少的搜索引擎开发者。互联网在最近几年的发展如同雨后春笋，出现了不少新兴事物和业务，以至于人们如此地看好互联网，导致出现了所谓的网络经济。在这阵热潮中，网站尤其是门户网站被罩上了五彩光环，大量的资金和人力被吸引到网站的发展上。很多搜索引擎开发公司随风而动，认为自己的搜索引擎能吸引大量用户，是创办门户网站得天独厚的条件。但是，事实上，开发和维护一个大型的搜索引擎系统并非一件简单的事情，需要投入大量的人力物力财力。现在，这些投入不再是以搜索引擎自身为中心进行运作，而是作为网站的一个补充。网络经济最终没有发展得如同人们想象的那样美好，因而也宣告了这些转型的搜索引擎公司的抉择失败。不过，人们从这个过程中还是得到了一些启示：其一就是在网络大潮中处在最高位置的Yahoo，它虽然是以搜索服务而闻名，但并没有自己进行搜索引擎的开发，而是租用别人提供的服务(原来是Inktomi，现在是Google)。其二就是坚守搜索引擎阵地的搜索引擎公司，也取得了巨大的成功，如Inktomi。这种租赁服务和提供服务的运作模式正是代表了互联网商务的发展方向。因此，现在搜索引擎的开发都集中在一些专业的搜索引擎公司，他们向别的网络公司出售搜索服务系统。著名的搜索服务提供公司有Inktomi、Google，而国内的百度也取得了很大的成功。另一方面，

网站也不再自主开发自己的搜索引擎系统，而是购买或租赁别人的服务。

·索引数据库规模的膨胀。数据库的覆盖面一直是各个搜索引擎追求的方向，尤其是门户网站的搜索引擎，每个搜索引擎在介绍自己时总会吹嘘自己为多少万、多少亿个页面建立了索引，力图覆盖所有的网络资源。

·搜索质量的逐步提高。搜索引擎除了努力扩大索引数据库的规模之外，还在极力改善检索结果的质量，而且的确取得了可喜的进步，这从 Google、Altavista 推出的 Raging Search、TEOMA 等搜索引擎可以看得出来。

·检索的多元化。现在的搜索引擎一般都提供多种类型的检索，如目录、网站、网页、新闻。走在更前面的则提供软件、音乐、游戏、商品等检索。

·检索的专业化：垂直搜索引擎。相对于追求"广博"的搜索引擎来说，这是一类求"精专"的搜索引擎。大量的用户是因为一个突发的原因而查找某条信息，对于他们来说并没有固定的检索方向，覆盖面广的搜索引擎对他们很有好处；但是，有不少的用户是另外一种类型，他们是"专业用户"，需要经常查询某个领域的信息。对于这类用户来说，搜索引擎的广博并不是至关重要的，他们需要的是一个尽量囊括他们所有的专业知识及相关信息的一个搜索引擎。这就是垂直主题搜索引擎或专业领域搜索引擎。例如专门提供医药信息检索、娱乐信息检索的搜索引擎。

·消除地域限制。现在的大型搜索引擎都尽量消除语言上的限制，提供多种语言接口，选择查询多种语言的结果。

·检索的用户定制特性。搜索引擎受到的很大一个限制就是不能提供充分的个性化服务。但是，现在不少搜索引擎系统正在朝这个方向努力，提供用户定制设置，包括一些常用的检索选项以及用户比较感兴趣的站点或常浏览的信息类型。

3.9 搜索引擎需要解决的问题和研究方向

搜索引擎是目前最重要的 Web 检索工具，自产生以来已经有了很大的发展，信息检索方面的研究不断为搜索引擎引入新的技术。但是目前的搜索引擎还远没有达到尽如人意的地步，有很多地方需要改善，总体说来包括用户界面、检索效果、处理效率。

3.9.1 *用户查询接口* 关于用户查询接口，各种搜索引擎给出了不同的实现方式。目前主要方式有以下几种：

·逻辑查询语法结合各种限制选项。一般来说，大多数搜索引擎提供了逻辑与、或、非的检索语法。但是通常只是简单的，而不能是任意复杂的逻辑表达式。因而这种查询表达方式有其限制性，而且，用户需要了解其逻辑语法表达方法（有的是用 and、or、not，而有的则是用＋、|、－等），而且要知道怎么将自己的要求表达成这种表述方式。而附加的各种限制选项通常是用来进一步限制检索范围：在某个领域内检索，检索某类语言的文档，检索在某个日期后更新的文档等等。由于可以提供各种选项，为了避免用户操作的复杂性，有些搜索引擎允许用户将这些选项作为自己的偏好设置加以保留供以后检索时使用，比如 Google。

·查询扩展和精化。主要有两种方式：一是通过一个词典[24]，二是通过用户的意见（相关反馈）[25]。

·逐次检索方式：允许用户对检索到的结果进行检索，一步步精化得到的内容。这种方式实际上是逻辑语法的一种实现方式，但是相对来说具有较好的用户友好性——对于不善于或不愿意使用逻辑查询语法的用户可以逐次得到要检索的结果。但它是以时间、网络带宽和计算资源作为代价的。

·自然语言查询接口，比如 Ask Jeeves，Chinaren 的孙悟空搜索。显然，现阶段的搜索引擎不可能实现真正的自然语言处理；它们的方法是：将自然语言提问进行分词处理，提取出关键词，然后进行逻辑与查询。尽管当前的技术并不能完全确切地理解所有提问方式，在效果上也可能存在偏差，自然语言提问查询的方法的确是提供了一个更为友好的用户界面。

总的说来，用户查询接口的发展趋势是让用户使用越来越方便，能精确地表达出很复杂的查询请求。从用户友好性的角度看，自然语言方式是最值得发展的一种方式，但目前的理解能力达不到理想的精度。人们希望最终能使用自然语言来查询，但这依赖于自然语言理解研究的发展。

3.9.2 *检索效果* 搜索引擎的检索效果主要从两个方面来考虑：查全率和查准率。为了提高查全率，必须要提高搜索引擎的数据库容量，对尽可能多的 Web 文档建立索引。为了做到这一点，搜索引擎一般是采用比较多的 spider 线程在 Web 上并行采集文档。但是由于 Internet 的飞速发展和瞬息万变的动态特征，单个搜索引擎的计算资源是不可能覆盖所有 Web 文档和保证其适时性的。我们可以想到利用分布式的数据库，在多个数据库中同时进行检索以提高查全率。事实上，元搜索引擎要做的就是这个工作。当然，对异地分布式的数据库进行检索需要付出额外的时间和计算代价，而这又涉及到搜索引擎的处理效率，这是一个需要权衡的问题。

查准率相对于查全率而言，显得更为重要。为了提高查准率，搜索引擎提出了各种处理方法，利用了和文档相关的各种信息：文档的来源，文档被链接的信息，文档的对别的文档的链接情况，文档的 HTML META TAG，标题以及乃至文档中的字体设置，文档的内容。目前优秀的搜索引擎的检索效果具有一定的质量，但有过检索经验的人都知道，一个搜索引擎对查询请求通常返回成千上万个检索结果，很显然，大部分的检索结果是没什么价值的；而且位置在前面的结果也有相当一部分是不相关的。这反映的是：搜索引擎的相关性确定算法还有待完善。

3.9.3 *处理效率* Web 搜索引擎是用户使用相当频繁的 Internet 应用，1994 年的 WWWW 平均每天要处理 1500 个查询请求，而 1997 年的 AltaVista 称其平均每天要处理两千万个检索请求。随着 Web 的膨胀和上网用户的普及，目前的主流搜索引擎的日处理检索请求量很可能是上亿的。如何处理如此繁重的任务，是一个优秀的搜索引擎必须要考虑的问题。

计算机技术的发展使现在的硬件具有越来越优异的性能，硬盘的存储能力有了很大提高，CPU 的计算能力和主存容量的提升也为繁重的并发处理提供了更好的支持，但我们必须看到，Web 的发展是急速的。这对搜索引擎的扩展性以及在短时间内处理高度并发请求的能力提出了挑战。

搜索引擎必须设计高效的存储结构，节省硬盘和主存空间，同时有利于程序的运行效率。搜索引擎为了提高运行效率、缩短请求反应时间可以采取的措施包括：查询请求处理缓存，建立子索引，磁盘读取和网络文件系统优化。

4. 元搜索引擎

4.1 元搜索引擎介绍

不同的搜索引擎的索引数据库包含不同的 Web 文档集，而且采用了不同的索引和检索技术，因而具有不同的检索效

果，在一个搜索引擎上找不到的东西却可能在另一个搜索引擎上找到。但通常单个搜索引擎能找到的相关信息不超过所有相关信息的45%。因此人们常常需要切换多个搜索引擎进行检索，为此也需要了解多个搜索引擎，在多个搜索引擎得到的检索结果中挑选需要的内容，这对用户而言是很不方便的。因此，如果有一个工具可以替代这些手工操作将会是很大的改善。

Web上出现过几种类型的以单一界面提供多个搜索引擎的检索服务的工具。

4.1.1 All In One Page 最简单的一种是在一个Web页面内列出多个搜索引擎的检索入口，最著名的就是All In One(http://www.allonesearch.com/all1srch.html)。这种工具实际上并没有自己的处理过程，只是集成了别的搜索引擎的检索界面。用户只能一次从中选择一个搜索引擎输入查询条件进行查询，唯一的好处是可以不用记住单个搜索引擎的地址。

这种方式的一个变种是一种看起来要更简洁一些的界面：一个地方输入查询条件，然后在一个下拉选择框里选择一个搜索引擎进行检索。在国内的一些小网站上经常有这种为用户提供服务的搜索引擎入口。

4.1.2 元搜索引擎 是一个同时对多个搜索引擎进行检索的工具。它得到用户的查询请求，将其转化为若干个搜索引擎的检索请求，然后同时对这多个搜索引擎进行检索，对得到的检索结果进行处理，将最后的结果返回给用户。

一个真正意义上的元搜索引擎对多个搜索引擎的搜索必须是并行的，而且应该对得到的结果进行信息融合。

4.1.3 独立元搜索软件 这是一类桌面型搜索软件[26,27]，同时对多个搜索引擎进行检索。可以理解为将元搜索引擎的功能直接实现到用户软件里，而不是采用B/S的功能结构。相应地，独立元搜索软件也有不同的处理方式：

· 在一个文件夹窗口里整合多个窗口页面，每个页面显示一个搜索引擎的检索结果。事实上，这种程序有点像All In One Page的搜索引擎列表，不过省去了用户的一些操作，察看也方便一些，但是并没有对多个搜索引擎的检索结果做进一步的分析处理。

· 同时对多个搜索引擎进行搜索，并对结果进行整合。

4.1.4 几类工具的优劣 毫无疑问，All In One Page式的搜索引擎列表对用户没有太大的帮助，除了可以让用户知道他可以使用哪些搜索引擎。

没有对结果进行融合处理的元搜索引擎和独立元搜索软件相对来说更进一步简化了用户切换多个搜索引擎进行检索的操作，但是，它们给出的是多个检索结果列表，用户需要花不少时间来从中查找需要的东西。

对搜索引擎的检索结果进行融合处理的元搜索引擎和独立元搜索软件才是我们想要的。它们有效地提高了单个搜索引擎的查全率，简化了用户操作，同时权衡了多个搜索引擎的结果信息，得出的是单一的融合的信息列表，排序更具有客观性。那么元搜索引擎和独立元搜索软件有什么区别，孰优孰劣呢？其实从可以采用的技术和可以达到的检索效果来说，两者是一样的。区别是在于：元搜索引擎是采用B/S架构，通过浏览器来对元搜索引擎服务器进行查询访问；而独立元搜索软件所有的处理都是在用户端进行的。毫无疑问，元搜索引擎的B/S结构会占用一些访问时间，结果相应会要慢一些。当然也不是绝对的，比如用户到元搜索引擎服务器和元搜索引擎服务器到别的搜索引擎网速都很快，而用户到某个搜索引擎的网速很慢的情况下。但是元搜索引擎也具有一些优点：网络的动态性很大，各个搜索引擎网站为了吸引更多的用户，经常进行升级改版，因而元搜索应用(元搜索引擎和元搜索软件)原来能识别和处理的搜索引擎变得不能处理了，因而必须进行一些组件的更新。元搜索引擎因为所有的处理都集中在元搜索引擎服务器上，因而用户不用为这种频繁的升级操心；而且现在的各种应用软件繁多，对太多的软件的下载、安装和管理是一个令用户头疼的问题。此外，在目前的条件下，独立元搜索软件会给网络带来巨大的带宽负担：少数的Robot对网络进行遍历和索引没有什么，但如果大量用户都使用Robot软件，网络将会瘫痪。

4.2 元搜索引擎实现原理

图1是元搜索引擎实现原理的简单示意图。用户通过元搜索引擎提供的检索界面向元搜索引擎提出一个检索请求，元搜索引擎选择一些搜索引擎进行检索，转化用户的查询条件并发送给这些搜索引擎，被检索的搜索引擎向元搜索引擎返回查询结果，元搜索引擎对这些检索结果进行融合整理之后返回给用户。

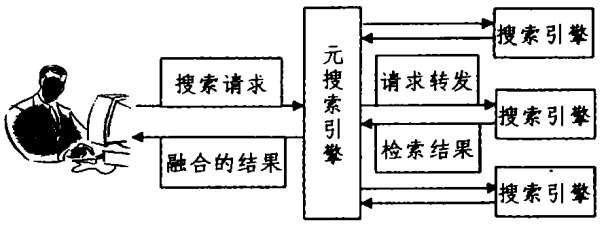

图1

4.3 元搜索引擎核心问题

元搜索引擎和搜索引擎的工作模式完全不同，它不是建立本地索引数据库，而是通过检索别的搜索引擎并处理其结果。从上述元搜索引擎的工作原理可以看出，元搜索引擎要处理的问题主要在于：

· 选择搜索引擎。Internet上的搜索引擎多如牛毛，元搜索引擎该选择哪些搜索引擎进行检索呢？实际上，(普通)搜索引擎相当于元搜索引擎的索引数据库，选择好的搜索引擎，就能获得高质量的原始检索结果，从而提高最终检索结果的质量。使搜索引擎的选择比较困难的几个因素[28]是：Web是由别的搜索引擎建立索引的，文档集不可得；这些搜索引擎既有通用搜索引擎也有专业搜索引擎，其专门技术变化很大；这些搜索引擎经常发生变化，尤其是其索引；作为Internet的一员，元搜索引擎应该在搜索结果质量和网络资源消耗之间作一个权衡。

· 检索条件转化。普通的搜索引擎是直接对用户的查询条件进行处理，而元搜索引擎在将它接收到的检索请求发送给别的搜索引擎之前必须将其分别转化成各个搜索引擎能处理的格式，因为别的搜索引擎接受的请求中变量参数设置不同。在此可以考虑针对不同的搜索引擎进行请求扩充处理。

· 结果融合。这是元搜索引擎最核心的问题，一个元搜索引擎的性能很大程度上是由这一部分决定的。每个搜索引擎都会向元搜索引擎返回一个检索结果列表，它们检索的标准和排序算法各不相同，如何将所有搜索引擎的反馈结果整合成一个单一的结果列表，客观地综合参考各搜索引擎的相关性评价，在最后结果中精确地体现相关性和重要性，是非常复

杂和值得研究的问题。

4.4 元搜索引擎关键组件

为了实现一个真正意义上的元搜索引擎，必须实现以下功能模块：

· 检索接口。一般也是 Web HTML 页面，但是元搜索引擎除了提供元搜索引擎检索接口的普通选项之外，还可以提供元搜索引擎特有的一些选项。结果的显示方式也应该有元搜索引擎的特点。

· 搜索引擎选择模块。有很多元搜索引擎忽视了这一过程，实际上，搜索引擎的选择很重要，直接影响到最后结果的质量。Internet 上的各种搜索引擎数量繁多，如何有效地选择高质量的引擎，尚有待研究。

· 检索模块。在选择了搜索引擎之后，检索模块负责将检索请求转化为这些搜索引擎能处理的请求格式，并将其发送给搜索引擎，随后获取这些搜索引擎的检索结果。检索模块中值得考虑的是检索条件的转化及扩充处理，以及获取结果的数量。一个高效的元搜索引擎的检索过程必须是并行的。

· 融合模块。融合模块要进行元搜索引擎的核心处理：检索模块从搜索引擎获取到检索结果后，融合模块负责将其融合为一个结果列表。这儿有很多方法可以考虑和进一步研究，而不是简单的顺序排列。

4.5 元搜索引擎技术

元搜索引擎的主要作用是对别的搜索引擎的检索结果作进一步处理。它没有自己的 Web 文档索引数据库，其信息来源是别的搜索引擎的结果输出。从元搜索引擎的工作过程来看，元搜索引擎的关键技术主要集中在对搜索引擎的选取、信息融合过程上。

4.5.1 *搜索引擎选择* Internet 上的搜索引擎为数不少，它们各有各的特色。就检索效果来说各不相同，每个搜索引擎都有自己性能比较突出的地方。因此，元搜索引擎的设计者已经想到，应该根据用户的查询条件的领域，来选择在这些领域内检索效果较好的搜索引擎来工作，尤其是专业搜索引擎，它们很适合专业领域。

但是在目前的元搜索引擎中，很少有充分意识到这一点的重要性的。有不少根本没有进行搜索引擎的选择，每次检索都是固定的几个搜索引擎，或者是随机选择。搜索引擎的选择算法上，比较有特色的是 SavvySearch 和 ProFusion。

SavvySearch 的做法是基于检索词来分析每个搜索引擎的检索效果。它通过一个 $t \times s$ 的矩阵保留每个搜索引擎关于其被检索的关键词的效率，t 是关键词的个数，s 是搜索引擎的个数。矩阵中每个元素 $E_{i,j}$（搜索引擎 j 关于关键词 i 的检索效率）的值是 j 在以 i 作关键词的历次检索活动中效率的积累。而一次检索活动中 $E_{i,j}$ 的更新是因为两种情况：一是 j 没有检索到和 i 相关的结果，此时 j 被认为不适合对 i 作检索，其 $E_{(i,j)}$ 会被削减；二是 j 检索出来的结果被用户选择浏览，此时认为 j 找到了和 i 相关的结果，其 $E_{i,j}$ 会增加。当用户提交包含 i 的查询的时候，$E_{i,j}$ 的排序情况说明了不同的搜索引擎关于 i 的检索效率的优劣。此外，由于搜索引擎经常有变动或发生故障，SavvySearch 还会分析最近几次的检索活动，看搜索引擎的返回信息量和反应时间是否正常，如果超出正常范围之外，其 $E_{i,j}$ 会被削减。

一个检索如果包含了多个检索词，那么每个检索词在矩阵中的效率值都会被计算出来。然后，搜索引擎就要按这些效率值进行排序。而关于排序的算法，SavvySearch 作了一个很有意思的引用：将传统信息获取中的 tf * idf 方法作了对应的替换，然后用其计算查询词和搜索引擎的相关性，所有查询词和搜索引擎的相关性的加权就是该查询条件和搜索引擎的相关性。

相关性排序确定之后，就是选择多少个搜索引擎进行查询的问题了。SavvySearch 主要参考三个条件来确定这一数量：一是当时的网络状况，这是通过分析 SavvySearch 服务器的日志纪录的以前几天的对应时刻的情况决定的；二是服务器的运行负载，这是通过运行系统命令得到的；三是查询条件的特殊性，如果查询的是一个比较普遍的、很多引擎都有内容的主题，那么可以用较少的搜索引擎检索，如果查询的是比较偏僻的主题，那么就应该启用较多的搜索引擎。

相对于 SavvySearch 的基于单个词进行分析的方法，ProFusion 的策略就比较抽象。它采取的是基于领域进行分析的方法：首先为 Internet 上的内容建立一个分类目录树，为了使这个分类目录和人们的兴趣相对应，ProFusion 是对新闻组的分类结构进行分析，然后结合经验建立分类目录。其次，对于这个目录的每一个子类，ProFusion 为其选择一些代表性的词和术语，这些词的来源也是新闻组。然后根据对搜索引擎以往检索效率的分析，为每个子类确定搜索引擎的效率值。当用户提交查询请求后，如果选择的是让系统自动选择搜索引擎，那么系统就会先分析每个查询词可能会属于哪些子类，每个搜索引擎关于这些子类的检索效率是多少；整个查询涉及到多少子类，每个搜索引擎在这些子类上的效率值的加权是多少；然后按这个加权对搜索引擎进行排序，选择排在最前面的三个搜索引擎进行检索。

4.5.2 *信息融合* 从各个搜索引擎返回的检索结果需要经过融合[29]才能反馈给用户。这个融合过程应该包含以下程序：删除重复，将结果合并到一个完整的列表中，按相关性排序。

由于不同的搜索引擎的索引数据库在一定程度上互相重叠，因此检索时会有一些重复结果，元搜索引擎必须尽量消除重复。一般说来，会有如下几种情况：最简单的重复情况是结果具有相同的 URL，可以很容易地排除。其次是同一文档存在常见的别名，如 http://xxx.xxx.xxx 和 http://xxx.xxx.xxx/index.htm 指向同一个文档页面。再次是同一文档被做了链接，因而具有差异比较大的别名：http://xxx.xxx.xxx/patha/namea.htm 与 http://xxx.xxx.xxx/pathb/nameb.htm 指的是同一个页面。第四种情况是重定向：本来是以 http://xxx.xxx.xxx/patha/namea.htm 发送的请求，被转移到 http://yyy.yyy.yyy/pathb/nameb.htm。第五种情况是同一文档具有不同的拷贝或版本，放在不同的位置，此时存放的主机也可能不相同。这种情况是最难以识别的。在现有的元搜索引擎中，MetaCrawler 和 ProFusion 为消除重复设计了比较好的算法。当然，想完全消除重复太困难了，除非是元搜索引擎亲自下载所有的结果页面进行分析，需要付出高昂的代价。

检索结果的排序需要综合考虑其在各个搜索引擎中的出现情况，原则上来说，结果出现在越多的搜索引擎里，它的价值越高；在单个搜索引擎中它的位置越靠前，它的价值越高；在检索效果越好的搜索引擎中出现，亦说明其参考价值越大。一般来说，现在的元搜索引擎在排序时都是采用不同的投票法（不对结果进行融合排序的除外）。最简单的就是看包含一条结果的搜索引擎的个数，如近来成绩不错的 ixquick。其次，

在考虑这个"票数"的同时，可以考虑每一票的分量——一条结果在一个搜索引擎的结果列表中的地位。这有两个指标：一是排列位置，二是相对精确的搜索引擎给出的得分评价[30]（并不是所有的搜索引擎都会给出评价）。如 Fusion 的融合就是基于各个引擎结果序列的排列位置；而 highway 61、Mamma、MetaCrawler 和 ProFusion 等的融合是基于各个搜索引擎对其结果给出的得分评价。如果考虑搜索引擎本身的效率，在计算时应该调整搜索引擎给出的得分评价。

文[31]中对不同的融合方法进行了比较，最后发现基于权值得分的融合方法只需要较小的计算代价，但是效果却很好。

为了更好地进行融合，关于元搜索引擎的一个共同的想法是：元搜索引擎亲自下载对找到的结果进行分析处理。这样做的好处是可以直接对文档的相关性和可用性做出判断，以及可以借此提供一些新的功能，如可以检查结果链接的存在，这一点对于检索结果的质量也是很重要的。

4.6 元搜索引擎系统评析

在用户看来，元搜索引擎和普通搜索引擎没有什么区别；但是，一个真正意义上的元搜索引擎必须具有一系列特征。在文[5]中，作者列出了他认为元搜索引擎应该具有的条件：

1)对所调用的搜索引擎的检索必须是并行的。

2)从不同搜索引擎得到的结果必须进行融合，而不是简单的顺序排列。

3)不同的搜索引擎找到的重复结果必须被删除。

4)最低程度要支持逻辑"与"和"或"查询。

5)如果被调用的搜索引擎为结果提供了描述或摘要，元搜索引擎也必须提供，而且内容不得少于被调用的搜索引擎的。

6)不让用户操心所调用的搜索引擎的细节情况。

7)元搜索引擎应该能获取所调用的搜索引擎检索出来的所有结果。

下面是按照这几个标准对一些主要的元搜索引擎进行考查的结果（表 1 中的 1、2、…、7 对应于上述条件，x 表示不满足，y 表示满足，p 表示部分满足）。

表 1

元搜索引擎	1	2	3	4	5	6	7
metasearch. com	x	x	x	x	x	x	x
http: //www. digiway. com/digisearch/	y	x	x	y	y	x	x
search. onramp. net	y	y	y	x	x	y	x
http: // www. designlab. ukans. edu/profusion/	y	y	y	y	y	x	x
search. cyber411. com	y	x	x	x	x	y	x
search. metafind. com	y	y	y	y	x	p	x
http: //www. inference. com/infind/	y	p	y	y	x	x	x
http: //www. dogpile. com/	y	x	x	y	x	y	x
http: //www. mamma. com/	y	y	x	y	y	y	x
guaraldi. cs. colostate. edu: 2000/form	y	x	x	y	y	y	x
http: //www. metacrawler. com/	y	y	y	y	y	y	n
mesa. rrzn. uni-hannover. de	y	y	y	y	n	y	n
meta. rrzn. uni-hannover. de	y	y	y	y	y	y	y
http: //www. highway61. com/	y	y	y	y	y	y	y

4.6.1 MetaCrawler[32] 是 1995 年 6 月在华盛顿大学开设的一个元搜索引擎，应该是最早的一个元搜索引擎了。MetaCrawler 支持逻辑查询、短语查询。对于一些普通搜索引擎不支持的功能，比如说短语查询等，MetaCrawler 采取的方法是直接将搜索引擎检索到的页面下载下来进行处理，还可以借此实现新的功能。

MetaCrawler 比较有特色的是它去除重复结果的算法：比较两个 URL 时，它先比较两个域名是否是同一个；然后检查路径名是否同一路径的别名；最后，如果域名是同一个而路径不同，就查看文档标题，如果相同则认为是别名，但是不会删除这些重复纪录，而是会将其 URL 列在后面。

MetaCrawler 对结果进行排序的方法是基于文档得分的投票方法：同一文档在不同的搜索引擎检索结果中的得分被累加起来。另外，MetaCrawler 还提供了按地理位置进行查看的选项。MetaCrawler 还提供死链接检查。

4.6.2 MetaGer 其特点在于它的元搜索引擎结合了一个本地数据库，被其称为 QuickTips，而本地数据库的内容包括两部分：一是 MetaGer 的开发和维护者手工维护的数据，二是早已存在于 Internet 上的分布式数据库：DNS 系统。当用户对他们的元搜索引擎进行检索的时候，MetaGer 会利用这些数据。之所以会利用 DNS 系统，使他们通过分析日志文件发现不少生手用户在查找的信息就是域名的一部分——尤其是当查询一个公司的时候。

另外，MetaGer 的开发者致力于开发一个自动工具，为有经验的专业用户提供服务。这些用户对他们的领域内的知识很熟悉，在检索时也需要搜索引擎提供专业内容，但这不是普通搜索引擎能做到的。MetaGer 的想法是根据这些用户输入的检索关键词来自动产生专业性的搜索引擎。他们把这称为三阶搜索引擎（元搜索引擎为二阶）。

MetaGer 依靠两种方法来保证其检索结果的质量，一是它所采用的 QuickTips，二是在对结果排序时除了参考底层搜索引擎的排序之外还依赖于文档标题、URL 及摘要里出现的关键词个数。在显示结果时用颜色来区分重要性。

4.6.3 ProFusion[33] 是肯萨斯大学电子工程与计算机科学系的研究项目。ProFusion 提供了 6 个搜索引擎，用户可以从中进行选择。也可以由系统分析查询的主题，然后选择 3 个进行查询。它的特色是它能根据对用户查询关键词的分析，找出要查询的主题，然后选择最合适的搜索引擎进行检索。ProFusion 实现这一技术的方法是：创建一个反映 Web 内容的分类目录；将术语词汇联系到目录主题；将搜索引擎联系到它比较擅长的主题。

ProFusion 消除重复结果的算法有点类似于 MetaCrawler：它将一个 URL 分为三部分：协议、主机和路径名，然后用 N-gram[34]检查其相似性，如果足够相似，则认为是重复的。为了减少这种方法的失误，ProFusion 下载相似网页的部分内容进行比较，但没有在实际运行版本中实现。ProFusion 消除重复的方法取得了很好的效果。

ProFusion 的融合是基于权值得分进行的，主要考虑两个因素：一是搜索引擎给出的检索结果和检索关键词的匹配情况，另一个是对搜索引擎的评价的准确程度的估计。这个估计称为搜索引擎的可信度因子（Confidence Factor，CF）。搜索引擎的 CF 是通过对其进行的一些检索结果的质量做出评价得到的，反映了前 10 个结果中相关结果及其排序情况。一条结果的权值是其相关度（由被调用的搜索引擎决定）与搜索引擎的 CF 的乘积。有重复时取权值最大的一个。如果是通过系统选择搜索引擎，这个结果将再乘以搜索引擎在该检索主题里的评价。

4.6.4 Fusion[3] 是一个相对简单的元搜索引擎。它所注重的是其C/S系统结构，区别于别的元搜索引擎的地方是它是从浏览器里启动一个Java Applet窗口，以这个窗口作为和用户交互的界面，而浏览器则用来显示被挑选查看的结果页面。但是，除了它采用了不一样的系统结构外，它所宣称要实现的用户相关反馈和查询扩展等技术却没有描述。实际上，这种采用Java客户端的方法并不被人们所看好[5]。

Fusion有一个多线程服务器，它接受客户端(Fusion的Java Applet用户接口)发送的请求。每当接受到一个连接请求，它就启动一个新的线程。当用户提交一个查询请求后，服务器启动一个数据融合线程，对各个搜索引擎进行并行检索，对检索结果进行分析、融合，将结果送回Applet用户接口，用户选择结果在浏览器中查看。

Fusion的融合操作是基于各个搜索引擎检索结果的排列位置的。但是它所调用的普通搜索引擎只是有限的那么几个：AltaVista，Excite，InfoSeek，Lycos，OpenText和WebCrawler，而且它只是从每个搜索引擎取回最前面的10条检索结果。

4.6.5 SavvySearch[28] 主要考虑对底层搜索引擎的选择，其方法是：建立一个元索引(Meta-Index)记录过去检索活动中搜索引擎选择的成功或失败经验；搜索引擎被按元索引中的信息以及最近的检索性能数据进行排序；同时启用的搜索引擎的数目(资源消耗)取决于系统状态和当前网络状况。

SavvySearch的另一个特点是对用户的检索给出建议，即其所谓的检索规划(Search Plan)，将可用的搜索引擎分为不同的层次，分别对应于不同的检索效果和检索代价，用户可以从中进行选择。

SavvySearch缺省情况下不对检索结果进行融合。如果用户选择了融合，那么融合是基于得分评价进行的(标准化为0到1.0之间)；如果搜索引擎没有给出得分评价，那么它的结果将会被给予0.5分的评价。

4.7 元搜索引擎需要进一步研究的问题

元搜索引擎是一种方便的检索工具，让用户通过单一接口能检索多个搜索引擎，并得到融合的结果。关于搜索引擎的各个环节，前述的元搜索引擎以及其他的元搜索系统提出了各种各样的技术，使元搜索引擎的技术达到了相当水平。但是，还是有不少问题有待解决和继续提高。其中最突出的是：

· 搜索引擎的选择和协作。Internet上的搜索引擎发展很快，数量会越来越多，其特点和专长都不相同。元搜索引擎架设在这些搜索引擎的基础之上，如何选择适合用户查询的、高效的搜索引擎将会影响到元搜索引擎的检索效率和结果质量。SavvySearch和ProFusion做了很有价值的研究，取得了不错的效果。为了更有效地利用底层的搜索引擎，有一个想法值得考虑：让这些搜索引擎和元搜索引擎合作，以便元搜索引擎能更好地了解其特性，更有效地利用其数据库。当然，如何才能这样做而不至于使搜索引擎和元搜索引擎的管理太复杂是一个需要研究的问题。

· 结果融合。结果融合是元搜索引擎的核心问题，现在的方法一般是基于排序位置或得分评价来进行的，能达到不错的效果。但是，重复文档、不相关文档还是在一定程度上存在，排序也并非能做到完全恰如其分，所以，有必要发现更先进有效的算法来进一步改善。

总结 Internet发展的步伐越来越快；网络，正以飞快的速度膨胀到世界的每个角落。通过网络进行的信息交流和业务合作已经深入到人们的日常生活和事务处理当中，在这种环境下，谁能及时地得到有价值的信息，谁就能把握先机，立于不败之地。网络搜索引擎以及元搜索引擎正以愈来愈高的质量为人们提供信息检索服务，成为人们使用最多的网络应用之一。尽管如此，搜索引擎还远没有达到理想的效果，还有不少问题需要继续研究，不少方面有待提高。

目前，搜索引擎、元搜索引擎的研究开发越来越呈专业化、集中化的态势，一些力量雄厚的公司集中掌握了尖端的搜索引擎技术，进行专业化的搜索引擎开发工作，为别的公司和网站提供搜索引擎系统，这正是目前流行的ASP运行模式。另一方面，如前文所述，搜索引擎的发展也呈现出比较明显的发展方向。

由于很少有文章介绍搜索引擎的核心技术等关键内容，本文调查了搜索引擎和元搜索引擎各方面的情况，力图给出一个详尽的报告。希望本文的内容对搜索引擎以及元搜索引擎的研究开发者有所帮助。

参考文献

1 archive. http://www.archive.org

2 Internet Domain Survey. http://www.isc.org

3 Smeaton A F, Agosti M. An Overview of Information Retrieval. in Information Retrieval and Hypertext. Kluwer Academic Publishers, 1996

4 McBryan O A. GENVL and WWWW: Tools for Taming the Web. First International Conference on the World Wide Web. CERN, Geneva (Switzerland), May 25-26-27 1994

5 Sander-Beuermann W, Schomburg M. Internet Information Retrieval - The Further Development of Meta-Searchengine Technology. In:Proc. of the Internet Summit, Internet Society, 1998

6 Salton G. Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer. Addison Wesley, 1989

7 TREC 96 In:Proc. of the fifth Text REtrieval Conf. (TREC-5). Gaithersburg, Maryland, November 20-22, 1996. Publisher: Department of Commerce, National Institute of Standards and Technology. Editors: D. K. Harman and E. M. Voorhees. Full text at: http://trec.nist.gov/

8 Witten I H, Moffat A, Bell T C. Managing Gigabytes: Compressing and Indexing Documents and Images. Von Nostrand Reinhold, New York, 1994

9 http://www.robotstxt.org

10 OpenDir. http://dmoz.org/

11 Pinkerton B. Finding What People Want: Experiences with the WebCrawler. The Second International WWW Conference Chicago, USA, Oct. 1994

12 Coster M. Guidelines for Robot Writers. http://www.robotstxt.org/wc/guidelines.html

13 Cho J, Garcia-Molina H, Page L. Efficient Crawling Through URL Ordering. Seventh Intl. Web Conf. (WWW 98). Brisbane, Australia, April 1998

14 Jing Y, Croft W B. An Association Thesaurus for Information Retrieval. In:Proc. of RIAO 1994, C. I. D., Paris, 1994. 146～160

15 Fletcher J. http://www.stir.ac.uk/jsbin/js

16 Marchiori M. The Quest for Correct Information on the Web: Hyper Search Engines. The Sixth Intl. WWW Conf. (WWW 97). Santa Clara, USA, April 1997

(下转第32页)

结果最优。算例3是选自文[8,9]中的例子，文[8]中最优解是：当点 $x^*=(0.875,0.625,1.875)$ 时，最优解：$f_3(x^*)=7.25$；文[9]中提供最优解为：$x^*=(0.874976,0.627653,1.869710)$，最优解：$f_3(x^*)=7.249980$；本文最优解是：当点 $x^*=(0.875019,0.624989,1.874998)$ 时，最优解：$f_3(x^*)=7.25$，结果显然优于文[9]并且达到真实的最优解，其它算例均优于有关文献提供最优解。特别值得一提是算例2和算例5是由当前世界上公认最好数值计算软件MATLAB软件计算的，原求解 $f_2(x)$ 最优解为：当点 $x^*=0.351700$ 时，最优解：$f_2(x^*)=0.827200$；而本文获得最优解为：当点 $x^*=0.351734$ 时，最优解：$f_2(x^*)=0.827184$；原求解 $f_5(x_1,x_2)$ 最优解为：当点 $x^*=(-9.5474,1.0474)$ 时，最优解：$f_5(x_1,x_2)=0.0236$；而本文获得最优解为：当点 $x^*=(-9.547368,1.047406)$ 时，最优解：$f_5(x_1,x_2)=0.023551$；此例说明本算法计算结果优于MATLAB软件计算结果。计算结果见表1。

表1 改进算法对5个函数测试结果

函数	原已知最优解 $x^*,f(x^*)$	本文算法最好解 $x^*,f(x^*)$
Max $f_1(x_1,x_2)$	$X^*=(11.631407,5.724824)$ $f_1(x^*)=38.818208$	$X^*=(11.625545,5.7250440)$ $f_1(x_1)=38.850294$
Min $f_2(x)$	$x^*=0.531700$ $f_2(x^*)=0.827200$	$x^*=0.351734$ $f_2(x^*)=0.827184$
Max $f_3(x_1,x_2,x_3)$	$x^*=(0.875,0.625,1.875)$ $f_3(x^*)=7.25$	$x^*=(0.875019,0.624989,1.874998)$ $f_3(x^*)=7.25$
Max $f_4(x_1,x_2,x_3,x_4)$	$X^*=(86.2,104.2,126.2,152.8)$ $F_4(x^*)=43.1$	$X^*=(88.04,105.72,123.4,153.1)$ $F_4(x^*)=43.1469$
Min $F_5(x_1,x_2)$	$X^*=(-9.5474,1.0474)$ $F_5(x^*)=0.0236$	$X^*=(-9.547368,1.047406)$ $F_5=0.023551$

结束语 在考察了遗传算法和文[1]中的演化算法后，本文提出了对该算法进行扩大杂交点和添加变异等改进策略，通过测试结果表明，本算法具有通用，精确，稳健，高效等特点，是求解优化问题较好方法。并对演化算法的整体性能的提高，同样具有重要作用。

参考文献

1 郭涛，康立山，等．一种求解不等式约束下函数优化问题的新算法．武汉大学学报，1999，45(5)：771～778

2 Michalewicz Z，Schoenauer M. Evolutionary algorithms for constrained parameter optimization problems[J]. Evolutionary computation，1996，4(1)：1～32

3 Michalewicz Z．Genetic Algorithm + Data Structures = Evolutionary Programms[M]. Berlin：Springer-Verlag，1992

4 [日]玄光男，程伟，著．汪定伟，唐加福，黄敏译．遗传算法与程序设计．北京：科学出版社，2000.5～11，100～200

5 刘勇，康立山，陈毓屏，著．非数值并行算法——遗传算法．北京：科学出版社，1995

6 潘正君，康立山，陈毓屏，著．演化算法，清华大学出版社，1998

7 [美]Z. 米凯利维茨著．周家驹，何险峰译．演化程序——遗传算法和数据编码．北京：科学出版社．2000.24～33

8 施光燕，董加礼．最优化方法，高等教育出版社，1999.3～9，27～31，90～93

9 瓦格纳著．邓三瑞，王元超，秋同译．运筹学原理与应用．北京：国防工业出版社，1992.412～420

10 黄豪，沈成武，雷建平．一种连续变量的遗传算法．武汉科技大学学报，1999，23(2)：123～125

（上接第12页）

17 Spertus E. ParaSite：Mining Structural Information on the Web. The Sixth Intl. WWW Conf.（WWW 97）. Santa Clara，USA，April 1997

18 Weiss R，et al. HyPursuit：A Hierarchical Network Search Engine that Exploits Content-Link Hypertext Clustering. In：Proc. the 7th ACM Conf. on Hypertext. New York，1996

19 Kleinberg J. Authoritative Sources in a Hyperlinked Environment. In：Proc. ACM-SIAM Symposium on Discrete Algorithms，1998

20 Page L，et al. The PageRank Citation Ranking：Bringing Order to the Web

21 Singhal A，Buckley C，Mitra M. Pivoted Document Length Normalization. In：Proc. of the 19th International ACM-SIGIR Conference on Research and Development in Information Retrieval（SIGIR96）

22 Brin S，Page L. The Anatomy of a Large-Scale Hypertextual Web Search Engine

23 Direct Hit. http：//www.directhit.com

24 Ginsberg A. A Unified Approach to Automatic Indexing and Information Retrieval. IEEE Computer，1993，8(5)：46～56

25 Efthimiadis E N. User Choices：A New Yardstick for the Evaluation of Ranking Algorithms for Interactive Query Expansion. Information Processing and Management，31(4)

26 Balabanovic M，Shoham Y，Yun Y. An Adaptive Agent for Automated Web Browsing. Journal of Image Representation and Visual Communication，1995，6(4)

27 Knoblock A，Arens Y，Hsu C. Cooperating Agents for Information Retrieval. In：Proc. of the Second Intl. Conf. on Cooperative Information Systems，Toronto，Canada，1994

28 Dreilinger D，Howe A E. Experiences with Selecting Search Engines using Meta-Search

29 Voorhees E M，Gupta N K，Johnson-Laird B. The Collection Fusion Problem. In：The Third Text REtrieval Conf.（TREC-3），Gaithersburg，MD 1994.95～104

30 Kantor P，et al. Combining the Evidence of Multiple Query Representations for Information Retrieval. Information Processing & Management，1995，31(3)：431～448

31 Callan J，Zhihong L，Croft W B. Searching Distributed Collections With Inference Networks. 18th Annual Intl. ACM SIGIR Conference on Research and Development in Information Retrieval，Seattle，WA 1995.21～28

32 Selberg E，Etzioni O. The MetaCrawler Architecture for Resource Aggregation on the Web

33 Gauch S，Wang Guijun，Gomez M. ProFusion：Intelligent Fusion from Multiple，Distributed Search Engines

34 Galescu L，Ringger K. Augmenting Words with Linguistic Information for N-gram Language Models

35 Smeaton A F，Crimmins F. Using A Data Fusion Agent for Searching the WWW