



(12) 发明专利

(10) 授权公告号 CN 101681377 B

(45) 授权公告日 2013. 03. 27

(21) 申请号 200880017123. 9

(22) 申请日 2008. 05. 16

(30) 优先权数据

11/805, 380 2007. 05. 23 US

(85) PCT申请进入国家阶段日

2009. 11. 23

(86) PCT申请的申请数据

PCT/US2008/063991 2008. 05. 16

(87) PCT申请的公布数据

W02008/147736 EN 2008. 12. 04

(73) 专利权人 微软公司

地址 美国华盛顿州

(72) 发明人 D·丘林 J·陆 N·哈诺卡

S·纳拉亚南 T·纳亚克 F·田

(74) 专利代理机构 上海专利商标事务所有限公
司 31100

代理人 陈斌 钱静芳

(51) Int. Cl.

G06F 17/30 (2006. 01)

(56) 对比文件

US 7181438 B1, 2007. 02. 20, 说明书第 3 栏第 45-67 行, 第 4 栏第 40-49 行, 第 6 栏第 36-56 行, 第 8 栏第 31-54 行, 第 9 栏第 6-29 行, 第 11 栏第 32 行-12 栏第 32 行, 第 18 栏第 62 行-第 19 栏

第 15 行, 19 栏第 36-55 行, 第 22 栏第 5 行-26 栏第 14 行, 第 25 栏第 39 行-26 栏第 36 行, 第 48 栏第 3-36 行, 第 53 栏第 15-42 行, 第 54 栏第 26-36 行, 第 63 栏第 51 行-第 64 栏第 13 行, 第 68 栏第 23-43 行, 72 栏第 1-54 行, 第 87 栏第 33-48 行、权利要求 27、说明书附图 3, 5, 6-10, 12.

CN 1755678 A, 2006. 04. 05, 全文.

CN 1517914 A, 2004. 08. 04, 全文.

US 2005/0177551 A1, 2005. 08. 11, 说明书第 85, 88 段.

US 7181438 B1, 2007. 02. 20, 说明书第 3 栏第 45-67 行, 第 4 栏第 40-49 行, 第 6 栏第 36-56 行, 第 8 栏第 31-54 行, 第 9 栏第 6-29 行, 第 11 栏第 32 行-12 栏第 32 行, 第 18 栏第 62 行-第 19 栏第 15 行, 19 栏第 36-55 行, 第 22 栏第 5 行-26 栏第 14 行, 第 25 栏第 39 行-26 栏第 36 行, 第 48 栏第 3-36 行, 第 53 栏第 15-42 行, 第 54 栏第 26-36 行, 第 63 栏第 51 行-第 64 栏第 13 行, 第 68 栏第 23-43 行, 72 栏第 1-54 行, 第 87 栏第 33-48 行、权利要求 27、说明书附图 3, 5, 6-10, 12.

审查员 刘光德

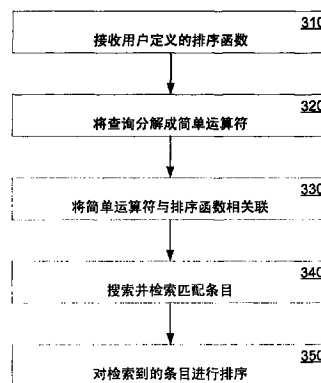
权利要求书 2 页 说明书 6 页 附图 5 页

(54) 发明名称

用于搜索的用户定义的相关性排序

(57) 摘要

本发明详细描述了一种特别地允许搜索引擎使用用户定义的关联函数的技术。在本技术的一种方法中, 描述了一种应用用户定义的关联函数的方法。在该方法中, 将复杂的搜索查询分解成简单运算符。将简单运算符与用户定义的关联函数相关联。检索与搜索查询匹配的文档, 并使用用户定义的关联函数来为该文档计算排名。



流程图 300

1. 一种应用用户定义的关联函数的方法,包括:

将复杂搜索查询分解成多个通过联结来连接的简单运算,所述多个简单运算包括包含同质逻辑运算符的简单运算符;

将所述简单运算符与所述用户定义的关联函数相关联;

检索与所述复杂搜索查询相匹配的匹配文档,包括使用流式表值函数 STVF 来执行索引访问以检索与对应于简单运算符的部分查询相匹配的行,所述行对应于匹配所述部分查询的文档;以及

使用所述用户定义的关联函数来为所述匹配文档计算排名。

2. 如权利要求 1 所述的方法,其特征在于,还包括:

接收所述用户定义的关联函数。

3. 如权利要求 1 所述的方法,其特征在于,还包括:

在数据库中搜索所述匹配文档。

4. 如权利要求 1 所述的方法,其特征在于,所述匹配文档包括由所述用户定义的关联函数用来计算所述排名的信息。

5. 如权利要求 1 所述的方法,其特征在于,所述计算包括所述用户定义的关联函数使用与所述匹配文档相关联的信息来计算所述排名。

6. 如权利要求 1 所述的方法,其特征在于,所述分解包括将同质逻辑表达式分解成多路跳跃归并联结运算符。

7. 如权利要求 1 所述的方法,其特征在于,所述复杂搜索查询包括布尔逻辑查询。

8. 如权利要求 1 所述的方法,其特征在于,所述分解包括将所述复杂搜索查询分解成运算符树。

9. 一种应用用户定义的关联函数的方法,包括:

接收用户定义的排序函数;

接收搜索查询;

将所述搜索查询分解成多个通过联结来连接的简单运算,所述多个简单运算包括包含同质逻辑运算符的简单运算符;

将所述多个简单运算中的每一个与所述用户定义的排序函数相关联;

检索对应于所述多个简单运算的多个结果,包括使用流式表值函数 STVF 来执行索引访问以检索与对应于简单运算符的部分查询相匹配的行,所述行对应于匹配所述部分查询的文档;以及

使用所述用户定义的排序函数来计算所述多个结果中的每一个的排名。

10. 如权利要求 9 所述的方法,其特征在于,还包括:

接收多个用户定义的排序函数;以及

将所述多个用户定义的排序函数与所述多个简单运算相关联。

11. 如权利要求 9 所述的方法,其特征在于,还包括:

组合所述多个结果的所述排名以产生所述多个结果中的每一个的总排名。

12. 如权利要求 11 所述的方法,其特征在于,所述组合包括:

从所述多个结果的所述排名中计算最大排名值。

13. 如权利要求 9 所述的方法,其特征在于,所述分解包括:

将所述搜索查询变换成搜索树；
标识所述搜索树中的同质运算符；以及
将所述同质运算符变换成多路跳跃归并联结运算符。

14. 如权利要求 13 所述的方法，其特征在于，所述多路跳跃归并联结运算符包括流式表值函数 STVF。

15. 如权利要求 9 所述的方法，其特征在于，所述搜索查询包括向量查询。

16. 如权利要求 15 所述的方法，其特征在于，所述向量查询包括多个关键字，且每一个关键字都与搜索权重相关联。

17. 一种应用用户定义的关联函数的方法，包括：

接收复杂搜索查询；
将所述复杂搜索查询解析成多个通过联结来连接的较简单的运算；
检索对应于所述多个较简单的运算的多个结果；以及
通过应用用户定义的关联函数来计算所述多个结果中的每一个的排名；
其中所述多个较简单的运算包括包含同质逻辑运算符的简单运算符；

其中检索对应于所述多个较简单的运算的多个结果包括使用流式表值函数 STVF 来执行索引访问以检索与对应于简单运算符的部分查询相匹配的行，所述行对应于匹配所述部分查询的文档。

18. 如权利要求 17 所述的方法，其特征在于，解析所述复杂搜索查询进一步包括：

将所述复杂搜索查询精简为搜索树；
将所述树中的同质运算符变换成多路跳跃归并联结运算符；以及
将所述搜索树的非叶节点变换成关系运算符。

19. 如权利要求 18 所述的方法，其特征在于，进一步包括将所述用户定义的关联函数作为所述搜索树的一部分来优化。

20. 如权利要求 17 所述的方法，其特征在于，所述用户定义的关联函数包括用声明性语言编写的函数。

21. 一种应用用户定义的关联函数的系统，包括：

用于接收复杂搜索查询的装置；
用于将所述复杂搜索查询解析成多个通过联结来连接的较简单的运算的装置；
用于检索对应于所述多个较简单的运算的多个结果的装置；以及
用于通过应用用户定义的关联函数来计算所述多个结果中的每一个的排名的装置；
其中所述多个较简单的运算包括包含同质逻辑运算符的简单运算符；

其中用于检索对应于所述多个较简单的运算的多个结果的装置包括用于使用流式表值函数 STVF 来执行索引访问以检索与对应于简单运算符的部分查询相匹配的行的装置，所述行对应于匹配所述部分查询的文档。

用于搜索的用户定义的相关性排序

[0001] 背景

[0002] 数字信息的基本问题之一在于如何有效地对大量数据进行排序,以便在任何给定时刻找出那些最相关的信息片段。为解决该问题,搜索引擎包括关联或排序函数。这些关联函数用于将不同的权重给予需要搜索准则的文档;所分配的权重或排名然后可用于进一步操纵信息池,例如通过按特定次序显示结果。

[0003] 许多搜索引擎,特别是那些用于与数据库交互的搜索引擎,使用 tf-idf 权重方案的某些变体,其中一项出现在特定文档中的频率针对逆文档频率,即对该项出现在文档池中的频率的度量来加权。不同的示例引擎将实现该方案的不同变体,且优化单独的搜索引擎以使用其特定关联函数。

[0004] 概述

[0005] 此处详细描述了一种特别地允许搜索引擎利用用户定义的关联函数的技术。在本技术的一种方法中,描述了一种应用用户定义的关联函数的方法。在该方法中,将复杂的搜索查询分解成简单运算符。将简单运算符与用户定义的关联函数相关联。检索与搜索查询匹配的文档,并使用该用户定义的关联函数来为该文档计算排名。

[0006] 在本技术的另一种方法中,描述了一种具有计算机可执行指令的计算机可读介质。在该方法中,接收用户定义的排序函数,并接收搜索查询。将该搜索查询分解成多个简单运算。这些简单运算中的每一个都与关联函数相关联。检索对应于这些较简单的运算的多个结果,并使用相关联的关联函数来为这些结果中的每一个计算排名。

[0007] 在本技术的另一种方法中,一种具有总线、处理器、存储器和数据存储设备的系统被配置成接收复杂搜索查询。该系统还被配置成将该复杂搜索查询解析成若干较简单的运算。该系统还被配置成检索对应于这些较简单的运算的结果,并通过应用用户定义的关联函数来为每一个结果计算排名。

[0008] 提供本概述以便以简化的形式介绍将在以下详细描述中进一步描述的一些概念。本概述并不旨在标识出所要求保护的主题的关键特征或必要特征,也不旨在用于限定所要求保护的主题的范围。

[0009] 附图简述

[0010] 合并在本说明书中并形成其一部分的附图示出了各实施例,并且与说明书一起用于解释所要求保护的主题的原理:

[0011] 图 1 是其上可实现各实施例的示例性计算系统的框图。

[0012] 图 2 是其上可实现各实施例的示例数据库表。

[0013] 图 3 是根据一个实施例的应用用户定义的关联函数的方法的流程图。

[0014] 图 4A 是根据一个实施例的示例数据库查询。

[0015] 图 4B 是根据一个实施例的搜索树的描绘。

[0016] 图 4C 是根据一个实施例的中间搜索树的描绘。

[0017] 图 4D 是根据一个实施例的查询树的描绘。

[0018] 详细描述

[0019] 现在将详细参考若干实施例。尽管本主题将结合各替换实施例来描述,但将理解,它们并不旨在将所要求保护的主体限于这些实施例。相反,所要求保护的主体旨在涵盖可被包括在权利要求书所定义的要保护的主体的精神和范围内的替换、修改和等效技术方案。

[0020] 此外,在以下详细描述中,阐明了众多具体细节以提供对所要求保护的主体的全面理解。然而,本领域的技术人员将认识到各实施例可以在没有这些具体细节或具有其等效物的情况下实施。在其它情况下,未详细描述公知的方法、过程、组件和电路以免不必要地使主题的各方面和特征显得晦涩难懂。

[0021] 以下详细描述中的各部分按照一种方法来呈现和讨论。尽管其步骤和排序在此处描述该方法的各操作的附图(例如,图3)中公开,但这些步骤和排序是示例性的。各实施例很好地适用于执行各种其他步骤或此处附图的流程图中所述的步骤的变型,并且按照与此处所描绘和描述的不同的顺序。

[0022] 本详细描述的某些部分按照过程、步骤、逻辑块、处理、以及对可在计算机存储器上执行的数据位的操作的其它符号表示来呈现。这些描述和表示是数据处理领域技术人员用来将其工作的实质最有效地传达给本领域的其他技术人员的手段。过程、计算机执行的步骤、逻辑块、处理等在此处且通常被认为是导致所需结果的自相一致的步骤或指令的序列。步骤是需要对物理量的物理操纵的那些步骤。尽管并非必需,但这些物理量通常采用能够在计算机系统中被存储、传输、组合、比较、或以其它方式操纵的电或磁信号的形式。原则上出于常见用法的原因将这些信号称作位、值、元素、符号、字符、项、数字等被证明有时是方便的。

[0023] 然而,应当记住,所有这些和类似术语都应适当的物理量相关联并且仅仅是应用于这些物理量的方便的标签。如从以下讨论中清楚的,除非具体地另外指明,否则可以理解,贯穿本发明,利用诸如“访问”、“编写”、“包括”、“存储”、“发送”、“遍历”、“关联”、“标识”等术语的讨论涉及计算机系统或类似的电子计算设备的动作和处理,其操纵计算机系统的寄存器和存储器中被表示为物理(电子)量的数据并将其变换成计算机系统存储器或寄存器或其他这样的信息存储、传输或显示设备中类似地被表示为物理量的其他数据。

[0024] 诸如计算系统环境10等计算设备通常包括至少某些形式的计算机可读介质。计算机可读介质可以是可由计算机访问的任何可用介质。作为示例而非限制,计算机可读介质可以包括计算机存储介质和通信介质。计算机存储介质包括以用于存储诸如计算机可读指令、数据结构、程序模块或其它数据这样的信息的任何方法或技术实现的易失性和非易失性、可移动和不可移动介质。计算机存储介质包括但不限于,RAM、ROM、EEPROM、闪存或其它存储器技术、CD-ROM、数字多功能盘(DVD)或其它光盘存储器、盒式磁带、磁带、磁盘存储或其它磁存储设备,或者任何其它可用于存储所需信息并可由计算机访问的介质。通信介质通常以诸如载波或其它传输机制等已调制数据信号来体现计算机可读指令、数据结构、程序模块或其它数据,并包括任意信息传送介质。术语“已调制数据信号”指的是以在信号中编码信息的方式设定或更改其一个或多个特征的信号。作为示例而非限制,通信介质包括有线介质,诸如有线网络或直接线连接,以及无线介质,诸如声学、RF、红外、和其它无线介质。以上的任何组合也应包括在计算机可读介质的范围内。

[0025] 某些实施例可以在由一个或多个计算机或其他设备执行的诸如程序模块等计算

机可执行指令的通用上下文中描述。一般而言,程序模块包括执行特定任务或实现特定抽象数据类型的例程、程序、对象、组件、数据结构等。通常,程序模块的功能可以按需在各个实施例中进行组合或分布。

[0026] 基本计算设备

[0027] 参考图 1,用于实现各实施例的示例性系统包括通用计算系统环境,如计算系统环境 10。在其最基本的配置中,计算系统环境 10 通常包括至少一个处理单元 12 和存储器 14。取决于计算系统环境的确切配置和类型,存储器 14 可以是易失性的(诸如 RAM)、非易失性的(诸如 ROM、闪存等)或是两者的某种组合。该最基本配置在图 1 中由虚线 16 来示出。另外,计算系统环境 10 还可具有附加的特征/功能。例如,计算系统环境 10 还可包含附加存储(可移动和/或不可移动),包括但不限于磁或光盘或磁带。这样的附加存储在图 1 中由可移动存储 18 和不可移动存储 20 示出。计算机存储介质包括以用于存储诸如计算机可读指令、数据结构、程序模块或其它数据等信息的任何方法或技术来实现的易失性和非易失性、可移动和不可移动介质。存储器 14、可移动存储 18 和不可移动存储 20 都是计算机存储介质的示例。

[0028] 计算系统环境 10 还可包含允许其与其它设备进行通信的通信连接 22。通信连接 22 是通信介质的一个示例。在所描绘的实施例中,计算系统环境 10 被示为通过通信连接 22 来与数据库 30 进行通信。数据库 30 的特性和内容可跨不同的实施例而变化。此外,在某些实施例中,数据库 30 可以是内置数据库,例如,在可移动存储 18 和/或不可移动存储 20 中实现。

[0029] 计算装置 10 还可具有输入设备 24,诸如键盘、鼠标、笔、语音输入设备、触摸输入设备等等。还可包括输出设备 26,诸如显示器、扬声器、打印机等等。此处所讨论的各具体实施例将触摸输入设备与显示器进行组合,例如,触摸屏。所有这些设备在本领域是公知的并且不必在此详细讨论。

[0030] 用户定义的关联函数

[0031] 虽然关联函数是搜索引擎的必需且有价值的部分,但单个关联函数可能不适用于每一种情形。如以下实施例中所描述的,具有允许用户修改由搜索引擎应用的关联函数的搜索引擎将会是有价值的。此外,在某些实施例中,用户将能够完全替换关联函数的部分或全部,以便定制搜索引擎以更好地反映用户的特定需求。

[0032] 因此,在以下的实施例中,将描述一种允许用户指定供与搜索引擎一起使用的排序函数的方法。在某些实施例中,这通过向排序函数的编写者展示关联函数的基本构建块(例如,多路归并联结逻辑运算符)来实现。

[0033] 数据库表

[0034] 现在参考图 2,描绘了根据一个实施例的示例性数据库表 200。虽然数据库表 200 被示为结合具体、枚举的特征和元素,但可以理解,各实施例很好地适用于涉及更多、更少或不同的特征或元素的应用。具体而言,虽然数据库表 200 被示为是只包含两列的简单表格,但可以理解,各实施例适用于跨各种各样的搜索类型和数据类型的应用。

[0035] 数据库表 200 被示为包括多个数据库行或条目,例如,行 210 到 260。每一数据库行都包括文档键列,例如,文档键 212,其包含存储在该行中的文档的唯一标识符。每一数据库行还包括一文档列,例如,文档 214,其包含对应于特定文档键的文档的文档文本。在所

描绘的实施例中,若干文档列条目被示为包括若干项,它们在以下将用于示出其他示例。

[0036] 关联函数

[0037] 在不同的实施例中,接受不同类型的查询或搜索。例如,在一个实施例中,支持布尔搜索,例如,使用诸如 AND(与)、OR(或)或 AND NOT(与非)等布尔逻辑运算符来组合项的搜索。在另一实施例中,支持向量指定,例如,搜索多个项,且这些项的部分或全部被分配不同的权重。在例如 web 搜索的另一实施例中,,对所返回的搜索文档的最终排名可部分地取决于与该文档相关联的外部考虑事项,例如,元数据。在另一实施例中,利用其它搜索方法。

[0038] 在许多应用中,用户能操纵或修改搜索引擎所应用的关联函数,而不依赖内置到特定搜索引擎中的关联函数是有利的,例如,用户可选择对不同的情况或数据集应用不同的关联函数,或者可以修改排序函数,以更改应用于搜索结果的排名以满足某一合乎需要的准则。这种操纵的自由向用户提供了优点。

[0039] 在某些实施例中,用户可选择提供对应于不同的可用运算符的若干排序函数。例如,在处理布尔逻辑运算符时,用户可为 AND 和 OR 运算符提供单独的关联函数。此外,在若干这样的实施例中,可向搜索引擎提供默认关联函数以便与未被用户提供的关联函数覆盖的运算符一起使用。

[0040] 应用用户定义的关联函数

[0041] 现在参考图 3,描绘了根据一个实施例的应用用户定义的关联函数的方法的流程图 300。尽管在流程图 300 中公开了具体步骤,但这些步骤是示例性的。即,这些实施例很好地适用于执行各种其它(附加)步骤或流程图 300 中所述的步骤的变体。可以理解,流程图 300 中的步骤可以按与所呈现的次序不同的次序执行,且并非流程图 300 中的所有步骤都要被执行。

[0042] 现在参考步骤 310,接收用户指定的排序函数。在不同的实施例中,利用提供这一关联函数的不同方法。例如,在某些实施例中,用户可提供用编程语言(例如,T-SQL)编写的关联函数。在若干这样的实施例中,对于用户定义的关联函数使用声明性语言,例如,SQL 或方言。而且,在不同的实施例中,排序函数的运算可以变化。例如,在一个实施例中,排序函数可应用于布尔运算符 AND,并对搜索结果应用 TF-IDF 方案。

[0043] 现在参考步骤 320,分解查询。在某些实施例中,将复杂查询分解成较简单的运算符的树。在其它一些实施例中,将查询分解成若干较简单的运算。例如,在一个实施例中,将全文查询分成包含通过联结来连接的若干简单运算的树。在这一实施例中,联结类型基于连接简单运算的逻辑运算符。例如,AND 被视作内联结,OR 被视作外联结,而 AND NOT 作为不存在来处理。在某些实施例中,每一较简单的运算都包含同质逻辑运算符,例如,仅 AND 或 OR。在一个这样的实施例中,用于同质逻辑运算符的排序函数被视作是可交换的。

[0044] 现在参考步骤 330,将简单运算与排序函数相关联。在某些实施例中,简单运算可与用户提供的排序函数相关联。在若干实施例中,简单运算可与用户提供的排序函数(如果有用户提供的排序函数可用的话)或默认排序函数链接。在某些实施例中,排序函数具有共同的签名,且将用于索引访问的语言元素用作基本工具;在这些实施例中,为排序函数提供上下文信息,该上下文信息进而被传递给执行索引访问的元素。此外,上下文信息可被传递到其它表值函数(TVF),以获取对于排序函数而言是重要的但不存在于索引中的信息。

[0045] 现在参考步骤 340, 执行搜索并检索对应的结果。在不同的实施例中, 该搜索和检索操作以不同的方式执行。可以理解, 各实施例很好地适用于具有许多不同的搜索和结果检索方法的应用。在某些实施例中, 检索到的结果包括关联函数计算这些结果的排名所必需的信息。这些信息可包括诸如文档计数、项频率、项权重和其它信息等元素。在某些实施例中, 利用上述分解的简单运算来执行多次搜索。在一个这样的实施例中, 提供上下文信息允许索引访问元素检索与对应于较简单的运算的部分查询相匹配的行。

[0046] 现在参考步骤 350, 根据一个或多个适当的排序函数来对检索到的结果进行排序。在某些实施例中, 排序函数使用诸如聚集和标量用户定义的函数等标准 SQL 编程构造来为每一个文档计算分数。

[0047] 在某些实施例中, 查询分解以及排序函数与简单运算符的关联在编译阶段期间执行, 而搜索、检索和排序在运行时阶段执行。在其它实施例中, 这些操作的定时可以变化。

[0048] 流式表值函数

[0049] 在实现上述方法的某些实施例中, 索引访问使用流式表值函数 (streamingtable valued function, STVF) 来提供。在这一实施例中, 对 STVF 的使用提供了某些优点。其中有: 多路 AND 和 OR 在倒排索引上的高效实现, 以及输出以递增文档标识符次序排序的高效倒置, 即使该文档标识符是倒排索引中的内部列。在其它实施例中, 可利用其它搜索结果排序, 例如, 相关性次序。此外, 使用 STVF 允许基于用户请求来在不同的层进行解压缩, 例如, 对于文档标识符解压缩在层 1 进行解压缩, 或者对于出现解压缩在等级 2 进行解压缩。另外, 使用 STVF 允许高效地检索排序所需信息, 例如, 文档计数、项频率、项权重或列权重。此外, 使用 STVF 允许准确的基数估计, 进而允许数据库查询优化器来进行有效优化。

[0050] 查询分解

[0051] 现在参考图 4A, 描绘了根据一个实施例的示例数据库查询 400。虽然查询 400 被示为结合具体、定义的元素, 但可以理解各实施例很好地适用于涉及更多、更少或不同元素或者表征不同的元素排列的应用。

[0052] 在所描绘的实施例中, 查询 400 被示为针对 SQL 数据库的全文查询。在该实施例中, 查询 400 经过多次变换, 这些变换以该查询被精简为若干较简单的运算并允许包括用户提供的排序函数的方式来构造。

[0053] 在所描绘的实施例中, T 是数据库表, 诸如表 200。文档和文档键是数据库表中的列。在所描绘的实施例中, 搜索串是 'apples AND oranges OR grapes AND pears'。CONTAINSTABLE(包含表) 是 T-SQL TVF, 它返回符合 (qualify) 搜索串的行以及文档相对于该搜索串的排名。在所描绘的实施例中, 该查询指示搜索引擎返回表 T 的所有行及其排名值, 其中文档列包含关键字 apples(苹果) 和 oranges(桔子) 或者关键字 grapes(葡萄) 和 pears(梨)。

[0054] 在一个实施例中, 解析并绑定查询。在解析和绑定过程期间, 标识 CONTAINSTABLE 函数, 并且调用全文查询编译器来处理搜索串。该全文查询编译器将搜索串变换成表示查询关键字和绑定这些关键字的逻辑运算符的树。图 4B 描绘了一种这样的棵树, 即, 树 401。

[0055] 如树 401 所示, 来自查询 400 的搜索串已被变换成树结构。组织树 401 以使得逻辑运算符绑定由搜索串关联的树元素。以此方式, OR 410 将 AND 420 与 AND 425 绑定。AND 420 进而将 apples 431 与 oranges 433 绑定, 而 AND 425 将 grapes 435 与 pears 437 绑

定。

[0056] 在某些实施例中,在将搜索字符串精简到简化的树结构后,标识同质逻辑运算符,诸如 AND 和 OR 运算符。在若干这样的实施例中,绑定简单项(例如,关键字)的同质逻辑表达式被变换成被实现为 STVF 的多路跳跃归并联结运算符。图 4C 将一个这样的实施例描绘为中间树 441。

[0057] 如中间树 441 所示,树 401 的 AND 部分已经用被实现为 STVF 的多路跳跃归并联结运算符来替换。OR 450 现在绑定元素 460 和 465,这两个元素中的每一个都是多路跳跃归并联结运算符。

[0058] 在某些实施例中,在按此方式变换查询后,系统将加载用户为适当的运算符提供的排序函数。例如,参考中间树 441,系统将加载用户为 AND 运算符提供的排序函数。在编译排序函数期间,系统将在适当时插入实际多路归并联结运算符。在其中用户没有为特定运算符提供排序函数的某些实施例中,改为加载并编译默认系统提供的函数。

[0059] 然后在更高的层,例如在图 4C 中的 OR 450 的层组合排序函数返回的排名值。在某些实施例中,该组合通过默认系统操作来处理,例如,计算组合排名作为最大值(在 OR 运算符的情况下),或者作为最小值(在 AND 运算符的情况下)。在其它实施例中,用户可按与用户指定的排序函数类似的方式提供用户指定的组合函数。

[0060] 在图 4D 中,描绘了根据一个实施例的查询树 471。查询树 471 示出了具有如上所述的组合函数的查询 400 的最终状态。在所描绘的实施例中,该全文查询树中的非叶节点被变换成关系运算符,例如,AND 被变换成内联结、OR 被变换成外联结或者 AND NOT 被变换成半联结。如图所示,OR 450 已被外联结 485 替换。已分别为多路跳跃归并联结运算符 492 和 497 加载排序函数 490 和 495。块 480 描绘了组合函数,此处被示为排序函数 490 和 495 所返回的排名值的最大值。

[0061] 在某些实施例中,该查询树被结合到总 SQL 查询中,并作为整体来优化。在若干这样的实施例中,该优化确保用户定义的排序函数无论多复杂都如同该函数是用户提供的查询的一部分那样得到系统的相同程度的优化。

[0062] 尽管已以针对结构特征和 / 或方法动作的语言描述了该主题,但可以理解,所附权利要求书中定义的主题不必限于上述的特定特征或动作。相反,上述特定的特征和动作作为实现这些权利要求的示例性形式而予以公开。

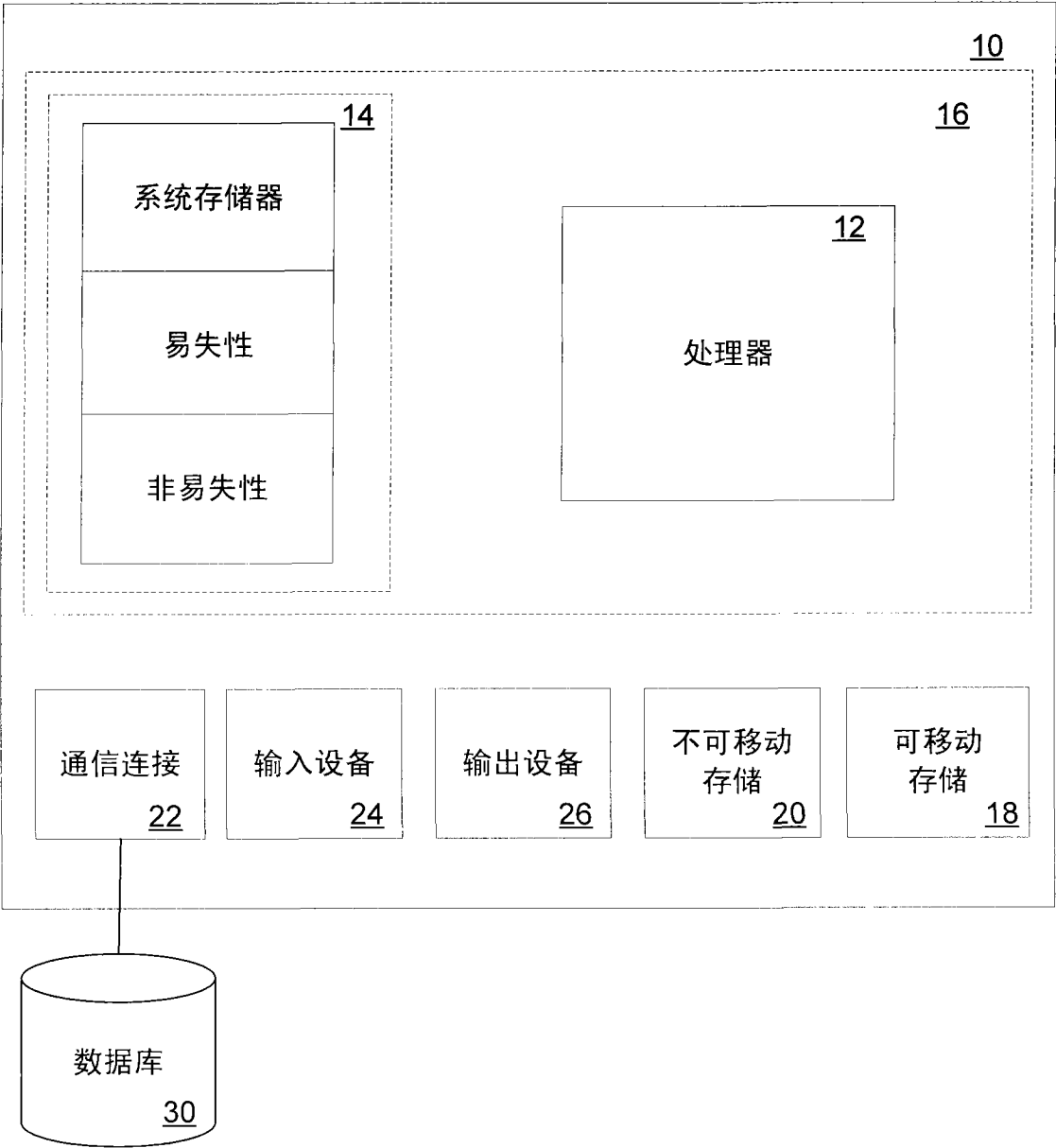
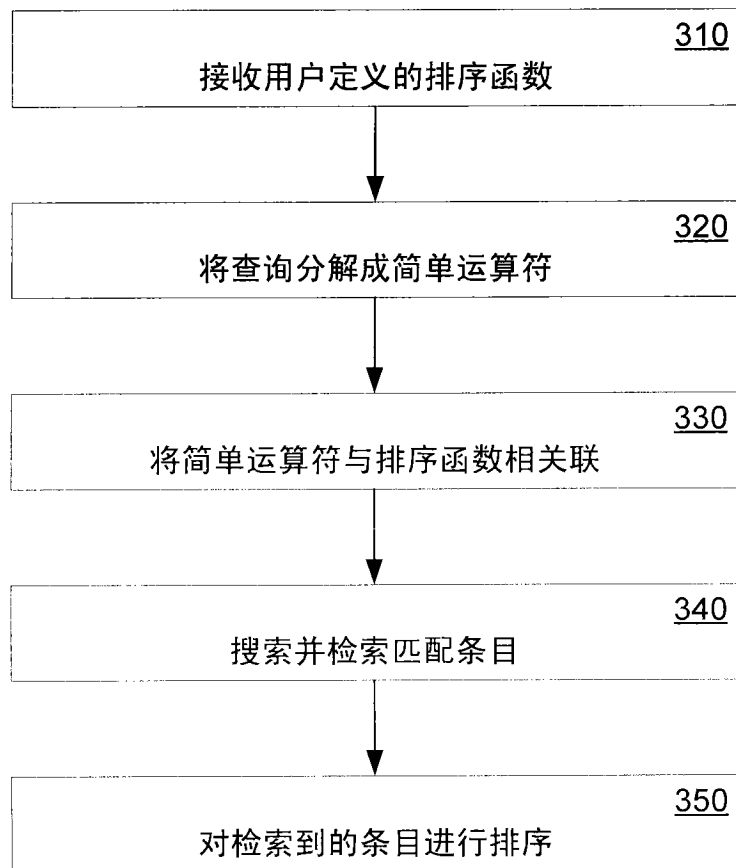


图 1

<文档键>	<文档>
<u>212</u>	<u>214</u> ...apples, oranges, grapes, pears....
<u>222</u>	<u>224</u> ...apples, grapes....
<u>232</u>	<u>234</u> ...oranges, grapes, pears....
<u>242</u>	<u>244</u> ...apples, oranges, pears....
<u>252</u>	<u>254</u> ...oranges, pears....
<u>262</u>	<u>264</u> ...apples, bananas, oranges, grapes, figs, pears....

数据库表 200

图 2



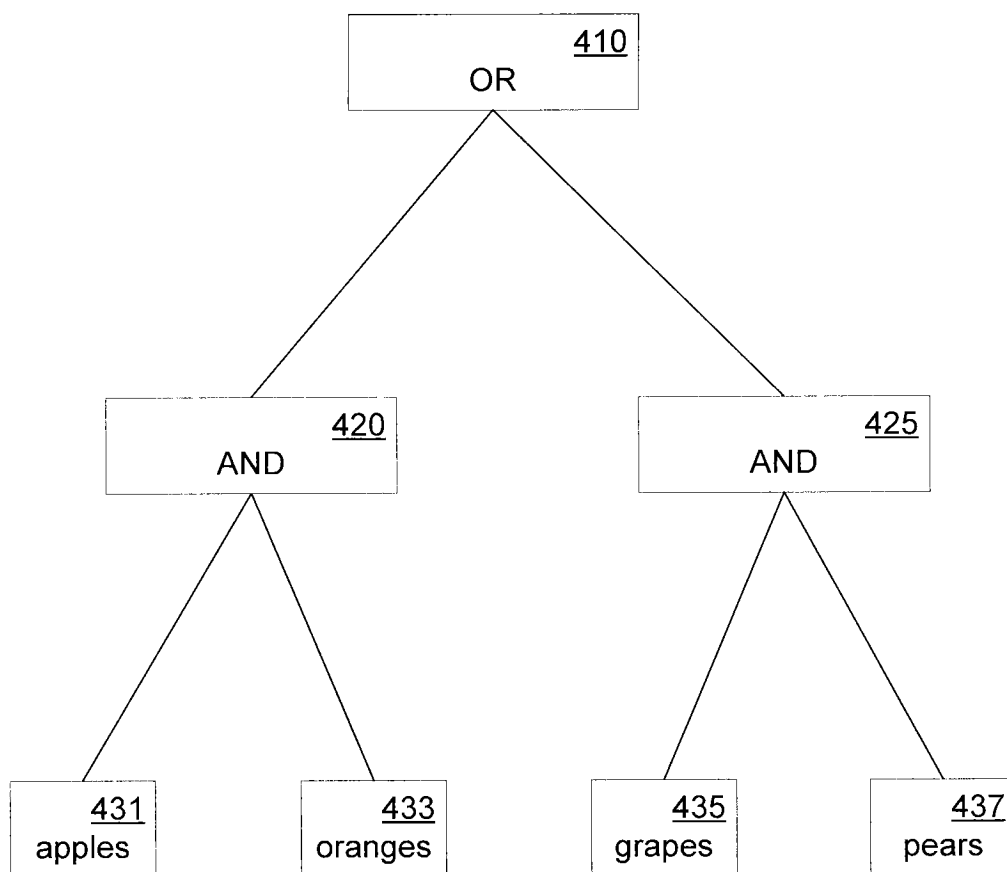
流程图 300

图 3

```
SELECT * FROM T, CONTAINSTABLE(T.document, 'apples AND oranges OR grapes AND  
pears') CT on T.ftkey=CT.FTKEY
```

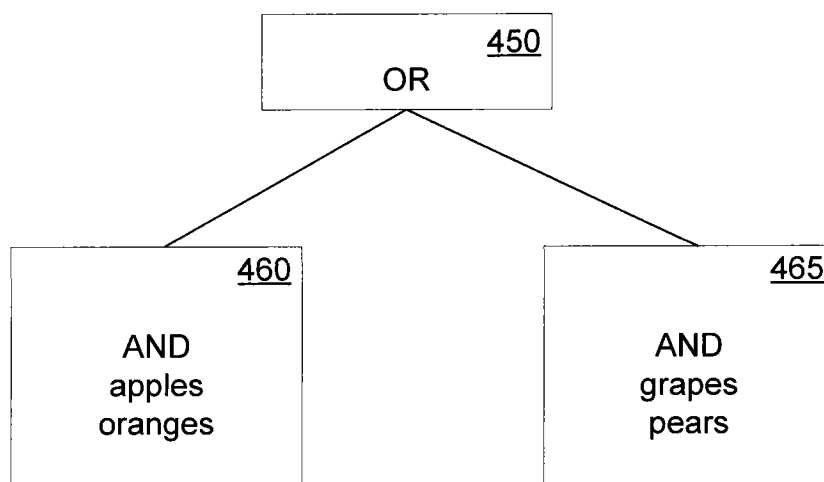
查询 400

图 4A



树 401

图 4B



中间树 441

图 4C

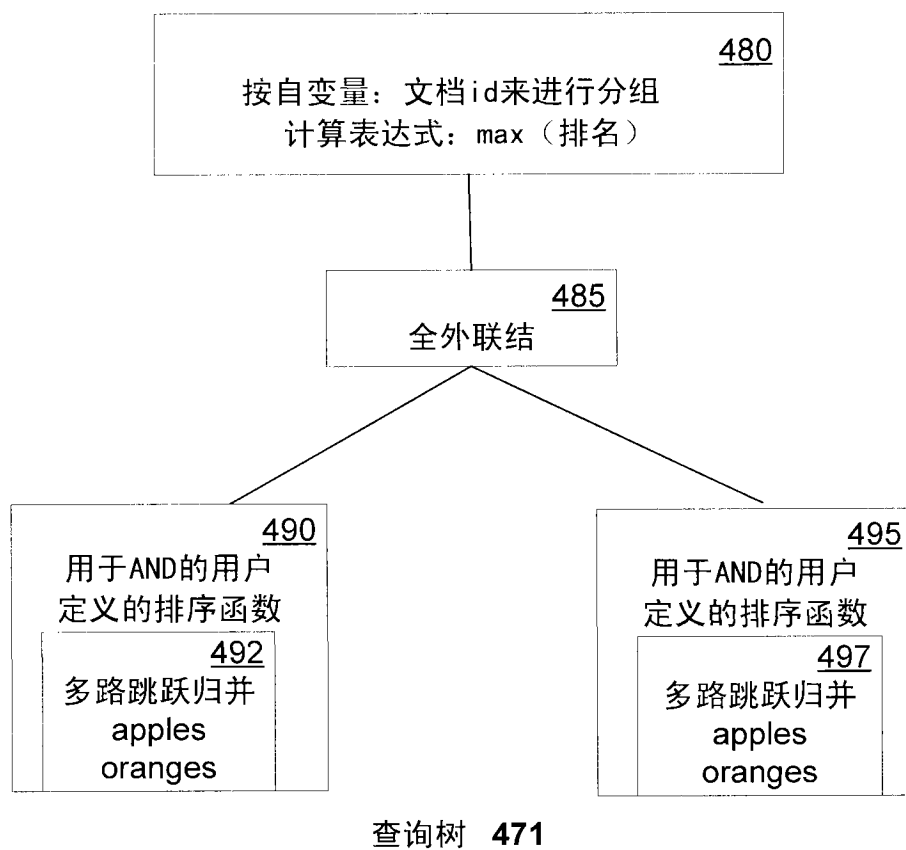


图 4D