基于相关术语集的搜索引擎选择*)

欧 洁

(中山大学博士后流动站)(广发证券股份有限公司博士后工作站 广州 510075)

Search Engines Selection Based on Relevance Terms

OU Jie

(Postdoctoral R&D Base, GF Securities Co., LTD., Guangzhou 510075, China)

Abstract Metasearch can effectively search distributed immense electronic resources. It is built on top of several search engines, providing user with uniform access to these engines. Metasearch first passes user's query to underlying useful search engines, and then collects and reorganizes the results from the search engines used. It is called search engines selection when metasearch selects underlying useful search engines. In this paper, we present a statistical method based on relevance terms to estimate the usefulness of a search engine for any given query, which is suitable for both Boolean query and vector query. Experimental results indicate that the proposed estimation method is quite accurate, especially when the critical similarity is high between the query and the results.

Keywords Metasearch, Information resource discovery, Search engine, Information retrieval, Relevance terms

1 引言

Web 从 1991 年出现以来,已经发展成为一个巨大的全球化信息空间,而且其信息容量仍在以指数形式飞速增长。面对海量 Web 信息资源,如何有效地检索 Web 信息,以帮助用户从大量文档信息集合中找到对给定查询请求有用的文档子集,也就成为一项重要而迫切的研究课题。

在过去几年中,Internet 上出现了很多能够帮助用户查询信息的搜索引擎。由于 Web 是一个如此浩瀚纷繁的信息海洋,使用单个综合性搜索引擎来进行 Internet 上所有网页的数据搜索已经不现实。首先,单个搜索引擎的数据处理能力和存储能力无法赶上 Web 数据的快速增长速度。其次,网页时时都在变化,及时更新大型搜索引擎的索引数据是一项极困难甚至是无法实现的工作。此外,很多网站虽然允许用户通过网站自己的界面访问其文件,但并不允许其它搜索引擎直接对其文件做索引。因此,我们不得不接受大量的专题搜索引擎共存的现实,每个专题引擎提供对 Web 一部分的搜索服务。把这些专题搜索引擎综合起来,使用户对它们进行统一查询也就成为了一个亟需解决的问题。

集成搜索引擎系统建立在多个现有的搜索引擎之上,提供对这些引擎进行统一访问的服务。当集成搜索引擎接到用户查询之后,将该查询分送到适当的底层搜索引擎,再合并和排列底层引擎返回的结果。对于用户来说,使用集成搜索引擎和使用搜索引擎是相同的,只是可供检索的资源更多而已。

集成搜索引擎自己并不存储和更新对所有文件的索引。但是,为了提供更好的服务,一个复杂的集成搜索引擎通常会收集和维护一些关于底层搜索引擎内容的信息,从而进行基于内容的路由安排,即根据这些信息选择对用户查询有用的底层搜索引擎(含有对用户查询有用的文件的搜索引擎)。集成搜索引擎选择对用户查询有用的底层搜索引擎的过程称为搜索引擎选择。本文提出了一种基于相关术语集的搜索引擎内容描述方法,有效提高了搜索引擎选择的精确度,并且该方法对于布尔查询和向量查询都有效。

我们将介绍已有的搜索引擎选择方法及其缺陷,介绍我们提出的基于相关术语集的搜索引擎内容描述方法以及进行搜索引擎选择的过程,将给出这种方法的实验结果。

2 已有搜索引擎选择方法的分析

集成搜索引擎是检索分布式的海量数字资源的有效工具,图 1 是它的示意图。

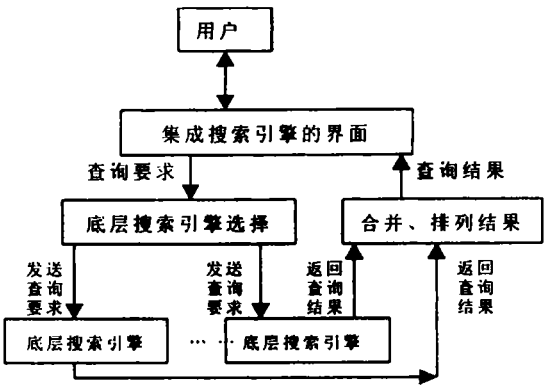

图 1 集成搜索引擎的组成示意图

底层搜索引擎选择程序将用户的查询要求发送给有用的底层搜索引擎,这是因为将查询送到无用的底层搜索引擎有很多的缺点。首先,传输查询到无用的底层搜索引擎并从这些引擎传输无用的文件到集成引擎形成了不必要的网络通信;其次,当无用的引擎处理查询时,其资源也被浪费了;第三,当无用的引擎返回大量文件时,集成搜索引擎需要花更大代价来识别出有用的文件。为了解决以上问题,我们应该尽量将查询只传送到有用的搜索引擎。下面我们给出搜索引擎对用户查询有用的定义。

*)本文工作得到 863 计划课题"中国数字图书馆示范系统"的支持,课题编号:863-306-ZD11-03。

对于支持布尔查询的搜索引擎，它对于用户查询有用性的度量标准为符合用户查询的文档的数目。

对于支持向量查询的搜索引擎，它对于用户查询有用性的度量标准可以为查询和搜索引擎中资源的相似度，更精确的度量标准为文[3]中所定义的方法，具体描述如下：

定义 1 假设存在一个相似度计算函数，该函数计算各文件对于给定查询的相似度，相似度的大小可以近似反映用户所提交的查询和文件的有用程度大小。对于给定查询 q 而言，一个文件 d 如果符合以下条件中的任意一个，则 d 被认为是对用户查询有用的：

①如果检索要求返回 n（n 是正整数）个文件，那么在所有文件对于 q 的相似度中，d 对于 q 的相似度在最大的 n 个相似度之中。

②如果检索要求返回所有与 q 的相似度大于某个临界值的文件，那么 d 对于 q 的相似度大于该值。

含有对用户查询有用的文件的搜索引擎被称为有用搜索引擎。搜索引擎对用户查询有用程度的度量标准为(NoDoc，AvgSim)，NoDoc 是搜索引擎中的有用文件的个数，AvgSim 是搜索引擎中的所有有用文件与用户查询的平均相似度。

为了计算出用户查询与搜索引擎的有用程度，集成搜索引擎需要知道能够反映每个引擎中的资源的信息，我们称其为引擎资源描述。现有的引擎资源描述方法分为以下 3 类：

定性的方法[7,8]。这类方法根据一定的评分函数针对给定的查询预测每个数据库的质量，其评分或质量往往不易理解。CORI NET 方法是一种定性的方法，其搜索引擎资源描述包括了其中所有术语的两个信息：文件频率和数据库频率。文件频率是指搜索引擎中包含该术语的文档的个数；数据库频率是指包含该术语的搜索引擎的个数。由此可得术语的权重，通过向量内积的方法即可求得用户查询与引擎之间的相似度。

定量的方法[1,4~6]。这类方法根据一些比定性的方法使用的度量标准更容易理解的标准来度量搜索引擎的有用程度。换言之，定量方法使用根据给定查询计算出的搜索引擎有用程度，相对于定性方法而言，更加直接和明晰。

gGLOSS 方法[1,2,6]。是一种定量的方法，它能支持集成不同类型(布尔查询和向量查询)的成员引擎。对于支持布尔查询的底层搜索引擎，因为用户的查询经常与术语的出现位置有关，所以 gGLOSS 描述搜索引擎中资源的方式为：引擎 d_i 中的文档数 db_i 和一组 $(T_{ij}, df_{ij}, Field_{ij})$，$T_{ij}$ 代表术语 t_j，$Field_{ij}$ 代表出现术语 t_j 的区域名，如标题、作者等。df_{ij} 是术语 t_j 出现在区域 $Field_{ij}$ 中的文件频率。gGLOSS 假设术语间相互独立，从而计算出引擎中满足查询 $t_1 \wedge \cdots \wedge t_n$ 的文档数为：$Es(t_1 \wedge \cdots \wedge t_n, db_i) = (\prod_{j=1}^{n} df_{ij}) / |db_i|^{n-1}$，满足查询 $t_1 \vee \cdots \vee t_n$ 的文档数为：$Es(t_1 \vee \cdots \vee t_n, db_i) = \max_{j=1}^{n} df_{ij}$，其中 $Field_{ij}$ 满足用户对术语 t_j 的出现区域的要求，从而计算出了引擎对于用户查询的有用程度。

对于支持向量查询的底层搜索引擎，gGLOSS 对其资源的简单描述由一组 (df_i, W_i) 组成，df_i 是术语 t_i 的文件频率，W_i 是术语 t_i 在该引擎资源的所有文件中的权重总和。假设术语出现在任何含有它的文件中的权重均为 W_i/df_i。将术语按照文件频率大小升序排列，即对于任意 $i<j$，有 $df_i<df_j$，df_i 表示术语 t_i 的文件频率。GLOSS 方法基于两种术语间关系的假设来估计引擎的有用程度：第一种是高度相关，即当 $j>i$ 时，每个含有 t_i 的文件必定含有 t_j。在此假设条件下，设 T 是用户查询所对应的向量 $(q_1, \cdots, q_k)$ 与文件相似的临界值，p 满足 $\sum_{i=p}^{k} q_i \cdot \frac{W_i}{df_i} > T$ 和 $\sum_{i=p+1}^{k} q_i \cdot \frac{W_i}{df_i} \leqslant T$，那么，该引擎的有用程度可估计为：$\sum_{j=1}^{p} (df_j - df_{j-1}) \cdot (\sum_{i=j}^{k} q_i \cdot \frac{W_i}{df_i})$。第二种是不相交的情况，即每个文件中至多含有一个查询术语，那么该引擎的有用程度可估计为：$\sum_{i=1,\cdots,k \mid (q_i \cdot \frac{W_i}{df_i}) > T} q_i \cdot W_i$。在这两种假设条件下也可以求得(NoDoc，AvgSim)，从而对引擎的有用程度做更精确的评估。

为了计算支持向量查询的底层搜索引擎的有用性，文[5]提出了一种术语间关系的假设，即术语间相互独立，从而计算出与用户查询所对应的向量 $(q_1, \cdots, q_k)$ 间的相似度大于 T 的文件的个数。考虑如下产生式：

$(p_1 * X^{q_1 * w_1} + (1-p_1)) * \cdots * (p_r * X^{q_r * w_r} + (1-p_r))$

其中 p_i 是术语 t_i 出现在引擎中的一个文件中的概率。设将上述产生式展开后所得公式为：$(a_1 * X^{b_1} + \cdots + a_r * X^{b_r})$，如果术语相互独立而且术语 t_i 在任何出现它的文件中的权重均为 w_i，则搜索引擎中与用户查询的相似度为 b_i 的文档数为 a_i。

基于学习的方法[9,10]。这类方法根据以往检索成员搜索引擎的经验来预测各引擎对新查询的有用性。

在 gGLOSS 中，搜索引擎的有用程度与所使用的相似度临界值紧密相关，因此 gGLOSS 方法可以区分搜索引擎中是否具有与查询高度相似的文件。相比之下，CORI Net 就无法做到这一点。

gGLOSS 方法用于评估引擎有用程度的两个假设条件显得过于严格，因此评估得到的有用程度并不准确。基于高度相关假设的评估倾向于过高地估计有用程度，而基于不相交假设的评估则倾向于低估有用程度，所以它们适合同时使用。

相对而言，文[5]所提出的术语间相互独立的假设比较符合现实情况，特别是在海量资源的情况下，实验结果也表示它能获得更高的引擎选择精确度。但是，展开产生式的计算复杂度是指数级的，因此它更适用于处理术语少的短查询。

3 基于相关术语集的搜索引擎选择

搜索引擎选择是集成搜索引擎中的一个很重要的问题，但我们的调查研究表明，尚没有能够解决这个问题的最佳方案。相对而言，术语间相互独立的假设比较符合现实，因此我们对这种方法做了改进，提出了一种基于相关术语集的资源描述方法，通过得到术语间的相关关系来提高搜索引擎选择的精确度，并减少计算复杂度。

3.1 设计思想

虽然术语间相互独立的假设比较符合现实，但是通过语言学常识可以知道，术语间并不是相互独立的，有些术语经常出现于同一个文件中，如信息和检索；有些术语几乎不可能出现于同一文件中，如梨子和计算机。我们认为，经常同时出现于一个文件中的术语具有某种相关关系，如上下位关系、所属领域相同关系等，因此我们将之称为相关术语。如果用户的查询包含多个术语，因为这些术语表示了用户的检索意图，所以通过信息检索的常识可知，这些术语很可能是相关术语，相关术语是术语间的一种重要关系。但是，已有的关于术语间的关系的假设都不能有效地得到相关术语，如高度相关假设把术语间的相关条件放宽了，它认为任何术语都是相关的，而不相

交假设则认为任何术语都不是相关的。术语间相互独立假设认为术语间的相关关系只与术语的文件频度有关。因此，我们提出了基于相关术语集的搜索引擎资源描述方法，即挖掘出相关术语集，并存储其相关性（出现了相关术语的文档的数目）。如果用户的查询中含有相关术语，则取出相关术语集中存储的相关性，并将之结合到 gGLOSS 方法和文[5]的计算方法中，从而有效地提高了搜索引擎选择的精确度，特别是对于系统设定的用户查询和文档间的相似度临界值高的情况，因为查询与文档间的相似度与术语间的相关性有密切联系，而且实验表明用于存储相关术语集及其相关性的存储容量并不大。

因为文档的标题是文档内容的概括，而且通常由一些与领域有关的术语组成，所以通过对文档标题的挖掘可得术语间的相关关系。用户在搜索文档的时候也经常使用标题中出现的术语，所以我们挖掘相关术语的依据为经常同时出现在文档的标题中的术语。

3.2 相关术语集挖掘

我们运用数据挖掘中的关联规则发现方法挖掘相关术语。在运用关联规则时，术语对应于项目（Item），文档的标题对应于交易（Transaction）。如果项目集的频度（frequency）大于 2，我们认为它是频繁项目集。用关联规则发现的方法挖掘出所有频繁项目集，我们把由所有频繁项目集及其频度组成的集合称为相关术语集。通过将相关术语集存储于引擎的资源描述中，可以提高搜索引擎选择的精度。为了减少存储空间，我们只存储了项目数目为 2 的频繁项目集及其频度。例如：信息和检索是相关术语，标题中出现这两个术语的文档数为 10 个；数字和图书馆是相关术语，标题中出现这两个术语的文档数为 8 个，那么相关术语集为：信息、检索，10；数字、图书馆，8。

我们选择项目数目为 2 的依据是：①根据文[5]的分析，用户的查询所包含的术语不多，一般不超过 6 个，大部分为一个或两个。②通过对项目数目为 2 的频繁项目集进行计算，可以估计出项目数目为 3 或 4 的项目集是否频繁项目集。如果是频繁项目集，还可以估计出其频度。因为项目集 m 的任何子集 m' 的频度一定大于或等于 m 的频度，所以频繁项目集的所有子集也是频繁项目集。如果项目集 m 的任何项目数为 2 的子集 m' 都是频繁项目集，则我们估计它是频繁项目集，否则我们可以肯定它不是频繁项目集。如果我们估计项目集 m 是频繁项目集，我们还可以估计 m 的频度为所有 m' 的频度中的最小值。

另外，相关术语也应该是经常在用户的查询中一块儿出现的术语，所以我们还定期用关联规则发现的方法得到用户查询中的频繁术语集，依此更新相关术语集。即如果用户查询中的频繁术语集不出现在相关术语集中，则将这个频繁术语集及其文件频度加入相关术语集。

3.3 基于相关术语集的搜索引擎有用性计算

相关术语集对于支持布尔查询和支持向量查询的搜索引擎选择都有效。

对于支持布尔查询的搜索引擎，使用 gGLOSS 方法和相关术语集相结合描述底层搜索引擎中的资源。其描述方式为：引擎 d_i 中的文档数 db_i、一组 $(T_j, df_{ij}, Field_{ij})$ 和引擎中的相关术语集，T_j 代表术语 t_j，$Field_{ij}$ 代表出现术语 t_j 的区域名，如标题、作者等。df_{ij} 是术语 t_j 出现在区域 $Field_{ij}$ 中的文件频率。如果用户所提交的布尔查询为标题中出现的相关术语，则根据相关术语集可以知道标题中出现相关术语的文档数，而且这个值是精确值，因而可以提高选择搜索引擎的精确度。如果用户查询中含有的术语数大于 2 个，用 3.2 节的方法估计它是否是频繁项目集。如果是频繁项目集，用 3.2 节的方法计算其频度。表 1 的结果表明，3.2 节的方法比用术语相互独立的假设计算出来的结果更接近实际情况。如果用 3.2 节的方法估计用户的查询不是频繁项目集，那么先用 3.2 节的方法估计出用户查询中的最大频繁项目子集，并估计出它的相关性，即标题中出现频繁项目子集中的术语的文档数，将这个频繁项目子集中的术语看成一个术语，采用文[2]中的方法来估计其有用性。例如：用户的查询由信息、检索和分布式组成，如果分布式与信息不是相关术语，且分布式出现在 20 个文档中，则该引擎对于查询的有用性为：$20 * 10/db_i$，其中 db_i 是引擎的文档数，10 是相关术语（信息和检索）对应的文档数。如果用户的查询不包含相关术语，采用 gGLOSS 方法来估计其有用性。

对于支持向量查询的搜索引擎，使用文[5]所介绍的方法和相关术语集相结合描述底层搜索引擎中的资源。其描述方式为：引擎 db 中的文档数 db_i、db 中的相关术语集和一组 $((p_{i1}, 1), (p_{i2}, W_i))$ 组成，其中 p_{i1} 是 db 中标题出现术语 t_i 的文档数。设 db' 是 db 的子集，db' 中的文档的标题不含 t_i。p_{i2} 是术语 t_i 出现在 db' 的文档中的概率。W_i 是 t_i 在 db' 中的权重总和除以 p_{i2} 和 db_i，即平均权重。假设 t_i 出现在所有 db' 中含 t_i 的文档中的权重均为 W_i。

下面首先介绍查询和文档的相似度计算，再介绍查询和引擎的有用性计算。

将查询 Q 表示为向量 $(q_1, q_2, \cdots, q_r)$，它满足条件：$\sum_{i=1}^{m} q_i = 1$，其中 q_i 表示术语 t_i 在 Q 中的权重。我们所采用的计算公式为：$q_i = \frac{0.7}{k} + \frac{0.6 * (k-j+1)}{k(k+1)}$，其中 k 表示在 Q 中出现的术语的个数，j 表示在这 k 个术语中术语 t_i 的出现位置。如果 t_i 不在 Q 中出现，q_i 的值为 0。将文档 D 表示为向量 $(d_1, d_2, \cdots, d_m)$，它满足条件：$0 = < d_i < = 1$，其中 d_i 表示术语 t_i 在 D 中的权重。在我们的实验系统中，如果 t_i 出现在标题和作者中，则其权重为 1；出现在其它地方，则根据文件频度和反文件频度取权重。计算公式为：$W_i = (0.5 + 0.5 \frac{tf(i)}{tf_{\max}})(\log \frac{n}{df(i)})$，其中 $tf(i)$ 为术语 t_i 在文档 D 中的出现次数，$df(i)$ 是搜索引擎中包含术语 t_i 的文档数，n 是搜索引擎中全部文档的数目，$tf_{\max}$ 是 D 中出现次数最多的术语的出现次数。由上可得，Q 与 D 的相似度如果以向量间的点积来度量的话，其相似度的取值在 0 与 1 之间。

在不相关假设下，根据文[5]中的分段法，可计算出搜索引擎 db 与用户查询 $Q: (q_1, q_2, \cdots, q_r)$ 间的相似度。考虑如下产生式：

$$(p_{11} * X^{q_1} + p_{12} * X^{q_1 * W_1} + (1 - p_{11} - p_{12})) * \cdots * (p_{r1} * X^{q_r} + p_{r2} * X^{q_r * W_r} + (1 - p_{r1} - p_{r2}))$$

设将上述产生式展开后所得公式为：$(a_1 * X^{b_1} + \cdots + a_c * X^{b_c})$，则搜索引擎中与用户查询的相似度为 b_i 的文档数为 a_i。

因为集成搜索引擎已经存储了相关术语集，所以我们通过相关术语集来提高查询与底层引擎有用性估计的精确度。如果用户的查询项为相关术语，因为根据相关术语集可以知道标题中出现相关术语的文档数，这些文档与用户查询的平

均相似度为1。如果用户的查询项用3.2节的方法估计为频繁项目集，则用3.2节的方法估计标题中出现这些术语的文档数，这些文档与用户查询的平均相似度为1。在这两种情况下，因为不用到产生式展开的方法进行计算，所以可以减少计算复杂度。如果在用户的查询项中有相关术语，也有其它术语，用户查询与引擎的有用性计算方法如下：因为根据相关术语集可以知道标题中出现相关术语的文档数，而产生式对这个文档数有一个估计值，即相关术语的出现概率乘以引擎中的文档数，所以用相关术语的文档数除以引擎中的文档数来替换产生式中的估计出现概率。设相关术语为 t_i 和 t_j，该替换所进行的具体计算为：将产生式中的

$$(p_{i1} * X^{q_i} + p_{i2} * X^{q_i * W_i} + (1 - p_{i1} - p_{i2})) * (p_{j1} * X^{q_j} + p_{j2} * X^{q_j * W_j} + (1 - p_{j1} - p_{j2}))$$

（记为Ge(db,Q,i,j)）替换为：$\mathrm{Ge}(db,Q,i,j) - p_{i1} * X^{q_i} * p_{j1} * X^{q_j} + N_{ij}/db_i * X^{q_i+q_j}$，其中 N_{ij} 是相关术语集中相关术语 t_i 和 t_j 所对应的文档数，db_i 是搜索引擎db中的文档数。

4 实验结果

我们所选的实验数据为786篇信息提取的与数字图书馆方面有关的文章，数据来源主要为CIIR所发表的文章和数字图书馆杂志中的文章。这些文档的标题中包括术语1816个，其中文件频度大于2的术语432个，术语数目为2的相关术语454对（标题中出现相关术语的文档的数目大于2）。其中有359对相关术语用术语间独立的假设条件计算出的相关度小于1，相关度即标题中同时出现这两个术语的文档的数目。大部分相关术语是专有名词。因为相关术语的数量相对于文章数目并不多，从而可知相关术语集所需要的存储空间并不多。

4.1 布尔查询结果

(1)对于用户的查询为相关术语的情况

因为我们存储了所有的454对相关术语及其相关度，所以可得相关术语的相关度的精确值。如果用术语间独立假设进行计算，在这454对相关术语中有359对相关术语的估计相关度小于1，即认为它们不同时出现在文档标题中。只有95对术语的估计相关度大于1，即认为它们同时出现在文档标题中。因此，大大提高了预测的精确度。

(2)对于用户的查询为3个术语，且这些术语为频繁项目集的情况

我们计算出这样的频繁项目集共296个。用我们的计算方法计算出它们的平均有用文档数为3.75，实际平均有用文档数为3.527，而在术语间相互独立的假设下计算出的平均有用文档数为0.114。我们任意选取了10个这样的查询，其计算结果见表1。

表1 布尔查询结果

查询 / 文档数	查询1	查询2	查询3	查询4	查询5	查询6	查询7	查询8	查询9	查询10
实际有用文档数	3	3	3	6	5	5	4	4	8	7
本文方法计算的文档数	4	3	3	7	7	6	5	5	9	8
在术语独立假设下计算出的文档数	1.014	0.36	0.17	0.98	0.81	0.91	0.03	0.71	0.5	0.51

4.2 向量查询结果

用户的查询为3个术语，其中有2个为相关术语，且用3.2节的方法估计这些术语不是频繁项目集的情况。取用户查询与文档间的相似度临界值为0.6，取10个这样的查询，我们的方法计算出的引擎有用性和用术语间独立假设计算出的引擎有用性比见表2。

表2 引擎有用性计算结果

查询 / 文档数/相似度	查询1	查询2	查询3	查询4	查询5	查询6	查询7	查询8	查询9	查询10
实际有用文档数	100	91	19	13	6	13	4	7	7	4
本文方法计算出的文档数	111.2	103	21	16	7.5	14	4.4	8.3	7.6	4.5
在术语独立假设下计算出的文档数	48.7	41	14.5	4	3.2	1	0.6	1.74	1	0.7
实际有用文档的平均相似度	0.73	0.72	0.72	0.74	0.72	0.72	0.76	0.76	0.79	0.72
本文方法计算出的平均相似度	0.72	0.72	0.71	0.72	0.7	0.72	0.72	0.71	0.72	0.71
在术语独立假设下计算出的平均相似度	0.70	0.7	0.7	0.66	0.69	0.67	0.67	0.68	0.67	0.66

从以上计算结果可知，我们的方法对于向量查询也是很有效的。

结论 集成搜索引擎提供对多个搜索引擎的统一访问，从而使用户可以方便快捷地对多个搜索引擎进行联合资源发现。开发大规模的集成搜索引擎的一个主要挑战是搜索引擎选择问题。术语间的相关性对于搜索引擎选择的精确度有重要影响。但是已有的关于术语间的关系的假设都不能有效地得到术语间的相关性，如高度相关假设把术语间的相关条件放宽了，认为任何术语都是相关的，不相交假设则认为任何术语都不是相关的。术语间相互独立假设认为术语间的相关关系只与术语的文件频度有关。因此，本文提出了一种基于相关

（下转第63页）

与基本流中的对象联系起来。OCI 中提供的内容信息，可以将基于内容的视频检索转化成基于文本的检索。缺点在于需要改造现有的编解码器，并且表示方式不够灵活多样。另外，MPEG-4 协议中定义的 XMT 也是可利用的对象，但是 XMT 的定义初衷是保留、维持原作者的创作意图，所以同样在使用上不能完全适应检索的需要。MPEG-4 格式的对象结构，使我们用结构化的 XML 可扩展标记语言来表示更为容易。利用 MPEG-4 标准和 XML 技术的结合，将基于内容的视频检索转化为对 XML 文档的检索，是一条捷径。已有的工作与此有相关性，但是还没有完全实现检索的例子。这里需要解决的问题是，怎样定义 XML 标签集，才能完全表示 MPEG-4 视频内容，且 XML 的表示文档要有合理的开销。合理的索引结构也是必要的。我们进一步的工作就是要解决这些问题。

参 考 文 献

1 Sun Xiaoming, Shi Zhi, Kuo C C J. XML-based MPEG-4 Video Representation, and Error Resilience. Circuits and Systems, 2002 IEEE International Symposium on, 2002, 2: 676～679

2 Kim M, Wood S, Cheok L-T. Extensible MPEG-4 Textual Format. In: Proc. ACM Multimedia 2000

3 Herpel C, Eleftheriadis A. MPEG-4 Systems: Elementary Stream Management. Image Communication Journal TUTORIAL ISSUE ON THE MPEG-4 STANDARD, 2000

4 Li C-S, Mohan R, Smith J R. Multimedia Content Description In The InfoPyramid. In: IEEE Proc. Int. Conf. Acoust., Speech, Signal Processing (ICASSP), Seattle, WA, May 1998

5 Law C, Furht B. The Mpeg-4 Standard for Internet-based Multimedia Application. Handbook of Internet computing, 1999. 41～65

6 Hunter J, Newmarch J. An Indexing, Browing, Search and Retrieval System for Audio visual Libraries. http: // archive. dstc. edu. au/RDU/staff/jane-hunter/ECDL3/paper. html

7 Salembier P. An overview of MPEG-7 Mulitimedia Description Schemes and of future visual information analysis challenges for content-based indexing. Int. Workshop on Content-based Multimedia Indexing, CBMI'01, Key note speech, Brescia, Italy, Sep. 2001. 351～361

8 王相海，张福炎. 多媒体视频编码研究. 计算机科学，2001，28 (11)：17～21

9 Chang S-F, et al. A fully automated content based video search engine supporting spatio-temporal queries. ACM Multimedia, 1997

10 Koenen R. Overview of the MPEG-4 Standard, ISO/IEO JT1/SC29/WG11 N4030, March 2001

11 Zhong D, Chang S-F. Spatio-temporal video search using the object based video representation. In: Proc. of IEEE Int'l Conference on Image Processing, Santa Barbara, CA, Oct. 1997. 21～24

12 Hampapur A, Semantic. Video Indexing: Approach and Issues. ACM Sigmod Record, 1999, 28(1): 32～39

13 Jose M. Martinez (UPM-GTI, ES). Overview of the MPEG-7 Standard(version 6. 0), ISO/IEC JTC1/SC29/WG11 N4509, Pattaya, Dec. 2001

14 Fallside D C. XML Scheme part0: Primer. March 2001. http: // www. w3. org/TR/xmlschema-0/

附 录

InfoPyramid 的 XML 表示：

```
〈NEWS-STORY〉
 〈Program〉ABC Evening News〈/Program〉
 〈date〉11/2/97〈/date〉
……
〈Video〉
 〈Component〉
  〈Content_type〉video/avi〈/Content-type〉
   〈URLL〉http: // foo. com/news1. avi〈/URL〉
   〈bit-rate〉1Mbs〈/bit-rate〉……
  〈/Component〉
  〈Component〉
    〈Content-type〉video/Bamba〈/Content-type〉
    ……
    〈/Component〉
    ……
    〈/Video〉
    〈transcript〉……〈/transcript〉
〈/NEWS-STORY〉
```

（上接第59页）

术语集的搜索引擎内容描述方法，并将之结合到 gGLOSS 方法和文[5]的计算方法中，使之对于支持布尔查询和支持向量查询的搜索引擎选择都有效。从实验结果可知，它有效提高了引擎选择的精确度。

参 考 文 献

1 Gravano L, Chang K, Garcia-Molina H, Lagoze C, Paepcke A. STARTS Stanford Protocol Proposal for Internet Retrieval and Search. In: Proc. of the 1997 ACM SIGMOD International Conference On Management of Data, 1997. 207～218

2 Gravano L, Garcia-Molina H, Tomasic A. Gloss: Text-Source Discovery over the Internet. ACM Transactions on Database Systems, 1999, 24(2)

3 孟卫一，吴宗寰. 集成搜索引擎的文本数据库选择. 计算机研究与发展，2001，38(4)：396～404

4 Meng W, Liu K L, Yu C, et al. Determine text databases to search in the internet. In: Proc. of Very Large Database Conference, New York City, 1998. 14～25

5 Meng W, Liu K L, Yu C, et al. Estimation the usefulness of search engines. In: Proc. of IEEE Int'l Conf on Data Engineering, Sydney, Australia, 1999. 146～153

6 Gravano L, Garcia-Molina H. Generalizing GLOSS to vector-space databases and broker hierarchies. In: Proc. of Very Large Database Conference, Zurich, Switzerland, 1995. 78～89

7 Callan J, Lu Z, Croft W. Searching distributed collections with inference networks. In: Proc of ACM SIGIR Conf. Seattle, 1995. 21～28

8 Gravano L, Garcia-Molina H. Generalizing GLOSS to vector-space databases and broker hierarchies. In: Proc of Very Large Database Conf. Zurich, Switzerland, 1995. 78～89

9 Dreilinger D, Howe A. Experiences with selecting search engines using metasearch. ACM Trans on Information System, 1997, 15 (3): 195～222

10 Fan Y, Gauch S. Adaptive agents for information gathering from multiple, distributed information sources. In: Proc of 1999 AAAI Symposium on Intelligent Agents in Cyberspace, Stanford University, 1999. 40～46