(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 101799834 A
(43)申请公布日 2010.08.11

(21)申请号 201010144526.9
(22)申请日 2005.08.08
(30)优先权数据
10/917,746 2004.08.13 US
(62)分案原申请数据
200580034128.9 2005.08.08
(71)申请人 谷歌股份有限公司
地址 美国加利福尼亚州
(72)发明人 杰弗里·A·迪安 保罗·G·哈尔
奥尔坎·瑟齐诺格鲁
阿米塔布·K·辛加尔
(74)专利代理机构 北京市柳沈律师事务所 11105
代理人 邵亚丽
(51)Int.Cl.
G06F 17/30(2006.01)

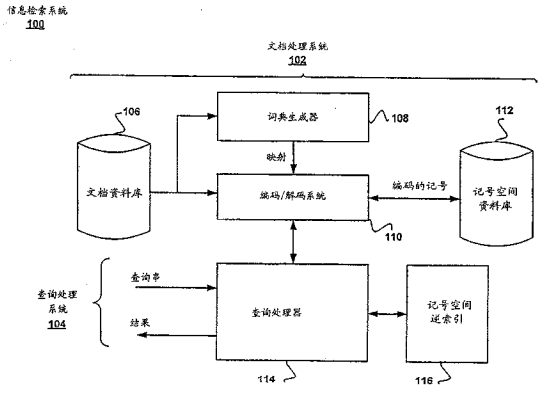
权利要求书 2 页 说明书 16 页 附图 10 页

(54)发明名称

用于记号空间资料库的多级查询处理系统与方法

(57)摘要

一种多级查询处理系统与方法，该多级查询处理系统与方法通过一个多层映射方案所简化的递增的文档重构，允许多级查询记分，包括“片段”生成。在多级查询处理系统的一个或多个级，使用一个相关性得分集合，选择作为一个有序列表向用户提交的文档的一个子集。可以部分地从多级查询处理系统的先前级中所确定的相关性得分的一个或多个集合中导出相关性得分集合。在某些实施例中，多级查询处理系统能够一或多遍地执行一个用户查询，并且能够使用来自每个遍的信息，扩展用户查询，以在相继遍中用于改进有序列表中的文档的相关性。



CN 101799834 A

1. 一种用于在多级查询处理系统中处理查询的方法，包括：

响应于一个或多个查询项，从索引中检索第一文档标识符集合；

针对相应于第一文档标识符集合的至少一个子集的压缩文档集合，生成相关性得分的第一集合；

解压缩所述压缩文档集合的至少一部分，以恢复第一记号集合，其中，所恢复的第一记号集合与所述相应于第一文档标识符集合的压缩文档集合中的位置相关联；

根据所恢复的第一记号集合，自动地生成附加查询项；

使用所述附加查询项，制定新查询；

处理所述新查询，以从所述索引检索第二文档标识符集合，并且至少部分基于所述附加查询项生成相关性得分的第二集合；以及

利用所述相关性得分的第二集合来选择作为有序列表用于呈现给用户的顶部文档的子集。

2. 根据权利要求 1 所述的方法，还包括：

解压缩所述压缩文档集合的至少一部分，以恢复第二记号集合，其中，所恢复的第二记号集合与所述相应于第二文档标识符集合的压缩文档集合中的位置相关联；以及

使用所恢复的第二记号集合，重构所述压缩文档集合的一个或多个部分。

3. 根据权利要求 1 所述的方法，还包括：把所重构的部分随至少部分地基于所述相关性得分的第二集合从所述压缩文档集合中选择的文档的有序列表一起呈现给用户。

4. 根据权利要求 1 所述的方法，其中，所述相关性得分的第二集合基于：在相应于第二文档标识符集合的所述压缩文档集合中的查询项的一个或多个位置。

5. 根据权利要求 1 所述的方法，其中，所述相关性得分的第二集合基于：在相应于第二文档标识符集合的所述压缩文档集合中查询项之间的距离。

6. 根据权利要求 1 所述的方法，其中，所述相关性得分的第二集合基于：其中在相应于第二文档标识符集合的所述压缩文档集合中使用查询项的上下文。

7. 一种在多级查询处理系统中处理查询的方法，包括：

响应于一个或多个查询项，检索第一信息集合；

根据第一信息集合，自动地生成至少一个附加查询项；

使用所述至少一个附加查询项，制定新查询，该新查询具有多个查询项；

处理所述新查询，以从索引中检索文档标识符集合；

针对相应于所述文档标识符集合的至少一个子集的压缩文档集合，生成相关性得分集合；

解压缩在所述记号空间资料库中的压缩文档集合的至少一部分，以恢复记号集合，其中，所恢复的记号集合与相应于所述文档标识符集合的所述压缩文档集合中的所述多个查询项的一个或多个查询项的位置相关联；

根据所述文档标识符集合的至少一部分，生成文档列表，该列表包括相应于所恢复的记号集合的至少一部分的信息；以及

利用所述相关性得分集合来选择作为有序列表用于呈现给用户的顶部文档的子集。

8. 如权利要求 7 所述的方法，其中，所述相关性得分集合基于：在相应于所述文档标识符集合的至少一个子集的压缩文档集合中的查询项的一个或多个位置。

9. 如权利要求 7 所述的方法，其中，所述相关性得分集合基于 ：在相应于所述文档标识符集合的至少一个子集的压缩文档集合中的查询项之间的距离。

10. 如权利要求 7 所述的方法，其中，所述相关性得分集合基于 ：其中在相应于所述文档标识符集合的至少一个子集的压缩文档集合中的使用查询项的上下文。

11. 一种多级查询处理系统，包括 ：

用于响应于一个或多个查询项，从索引中检索第一文档标识符集合的部件 ；

用于针对相应于第一文档标识符集合的至少一个子集的压缩文档集合，生成相关性得分的第一集合的部件 ；

用于解压缩所述压缩文档集合的至少一部分，以恢复第一记号集合的部件，其中，所恢复的第一记号集合与相应于第一文档标识符集合的所述压缩文档集合中的位置相关联 ；

用于根据所恢复的第一记号集合，自动地生成附加查询项的部件 ；

用于使用所述附加查询项，制定新查询的部件 ；

用于处理所述新查询，从所述索引检索第二文档标识符集合，并且至少部分基于所述附加查询项生成相关性得分的第二集合的部件 ；以及

用于利用所述相关性得分的第二集合来选择作为有序列表用于呈现给用户的顶部文档的子集的部件。

12. 根据权利要求 11 所述的系统，还包括 ：

用于解压缩所述压缩文档集合的至少一部分，以恢复第一记号集合的部件，其中，所恢复的第二记号集合与相应于第二文档标识符集合的所述压缩文档集合中的位置相关联 ；以及

用于使用所恢复的第二记号集合，重构所述压缩文档集合的一个或多个部分的部件。

13. 根据权利要求 11 所述的系统，还包括 ：

用于把所重构的部分随至少部分地基于所述相关性得分的第二集合从所述压缩文档集合中选择的文档的有序列表一起呈现给用户的部件。

14. 根据权利要求 11 所述的系统，其中，所述相关性得分的第二集合基于 ：在相应于第二文档标识符集合的所述压缩文档集合中的查询项的一个或多个位置。

15. 根据权利要求 11 所述的系统，其中，所述相关性得分的第二集合基于 ：在相应于第二文档标识符集合的所述压缩文档集合中查询项之间的距离。

16. 根据权利要求 11 所述的系统，其中，所述相关性得分的第二集合基于 ：其中在相应于第二文档标识符集合的所述压缩文档集合中使用查询项的上下文。

用于记号空间资料库的多级查询处理系统与方法

[0001] 本申请是申请日为 2005 年 08 月 08 日、申请号为“200580034128.9”、发明名称为“用于记号空间资料库的多级查询处理系统与方法”的发明专利申请的分案申请。

[0002] 相关申请

[0003] 本申请涉及 2004 年 8 月 13 日提出的、申请号为 10/917,745、名为“Systemand Method For Encoding and Decoding Variable-Length Data”(“用于对可变长度数据进行编码和解码的系统和方法”)的美国专利申请,以及 2004 年 8 月 13 日提出的、申请号为 10/917,739 的、名为“Document Compression System andMethod For Use With Tokenspace Repository”(“用于记号空间资料库的文档压缩系统和方法”)的美国专利申请,将这些申请的全部内容并入此处,以作参考。

技术领域

[0004] 总体上讲,所公开的实施例涉及数据处理系统与方法,具体地讲,涉及用于具有相关索引的文档集合(以下,将其称为“记号(token)空间资料库”)的多级查询处理系统与方法。

背景技术

[0005] 信息检索系统(例如,搜索引擎)使得查询与根据文档集(例如万维网)所生成的文档的索引相匹配。典型的逆索引(inverse index)包括每个文档中的单词,以及指向它们在文档中的部位的指针。文档处理系统通过使用自动的或手动进程处理从文档集所检索到的文档、页或地址的内容,来制备倒排索引(inverted index)。文档处理系统还可以把文档的内容或内容的各部分存储在资料库中,以供查询处理器在响应查询时使用。

[0006] 一直存在着对更复杂的搜索和记分技术(scoring technique)的需求,以确保查询结果与查询相关联。某些记分技术可能要求对候选文档进行部分重构,例如确定文档中所发现的查询项或关键字的上下文。令人感到遗憾的是,引入这样复杂的技术可能会由于所涉及的额外的处理和开销而导致搜索性能的降低。

发明内容

[0007] 所公开的实施例包括与一种用于记号空间资料库的多级查询处理系统与方法。所述多级查询处理系统与方法通过由多层映射方案所简化的递增文档重构而能够多级查询记分,包括“片段(snippet)”生成。在多级查询处理系统的一或多级,使用相关性得分集合,选择作文档子集作为有序列表呈现给用户。该相关性得分集合可以部分地从所述多级查询处理系统的先前级中所确定的一个或多个相关性得分集合中导出。在某些实施例中,多级查询处理系统能够对用户查询执行一或多遍,并且能够使用来自每个遍的信息,扩展用于后一遍中的用户查询,以改进有序列表中的文档的相关性。

[0008] 具体来讲,根据本发明的一个方面,提供了一种用于在多级查询处理系统中处理查询的方法,该方法包括:响应于一个或多个查询项,从索引中检索第一文档标识符集合;

针对相应于第一文档标识符集合的至少一个子集的压缩文档集合，生成相关性得分的第一集合；解压缩所述压缩文档集合的至少一部分，以恢复第一记号集合，其中，所恢复的第一记号集合与所述相应于第一文档标识符集合的压缩文档集合中的位置相关联；根据所恢复的第一记号集合，自动地生成附加查询项；使用所述附加查询项，制定新查询；处理所述新查询，以从所述索引检索第二文档标识符集合，并且至少部分基于所述附加查询项生成相关性得分的第二集合；以及利用所述相关性得分的第二集合来选择作为有序列表用于呈现给用户的顶部文档的子集。

[0009] 根据本发明的另一个方面，提供了一种在多级查询处理系统中处理查询的方法，该方法包括：响应于一个或多个查询项，检索第一信息集合；根据第一信息集合，自动地生成至少一个附加查询项；使用所述至少一个附加查询项，制定新查询，该新查询具有多个查询项；处理所述新查询，以从索引中检索文档标识符集合；针对相应于所述文档标识符集合的至少一个子集的压缩文档集合，生成相关性得分集合；解压缩在所述记号空间资料库中的压缩文档集合的至少一部分，以恢复记号集合，其中，所恢复的记号集合与相应于所述文档标识符集合的所述压缩文档集合中的所述多个查询项的一个或多个查询项的位置相关联；根据所述文档标识符集合的至少一部分，生成文档列表，该列表包括相应于所恢复的记号集合的至少一部分的信息；以及利用所述相关性得分集合来选择作为有序列表用于呈现给用户的顶部文档的子集。

[0010] 根据本发明的再一个方面，提供了一种多级查询处理系统，该系统包括：用于响应于一个或多个查询项，从索引中检索第一文档标识符集合的部件；用于针对相应于第一文档标识符集合的至少一个子集的压缩文档集合，生成相关性得分的第一集合的部件；用于解压缩所述压缩文档集合的至少一部分，以恢复第一记号集合的部件，其中，所恢复的第一记号集合与相应于第一文档标识符集合的所述压缩文档集合中的位置相关联；用于根据所恢复的第一记号集合，自动地生成附加查询项的部件；用于使用所述附加查询项，制定新查询的部件；用于处理所述新查询，从所述索引检索第二文档标识符集合，并且至少部分基于所述附加查询项生成相关性得分的第二集合的部件；以及用于利用所述相关性得分的第二集合来选择作为有序列表用于呈现给用户的顶部文档的子集的部件。

附图说明

[0011] 图 1 为信息检索系统的实施例的方框图。

[0012] 图 2 为图 1 的词典生成器的实施例的概念性方框图。

[0013] 图 3A 为用于记号空间资料库的、对文档进行编码的编码系统的实施例的方框图。

[0014] 图 3B 为用于记号空间资料库中的、对文档进行解码的解码系统的实施例的方框图。

[0015] 图 3C 为用于对文档属性进行编码 / 解码的属性编码 / 解码系统的实施例的方框图。

[0016] 图 4 为用于记号空间资料库的查询处理系统的实施例的方框图。

[0017] 图 5 为用于记号空间资料库的多级查询处理系统的实施例的方框图。

[0018] 图 6 为记号空间资料库服务器的实施例的方框图。

[0019] 图 7 为查询处理服务器的实施例的方框图。

[0020] 图 8A 为记号化文档资料库的第二实施例的方框图；以及图 8B 为图 1 的词典生成器的第二实施例的概念方框图。

[0021] 图 9A 为用于词典生成器的实施例中的编码进程的概念图，以及图 9B 描述了用于表示编码的记号的示例性数据结构。

[0022] 在这些图中的多个图中，相同的参照标记指示相应的部分。

具体实施方式

[0023] 系统概述

[0024] 图 1 为信息检索系统 100 的实施例的方框图。信息检索系统 100 包括文档处理系统 102 和查询处理系统 104。信息检索系统 100 可以为任何一种能够响应查询而检索信息（例如，文件、电子邮件、应用程序等）的系统，包括但不局限于，用于在诸如 Internet（例如，经由万维网）或者内部网 (Intranet) 的一个或多个网络上或在本地的用户的计算机上执行明确的或隐含的文档搜索的一个或多个计算机系统。注意，项“文档”意指文档、网页、电子邮件、专用 (application specific) 文档和数据结构、即时消息处理 (IM) 消息、音频文件、视频文件、以及可以驻留在一个或多个计算机系统上的任何其它数据或应用程序。

[0025] 文档处理系统

[0026] 文档处理系统 102 通常包括一个或多个文档资料库 106、词典生成器 108、编码 / 解码系统 110 以及记号空间资料库 112。编码 / 解码系统 110 从一个或多个文档资料库 106 中检索文档，将文档解析为记号，使用来自词典生成器 108 的映射，把记号编码为一种压缩的格式，然后把编码的记号存储在记号空间资料库 112 中。

[0027] “记号”可以为通常在文档中所发现的任何对象，包括但不局限于，项、短语、标点、HTML 标签等。在解析之后，文档的集合表示为记号序列。而且，该记号序列中的每个记号都具有记号位置，该记号位置也代表该记号在文档集合中的位置。例如，该文档集合中的第一个记号可以被赋予位置 0，该文档集合中的第二个记号可以被赋予位置 1，等等。

[0028] 需要注意的是，在某些实施方式中，采用与用于对文档进行解码的计算机完全不同的计算机集合来对文档进行编码。例如，网蠕动 (crawling) 系统可以包括对文档进行编码的文档处理系统 102，而查询处理系统 104 可以对编码文档的所选择部分进行解码。在这样的实施方式中，由文档处理系统 102 所编制的文档逆索引和记号空间资料库 112 或其拷贝，由查询处理系统 104 使用。

[0029] 词典生成器 108 通过对文档进行解析，生成用于对文档集合进行编码的映射 (mapping)。此处，把词典生成器 108 所产生的第一映射称为全局词典，全局词典标识文档集合中的所有不同的记号（此处，称为唯一记号），并且把全局记号标识符赋予每个唯一记号。由词典生成器 108 所产生的第二映射实际上为映射的序列，此处，将其中的每个映射称为小词典。每个相应的小词典仅用于对文档集合中位置的相应范围进行编码和解码。以下将更详细地解释全局词典和小词典的生成和使用。

[0030] 查询处理系统

[0031] 查询处理系统 104 包括耦接于编码 / 解码系统 110 的一个或多个查询处理器 114 以及记号空间逆索引 116。记号空间逆索引 116 把文档集合中的所有 GTokenID 映射到它们在文档中的位置。从概念上讲，逆索引 116 包括针对每个 GTokenID 的记号位置的列表。为

有效起见，对每个 GTokenID 的记号位置的列表进行编码，以减小逆索引所占空间量。

[0032] 在某些实施例中，一个或多个查询处理器 114 把一个查询进行解析为多个查询项 (terms)，这些查询项被一个或多个查询处理器 114 变换成查询表达式（例如，布尔树表达式）。使用查询项对记号空间逆索引 116 编索引，以检索记号位置，以下将参照图 4 更全面地对此加以描述。在某些实施例中，把记号位置用于多级查询处理系统，该多级查询处理系统用于对与查询相关的文档进行记分，如针对图 5 所描述的。相应于查询项，查询处理器 114 生成经由一个或多个通信模式（例如显示设备、音频等）呈现给用户的文档的有序列表。

[0033] 词典生成器

[0034] 图 2 为图 1 的词典生成器 108 的实施例的概念性方框图。词典生成器 108 包括全局词典编制器 (builder)202 和小词典编制器 204。

[0035] 全局词典编制器

[0036] 全局词典编制器 202 从文档资料库 106 中检索文档，并且通过把唯一全局记号标识符 (GTokenID) 赋予包含在文档中的每个唯一记号，生成全局词典 206。在某些实施例中，文档资料库 106 被逻辑地或物理地划分成多个部分，有时将它们称为分区 (partition)，并且针对每个分区生成独立的全局词典 206。在一个实施例中，数十亿个文档的集合划被分成数千个分区，对每个分区进行处理，以生成全局词典 206。典型的全局词典 206 可以包括数百万个唯一记号。

[0037] 在某些实施例中，在把文档解析成记号以及对记号进行处理之前，要被编码的文档集合（例如，一个分区中的文档）根据一个或多个准则进行分类。对文档的这种分类有助于对记号化的文档进行有效的编码，因为使用了类似的单词集合的文档将被彼此相近地放置在文档集合中。因此，与相反的情况相比，，每个小词典（以下所描述的）将平均覆盖文档集合的一个较大的部分，而且，文档的编码将占用较少的空间。在一个实施例中，首先根据语言对文档集合进行分类，然后，针对每种语言，根据 URL 对文档进行分类，其中 URL 的主机名部分的字段依次反转。例如，在根据语言进行分类之后，把所有的法语文档集合在一起，然后将根据 URL 对法语文档进行分类。当根据 URL 进行分类时，每个 URL 最初包括一个 h1.h2…hy.hz/n1/n2…的模式，其中，h1.h2…hy.hz 包括 URL 的主机名部分，而 /n1/n2 代表 URL 的余留部分。在根据 URL 进行分类之前，把 URL 重新映射到模式 hy.hz…hy.hz/n1/n2…。例如，把 URL “www.google.com/about.html” 重新映射到 “com.google.www/about.html”。通过在根据 URL 进行分类之前反转 URL 的主机名字段，可以根据文档相互间的逻辑接近性对文档进行分类。因此，可以把相似类型的文档（在针对某一具体语言的文档组中）集合在一起；在针对每个文档类型的文档组中，把每个网点上的文档集合在一起；在针对每个网点的文档中，把针对网点的不同分支的文档集合在一起；等等。

[0038] 在某些实施例中，使用一种或多种群集技术对文档进行排序。可以使用包含在文档中的项 (term)、单词或短语，把文档组织成与各种概念相关的群集。例如，可以把关于文档的一般信息（例如，嵌入所标识的文档或与所标识的文档相关的元数据）、从所标识的文档所取样的内容、和 / 或关于文档的类别信息用于对文档排序。

[0039] 在某些实施例中，当对文档进解析时，全局词典编制器 202 存储有关每个所标识的唯一记号的信息（未在图 2 中加以描述），例如，文档集合中的每个唯一记号的出现的次数，以及与所述唯一记号相关的语言（如果存在的话）。可以根据与其中发现所述唯一记号

的文档相关的语言，来确定与该记号相关的语言。当在与一种以上的语言相关的文档中发现特定记号时，可以使用任何适当的方法确定与该记号相关的语言。一种适当的方法是统计方法，该方法在解析文档集合以便标识唯一记号时使用。最初，把每个记号赋予其中发现该记号的第一文档的语言，然后，针对出现在与赋予该记号的当前语言不同的一种语言的文档中的记号的每次相继的出现，仅当在 0 和 1 之间随机（伪随机）选择的数小于 1/N 时，该记号都被重新赋予另一种语言，其中，N 为所述记号的出现的当前次数。在其它一些实施例中，可以使用任何类似或适当的语言赋予机制，使得语言与每个唯一记号相关联。在某些实施例中，不把语言与代表标点符号的唯一记号相关联。在又一个实施例中，当可以把语言与每个唯一记号相关联时，当处理 N 个（例如 256 个）最频繁出现的记号时，忽略语言关联。因此，有效地忽略了与标点记号相关的语言。

[0040] 在某些实施例中，根据唯一记号出现的频度，对唯一记号的列表、以及相关的频度和语言信息进行分类。作为选择，接下来，还可以对条目进行分类，以有助于对文档集合进行节省空间的编码。例如，在一个实施例中，首先根据出现频度对所有唯一记号进行分类。然后把唯一记号的所得到的分类列表划分成频带 (band)。例如，顶级频带，即，频带 0，可以包括最多 255 或 256 个记号（即，那些具有最高频度的记号）。第二个频带，即频带 1，可以包括最多 2^{14} 个（即 65536 个）记号，不含频带 0 中的记号。第三个频带，即频带 2，可以包括所分类的唯一记号的列表中接下来的 2^{14} 个（即 65536 个）记号。当然，在其它一些实施例中，每个频带中的记号的数目可以不同。接下来，根据第二准则集合，对每个频带中的记号进行分类。例如，在一个实施例中，按字母顺序、即按数字和字母的值，对第一频带中的记号进行分类。首先，按语言对每个其它频带中的记号进行分类，然后按字母顺对它们进行分类。因此，除频带 0 之外，根据语言，对每个频带中所分类的记号进行了分组，而且在每个语言组中，按字母顺对记号进行了分类。在其它一些实施例中，可以使用其它分类准则，对每个频带中的唯一记号的分类。

[0041] 分类过程产生了唯一记号的分类的列表，每个唯一记号在列表中具有相应的位置。然后，向每个所分类的唯一记号赋予唯一的全局记号标识符（以下，也将其称为“GTokenID”）。GTokenID 根据用于实现文档处理系统 102 的平台可以包括任何适当的数据类型与宽度（例如，32 个比特的无符号整数）。在某些实施例中，按递增次序把 GTokenID 赋予所分类的唯一记号，以致可以向高频度记号赋予小值的 GTokenID，以及向低频度记号赋予大值的 GTokenID。更具体地讲，在一个实施例中，向所分类的记号列表中的每个记号赋予等于其在唯一记号的分类列表中的数值位置的 32 个比特的全局记号标识符。于是，向列表中的第一个记号赋予等于 0（即，十六进制格式的 00000000）的 GtokenID，向列表中的第二个记号赋予等于 1 的 GtokenID，等等。此处，把 GTokenID 映射到唯一记号值的结果集合称为全局词典 206。在某些实施例中，全局词典 206 实际上包括两个映射结构，一个把 GTokenID 映射到记号，另一个把记号映射到 GTokenID。在编码过程期间使用记号向 GTokenID 的映射，而在对文档的一些部分进行解码时，使用 GTokenID 向记号的映射。

[0042] 如以下将更全面地加以解释的，根据频度对唯一记号进行排序，有助于减小存储小词典 208 所需的空间量。在那些其中根据除出现频度之外的其它准则对唯一记号的区进行分类的实施例中，情况更是如此，因为赋予较低 GTokenID 的频带中的记号比赋予较高 GTokenID 的频带中的记号具有更高的出现频度。

[0043] 在某些实施例中，向那些比普通记号更频繁出现的“特殊”记号，例如 HTML 标签和标点，赋予占据了全局词典 206 中的 GtokenID 的前缀 205 部分（例如，$GTokenID_0$-$GTokenID_{N-1}$）的 GtokenID。可以把所有其它的 GTokenID 位移分配给前缀 205 的最后特殊的 GTokenID。

[0044] 在以上的讨论中，把 GTokenID 描述为固定长度值，例如 32 个比特的无符号整数值。然而，也可以把这些同样的 GTokenID 视为可变长度标识符，因为当为了存储对 GTokenID 进行编码时，可以在编码期间截断或屏蔽等于 0 的最高有效字节（或比特）。例如，在某些实施例中，把具有小于 2^8 的值的所有 GTokenID 编码为单字节值，把具有小于 2^{16} 的值的所有 GTokenID 编码为二字节值，以及把具有小于 2^{24} 的值的所有 GTokenID 编码为三字节值。在这一方式下，与具有低出现频度的记号相比，用较短长度的 GTokenID 表示文档集合中具有最高出现频度的记号。

[0045] 在以下所描述的实施例中，记号空间资料库中填充了有固定长度的 LTokenID，而不是可变长度的 GTokenID。然而，把记号空间资料库中的 LTokenID 映射回原始记号（当然，它们也为可变长度的），需要存储大量的“小词典”，而且小词典内容包括 GTokenID。为了有效存储小词典，可以把每个小词典中的 GtokenID 当作可变长度值。作为选择，每个小词典中的 GtokenID 也可以被作为这样列表：首先对其进行 Delta 编码，然后使用可变长度的编码方案对所得到的 Delta 值进行编码。

[0046] 小词典编制器

[0047] 在生成了全局词典 206 之后，由小词典编制器 204 生成小词典集合 208，供编码 / 解码系统 110 使用。小词典 208 中的每个条目包括 GTokenID 和相应的局部记号标识符 (LTokenID)。小词典 208 中的条目的位置暗示了每个条目的 LTokenID，因此不需要显式地存储。每个相应的小词典 208 仅用于对记号化文档中的记号位置的不同的、相应的具体范围进行编码和解码，从而允许每个小词典 208 使用相同的 LTokenID 集合。例如，当小词典编制器 204 对整个文档进行解析时，针对小词典编制器 204 所遇到的前 P 个唯一记号，生成具有 P 个（例如 256 个）条目的第一小词典 208（即，小词典 A）。一旦已经遇到前 P 个唯一记号，则针对第一小词典 208 对于其为有效的记号位置的范围，产生包括开始记号位置 $Start_Pos_A$ 的“有效范围映射”210 中的第一条目。把第一小词典 208 中 P 个 LTokenID 中的每个 LTokenID 赋予唯一的 GTokenID。当已经把所有 LTokenID 赋予 GTokenID 时，针对小词典编制器 204 所有遇到的下 P 个唯一记号，生成第二小词典 208（例如，小词典 B），并且在有效范围映射 210 中产生第二条目，其中有效范围映射 210 包括第二小词典 208 对于其为有效的位置范围的开始记号位置 $Start_Pos_B$。因此，可以使用小词典 B 对具有落入范围 $Start_Pos_B \sim Start_Pos_{C-1}$ 中的记号化的文档中的一个位置的记号进行解码，如图 2 中所示。

[0048] 为了提供具体的例子，在一个实施例中，每个小词典中的 LTokenID 具有值 0 ～ 255，每个值均由 8 个比特的无符号整数加以表示，而 GTokenID 为 32 个比特的无符号整数。通过从记号位置 0 开始，直至标识了预先规定数目的 P 个（例如 256 个）不同的记号，扫描文档集合，生成第一小词典。把这 P 个不同记号的 GTokenID 汇编 (assemble) 于列表中。在某些实施例中，根据数值值对列表中的 GTokenID 进行分类，其中，最小的 GTokenID 处于列表的顶部。然后，根据列表中 GTokenID 的位置，把 LTokenID 赋予列表中的 GTokenID。例

如，把一个为 0 的 LTokenID 赋予列表中的第一 GTokenID，把一个为 1 的 LTokenID 赋予列表中的下一个 GTokenID，等等。把所得到的从 LTokenID 至 GTokenID 的映射称为小词典 208。从 $Start_Pos_A$ ～ $Start_Pos_B$ 的记号位置的范围与该小词典相关联。通过从紧随与第一小词典相关联的最后位置的位置 $Start_Pos_B$ 开始扫描文档集合，生成第二小词典。这一扫描继续，直至识别出了预定数目的 P 个不同的记号，在这一位置，使用与以上所描述的相同的过程，生成第二小词典。小词典编制器 204 继续针对文档集合中的相继的记号位置范围生成小词典 208 的序列，直至文档中的所有记号都被映射到小词典 208。

[0049] 在一个可选的实施例中，每个小词典 208 中的前 F 个 LTokenID 保留给文档集合中 F 个最流行 (popular) 的记号。对于这 F 个 LTokenID，LTokenID 总是等于 GTokenID。该赋值方案有助于对文档的快速解码。凡当对具有 F-1 或小于 F-1 的值的 LTokenID（在记号空间资料库中）进行解码时，可以根据全局词典将其直接映射到记号，而无需首先把 LTokenID 映射到相应的 GTokenID。

[0050] 在每个小词典 208 使用相同的 LTokenID 集合（例如 0 ～ 255 个）。为了有助于压缩文档，与 GTokenID（例如 4 个字节）相比，LTokenID 具有较小的宽度（例如 1 个字节）。该宽度差 (3 个字节）表明用于把记号化的文档存储于记号空间资料库 112 中的每记号的字节数据的减少。在一个其中每个 LtokenID 占用一个字节的实施例中，在忽略其它支持数据结构所占空间（本文献中，以下将对它们加以描述）的情况下，具有 10 亿个记号的文档集合将占用记号空间资料库 112 中 10 亿个字节 (1GB)。

[0051] 当完成了生成小词典 208 的过程时，根据每个记号在记号化的文档中的位置，把记号化的文档中的每个记号与小词典 208 相关联。注意，如果记号化的文档中的每个唯一记号出现在一个以上的位置范围中，则可以把该记号与一个以上的小词典 208 相关联。在一个实施例中，一般的文档具有大约 1100 个记号，一般的小词典 208 囊括大约 1000 个记号。

[0052] 在生成了每个小词典 208 之后，编码 / 解码系统 110 把文档集合的相应部分中的记号映射到 LTokenID，并且将它们存储在记号空间资料库 112 中，以用于相继的检索。通过这一映射，把文档资料库 106 中的每个记号映射到记号空间资料库 112 中的固定长度（例如一个字节）的 LTokenID。于是，在解码 / 解压缩期间，在记号空间资料库 112 中，从一个记号位置跳至另一个记号位置，而无需可能减慢解码过程的跳跃表或等效的数据结构，是可能的。

[0053] 在某些实施例中，在需要进行文档重构之前，以压缩的格式对小词典 208 进行编码，并且将其加以存储。在一个实施例中，对每个小词典 208 中所分类的 GTokenID 列表进行 Delta 编码，然后按压缩的格式，较佳的做法是按有助于对小词典的快速与有效的解码和重构的格式，对所得到的 Delta 值的列表进行编码。2004 年 8 月 13 日提出的、序号为 10/917,745 的、名为“System andMethod for Encoding and Decoding Variable-Length Data”（“用于对可变长度数据进行编码和解码的系统和方法”）的待审美国专利申请中，描述了一种适当的数据结构和编码 / 解码方法。

[0054] 为了对特定文档进行解压缩，把与该文档的记号位置范围相关的小词典 208 解压缩成把 LTokenID 转换成相应 GTokenID 的转换表或根据小词典 208 的条目所编制 (build) 的映射。于是，可以通过读取存储在针对该文档的记号空间资料库 112 中的固定长度的

LTokenID，以及访问该文档中针对每个记号位置的小词典以便把 LTokenID 转换成相应的 GTokenID，来完成对记号空间资料库 112 中的记号化的文档的解码。然后，使用全局词典 206 把 GTokenID 映射成相应的记号（例如，文本和标点），从而重构了整个文档或文档的某一部分。

[0055] 编码系统

[0056] 图 3A 为对记号空间资料库的文档进行编码的编码系统 300 的实施例的方框图。编码系统 300 包括可选的前置处理器 302、可选的 Delta 编码器 304 以及可变长度数据编码器 306。例如，可变长度数据类型可以包括各种数据类型，但不局限于整数、字符串、浮点数、定点数等。可变长度数据包括，但不局限于文本、图像、图形、音频样本等。

[0057] 在某些实施例中，前置处理器 302 接收信息的列表，该前置处理器为了进行有效编码对信息进行排序。前置处理器 302 可以使用一种或多种分类算法，把数据排序为单调序列。例如，如果根据值对整数集合进行分类，则就大小而言相邻的整数将互相靠近，于是，可以使 Delta 编码器 304 生成用于编码的、为小数值整数的 Delta 值。Delta 编码器 304 接收排序的数据，Delta 编码器 304 计算排序的数据的相邻对儿之间的差，以得到小数值的整数。可变长度数据编码器 306 接收小数值的整数，并且把这些数据编码为一种可以有效加以解码的压缩格式。2004 年 8 月 13 日提出的、序号为 10/917,745 的、名为“System and Method for Encoding and Decoding Variable-Length Data”的待审美国专利申请中，描述了适当的可变长度数据编码器 306 的一个例子。

[0058] 可以使用编码系统 300 的全部或一部分，对文档处理系统 102 所生成的各种信息进行编码。在某些实施例中，使用前置处理器 302 对每个小词典 208 中的 GTokenID 进行分类，以确保将对大小上最接近的整数值进行 Delta 编码。然后，由 Delta 编码器 304 对所排序的 GTokenID 进行 Delta 编码，以提供差值或余值。然后，使用可变长度数据编码器 306，按组（例如，4 个值的组）把差值编码为一个压缩格式。在某些实施例中，按一个逆索引，对记号位置的列表进行类似的编码，以有助于对位置的快速与有效的解码，如参照图 4 更全面加以描述的。

[0059] 尽管可变长度数据编码器 306 提供了一种有助于快速和有效解码的压缩的格式，但也可以把其它已知的编码方案用于文档处理系统 102，以压缩信息的列表（例如，CCITT-G4、LZW 等）。

[0060] 解码系统

[0061] 图 3B 为对记号空间资料库中的文档进行解码的解码系统 308 的实施例的方框图。解码系统 308 包括可变长度数据解码器 310 和可选的 Delta 解码器 312。在某些实施例中，由可变长度数据解码器 310 接收所编码的数据组，可变长度数据解码器 310 借助一个或多个位移 / 屏蔽表对各组进行解码。Delta 解码器 312 接收所解码的数据，Delta 解码器 312 计算运行和，从而产生 Delta 解码的数据，这等效于原始的信息列表。2004 年 8 月 13 日提出的、序号为 10/917,745 的、名为“System and Method for Encoding and DecodingVariable-Length Data”的待审美国专利申请中，更全面地描述了在对组编码的可变长度整数值进行解码的过程中对位移 / 屏蔽表的使用。

[0062] 属性编码 / 解码系统

[0063] 图 3C 为用于对文档属性进行编码 / 解码的属性编码 / 解码系统 314 的实施例的方

框图。属性编码 / 解码系统 314 包括编码 / 解码系统 320，编码 / 解码系统 320 把属性信息 322 编码为属性记录 318，以存储在属性表 316 中。逐个记号地确定文档的属性，其中采用 0 或 1 的比特值来表示给定记号的每个属性的存在或不存在。例如，可以把属性表中的属性记录 318 概念性地表示为 A×K 个比特的映射，其中，A 为所编码的属性的数目，K 为其属性由属性记录 318 表示的记号的数目。如果 A 为 8，K 为 32，那么每个属性记录 318 可以针对 32 个记号的每个存储 8 个属性。可以对每个属性记录 318 进行编码，以压缩属性表所占的空间量，同时能够在查询处理期间非常快地对所选择的属性记录进行解码。2004 年 8 月 13 日提出的、序号为 10/917,745 的、名为“System and Method for Encoding and Decoding Variable-Length Data”的待审美国专利申请中，描述了一种用于对属性记录 318 进行编码和解码的适当的方法。作为选择，也可以对每个属性记录中的信息进行运行长度编码。

[0064] 记录在属性表 316 中的属性集合，可以包括一个或多个字体属性（例如，粗体、下划线等）、一个或多个文档位置属性（例如，题目、标题等）、元数据以及可用于对文档集合中的记号之间加以区别的任何其它特性或特征。在某些实施例中，在对记号化的文档进行编码，并且将它们存储在记号空间资料库中的同时，对文档集合中的记号的属性进行标识，并且对它们加以编码，如以上所描述的。把所编码的属性用于一个或多个相关性得分的级别，如参照图 5 更全面地加以描述的。

[0065] 文档资料库编码与解码系统 -- 第二实施例

[0066] 图 8A 和 8B 为一个实施例的方框图，其中，按与以上所描述的方式略有不同的方式对一组记号化的文档（“记号空间资料库”）进行编码。如以上所描述的，全局词典编制器 202 对文档集合 106 进行记号化、标识所有唯一记号、以及把全局记号标识符赋予所有唯一记号。结果为全局词典 206。接下来，区域词典编制器 804 处理该文档集合（已被记号化）。概念地把文档集合划分为区域 820，然后把每个区域 820 划分成块 822。区域词典编制器 804 针对每个区域编制“词典”，即字典 830，编码系统 810 为每个区域生成被编码的记号 832 的集合以及用于每个区域的块位移 834 的集合。区域词典 830、编码的记号 832 以及块位移 834（以下，将更详细地对它们分别加以描述），共同形成文档集合的各个区域 820 的编码表示法。

[0067] 在一个实施例中，文档集合被划分成区域 820，每个区域 820（或许除了最后区域之外）具有预定的固定大小，诸如 8192 个记号（或任何其它适当的大小）。区域 820 的每个块 822 也具有一个预定义的固定大小，诸如 64 个记号（或任何其它适当的大小）。

[0068] 在一个实施例中，针对相应区域 820 的“词典”830 为具有最高重复率的最长记号序列的有序列表，或任何类似的结构。可以通过在所述区域中编制候选记号串的表，确定它们在所述区域中的重复次数，然后选择最佳候选，直至达到最大词典大小，编制词典 830。在示例性实施例中，最大词典大小为 64 个记号，然而，在其它实施例中，也可以使用任何其它适当的大小限制。如以下将加以描述的，词典 830 用作对相应区域 820 的每个块 822 进行编码的上下文，从而能够高度压缩区域的表示。在某些实施例中，可以按一种压缩的格式对一个或多个区域词典 830 进行编码，例如，使用本文献中先前所参照的 2004 年 8 月 13 日提出的、序号为 10/917,745 的、名为“System andMethod for Encoding and Decoding Variable-Length Data”的美国专利申请中所描述的编码方法。

[0069] 参照图 9A 和 9B，在一个实施例中，编码系统 810 按如下方式对记号的每个块 822

进行编码。把针对相应区域的词典 830 作为紧在所述块的记号之前的一个记号集合加以对待。从第一个到最后一个，顺序地处理所述块的记号，把每个记号，并且尽可能多地把相继的记号与先前记号序列中的最长匹配记号序列，包括区域词典 830，相匹配。如果发现匹配的先前序列，则生成“拷贝代码”。否则，生成“文字代码”，以表示记号。然后把当前代码所覆盖的所有记号作为先前记号加以对待，以用于块中下一个记号（如果存在的话）的相继处理。如图 9B 中所示，每个代表块中的记号集合的“代码”可以包括一个类型字段 902。如果代码为“文字代码”，则代码的第二部分 904 代表全局记号标识符。在某些实施例中，这一类型字段 902 表示代表全局记号标识符所需的比特的数目。例如，在一个实施例中，类型代码 902 可以最多表示 7 种不同的文字代码，每个种都具有相应的全局记号标识符长度。在其它一些实施例中，不同类型代码的数目可以多于或少于 8 个（例如，一个表示拷贝代码，其余的表示文字代码）。如果文字代码为“拷贝代码”，则代码的第二部分 906 可以包括指针 908 和长度 910，其中，指针 908 指出从先前文本的哪一个地方开始，长度 910 指出匹配的序列的长度（即，在解码期间将加以拷贝的记号的数目）。于是，比如，如果编码系统 810 发现了 4 个从当前位置之前的 31 个记号的部位开始的记号的一个匹配序列，则针对这一序列的代码将为：

[0070] <type = copy, ptr = 31, length = 4>

[0071] 拷贝代码的长度（按比特加以测量）将依赖于区域词典 830 的最大记号长度和块的最大记号长度、匹配的序列的最大允许长度、以及不同代码的数目。在一个例子中，类型字段 902 为 3 个比特（允许 8 个类型代码）、指针字段 908 为 7 个比特，以及长度字段 910 为 2 个比特，总共 12 个比特。在其它一些实施例中，也可以使用针对拷贝代码的每个字段的其它比特长度。由文字代码的类型指出每个文字代码的长度（按比特加以测量）。

[0072] 回过头来参照图 8B，当编码系统 810 对区域的块进行编码时，编码系统 810 生成指示针对该区域的每个块的编码的记号的部位的块位移 834 的集合。在一个实施例中，该区域的第一个块的块位移为进入记号空间资料库的指针，而且针对该区域的每个其它块位移为相对该区域中的第一块的开始位置的相对位移。在一个实施例中，把区域词典 830 和块位移 834 存储在根据按固定区域大小所划分的区域 820 的开始位置加以索引的表或等效的数据结构中。从另一个角度来看，向每个区域 820 赋予了区域号码，该区域号码包括其按固定区域大小所划分的开始位置，并且按区域号码对其中存储了区域词典 830 和块位移 834 的一个或多个数据结构加以索引。

[0073] 通过对相应区域的区域词典 830 的定位，使用针对区域 820 的块位移 834 对编码的块进行定位，然后对针对所述块的代码集合进行解码，以产生全局记号标识符的序列，实现对该区域的块 822 的解码。接下来，可以使用全局词典 206，把所得到的全局记号标识符的序列或其任何子集转换成相应的符号或项集合。

[0074] 查询处理系统

[0075] 图 4 为用于记号空间资料库的查询处理系统 104 的第一级的实施例的方框图。查询处理系统 104 包括全局词典 402、记号空间逆索引 408、第一级查找表 406 以及第二级查找表 410。全局词典 402 接收查询项或串，全局词典 402 使用根据全局词典 402 的条目所编制的表或映射，把查询项转换成 GTokenID。逆索引 408 接收 GTokenID，逆索引 408 包括映射 404，映射 404 把 GTokenID 映射到存储在逆索引 408 中的索引记录 412。使用映射 404 所

标识的每个索引记录 412 包括记号位置的列表，该列表直接对应于记号空间资料库 112 中的记号位置。在某些实施例中，在生成全局词典之后，生成逆索引 408，并且可以在与用于生成小词典而遍历文档的同一遍期间，生成逆索引 408。

[0076] 在某些实施例中，逆索引 408 提供了位置列表，该位置列表可用作进入第一级查找表 406 的索引。当查询包含多个项时，逆索引 408 产生多个位置列表。为了避免必须针对相应于一个或多个位置列表中的每个位置的一个条目搜索整个 DocID 映象图 410，第一级查找表 406 具有针对记号空间资料库中每个位置块的条目。例如，每个块可以具有 32768 个位置的大小，而且每个条目可以具有指向针对相应位置块的 DocID 查找表 410 中的第一条目的指针。于是，第一级查找表 406 可以把一个或多个位置列表转换成第二级查找表 410 中的文档标识符 (DocID 条目)412 的开始点位置。作为选择，也可以把表 406 和 410 统称为 DocID 查找表。第二级查找表 410 中的每个条目 412 包括 DocID（文档标识符）以及相应文档的开始资料库位置。在第二级查找表 410 中，任何文档中的最后一个记号均处于紧在下一个标目 412 所标识的开始位置之前的位置。第二级查找表 410 接收针对 DocID 的开始点位置 $Start_Pos_{A-Z}$，第二级查找表 410 把开始点位置转换成针对每个查询项的一个 DocID 的列表。

[0077] 在某些实施例中，第一级查询处理器 416 包括用于产生结果集合的逻辑 416。逻辑 416 根据查询或查询树所指定的布尔逻辑合并 DocID 的列表，以形成 DocID 的结果集合。逻辑 416 还可以有选择地过滤记号位置的列表，以消除没有位于相应于结果集合中的 DocID 的文档中的记号位置。而且，还可以使用 DocID 的所标识的每个文档中的 DocID 和记号位置，将记分功能施加到结果集合，以使得得分（有时将其称为查询得分）与结果集合中的每个 DocID 相关联。

[0078] 多级查询处理

[0079] 图 5 为利用记号空间资料库 524 的多级查询处理系统 500 的实施例的方框图。在某些实施例中，查询处理系统 500 包括 4 个查询处理和相关性得分生成级，即包括第一级查询处理器 510、第二级查询处理器 514、第三级查询处理器 518 以及第四查询级处理器 520。注意，在系统 500 中，可以使用或多或少的查询处理器级，取决于具体的应用。根据应用场合，每级计算一个或多个可以返回给用户的相关性得分集合与 / 或把它们与先前级中所生成的相关性得分加以组合。

[0080] 查询处理 -- 级 1

[0081] 已针对图 4 一般性地描述了第一级查询处理器 510。查询解析器 504 对查询串 502 进行记号化，并且将其解析成查询项（即，把查询中的每个不同的项作为记号加以对待）。记号化的查询项被全局词典映射 508 使用转换表或映射转换成相应的 GTokenID，如先前参照图 2 和 4 所描述的。由于用户可以在他们的查询串中使用特定的操作符 (operator)，包括布尔、邻接、或相近性操作符，所以系统 500 可以将查询解析成查询项和操作符。操作符可以按特定化格式（例如，AND、OR），以保留的标点（例如，问号）或者保留的项 (term) 的形式出现。在自然语言处理 (NLP) 系统的情况下，无论可以如何表达操作符，都能够在所使用的语言中隐含地识别操作符（例如，介词、连接词、次序关系等）。第一级查询处理器 510 中也可以包括其它的查询处理，例如删除结束单词（例如“a”、“the”等）以及项干（即，去除单词前缀）。

[0082] 接下来，查询扩展器 506 处理 GTokenID 的列表，查询扩展器 506 生成查询树或其它的查询表示，并且考虑到查询串中所使用的任何操作符（例如布尔表达式）。作为选择，查询扩展器 506 还可以按各种方式扩展查询。例如，可以把查询项转换成子树，该子树包含所述项和一个或多个同义词项，或者与查询项相关的其它项，子树中的项通过 OR 操作符或者父结点而彼此相关。

[0083] 如以下将更详细地加以描述的，在某些实施例中，根据图 5 中所示的查询处理级的序列，一次或多次地处理查询。在每个遍（除最后一遍）中，生成附加查询扩展项（以下将对此加以解释），然后，把这些附加项添加到查询树。也可以把查询树用作记分树，记分树具有与查询树中的项相关的权重。所扩展的查询树还可以包括不要求出现在相应于查询的文档中，但将它们用于对相应于查询的文档的相关性的记分的补充的项和项的子树。如果存在一个以上的查询项，则在第一遍期间，可以针对查询项计算权重，以改进搜索结果。

[0084] 在某些实施例中，遍历系统 500 的第一遍处理来自文档集的文档的随机样本。可以根据能够由系统 500 使用的一个或多个较小的随机样本，选择随机样本的大小，以估计整个文档集中匹配查询的文档的数目。在其它一些实施例中，在遍历系统 500 的第一遍中，使用第一文档集（例如，查询对话期的一个集合），以及在遍历系统 500 的第二或相继的遍，使用第二、不同的文档集。使用先前的查询对话期集合，可以使系统 500 能够确定通常共同出现在类似查询中的其它相关的项。查询扩展器 506 可以使用这些相关项扩展相继各遍的查询。

[0085] 第一级查询处理器 510 使用查询项搜索一个记号空间逆索引 512，并且标识与查询相匹配的文档。第一级查询处理器 510 访问记号空间逆索引 512，以产生查询树中项的记号位置（也称为记号空间资料库位置）的列表，并且访问 DocID 映象图 516，以产生针对相应于记号位置的文档的 DocID 的集合。另外，第一级处理器 510 还执行由查询或查询树所指出的布尔逻辑，以生成相应于查询的 DocID 的集合。在某些实施例中，第一级查询处理器 510 还根据一个或多个记分算法，计算查询和每个文档之间的相关性得分的第一集合 S_1。总之，记分算法根据一个或多个查询特性向每个匹配的文档提供相关性，所述查询特性包括，但不局限于一个或多个查询项的存在或不存在、项频度、布尔逻辑实现、查询项权重、文档的流行性（例如，一个独立于文档的重要性、或流行性、或互连性的得分）、查询项相互间的相近性、上下文、属性等。在一个实施例中，相关性得分 S_1 的第一集合基于包括查询项的存在性、项频度以及文档流行性的一组因素。

[0086] 在某些实施例中，相关性得分的第一集合 S_1 可用于选择作为有序列表呈现给用户的文档，然后，用户可以简单地点击和跟随指向所选择的文档的内部指针。在其它一些实施例中，把相关性得分的第一集合 S_1 以及 DocID 和相应的位置提供于第二级查询处理器 514，以进行进一步的处理。

[0087] 查询处理 -- 级 2

[0088] 第二级查询处理器 514 从第一级查询处理器 510 接收 DocID 的集合、针对相应文档的记号空间资料库位置的列表、以及相关性得分的第一集合 S_1。第二级查询处理器 514 使用所述位置列表，根据文档中所发现的查询项的相近性或相对位置，生成相关性得分的第二集合 S_2。当在一个文档中，查询中的项相互靠近地出现时，与各项按较大距离出现时相比，所述文档更可能与所述查询相关。因此，与其中项按某一距离出现的文档相比，如果查

询项互相相邻地出现，即相近地出现，则相关性得分的第二集合 S_2 被用于将文档的等级排得较高。在某些实施例中，相关性得分的第二集合 S_2 可用于选择作为有序列表用于呈现给用户的顶部的 X 文档，然后，用户可以简单地点击和跟随指向所选择的文档的内部指针。在某些实施例中，部分地根据相关性得分的第一集合 S_1 导出相关性得分的第二集合 S_2（例如，通过根据第二级查询处理器 514 所使用的附加的得分因素，调整 S_1 得分），以生成一个向用户提交的、与 / 或由第三级查询处理器 518 进一步处理的文档的有序列表。

[0089] 查询处理 -- 级 3

[0090] 在某些实施例中，把第二级查询处理器 514 耦接于第三级查询处理器 518，以处理已经在一个属性表 522 中加以编码的项属性（例如字体属性、题目、标题、元数据等），如以上参照图 3C 所描述的。第三级查询处理器 518 从第二级查询处理器 514 接收 DocID 的集合、针对相应文档的记号空间资料库位置的列表，以及相关性得分的第二集合 S_2。作为选择，第三级查询处理器也可以接收相关性得分的第一集合 S_1 以及相关性得分的第二集合 S_2。

[0091] 某些研究表明，文档中的项的部位表示其对文档的重要性。例如，在权重方面，出现在与某一查询相匹配的文档的题目中的项，可能重于出现在该文档的体中的查询项。相类似，出现在章节标题或文档第一段中的查询项与出现在文档中的不太重要位置中的项相比，很可能更能说明文档与所述查询的相关性。可以用于指示相关性的其它属性包括粗体文本、下划线文本以及字体大小。因此，使用与所述查询项相匹配的文档中的记号的属性，确定相关性得分的第三集合 S_3。参照图 3C，为了访问文档中的查询项的属性（即，与查询项相匹配或相关联的记号的属性），使用该文档中的查询项的记号位置，执行进入属性表 316（图 5 中的 522）的索引操作。更具体地讲，如果由每个属性记录 318 对其属性进行编码的记号的数目为 K，则把除以 K 的记号位置用于进入属性表 316 的索引。在某些实施例中，把所标识的属性一个或多个记录 318 以编码的、压缩的格式加以存储，因此，为了确定与每个查询项相关的属性，必须对其进行解码。

[0092] 在某些实施例中，相关性得分的第三集合 S_3 可用于选择作为有序列表向用户提交的顶部的 Y 个文档，然后，用户可以简单地点击和跟随指向所选择的文档的内部指针。在某些实施例中，部分地根据一个或多个相关性得分的第一和第二集合 S_1 和 S_2 导出相关性得分的第三集合 S_3，以生成呈现给用户的、与 / 或由第四查询级处理器 520 进一步处理的文档的有序列表。在某些实施例中，通过根据第三级查询处理器 518 所产生的附加的得分因素，调整 S_2 得分，产生 S_3 得分。

[0093] 查询处理 -- 级 4

[0094] 第四查询级处理器 520 从第三级查询处理器 518 接收 DocID 的集合、相应于所述 DocID 的文档中的位置的列表、以及相关性得分的第三集合 S_3。第四查询级处理器 520 还可以有选择地接收相关性得分的第一与 / 或第二集合 S_1 和 S_2。第四查询级处理器 520 耦接于解码系统 527，接着解码系统 527 耦接于一个或多个小词典映射 523、记号空间资料库 524 以及一个或多个全局词典映射 508。以上，针对图 1 和图 2，描述了小词典映射 523、记号空间资料库 524 以及全局词典映射 508。

[0095] 第四查询级处理器 520 根据上下文生成相关性得分的第四集合 S_4，并且还可以为结果集合中所列的一个或多个文档生成“片段”。片段来自文档的小部分文本，通常包括出现在将加以搜索的关键字周围的文本。在一个实施例中，为了生成针对结果集合中所列的

文档的片段，查询处理器对位于出现在文档中的每个查询项第一次出现之前和之后的预确定数目的记号进行解码，从而重构了文档的一个或多个文本部分，然后选择将包括在片段中的文本部分的子集。使用结果集合中的位置列表，解码系统 527 可以选择对文档的一些部分进行解码所需的小词典 523，其中文档的所述一些部分为文档中查询项出现之前和之后的一些部分。所选择的小词典 523 和全局词典 508 用于把记号空间资料库中的 LTokenID 转换成 GTokenID，然后用于把 GTokenID 转换成记号，如以上针对图 2 所描述的。

[0096] 在某些实施例中，相关性得分的第四集合 S_4 可用于选择作为有序列表向用户提交的顶部的 Z 个文档，然后，用户可以简单地点击和跟随指向所选择的文档的内部指针。在某些实施例中，部分地根据一个或多个相关性得分的第一、第二以及第三集合 S_1、S_2 以及 S_3 导出相关性得分的第四集合 S_4，以生成向用户提交的、与 / 或由相关性反馈模块 517 进一步加以处理的文档的有序列表。在一个可选的实施例中，最后级查询处理器为那些在先前查询处理器级所产生的相关性得分方面具有最高得分的文档生成片段，但不生成相关性得分的一个新的集合 S_4。

[0097] 在某些实施例中，把相关性得分的最终集合提供于一个相关性反馈模块 517，相关性反馈模块 517 根据最后查询级所产生的结果集合中的文档，生成一个或多个新的查询扩展项。例如，相关性反馈模块 517 可以实现一个或多个已知的相关性反馈算法，包括，但不局限于基于一个全文档方案的伪相关性反馈算法（基于某一完整 Web 页的伪相关性反馈算法），文档对象模型 (DOM) 分段、基于页分段的显示 (VIPS)、使用概念点阵的概念相关反馈等。相关性反馈算法可以分析根据先前查询处理级所核实 (vet) 的文档，并且可以根据分析结果生成查询扩展项。把新的查询扩展项提供于查询扩展器 506，查询扩展器 506 生成将由一个或多个查询处理器 510、514、518 以及 520 所处理的新的查询表达式。于是，多级查询处理系统 500 能够针对一个查询执行两遍或两遍以上，并且使用来自每个遍的信息，生成改进了的查询，从而将最终致使用户可以接收更多相关的文档。

[0098] 在一个实施例中，当执行查询的第一遍处理时，最后查询级处理器 520 产生一些长的片段，例如包括文档中查询项每次出现之前和之后的 N 个（例如 10 ～ 40 个）记号。如果片段超过一个预先定义的长度，则可以截断该片段。把查询和最后查询级 520 所产生的片段与相关性得分一起提供于相关性反馈模块 517，以生成查询扩展项的集合，而且作为选择，也可生成查询项权重的集合。在扩展的查询的第二遍处理期间，最后查询级 520 产生一些长度适当的短的片段，以及用于随具有最高，即最佳得分的结果集合中的文档列表一起显示的内容。

[0099] 在一个实施例中，查询处理系统包括 L 个并行的查询处理子系统，每个查询处理子系统包括针对一组文档的相应子集的逆索引 512 和记号空间资料库 524。例如，查询处理系统可以包括 1000 多个并行查询处理子系统。所有查询处理子系统可以共享相关性反馈模块 517（图 5）。在遍历查询处理系统的第一遍期间，由并行查询处理子系统的一小部分处理查询，而在第二遍期间，由整个查询处理系统处理查询。例如，可以把查询处理系统划分成 S 个子集（例如 32 个子集），并且根据把杂凑函数施加到查询的正常化的版本的结果，把每个查询赋予这些子集之一，然后把取模函数施加到杂凑函数所产生的结果。可以把查询处理系统的每个子集称为查询处理系统的“分区”，并且可以把每个查询处理子系统称为“子分区”。

[0100] 查询的第一遍处理的主要目的是，产生查询扩展项的集合，以及查询项权重，以改进由查询的第二遍处理所产生的查询结果的质量。只要在查询处理系统中把文档适当、随机地在所有查询处理子系统之间加以分布，则查询仅需少数子系统加以处理，就可以产生查询扩展项的一个集合。查询扩展器 506 使用查询扩展项产生扩展的查询树或查询表达式，然后由查询处理级（在查询的一个第二遍处理中）对它们加以处理，如以上所描述的。例如，可以把查询“纽约的照片”扩展为“纽约（照片或图像，或者图像或照片）”。可以对第二遍期间最后查询级所产生的结果集合和片段进行格式化，以由从其处接收查询的计算机或设备加以显示（或者更一般地讲，加以提交）。

[0101] 在一个实施例中，在与相继遍不同的数据资料库上执行查询的第一遍处理。例如，针对第一遍的初始数据资料库，可以为先前处理过的查询的一个数据资料库，而用于相继遍的数据资料库可以为具有用于把查询映射到该数据资料库中的文档的一个逆索引的文档集合。

[0102] 文档处理服务器

[0103] 图 6 为记号空间资料库服务器 600 的实施例的方框图。服务器 600 可以为独立的计算机系统或是包括多个计算机系统的分布式处理系统的一部分。通常，服务器 600 包括一个或多个处理单元 (CPU)604、一个或多个网络或其它通信接口 608、存储器 602、以及用于互连这些部件的一条或多条通信总线 606。服务器 600 可以有选择地包括用户接口，例如，显示器和键盘。存储器 602 可以包括高速随机存取存储器，还可以包括非易失存储器，例如一个或多个磁盘存储设备。存储器 602 可以包括远离一个或多个中央处理器 604 的海量存储器。

[0104] 存储器 602 存储操作系统 610（例如，Linux 或者 Unix）、网络通信模块 612、词典生成器 614（例如词典生成器 108）、编码系统 616（例如，编码系统 300）、一个或多个全局词典 618（例如，全局词典 206）、一个或多个小词典 620（例如，小词典 208）、记号空间资料库 622（例如，记号空间资料库 112）、属性记录 624（例如，属性记录表 316）、以及有效范围映射 626（例如，有效范围映射 210）。以上已经针对图 1 ～ 5 描述了这些部件中每个部件的操作。

[0105] 查询处理服务器

[0106] 图 7 为查询处理服务器 700 的实施例的方框图。服务器 700 可以为独立的计算机系统或为包括多个计算机系统的分布式处理系统的一部分。通常，服务器 700 包括一个或多个处理单元 (CPU)704、一个或多个网络或其它通信接口 708、存储器 702、用于互连这些部件的一条或多条通信总线 706。服务器 700 可以有选择地包括用户接口，例如，显示器和键盘。存储器 702 可以包括高速随机存取存储器，还可以包括非易失存储器，例如一个或多个磁盘存储设备。存储器 702 可以包括远离一个或多个中央处理器 704 的海量存储器。

[0107] 存储器 702 存储操作系统 710（例如，Linux 或者 Unix）、网络通信模块 712、记号空间逆索引 714（例如，记号空间逆索引 408）、解码系统 716（例如，解码系统 308）、一个或多个词典转换表或映射 718（例如，从全局词典 206 和小词典小词典 208 所导出的）、有效范围映射 720（例如，有效范围映射 210）、DocID 映射 722（例如，DocID 映射 410）、查询解析器 724（例如，查询解析器 504）、查询树 726、一个或多个查询处理器 728（例如，查询处理器 510、514、518 以及 520）、属性记录 730（例如，属性记录表 316）、以及记号空间资料库

732(例如,记号空间资料库 112)。以上已经针对图 1 ～ 5 描述了这些部件中每个部件的操作。

[0108] 已参照具体的实施例,解释性地进行以上的描述。然而,这些说明性的讨论,不旨在穷举性地描述本发明,也不旨在把本发明限制于所公开的精确形式。鉴于以上的描述,对本发明的许多修改与改变是可能的。这些实施例的选择与描述,旨在充分地解释本发明的原理及其实际的应用,从而可使这一技术领域中的其他熟练技术人员能够对本发明和不同的实施例进行多方面的修改,以适应所考虑的具体应用,从而可充分利用本发明和不同的实施例。

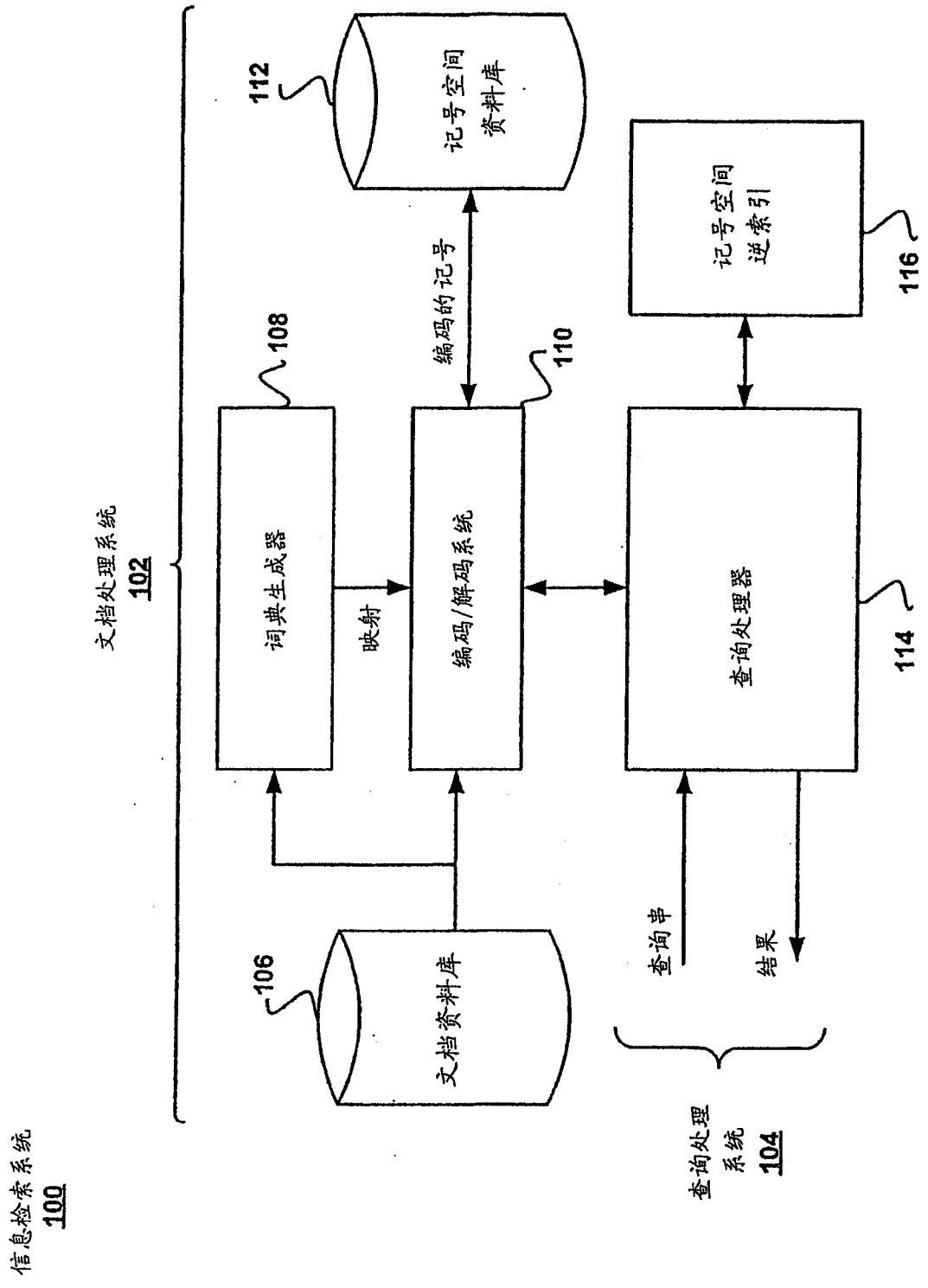

信息检索系统
100
文档处理系统
102
查询处理系统
104
文档资料库
106
词典生成器
108
编码/解码系统
110
记号空间
资料库
112
查询处理器
114
记号空间
逆索引
116
编码的记号
映射
查询串
结果

图 1

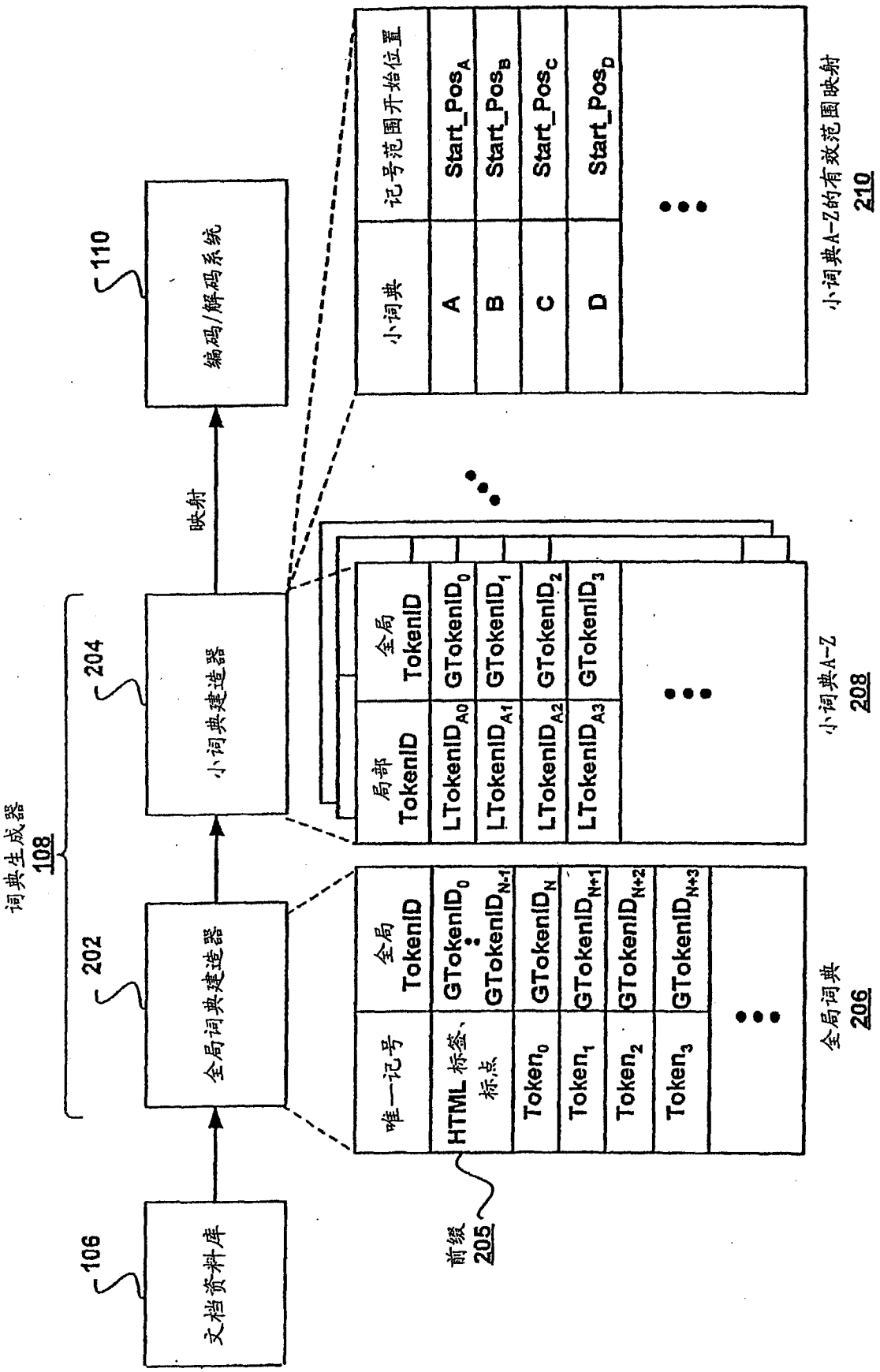

词典生成器
108
106
文档资料库
202
全局词典建造器
204
小词典建造器
110
编码/解码系统
映射
唯一记号
全局 TokenID
HTML 标签、标点
Token0
Token1
Token2
Token3
GTokenID0 ... GTokenIDN-1
GTokenIDN
GTokenIDN+1
GTokenIDN+2
GTokenIDN+3
前缀
205
全局词典
206
局部 TokenID
全局 TokenID
LTokenIDA0
LTokenIDA1
LTokenIDA2
LTokenIDA3
GTokenID0
GTokenID1
GTokenID2
GTokenID3
小词典A-Z
208
小词典
记号范围开始位置
A
B
C
D
Start_PosA
Start_PosB
Start_PosC
Start_PosD
小词典A-Z的有效范围映射
210

图 2

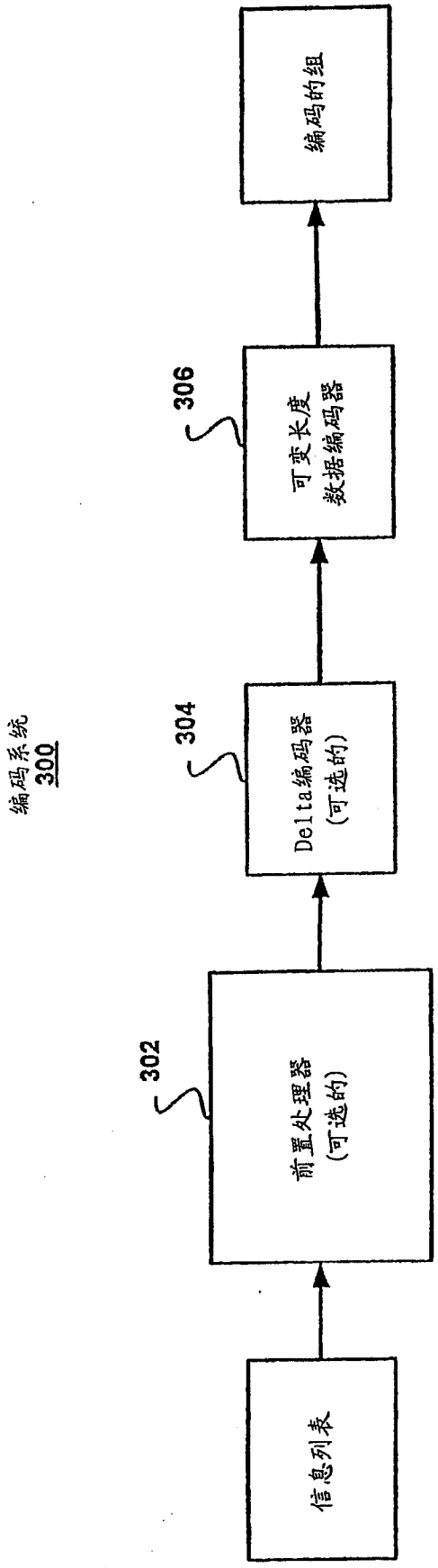

编码系统
300
信息列表
前置处理器
(可选的)
302
Delta编码器
(可选的)
304
可变长度
数据编码器
306
编码的组

图 3A

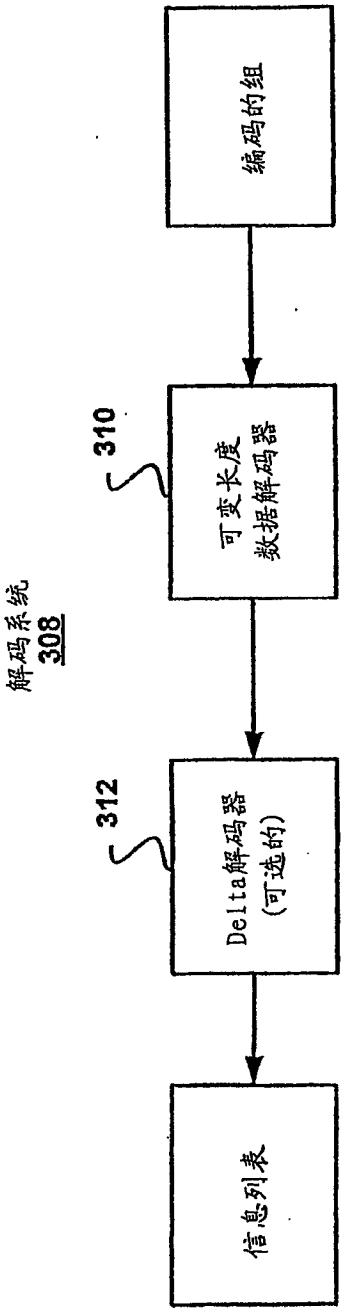

解码系统
308
编码的组
可变长度
数据解码器
310
Delta解码器
(可选的)
312
信息列表

图 3B

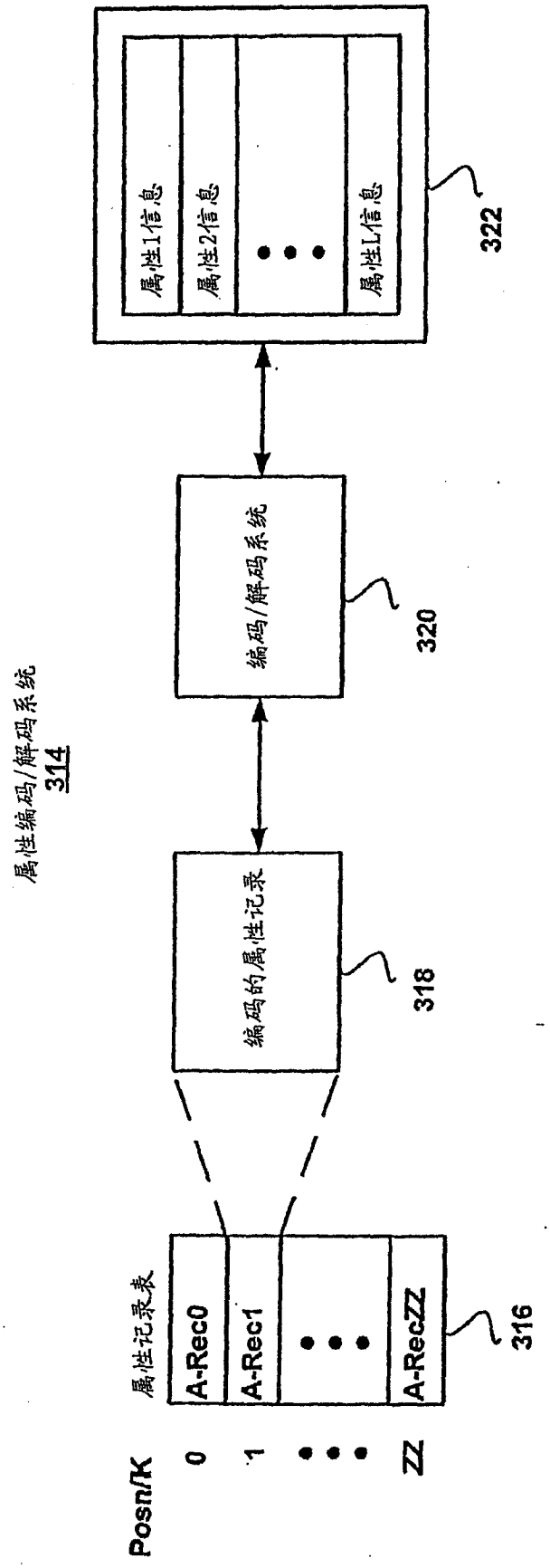

属性编码/解码系统
314
属性1信息
属性2信息
属性L信息
322
编码/解码系统
320
编码的属性记录
318
属性记录表
A-Rec0
A-Rec1
A-RecZZ
316
Posn/K
0
1
ZZ

图 3C

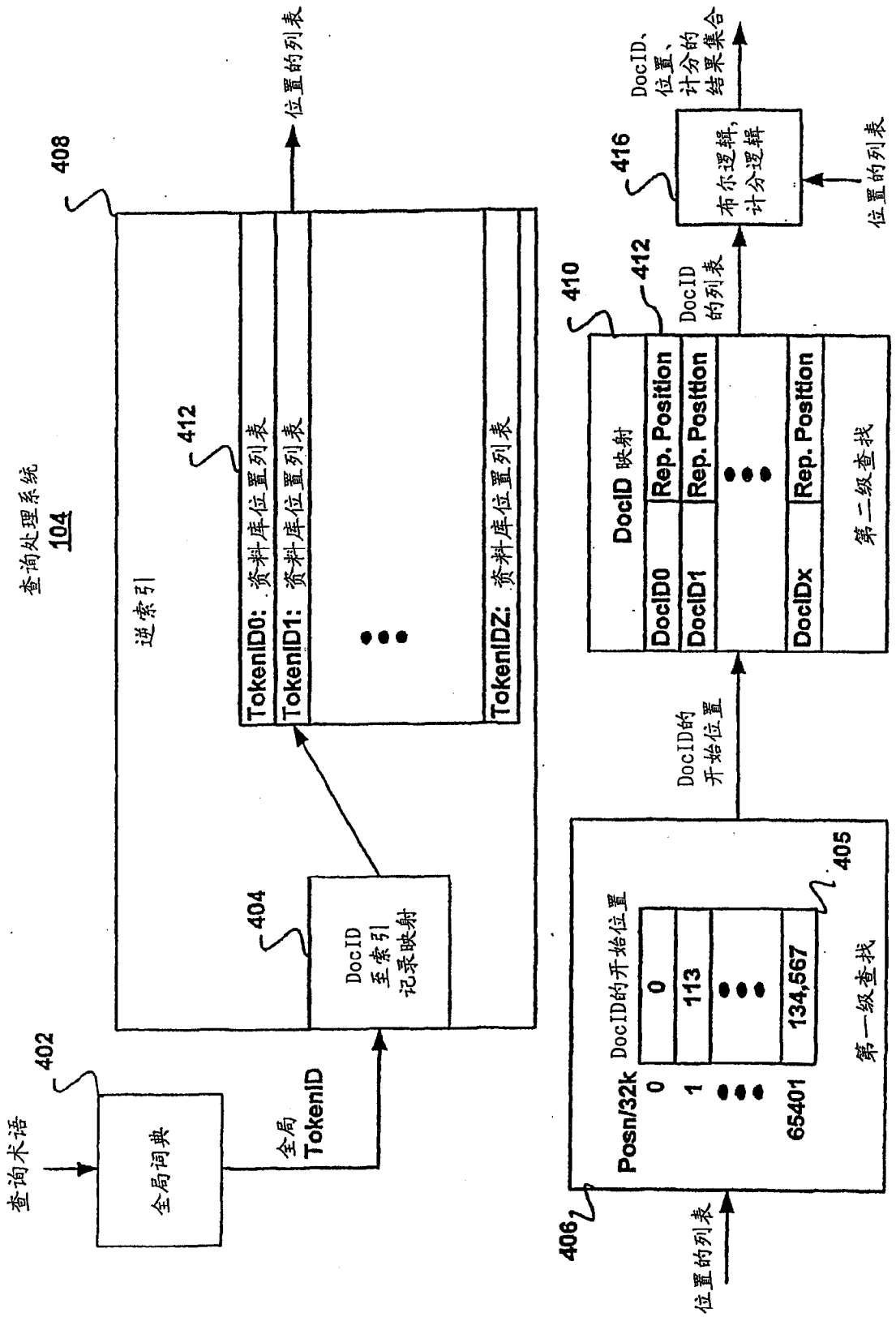

查询处理系统
104
408
412
逆索引
TokenID0: 资料库位置列表
TokenID1: 资料库位置列表
TokenIDZ: 资料库位置列表
位置的列表
404
DocID
至索引
记录映射
402
查询术语
全局词典
全局
TokenID
DocID的
开始位置
410
412
DocID 映射
DocID0
DocID1
DocIDx
Rep. Position
Rep. Position
Rep. Position
第二级查找
DocID
的列表
416
布尔逻辑,
计分逻辑
DocID、
位置、
计分的
结果集合
位置的列表
406
Posn/32k
DocID的开始位置
0
1
65401
0
113
134,567
405
第一级查找
位置的列表

图 4

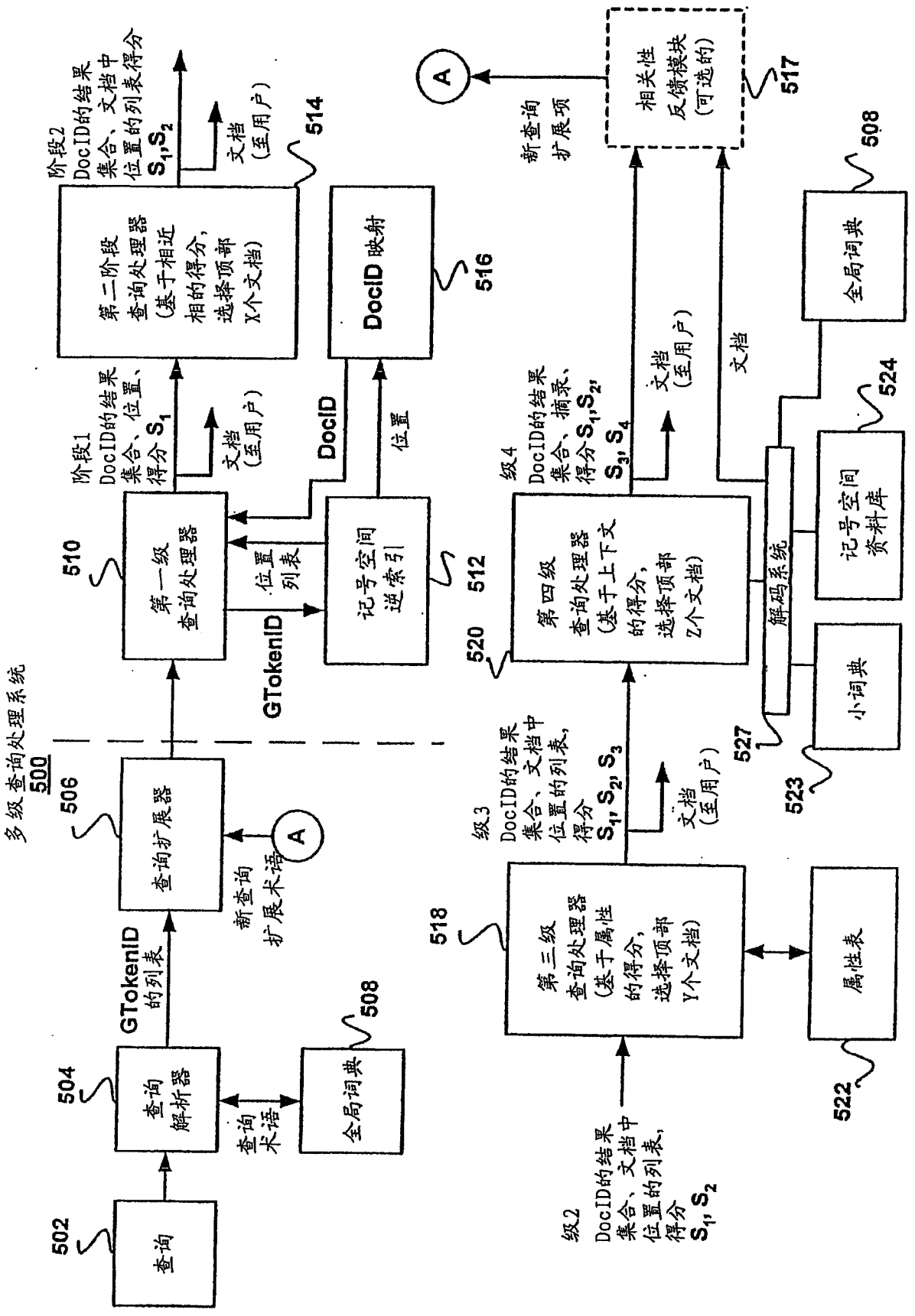

多级查询处理系统
500
查询
502
查询解析器
504
查询术语
全局词典
508
GTokenID的列表
查询扩展器
506
新查询扩展术语
A
第一级查询处理器
510
GTokenID
位置列表
记号空间逆索引
512
阶段1 DocID的结果集合、位置、得分 S_1
文档(至用户)
第二阶段查询处理器(基于相近相的得分,选择顶部X个文档)
阶段2 DocID的结果集合、文档中位置的列表得分 S_1, S_2
文档(至用户)
514
DocID
位置
DocID映射
516
A
新查询扩展项
相关性反馈模块(可选的)
517
级2 DocID的结果集合、文档中位置的列表,得分 S_1, S_2
第三级查询处理器(基于属性的得分,选择顶部Y个文档)
518
属性表
522
级3 DocID的结果集合、文档中位置的列表,得分 S_1, S_2, S_3
文档(至用户)
第四级查询处理器(基于上下文的得分,选择顶部Z个文档)
520
级4 DocID的结果集合、摘录、得分 S_1, S_2, S_3, S_4
文档(至用户)
文档
解码系统
527
小词典
523
记号空间资料库
524
全局词典
508

图 5

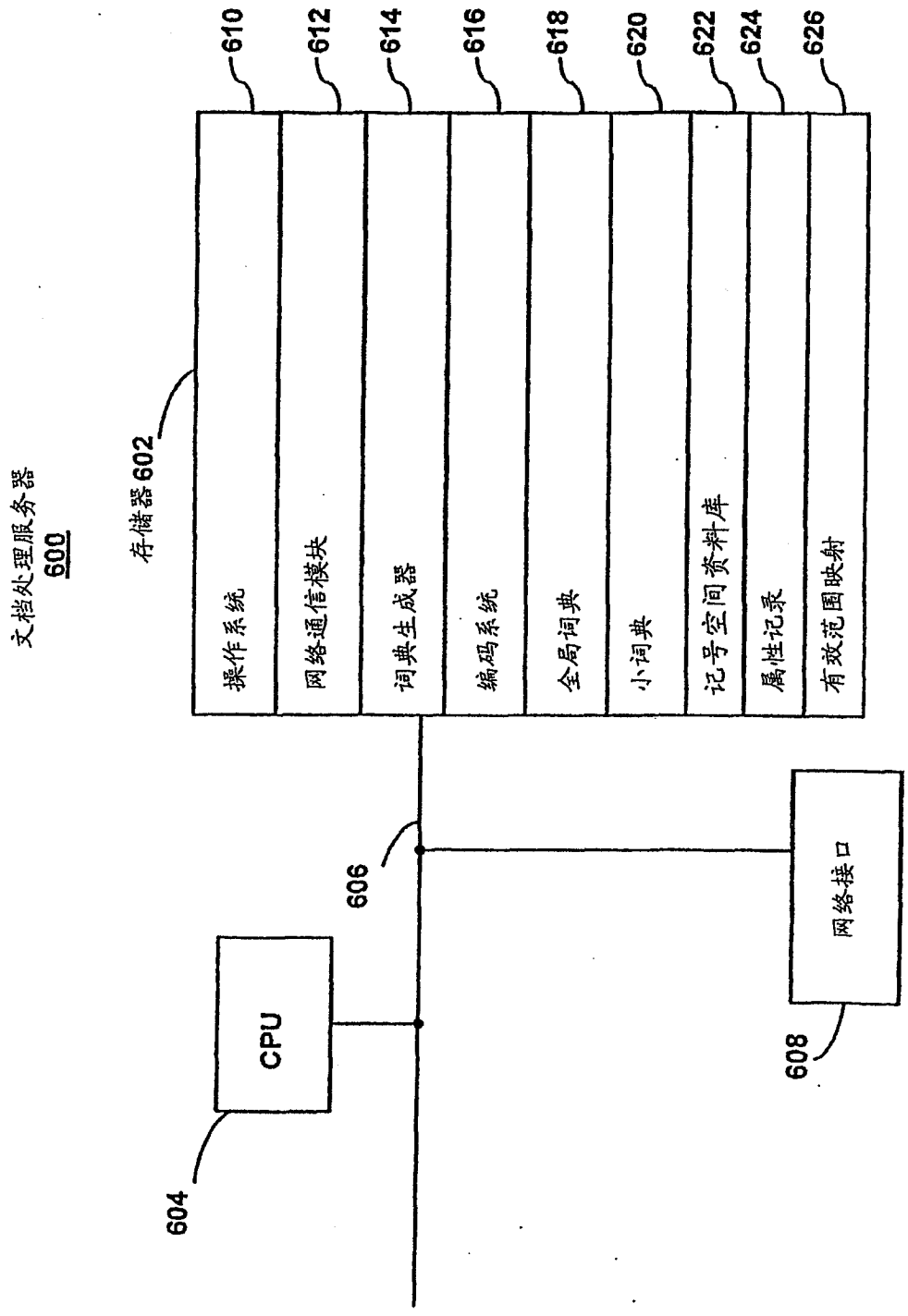

文档处理服务器
600
存储器 602
610
612
614
616
618
620
622
624
626
操作系统
网络通信模块
词典生成器
编码系统
全局词典
小词典
记号空间资料库
属性记录
有效范围映射
606
CPU
604
网络接口
608

图 6

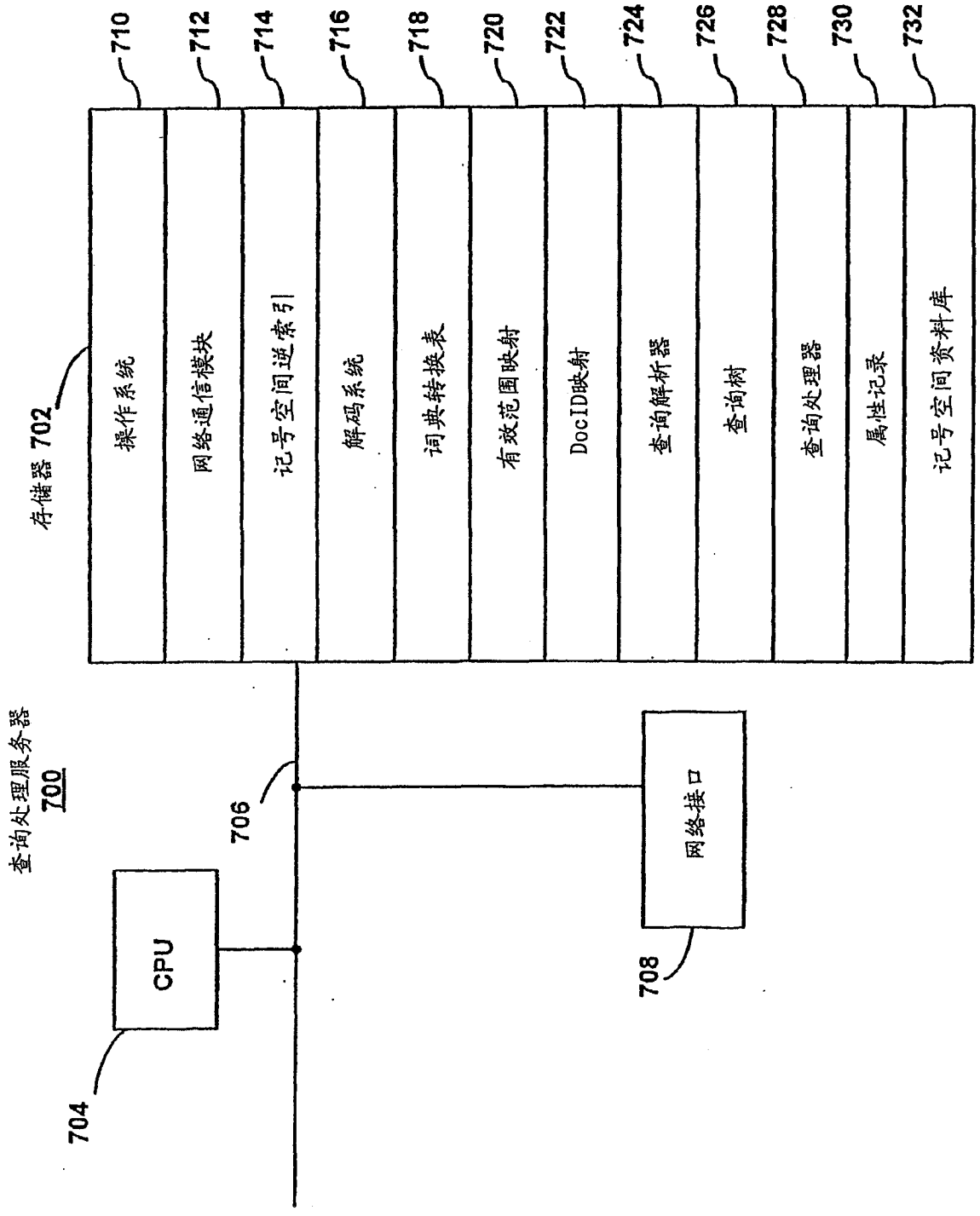

查询处理服务器
700
存储器 702
操作系统
网络通信模块
记号空间逆索引
解码系统
词典转换表
有效范围映射
DocID映射
查询解析器
查询树
查询处理器
属性记录
记号空间资料库
710
712
714
716
718
720
722
724
726
728
730
732
706
CPU
704
网络接口
708

图 7

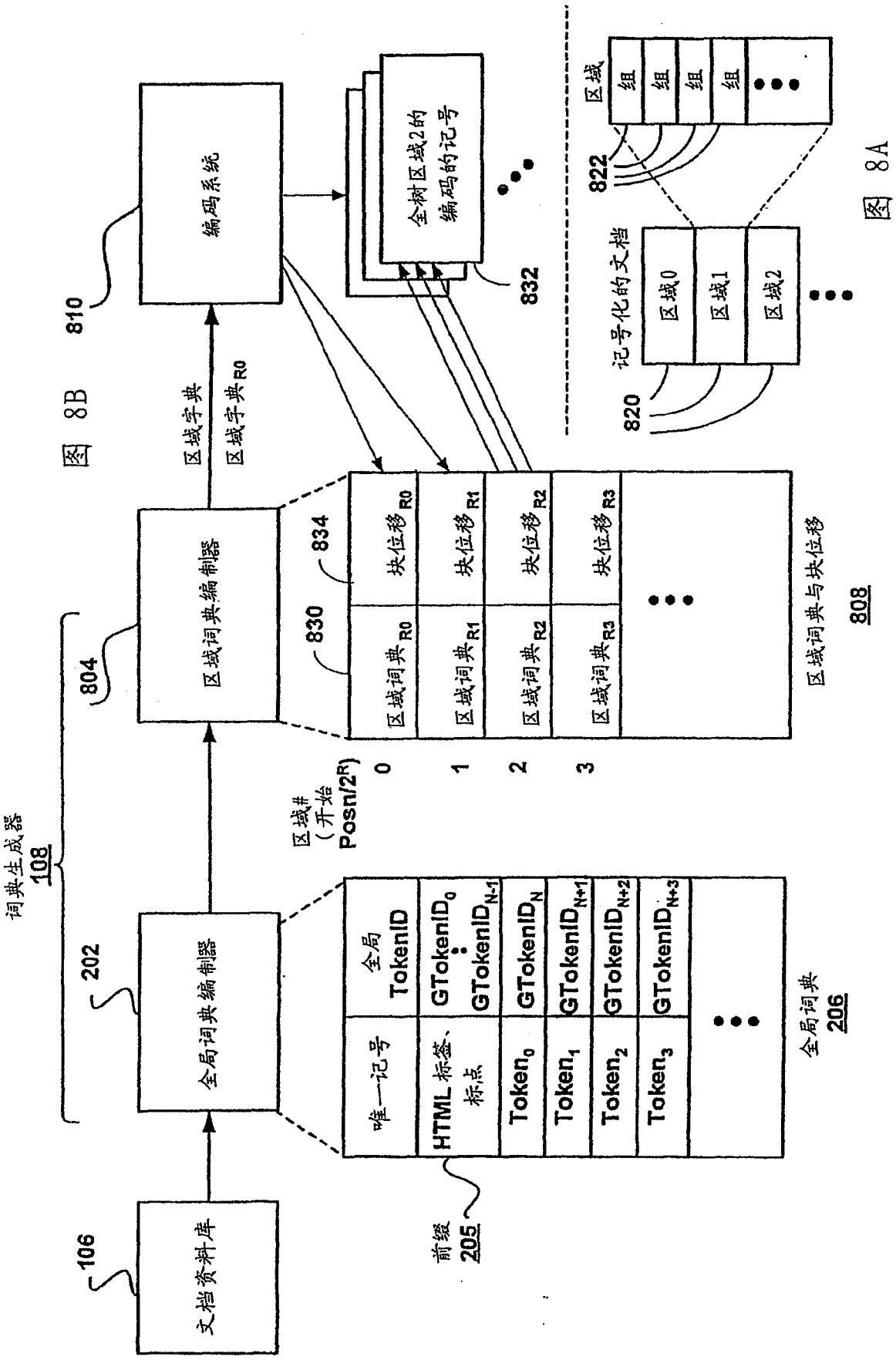

词典生成器
108
106
文档资料库
202
全局词典编制器
804
区域词典编制器
810
编码系统
区域字典
区域字典R0
全树区域2的编码的记号
832
区域
组
822
记号化的文档
区域0
区域1
区域2
820
唯一记号
HTML 标签、标点
全局 TokenID
Token0
Token1
Token2
Token3
GTokenID0 : GTokenIDN-1
GTokenIDN
GTokenIDN+1
GTokenIDN+2
GTokenIDN+3
前缀
205
全局词典
206
区域# (开始 Posn/2R)
0
1
2
3
区域词典R0
区域词典R1
区域词典R2
区域词典R3
830
834
块位移R0
块位移R1
块位移R2
块位移R3
区域词典与块位移
808

图 8B

图 8A

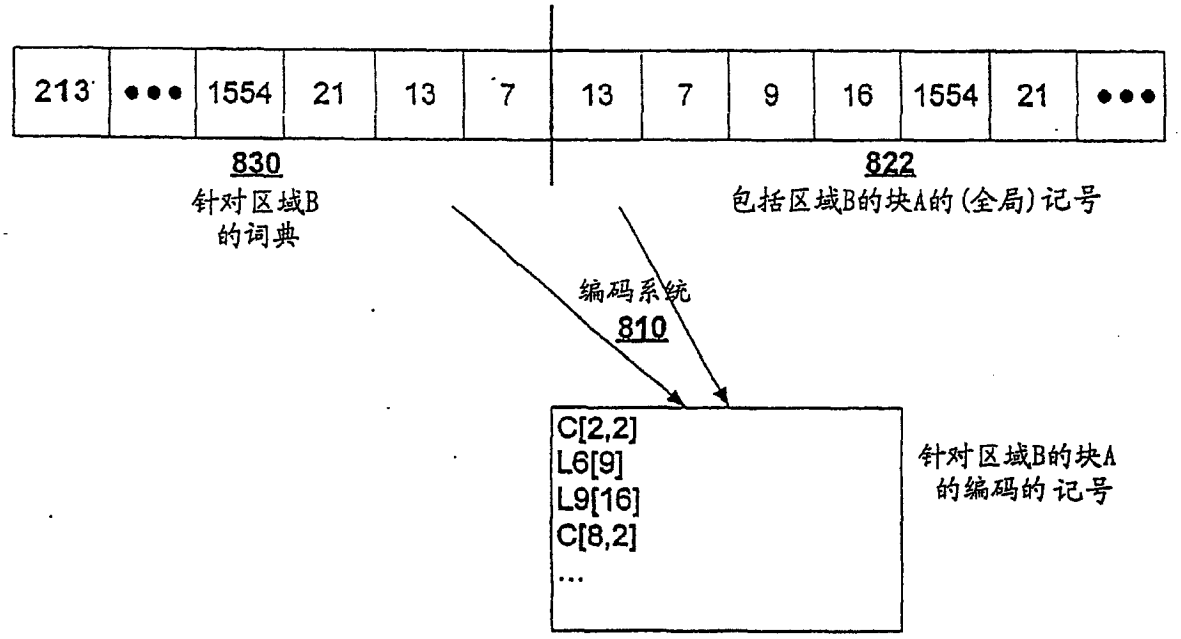

213
•••
1554
21
13
7
13
7
9
16
1554
21
•••
830
针对区域B
的词典
822
包括区域B的块A的(全局)记号
编码系统
810
C[2,2]
L6[9]
L9[16]
C[8,2]
...
针对区域B的块A
的编码的记号

图 9A

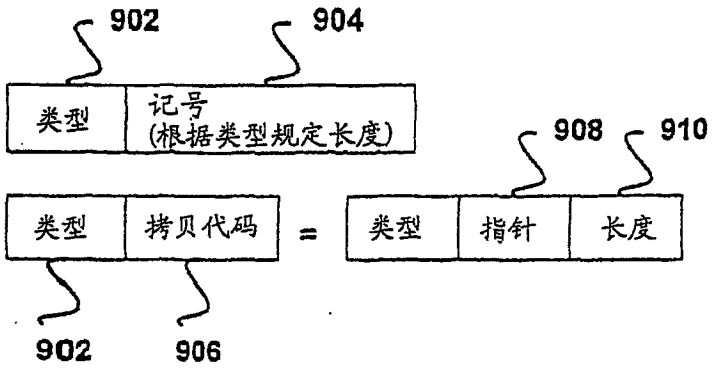

902
904
类型
记号
(根据类型规定长度)
908
910
类型
拷贝代码
=
类型
指针
长度
902
906

图 9B