一种适合我国动态办公环境的正文数据库检索方法*

刘 怡
(中国人民大学)

A TEXT DATA BASE ACCESS METHOD FOR CHINESE DYNAMIC OFFICE ENVIRONMENT

Liu Yi
(*Institute of Data and Knowledge Engineering, People's University of China*)

ABSTRACT

With the rapid growth of text database applications, the research of text access methods has become one of the most interesting topics in database area. According to the characters of Chinese office environment, this paper proposes a new text data base retrieval method which organizes the word signature file in the light of their superimposed coding signature. The method has no false drop that is unavoidable in other signature file methods. So it does not need to scan the full text obtained by searching the signature index. When a query is processed, only range scan to the signature file is required. Therefore, within a rather wide scope, the performance of our method is approximate to the inversion which is the fastest one among all the text access methods. In the meantime, it also maintains the merit of fitting dynamic information retrieval environment, that can not be found in other kinds of text access method.

摘 要

随着正文数据应用的迅速增长，对正文数据检索方法的研究已成为数据库领域中令人感兴趣的课题之一．本文结合我国办公环境的特点，提出一种新的按附加码方法组织字标

* 1989年3月20日收到．

识文件的正文数据库检索方法．这种方法不仅没有一般标识文件方法特有的"误选"现象，不必对查找标识文件后得出的文献进行全文扫描，而且处理查询时只需对标识文件进行范围查找．因此，其查询效率在较大范围内可以与正文检索方法中效率最高的倒排文件方法相近，并同时保持其它正文检索方法不具有的适应动态信息检索环境的优点．

§1. 引 言

传统的数据库管理系统是为处理格式化数据而设计的．近年来，随着办公信息系统和其它信息检索系统的发展，越来越多的应用环境要求扩展传统的数据库管理系统的功能，使其能有效地处理正文数据．因此，对正文数据检索方法的研究逐渐引起人们的兴趣，至目前为止已经提出了一些正文检索方法，其中有些算法也已在商品化或试验性系统中实现，但这些方法均在某些方面存在着不同的缺欠，研究设计更好性能的正文检索方法已成为数据库技术中一个新的重要课题[2],[3],[4],[5],[6],[7]．

本文第二部分介绍了正文数据检索的一般特点和现有正文检索技术，第三部分根据我国办公环境的特点提出了一种新的标识文件方法，最后讨论了算法在一个具体系统的实现。

§2. 现有正文数据检索方法

2.1 正文检索的环境特点

在正文数据库中，不仅数据的结构与传统的格式化数据结构不同，而且由于应用环境的差异，操作方面的特点也不同，主要表现在以下几个方面：

(1)数据量大．一般在数十兆至数百兆以至上千兆．

(2)数据的插入量很大，但删除和更新量均很少，且删除可以是批处理方式．

(3)用户的查询要求可能是归确给出的，也可能是含意不清的．

2.2 正文检索的一般方法

至今为止，已经提出了多种正文检索方法，主要有：全文扫描方法、倒排文件方法、标识文件方法、物理聚集方法和多属性散列方法．其中全文扫描方法最节省空间，但查询效率低．倒排文件方法检索效率最高，但所需索引空间大，约占数据文件的 50%～300%[2]，而且对于插入量大的动态环境可能由于并发控制的影响使查询速度大大降低．标识文件方法检索速度快于全文扫描，慢于倒排文件方法，但节省空间，插入处理简便，更适用于动态信息检索环境．

2.3. 两种基本的标识文件方法

标识文件的方法基本分为两种：一种是字标识文件的方法(简称 WS 方法)，另一种是附加码标识方法(简称 SC 方法)．

字标识文件方法是将文献中每个主题词散列为一个长度为 f 的二进制串，文献中所有字标识串接在一起，形成文献的标识．

附加码标识方法是将文献分为若干逻辑块每个逻辑块定义为包含 D 个不同的主题词的一段正文．每个主题词散列为一个长度为 F，其中 M 位为 1，其余位为 0 的二进制标

识．同一逻辑块中 D 个二进制标识以“或”运算方式形成逻辑块的标识，如图 1 所示．这些逻辑块标识连接在一起，构成了文献标识．

主题词	字标识			D=3
中文	0011	0100	1001	M=5
信息	1000	0010	1110	F=12
检索	1001	0110	0001	
逻辑块标识	1011	0110	1111	

图 1 附加码标识方法示意图

由于标识文件是用散列方法产生的，某个文献标识指出文献包含某个主题词，而实际上此文献并不包含这个主题词的情况是不可避免的．这种情况称为“误选”．对于字标识文件方法，其误选概率 $F_{d,ws}$ 为

$$F_{d,ws}=1-[1-\frac{1}{S_{max}}]^{D} \tag{1}$$

其中 S_{max} 为不同的字标识的最大可能数．对于附加码标识方法，误选概率 $F_{d,ac}$ 为

$$F_{d,ac}=[\frac{1}{2}]^{\frac{F\ln 2}{D}} \tag{2}$$

其中 F 为块标识的长度，D 为每一块包含的主题词数．关于公式(1),(2)的推导可参阅[2].

§3. 适合我国动态办公环境的标识文件方法

3.1 我国办公环境的特点

我国办公环境除具有 2.1 中所述的动态信息检索的特点之外，还有两个与其它国家不同的特点．(1)我国办公环境中，虽文献一般比国外长[8]，但每篇文献的主题词却相对较少．国外一般为 10%左右，而我国只占 2–4%或更少．(2)虽数据库中，每篇文献具有相当数量的主题词，但整个应用系统所涉及的主题词却不多．

这些特点，使我们能设计一种新的标识文件，更有效地处理我国办公环境中信息检索事务．

3.2 一种适合我国办公环境特点的标识文件方法

在一个正文数据库中可以存储多种类型的文献，如公文、备忘录、情报等等．这些文献可以形式化地表示为$(a_1, a_2, \cdots a_n, b)$．其中 $a_1 \cdots a_n$ 是文献首部，为一些格式化数据，b 是不定长的正文数据，称为文献体．不同类型的文献可有不同类型首部．如公文类文献首部可如图 2(a)所示；情报类文献，其首部可如图 2(b)所示．

a_1：文件标题　　a_1：标题

a_2：发文机关　　a_2：收到日期

a_3: 发文日期 a_3: 作者(译者)

a_4: 文件密级

(a)公文类文献首部 (b)情报类文献首部

图 2 文献首部示例

对于文献可以根据首部各项进行检索，也可以根据文献体内容进行检索．文献首部是格式化数据，对这类数据的检索已有许金成熟的技术和方法．文献体是不定长的正文数据，为支持对文献的按内容查找，可用第 2 节中介绍的各类方法．现有一些商品化系统多是采用倒排文件的索引结构，这在我国多是 PC 与高档微机组成的分布办公信息系统来说，首先遇到的问题是外存空间不足以建立必要的主题词索引，其次是对于信息加人较多的系统，索引维护量大，而且在多用户并行运行情况下，可能使查询延迟时间过长，这种情况下，标识文件方法显然是较好的替代方法之一．

为改进标识文件的性能，提高其检索速度，可从两个方面入手．一是尽量降低误选率，二是缩短标识文件的长度．从 3.1 中所述我国办公环境中主题词总数不多的特点出发，我们可以在系统中设立一张转换表，表中存放每个主题词和它的字标识．这种转换表方式生成的字标识与主题词是一一对应的，因此不会产生一般标识文件方法特有的误选现象．这种转换表如能与以词方式输人的汉字中文系统相结合则可取得更好的效果．

对数据库中所有主题词建立字标识后，每个文献所有主题词的字标识与文献的逻辑地址构成了整个文献的标识，如图 3 所示．

图 3 文献标识结构

但如果直接用这样的文献标识组成的标识文件作为索引检索，则对于一个大数据库说来，每次查询要扫描的标识文件就很长．如：设数据库中存有 10 万篇文献，每篇文献的主题词为 12 个，每个主题词的字标识为 2bytes，文献的逻辑地址为 10bytes，则标识文件的长度为 3.4MB．这种情况下其查询速度比倒排文件方法慢几个数量级．为降低查询时扫描全部标识文件的代价，我们对字标识形式的文献标识再建立附加码方式的索引，以缩小查找标识文件的范围．

索引建立时，对每篇文献用 hash 方法生成其附加码标识，再将用 WS 方法产生的文献标识按其附加码标识排序，具有相同附加码标识的文献标识组成一块，每块的地址及相应的附加码记人上一级索引块．若有某个附加码标识对应的文献标识多于一块，则将这些文献标识块链接起来．整个索引结构如图 4 所示．

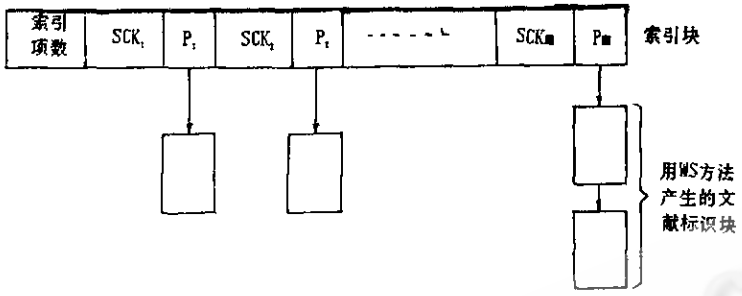

图 4 标识文件的索引结构

为处理方便，建立索引时可首先估计数据库中数据量，留有一定的余量后确定大致的文献标识块数，再根据应有块数确定附加码的位数。

查询时，系统同时生成查询表达式附加码标识和字标识，然后先根据附加码标识确定标识文件的查找范围，再将应查找的页块调入内存按字标识查找，文献标识块结构如下图所示。

图 5 文献标识块结构

其中 Pt 是指下一页块的指引元，而当前文献标识数是为处理并行的检索和插入操作设立的。

下面给出具体的索引插入和检索算法。由于索引的建立在 SC 索引块建立后就与插入算法完全一致，所以插入算法实际上是索引建立算法的基础。

为处理对索引的并发操作，系统设立页级更新锁，它是只对更新操作有意义的排它锁，其封锁与解锁过程为 ILOCK 与 UNILOCK。

算法 1：索引插入算法

在新加入的文献插入正文文件后，索引插入算法为：

①根据用户给出的主题词生成如图 4 所示结构的文献标识和文献的附加码标识 SCK_i。

②根据 SCK_i 值找到相应页指引元 P_i，ILOCK(P_i)。

③将文献标识加入以 P_i 为首的文献标识链块末尾，将 P_i 中当前最大文献标识数加 1。

④UNLOCK(P-[i])。

⑤将更新页写回磁盘。

算法 2：索引查找算法

①生成查询表达式的字标识与附加码标识 KSC。

②读取索引块，对索引项的第一项至最后一项，执行③。

③若 KSC > SCK_i 则转⑤，否则执行④。

④若对 KSC 所有为 1 的位，SCK_i 相应位也为 1，检查 P_i，若 $P_i>0$，则取出 P_i 指向页，根据查询表达式的字标识查找文献标识块，并将找出的满足条件的文献逻辑地址送至结果表列.

⑤根据结果表列找出满足条件的文献送给用户.

由于办公环境和其它正文检索环境中一般删除操作均为批处理，这相当于对索引进行再组织，这里不再给出删除算法.

从算法 1、算法 2: 可以看出，采用本节所述的标识文件方法建立的索引结构有以下几个优点; (1)用转换表产生的 WS 形式的文献标识与主题词是一一对应的，因而避免了"误选"现象，避免了查找标识文件后再对部分正文数据文件进行全文扫描. (2)用附加码方法组织文件标识索引后，由算法 2 可知，查询时只要对标识文件中 SCK_i 大于查询条件的附加码标识且对查询条件的附加码标识所有为 1 位，SCK_i 相应位也为 1 文献标识块才调入内存进一步查找，在相当程度上缩小了对标识文件的查找范围. (3)由于使用附加码索引，可简化对查询表达式中多个关键词的处理，免于对于指引元的交并运算，提高了查询效率. (4)附加码标识索引结构插入操作简便，并且不影响查询处理，保持了标识文件方法适用于动态信息检索环境的特点.

3.3 在已知主题词查询概率时对算法的进一步改进

在存有大量自然语言文献的正文数据库中，各主题词在文献中出现的频率是不同的，它服从 Zipf 分布，即第 n 个常用词出现的频率与 n 成反比. 在信息检索环境中，另一个接近实际的分布是"80-20"规则，即约 20%的主题词出现在 80%的查询中，同样的情况也出现在这 20%中. 对于整个库说来，约 80%的检索是涉及数据库中 20%最活动的记录的. 根据这种情况，[6]中提出对不同查询概率的主题词分配不同的置 1 位的方法来降低最常查询的主题词的误选率，以提高应答速度. 但这种方法仍需查找全部标识文件. 这里我们提出另一种改进方法，即采用对附加码标识设立查询概率加权位方式组织标识文件索引，进一步缩小对标识文件的查询范围，以期在较大程度上改进标识文件方法的查询效率.

为实现上述设想，首先在主题词转换表中为每个主题词增加一个计数单元和一个 bit 标志位. 计数单元用以统计特定时间间隔中该主题词被检索的概率，而 bit 标志位则表示在本次时间间隔中，该主题词是否属于最常查询的 20%主题词.

在索引建立或插入生成文献的附加码标识时，对每个附加码标识前增加一个 bit 做为查询概率加权位. 凡文献中包含经常查询的主题词时，此加权位置 1，否则置 0. 这样凡包含经常查询的文献标识都集中到图 4 所示的索引结构的最左边的文献标识块中. 这部分加权位为 1 的文献标识约占全部文献标识 20%，因此，对最经常的查询只需查找标识文献 20%左右即可，这大大降低了查询代价. 而对于"90-10"规则成立的环境，查询效率还可以进一步提高.

根据"80-20"规则，还可以将附加码标识和主题词转换表中的标志位增加到 2 个 bit，以进一步区分不同查找频率的主题词，降低最常查找文献的存取开销.

[6]中的算法降法经常查找主题词的误选率是以增加不常查找主题词的误选率为代价的，这样在查找含有属于 80%的不常查找的主题词和文献时，就要花费更多的时间. 而我们的方法即提高了经常查询的主题词的查找速度，又不降低查询概率低的主题词的查找

速度，较好地解决了不同概率的主题词的查询问题.

在数据库运行过程中，由于应用环境的改变，部分主题词的查询概率也会随之改变，为保证系统一直具有较高的运行效率，可在适当的时间对附加码标识索引进行再组织. 因系统中需定期对数据库做批处理的删除工作，附加码标识索引的再组织工作可以与之同时进行. 这种再组织处理也非常简单. 首先根据主题词对照表找出查询概率升高和降低的主题词，修改其相应的 bit 位，以备下次再组织之用. 然后在附加码索引中分别找出查询概率升高和降低的文献标识，将其在原索引中删除，再按转换表修改找出的文献标识的加权值后，重新将其加入索引即可.

§4. 一个实例

下面我们结合一个具体系统，将附加码标识索引方法与倒排文件方法作一个初步比较.

在我们为中央某部研制的办公信息检索系统中要存储 50 万篇文献，用户要求的文献检索项是每篇文献 12 个主题词(但整个系统的主题词仅为 1 万左右)，以及作为文献首部的标题，作者和时间. 其中作者包括第一作者，第二作者及译者. 系统配制 3 台 ALTOS986，内存 1MB，硬盘 80MB，配有 informix 数据库. 24 台 IBM PC 内存 512KB，硬盘 10MB，也配有 informix 数据库.

PC 与 ALTOS 用 worknet 网作星形连接. 50 万篇文献分别存放在 24 台 PC 上，另有一些文献资料存放在作为中心结点的机的 ALTOS 公共库中. 用户要求这样的分布系统具有充分的结点自治性.

面对这样的环境，我们分别讨论一下 B 树类结构的倒排文件方法和附加码标识索引方法对系统的影响.

首先考虑一下两种方法的空间要求.

由于用户要求充分的结点自治性，系统对每台 PC 设立了自己的局部主题词对照表，每个主题词的字标识为 2bytes，为了便于查询，每个中心结点机的 ALTOS 上也设立一个主题词对照表，每个主题词的字标识也是 2bytes.

其它索引项中时间为 4bytes，作者项中 3 个作者共 58 个 bytes，标题为 104bytes.

由于全局查询时，用户最常使用的检索项是主题词，因此除在每个结点的 PC 上设立主题词索引外，还在 ALTOS 上设立了全库的主题词索引，以提高全局查询效率. 而对于按时间、作者、标题等项进行的查询多是局部的，已知地点的，所以只在每台 PC 的数据库中对这些项建立索引.

若采用 B 树类索引结构，则每台 PC 上索引所需空间可按如下方法推算:

设每篇文献的逻辑地址为 10bytes(按用户要求)，2 万篇文献的主题词索引叶结点所需空间为:

$(2+10)\times 12\times 2\times 10^4=2.88$(MB). 时间项索引叶结点所需空间为: $(4+10)\times 2\times 10^4=0.28$(MB). 作者项索引叶结点所需空间为: $(58/3+10)\times 3\times 2\times 10^4=1.74$(MB). 标题项索引叶结点所需空间为: $(104+10)\times 2\times 10^4=2.28$(MB). 全部索引项叶结点所需空间约为 7.5MB. 而文献首部，包括: 总编号、分类号、时间、作者、资料来源、标题等

项，所需空间共为：

$(10+10+4+58+24+104)\times 2\times 10^4=4.2(MB)$.

因此，即使全部文献存在软盘上，硬盘只存最常查询的文献首部和各类索引，而且尚未考虑任何系统程序，所需空间就已超过 10MB，这显然是行不通的。

再考虑 ALTOS 上主题词的索引空间，50 万篇文献，其索引项叶结点需空间为

$144\times 5\times 10^5=72(MB)$

而 ALTOS 上加上网络软件后硬盘空间只余 50MB，因此也不能容纳全部主题词索引。

从上面粗略的推算已经可以得知采用 B 树类索引结构，在用户给定的配置条件下是不能满足其设计要求的。

若采用附加码标识的索引方法，则每台 PC 上所需索引空间可推算如下：

主题词索引采用第 3 节中所述的附加码标识索引法，文献标识所需空间为：

$(2\times 12+10)\times 2\times 10^4=0.68(MB)$

留有一定余量可按 0.8MB 计算。

文献首部各索引项虽然是格式化数据，可直接利用 informix 数据库的索引，但为节省空间，也可采用附加码方法建立索引。

设对文献首部各索引项用附加码方法建立索引，因共有时间、总编号、分类号、作者、标题等 5 项要建立索引，所以 D＝5。设每一项可有 8 个 bit 位置 1，则每个文献的附加码标识为 10tytes，2 万篇文献所需空间为：

$(2\times 12+10)\times 5\times 10^5=17(MB)$

留有一定余量可按 18MB 估算，ALTOS 上还有 62MB 空间可供系统程序和公共库使用。

再比较一下两种方法的查询效率。

为在同等条件下相比较，我们设全库共存储 24 万篇文献，每台 PC 为 1 万篇文献。

在上述假设下，采用附加码标识索引方法的标识文件所需空间约 340K。对于只作信息检索的局部结点来说，至少可将其 1／2 放在内存，若已知主题词查询概率，则将 70K 的标识文件放内存即可满足 80%的查询要求，且对多主题词的包含“与”、“或”条件的查询也不必做指引元的交并运算。

对于采用 B 树类索引结构的倒排文件方法的情况，1 万篇文献的主题词索引约有 12 万索引项，设 B^+树结点为 2KB，则 B^+树要有三级，设根结点常驻内存，则查找一个主题词至少要两次 I／O 操作。若查询条件包括多个主题词，则 I／O 操作还要成倍增加，且需作相当数量的指引元交并运算。

从上面的比较看，在 PC 的数据量不很大的条件下，附加码标识索引方法，可以在较大范围内获得较为理想的查询效率。

再考虑 ALTOS 上的情况，24 万篇文献若采用 B^+树索引，树的高度为 4，假设根结点常驻内存，则查找一个主题词需 3 次 I／O 操作。

若采用附加码标识索引方法，则标识文件空间约为

$(2\times 12+10)\times 24\times 10^4=8.16(MB)\approx 8.2MB$

在已知主题词查询概率的条件下，约 1.6MB 的索引包含了 80%查询涉及的文献标

识，若将其中约 320K 索引放在内存，在"80—20"规则成立时，不用 I／O 操作，即可满足约 60%的查询要求，对于"90—10"规则成立的环境，即可满足 80%的查询要求．再考虑包含多个主题词的"与"、"或"条件的查询，可以看到附加码标识索引方法在比较大的范围内可获得与倒排文件方法相当，甚至更高的效率．

§5. 结束语

标识文件的方法是正文数据库物理存储结构之一，随着正文信息检索应用的发展而受到人们的重视．但这种方法明显的缺点是查询效率低．本文结合我国办公环境特点，提出一种无"误选"的附加码标识索引的标识文件方法．在已知主题词查询概率时，进一步提出了设立附加码加权位的改进方法，并根据一个具体系统将此方法与倒排文件方法作了一个简单比较．文中提出的算法还可进一步改进，由于篇幅所限，这里不再讨论．

参考文献

[1] J. D. Ullman, 《Principle of Database System》, 1982.

[2] Faloutsos C. and Christodovlakis, S. "Signature files: An access method for documents and its analytical performance evaluation", ACM TOOIS 1984 oct.

[3] Cristos Faloutsos, "Access methods for text", Computing Surveys, Vol. 17, No.1, 1985.

[4] Christodovlakis, S. et al.,"Design considerations for a message file server", IEEE Trans. Softw. Eng. SE—10, 2, 1984.

[5] Tsichritzis D. et al., "A multimedia office filing system", Proceedings of the 9th International Conference on Very Large Data Base, 1983.

[6] Faloutsos, C., "Design of a signature file method that accounts for nonuniform occurrence and query frequencies," Proceedings of the 11th Conference on VLDB, 1985.

[7] G. Pavlovic—Lazetic and E. Wong, "Managing text as data," Proceedings of the 12th Conference on VLDB, 1986.