(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 101133388 B
(45)授权公告日 2011.07.06

(21)申请号 200680007173.X

(22)申请日 2006.01.25

(30)优先权数据
11/043,695 2005.01.25 US

(85)PCT申请进入国家阶段日
2007.09.05

(86)PCT申请的申请数据
PCT/US2006/002709 2006.01.25

(87)PCT申请的公布数据
WO2006/081325 EN 2006.08.03

(73)专利权人 谷歌公司
地址 美国加利福尼亚州

(72)发明人 A·L·帕特森

(74)专利代理机构 北京市金杜律师事务所 11256
代理人 王茂华

(51)Int.Cl.
G06F 7/00(2006.01)
G06F 17/30(2006.01)

(56)对比文件
US 2003/0195877 A1,2003.10.16,说明书第[0010]-[0035],[0124]-[0128],[0071]-[0093]段、权利要求17、说明书附图1.
WO 03/060767 A2,2003.07.24,全文.
US 6499030 B1,2002.12.24,全文.
CN 1363069 A,2002.08.07,全文.

审查员 胡雅娟

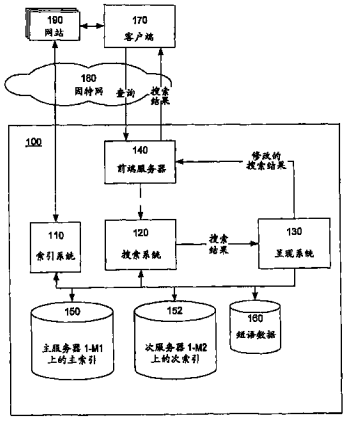
权利要求书 2 页 说明书 19 页 附图 5 页

(54)发明名称

基于多索引的信息检索系统

(57)摘要

一种信息检索系统使用短语索引、检索、组织并且描述文档。识别预测文档中其他短语的出现的短语。根据它们包括的短语来索引文档。文档索引被分成多个索引,包括主索引和次索引。主索引存储具有相关性次序排列的文档的短语置入列表。次索引按照文档顺序存储来自于置入列表的额外文档。



CN 101133388 B

1. 一种针对短语来索引文档的计算机实现的方法，其中每个文档具有文档标识符，该方法包括：

建立包含该短语的文档列表；

通过相关性分值来对所述列表中的文档进行排序，将所述列表中的文档划分为包括所述列表中较高排序文档的第一部分和包括所述列表中较低排序的文档的第二部分；

按照所述排序文档的相应相关性分值的排序次序，将所述第一部分存储在主索引中，所存储的所述第一部分的较高排序文档在所述主索引中彼此相关；以及

基于所述划分，按照所述排序文档的相应文档标识符的数字次序，将所述第二部分存储在次索引中，所存储的第二部分的较低排序的文档在所述次索引中彼此相关。

2. 根据权利要求 1 的方法，其中所述相关性分值包括基于页面排序的类型分值。

3. 根据权利要求 1 的方法，还包括：针对每个文档，将所述文档的相关性属性存储在所述主索引中。

4. 根据权利要求 3 的方法，其中所述相关性属性包括以下内容中的至少一个：该短语在文档中出现的总数目、也包含该短语并且指向该文档的锚文档的排列次序的列表、文档中每个短语出现的位置、一个或多个标记的集合，所述标记表示出现的格式或者包含该出现的文档部分。

5. 根据权利要求 3 的方法，其中将列表的第二部分存储在次索引中包括仅存储文档识别信息。

6. 根据权利要求 1 的方法，其中将列表的第一部分存储在主索引中包括按照相关性分值的排列次序将列表的第一部分存储在物理存储设备上。

7. 根据权利要求 1 的方法，其中将列表的第二部分存储在次索引中包括按照文档标识符的数字次序将列表的第二部分存储在物理存储设备上。

8. 根据权利要求 1 的方法，其中每个文档列表的第一部分包括第一分段和第二分段，其中在第一分段中列出的每个文档包括第一多个相关性属性，以及在第二分段中列出的每个文档包括第二多个相关性属性，所述第二多个相关性属性是第一多个相关性属性集合的子集，并且其中在第一分段中列出的文档的排序高于在第二分段中列出的文档。

9. 根据权利要求 8 的方法，其中每个文档列表的第一部分包括第三分段，其中在第三分段中列出的每个文档包括第三多个相关性属性，所述第三多个相关性属性是第二多个相关性属性的子集，并且其中在第二分段中列出的文档的排序高于在第三分段中列出的文档。

10. 根据权利要求 8 的方法，其中每个列表的第一部分包含 n 个条目，其中该列表的第二部分包含 m*n 个条目，其中 $m > 2$，并且该列表的第三部分包含 1*n 个条目，其中 $1 > 4$。

11. 一种提供信息检索系统的方法，该方法包括：

存储包括主短语置入列表的主索引，每个置入列表与一个短语相关联并且包括多达最大数目的包含该短语的文档，所述文档按照各自的相关性分值来排列次序；

存储包括次短语置入列表的次索引，每个置入列表与主索引中的主短语置入列表相关联，并且包括包含该短语并且相关性分值比针对该短语的主短语置入列表中最低排序文档的相关性分值小的文档，所述文档按照文档标识符来排序；

接收包括至少一个短语的搜索查询；

响应于包含具有主短语置入列表和次短语置入列表的第一短语以及仅具有主短语置入列表的第二短语的搜索查询，对第二短语的主短语置入列表和第一短语的主短语置入列表取交集，以获得第一共同文档集合，并且对第一短语的次短语置入列表和第二短语的主短语置入列表取交集，以获得第二共同文档集合，并且结合第一和第二共同文档集合；并且

排序来自所述第一共同文档集合和所述第二共同文档集合的结合的文档。

12. 一种信息检索系统，包括：

第一存储装置，用于存储包括主短语置入列表的主索引，每个置入列表与一个短语相关联并且包括多达最大数目的包含该短语的文档，所述文档按照各自的相关性分值来排列次序；

第二存储装置，用于存储包括次短语置入列表的次索引，每个置入列表与主索引中的主短语置入列表相关联，并且包括包含该短语并且相关性分值比针对该短语的主短语置入列表中最低排序文档的相关性分值小的文档，所述文档按照文档标识符来排序；

接收装置，用于接收包括至少一个短语的搜索查询；

结合装置，用于响应于包含具有主短语置入列表和次短语置入列表的第一短语以及仅具有主短语置入列表的第二短语的搜索查询，对第二短语的主短语置入列表和第一短语的主短语置入列表取交集，以获得第一共同文档集合，并且对第一短语的次短语置入列表和第二短语的主短语置入列表取交集，以获得第二共同文档集合，并且结合第一和第二共同文档集合；以及

排序装置，用于排序来自所述第一共同文档集合和所述第二共同文档集合的结合的文档。

基于多索引的信息检索系统

[0001] 相关申请的交叉引用

[0002] 本申请要求 2005 年 1 月 25 日提交的、名称为“基于多索引的信息检索系统”的美国实用专利申请 11/043,695 的利益和优先权，其中将该申请的公开内容通过参考文件的形式合并于此。本申请也是共同拥有的 2004 年 7 月 26 日提交的申请号为 10/900,021 的申请的部分继续申请，在此通过参考将申请号为 10/900,021 的申请引入。

技术领域

[0003] 本发明涉及一种用于对诸如因特网的大型资料库中的文档进行索引、搜索和分类的信息检索系统。

背景技术

[0004] 当前，信息检索系统（通常称为搜索引擎）是用于在大型、分散并且不断增长的资料库（诸如因特网）中寻找信息的必备工具。通常，搜索引擎创建索引，其中该索引将文档（或者“页”）和出现在每个文档中的各个词联系起来。响应于包含多个查询词的查询，典型地，基于出现在文档中的一些数量的查询词来检索文档。然后，根据其他统计方法，诸如查询词的出现频率、主域 (host domain)、链接分析等等，来对检索的文档进行排序。然后，检索的文档典型地按照它们排列的次序被呈现给用户，并且没有任何进一步的分组或者强加的层级。在某些情况中，呈现文档文本的选择部分，以将文档内容的概览提供给用户。

[0005] 查询词的直接“布尔”匹配具有众所周知的限制，并且特别地，不能识别不具有查询词但具有相关词的文档。例如，在典型的布尔系统中，对“澳大利亚牧羊犬”的搜索将不会返回不具有确切查询词的关于其他牧羊犬（诸如边境牧羊犬）的文档。相反，这样的系统通常很可能也检索并且高度地排序关于澳大利亚的文档（与狗无关）以及关于“牧羊犬”的文档。

[0006] 这里的问题是：传统系统基于各个单词而不是根据概念来对文档进行索引 (index)。概念经常以短语的形式来表述，诸如“澳大利亚牧羊犬”、“美国总统”或者“圣丹斯电影节”。最多，某些现有系统针对预定的以及非常有限的‘已知’短语集合来对文档进行索引，这通常是由人类操作员来选择的。由于用于识别所有三个、四个或者五个或者更多单词的可能短语的可感知的计算和存储需要，所以通常避免对短语进行索引。例如，假设任何五个单词可以组成一个短语，并且大资料库将至少具有 200,000 个唯一词，则存在大概 3.2×10^{26} 个可能的短语，明显地超过任何现存系统可以存储在存储器中或者可编程操作的数量。进一步的问题是：短语根据它们的使用持续地输入并且离开词典，会更频繁地发明新的单个单词。将总是从诸如技术、艺术、世界事件和法律的源生成新的短语。其他短语将随着时间在使用上将减少。

[0007] 一些现存的信息检索系统通过使用各个单词的同现模式来试图提供概念检索。在这些系统中，对诸如“总统”的一个单词的搜索将也检索具有随“总统 (president)”频繁出现的其他单词，诸如“白 (white)”和“宫 (house)”的文档。虽然此方法可以产生具有在各

个单词级别上概念相关的文档的搜索结果，但是典型地，它不能捕获同现短语间的固有的主题关系。

[0008] 因而，需要一种信息检索系统和方法，其可以根据短语来综合地识别大型资料库中的短语，根据文档的短语来搜索和排序文档，以及提供关于文档的额外分组和描述信息。

[0009] 传统信息检索系统的另一个问题是：它们仅可以索引在因特网上可用的文档的一相对小的部分。根据当前估计，在如今的因特网上存在超过 2000 亿网页。然而，甚至最好的搜索引擎仅索引 60 到 80 亿网页，因而错过多数可用网页。存在几个限制现存系统索引能力的原因。最主要的是，典型系统依靠反向索引的变型，其中反向索引针对每个词（如上面讨论所讨论的）保持出现该词的每个网页以及识别该词在网页上的每个出现的确切位置的位置信息的列表。索引各个词和索引位置信息的组合需要非常大的存储系统。

[0010] 许多用于搜索因特网的信息检索系统的进一步的问题是：它们没有能力对随时间变化的网页存档。传统地，大部分因特网搜索引擎仅存储针对给定网页的当前实例（或版本）的相关性信息，并且在每次重新索引该网页时更新此信息。作为结果，给定的搜索仅返回满足查询的网页的当前版本。这样，用户不能搜索网页的现先实例，或者具有特定日期间隔的最近的网页。而且，当评估搜索查询或者呈现搜索结果的时候，搜索引擎同样不使用与版本或日期相关的相关性信息。

[0011] 因而，希望提供一种信息检索系统，其可以有效地索引数百亿并且实际超过 1000 亿网页内容，而没有现存系统的大量存储需要。

发明内容

[0012] 一种信息检索系统和方法使用短语索引、搜索、排序并且描述文档集合中的文档。该系统适合识别在文档集合中具有足够频繁和 / 或明显使用的短语，以表示它们是“有效的”或者“好的”短语。以此方式，可以识别多单词短语，例如四个、五个或多个词的短语。这避免了不得不识别并且索引由给定数目单词的所有可能序列产生的每个可能短语的问题。

[0013] 该系统还适合于根据一个短语预测在文档中其他短语的出现的能力，来识别彼此相关的短语。更具体地，可以使用这样的预测方法，即该预测方法将两个短语的实际同现率和两个短语的期望同现率联系起来。信息增益是一个这样的预测方法，其中信息增益是实际同现率与期望同现率的比值。预测方法超过预定的阈值，则两个短语是相关的。在这样的情况中，第二短语针对于第一短语具有明显的信息增益。语义上，相关短语将是那些通常用于讨论或者描述给定主题或者概念的短语，诸如“美国总统”和“白宫”。对于给定的短语，可以基于它们各自的预测方法，根据它们的相关性或者重要性，来排序相关短语。

[0014] 一种信息检索系统通过有效的或者好的短语来索引文档集合中的文档。对于每个短语，置入列表识别包含短语的文档。另外，对于给出的短语，第二列表、向量或者其他结构被用于存储数据，该数据表示给定短语的哪些相关短语也出现在包含给定短语的每个文档中。以此方式，系统不仅可以容易地识别包含响应于搜索查询的短语的文档，而且可以识别也包含与查询短语相关的短语的文档，而且因此更可能是明确关于表达于查询短语中的主题或者概念。

[0015] 当响应于查询来搜索文档时，该信息检索系统也适合于使用短语。进行查询，以识别任何出现在查询中的短语，以便检索查询短语的相关置入列表，以及相关短语信息。

另外，在某些情况中，用户可以在搜索查询中输入不完整短语，诸如“的总统 (President ofthe)”。可以识别诸如这些的不完全短语，以及可以由短语扩展（诸如“美国的总统 (President of United States”) 来替换的不完整短语。这有助于保证实际执行用户最可能的搜索。

[0016] 本发明的另一方面是能够使用多索引结构来索引 1000 亿或者更多数量级的大量文档。在一个实施例中，提供主和次索引。主索引存储针对短语的索引数据，其中对于每个短语，索引有限数目的文档。对于特定短语，文档的索引数据按照文档与短语的相关性的排序次序来存储。优选地，此存储安排是逻辑和物理的（即，怎样将数据存储在主存储设备上）。在多于有限数量的文档包括特定短语的情况，用于这些剩余文档的索引数据存储在次索引中，但是这里由文档号代替相关性排序来进行排序，以及例如使用分散聚集类型方法，来进行检索。

[0017] 例如，主索引可以构造为对于每个短语存储 32k 文档条目，并且次索引可以构造为存储针对包含短语并超过 32k 的其他文档的任何其他文档条目。为了获得对于主索引的文档条目，对文档对于短语的相关性进行评分，并且通过它们的相关性分值来对他们进行排序。可选地，文档可以通过针对相关性分析有用的各种文档特征来排序。用于文档条目的排序被用于在主索引和次索引间划分条目。在存在少于有限数量的包含短语的文档的情况下，则再次按照相关性排序将所有条目存储于主索引。索引安排使用于索引的存储能力能够增加十到十五倍，并且由于最优化的索引信息，服务器性能产生十倍增长。

[0018] 本发明的另一个方面是能够索引用于存档的文档的多个版本和实例。此能力使用户能够搜索在特定日期范围内的文档，能够使日期或者版本相关的相关性信息用于响应于搜索查询评估文档以及组织搜索结果。在一个实施例中，文档与一个或多个日期范围相关。每个日期范围与相关性数据相关联，该相关性数据来自文档以及在日期范围期间针对该文档视为有效。当前日期范围与文档的当前实例相关联，从最近索引文档的日期开始。当在索引进行期间遇到文档时，将其与先前版本相比较，以确定文档是否已经改变。如果文档没有改变，那么保留索引的相关性数据。如果文档改变，那么文档的当前日期范围被关闭，并且重索引文档，并且建立新的当前日期范围，并且与当前相关性数据相关联。

[0019] 在系统和软件架构、计算机程序产品和计算机实现方法以及计算机生成用户接口和呈现中，本发明具有进一步的实施例。

[0020] 前面只是基于短语的信息检索系统和方法的一些特征。信息检索系统领域的那些技术人员将会理解，短语信息一般性的灵活性允许在索引、文档注释、搜索、排序以及文档分析和处理的其他领域中大量使用和应用。

附图说明

[0021] 图 1 是本发明一个实施例软件架构的框图。

[0022] 图 2 说明了识别在文档中的短语的方法。

[0023] 图 3 说明了带有短语划分和次划分的文档。

[0024] 图 4 说明了识别相关短语的方法。

[0025] 图 5 说明了针对相关短语索引文档的方法。

[0026] 图 6 说明了基于短语检索文档的方法。

[0027] 附图描述了本发明的优选的实施例，仅用于说明目的。本领域技术人员将从下列讨论中容易地认识到，在不偏离这里描述的本发明的原理的情况下，可以使用这里说明的结构和方法的可选的实施例。

具体实施方式

[0028] I. 系统概括

[0029] 现在参考图 1，示出了根据本发明的一个实施例的搜索系统 100 的实施例的软件架构。在此实施例中，该系统包括索引系统 110、搜索系统 120、呈现系统 130 和前端服务器 140。

[0030] 索引系统 110 负责通过访问各种网站 190 和其他文档集合来识别文档中的短语，并且根据它们的短语来索引文档。前端服务器 140 接收来自于客户端 170 的用户的查询，并且向搜索系统 120 提供这些查询。搜索系统 120 负责搜索与搜索查询相关的文档（搜索结果），包括识别搜索查询中的任何短语，以及然后使用短语的出现影响排列顺序来对搜索结果中的文档进行排列。搜索系统 120 向呈现系统 130 提供搜索结果。呈现系统 130 负责修改搜索结果，包括移除最近复制文档并且生成文档的主题描述，并且将修改的搜索结果提供返回给前端服务器 140，前端服务器 140 向客户端 170 提供结果。系统 100 还包括存储与文档有关的索引信息的主索引 150 和次索引 152，以及存储短语以及相关统计信息的短语数据存储器 160。主索引 150 分布在多个主服务器 1...M1 上，以及次索引 152 同样分布在多个次服务器 1...M2。

[0031] 本申请的上下文中，“文档”应被理解为可以由搜索引擎索引和检索的任何媒体类型，包括网页文档、图像、多媒体文件、文本文档、PDF 或者其他图像格式文档等等。文档可以具有一或多页、分区、分段或者其他组成部分，如适合它的内容和类型。同样地，文档可以被称为“页”，如通常用来指代代因特网上的文档的“页”。使用通用术语“文档”并不隐含对于本发明的范围的限制。搜索系统 100 操作在大型文档资料库上，诸如因特网和万维网，但是同样可以被用于更多限制的集合中，诸如针对图书馆或者私人企业的文档集合。在任一上下文中，应该理解的是，文档典型地分布在很多不同的计算机系统和地点。那么不失一般性，不论格式和位置（例如，哪个网站或者数据库），文档通常集体地被称作资料库或者文档集合。每个文档具有相关联的标识符，用于唯一识别文档；优选地，该标识符是 URL，但是其他类型的标识符（例如，文档号）也可以使用。在此公开中，假设使用 URL 识别文档。

[0032] II. 索引系统

[0033] 在一个实施例中，索引系统 110 提供三个主要的功能操作：1) 识别短语和相关短语；2) 关于短语来索引文档；以及 3) 生成并且保持基于短语的排序。本领域技术人员应该理解的是，索引系统 110 将执行其他功能以及支持传统索引功能，因此在这里不再描述这些其他操作。索引系统 110 对主索引 150 和次索引 152 以及短语数据的数据仓库 160 进行操作。下面将进一步描述这些数据仓库。

[0034] 1. 短语识别

[0035] 索引系统 110 的短语识别操作识别文档集合中“好的”和“坏的”短语，其中短语对于索引和搜索文档是有用的。在一个方面中，好的短语是在文档集合中趋向以大于文档的特定比例出现，和 / 或被表示作为显著出现在所述文档中的短语，诸如由标记标签或者

其他形态、格式或者语法标记界定的那样。好的短语的另一个方面是它们可预测其他好的短语，并且不仅仅是出现在词典中的单词的序列。例如，短语“美国总统”是预测诸如“乔治布什”和“比尔克林顿”的其他短语的短语。然而，其他短语是不可用于预测的，诸如“fell down stairs(跌下楼梯)”、“top of the morning(王牌投手)”、或者“out of the blue(突然)”，这是因为类似这些的习语和口语趋向于与很多其他不同的和无关的短语一起出现。因此，短语识别阶段确定哪个短语是好的短语以及哪个是坏的（即，缺乏预测能力）。

[0036] 现在参考图 2，短语识别过程具有下列功能阶段：

[0037] 200：收集可能的并且好的短语，连同这些短语的频率和同现统计。

[0038] 202：基于频率统计将可能的短语分类为好的或坏的短语。

[0039] 204：基于源自同现统计的预测方法来整理好的短语列表。

[0040] 现在将更详细地描述这些阶段的每一个。

[0041] 第一阶段 200 是这样一个过程，即通过该过程，索引系统 110 爬过文档集合中的一组文档，随时间重复文档集合的分区。每单程 (pass) 处理一个分区。每单程爬过的文档数目可以变化，并且优选地，每分区大概是 1,000,000。优选地，每个分区中仅处理之前未爬过的文档，直到已经处理所有文档，或者符合某些其他的终止标准。在实践中，当将新文档持续加入文档集合时，持续所述爬过。针对被爬过的每个文档，由索引系统 110 进行下列步骤：

[0042] 使用长度为 n 的短语窗口来遍历 (traverse) 文档的单词，其中 n 是希望的最大短语长度。典型地，窗口的长度将至少是 2，并且优选地为 4 或 5 词（单词）。优选地，短语包括短语窗口中的所有单词，包括可表征为停止单词的那些单词，诸如“a”、“the”等等。可以由行尾、段回车、标记标签或者内容或格式变化的其他表示来终止短语窗口。

[0043] 图 3 说明了在遍历其间文档 300 的一部分，示出了开始于单词“stock”并且向右延伸 5 个单词窗口 302。窗口 302 中的第一单词是候选短语 i，并且序列的每一个 i+1、i+2、i+3、i+4 和 i+5 同样是候选短语。因此，在此例子中，候选短语是：“stock”、“stock dogs”、“stockdogs for”、“stock dogsfor the”、“stock dogsfor the Basque”和“stockdogsfor the Basque shepherds”。

[0044] 在每个短语窗口 302 中，每个候选短语被依次检查，以确定是否它已经出现在好的短语列表 208 或者可能的短语列表 206 中。如果候选短语既没有出现在好的短语列表 208 中也没有出现在可能的短语列表 206 中，那么候选短语已经被确定为“坏的”并且被忽略。

[0045] 如果候选短语在好的短语列表 208 中，作为条目 g_j，那么短语 g_j 的索引 150 条目被更新，以包括该文档（例如，它的 URL 或者其他文档标识符），以表示此候选短语 g_j 出现在当前文档中。在短语 g_j（或词）的索引 150 中的条目被称作短语 g_j 的置入 (posting) 列表。该置入列表包括文档 d 的列表（由它们的文档标识符，例如文档号，或者可选地，URL），其中短语出现在这些文档 d 中。在一个实施例中，使用例如 MD5，通过 URL 的单向散列来产生文档号。

[0046] 另外，如下面进一步解释，更新同现矩阵 212。在最先的单程中，好的和坏的列表将是空的，并且因此，大部分短语将趋向于加入可能的短语列表 206。

[0047] 如果候选短语不在好的短语列表 208 中，那么它被加入可能的短语列表 206，除非

它已经出现在其中。可能的短语列表 206 上的每个条目 p 具有三个相关的计数：

[0048] P(p)：出现可能的短语的文档的数目；

[0049] S(p)：可能的短语的所有实例的数目；并且

[0050] M(p)：可能的短语的感兴趣实例的数目。在可能短语可通过语法或者格式标记（例如通过黑体字、或者下划线、或者作为超级链接中的锚文本，或者引号）与文档中相邻的内容区分出来的情况下，可能的短语的实例是“感兴趣的”。由各种 HTML 标记语言标签和语法标记来表示这些（以及其他）区别表现 (appearance)。当将短语置于好的短语列表 208 中时，保持针对短语的这些统计。

[0051] 另外，保持用于好的短语的各种列表，即同现矩阵 212(G)。矩阵 G 具有 m×m 维，其中 m 是好的短语的数目。矩阵中每个条目 G(j,k) 代表一对好的短语 (g_j,g_k)。同现矩阵 212 逻辑地（虽然不需要物理地）为好的短语的每个对 (g_j, g_k) 保持相对于次窗口 304 的三个分开的计数，其中次窗口 304 的中心定位在当前单词 i，并且延伸 +/-h 个单词。在一个实施例中，诸如图 3 所描述的，次窗口 304 是 30 个单词。同现矩阵 212 因此保持：

[0052] R(j,k)：自然同现计数。短语 g_j 与短语 g_k 一起出现在次窗口 304 中的次数；

[0053] D(j,k)：分离感兴趣计数。短语 g_j 或者短语 g_k 在次窗口中出现作为显著文本的次数；以及

[0054] C(j,k)：联合感兴趣计数。短语 g_j 和短语 g_k 在次窗口中出现作为显著文本的次数。对于避免短语（例如，版权提醒）频繁出现在工具条、页脚或者页眉中并且因此不是其他文本的实际预测的情况，使用联合的感兴趣计数是有益的。

[0055] 参考图 3 的例子，假设“stock dogs（斯托克狗）”在好的短语列表 208 上，短语“Australian Shepherd（澳大利亚牧羊犬）”和“AustralianShepherd Club of America（美国的澳大利亚牧羊犬俱乐部）”也在好的短语列表 208 上。后两个短语围绕当前短语“stock dogs”出现在次窗口 304 中。然而，短语“Australian Shepherd Club of America”作为用于到网站的超级链接（由下划线表示）的锚文本出现。因此增加用于对｛“stock dogs”,“Australian Shepherd”｝的自然同现计数，并且同时增加用于｛“stock dogs”,“Australian Shepherd Club of America”｝的自然出现计数和分离感兴趣计数，因为后者作为显著文本出现。

[0056] 用于分区中的每个文档，使用序列窗口 302 和次窗口 304 遍历每个文档的过程被重复。

[0057] 一旦已经遍历了分区中的文档，索引操作的下个阶段是从可能的短语列表 206 更新 202 好的短语列表 208。如果短语出现的频率和短语出现在其中的文档的数目表示它作为语义上有意义的短语具有足够的使用，则可能的短语列表 206 上的可能短语 p 被移动到好的短语列表 208。

[0058] 在一个实施例中，对此进行如下测试。如果：

[0059] a)P(p)>10 并且 S(p)>20（包含短语 p 的文档数大于 10 并且短语 p 的出现数目大于 20）；或者

[0060] b)M(p)>5（短语 p 的感兴趣实例数目大于 5），则可能短语 p 被从可能的短语列表 206 移除并且被置于好的短语列表 208 上。

[0061] 这些阈值由分区中文档的数目调分段；例如，如果 2,000,000 个文档在分区中被

爬过，那么阈值大概是双倍。当然，本领域技术人员应该理解，阈值的特定值，或者测试它们的逻辑可以如希望变化。

[0062] 如果短语 p 不具有好的短语列表 208 的资格，那么检查它作为坏短语的资格。如果：

[0063] a) 包含短语的文档的数目，P(p)<2；并且

[0064] b) 短语的感兴趣实例数目，M(p) = 0，则短语 p 是坏的短语。

[0065] 这些条件表示该短语既是不频繁的，又没有作为重要内容的表示使用，并且这些阈值再次根据分区中的文档数目调整。

[0066] 如上所述，应该注意，好的短语列表 208 将自然包括各个单词作为短语，以及多单词短语。这是因为短语窗口 302 中的每个第一单词总是候选短语，并且适当的实例计数将被积累。因此，索引系统 110 可以自动索引各个单词（即，具有单个单词的短语）和多单词短语。好的短语列表 208 也将显著短于基于 m 个短语的所有可能组合的理论最大值。在典型的实施例中，好的短语列表 208 将包括大概 6.5×10^5 个短语。当系统仅需要跟踪可能和好的短语时，坏短语列表不需要存储。

[0067] 在最后单程通过文档集合之前，由于大资料库中短语使用的期望分布，可能短语的列表将相对短。因此，如果假设在第十次单程（例如，10,000,000 个文档）中短语最初出现，则此时短语不可能是好的短语。它可能是刚进入使用的新短语，并且因此在随后爬过期间变得逐渐普通。在这种情况中，它的各个计数将增加并且可以最终满足作为好的短语的阈值。

[0068] 索引操作的第三阶段是使用源自于同现矩阵 212 的预测方法来整理 204 好的短语列表 208。如果不整理，好的短语列表 208 很可能包括很多虽然合理地出现在词典中但是它们自己不足以预测其他短语的出现，或者它们自己是更长短语的字序列的短语。移除这些弱的好短语很可能使好的短语的非常鲁棒。为了识别好的短语，使用这种的预测方法，该预测方法表达一个短语相对于另一个短语出现在文档中的增加的可能性。在一个实施例中，这通过以下方式来完成：

[0069] 如上提到的，同现矩阵 212 是存储与好的短语相关联的数据的 m×m 矩阵。矩阵中每行 j 代表好的短语 g_j 并且每列 k 代表好的短语 g_k。对于每个好的短语 g_j，计算期望值 $E(g_j)$。期望值 E 是集合中期望含有 g_j 的文档的百分比。计算该百分比，例如，为包含 g_j 的文档数目与已经爬过过的集合中的文档总数目 T 的比率：P(j)/T。

[0070] 如上提到的，在每次 g_j 出现在文档中时，更新包含 g_j 的文档数目。可以在每次 g_j 的计数增长时，或者在此第三阶段期间更新 $E(g_j)$ 的值。

[0071] 接下来，对于每个其他好的短语 g_k（例如，矩阵的列），确定是否 g_j 预测 g_k。g_j 的预测方法确定如下：

[0072] i) 计算期望值 $E(g_k)$。如果么 g_j 和 g_k 是不相关短语，那么 g_j 和 g_k 的期望同现率 E(j,k) 是 $E(g_j)*E(g_k)$；

[0073] ii) 计算 g_j 和 g_k 的实际同现率 A(j, k)。这是自然同现计数 R(j, k) 除以文档的总数目 T；

[0074] iii) 在实际同现率 A(j,k) 超过期望同现率 E(j,k) 一阈值总量的情况下，g_j 被认为预测 g_k。

[0075] 在一个实施例中，预测方法是信息增益。因此，当在 g_j 的出现时 g_k 的信息增益 I 超过阈值时，短语 g_j 预测另一个短语 g_k。在一个实施例中，计算如下：

[0076] $I(j,k) = A(j,k)/E(j,k)$

[0077] 并且在 I(j,k)> 信息增益阈值的情况下，

[0078] 好的短语 g_j 预测好的短语 g_k。

[0079] 在一个实施例中，信息增益阈值是 1.5，但是优选地在 1.1 和 1.7 之间。提高阈值超过 1.0 用作减少两个不相关短语同现超过随机预测的可能性。

[0080] 正如所强调的，关于给定的行 j，针对矩阵 G 的每列 k，重复计算信息增益。一旦完成了行，如果没有好的短语 g_k 的信息增益超过信息增益阈值，那么这意味着短语 g_j 不预测任何其他好的短语。在这个情况中，从好的短语列表 208 中移除 g_j，本质上成为坏的短语。注意：当此短语自己可以被其他好的短语预测时，不移除该短语 g_j 的列 j。

[0081] 当已经评估了同现矩阵 212 的所有行时，此步骤结束。

[0082] 此阶段的最终步骤是整理好的短语列表 208，以移除不完整短语。不完整短语是仅预测自己的短语扩展的短语，并且其开始于短语的最左侧（即，短语的开始）。短语 p 的“短语扩展”是开始于短语 p 的超序。例如，短语“President of”预测“President of the UnitedStates”、“President of Mexico”、“President of AT&T”等。由于这些短语开始于“President of”并且是它的超序，后面所有它们是短语“President of”的短语扩展。

[0083] 因而，基于前面讨论的信息增益阈值，保持在好的短语列表 208 上的每个短语 g_j 将预测一些数目的其他短语。现在，对于每个短语 g_j，检索系统 110 执行与每个预测短语 g_k 的串匹配。串匹配测试是否每个预测短语 g_k 是短语 g_j 的扩展。如果所有预测短语 g_k 是短语 g_j 的短语扩展，那么短语 g_j 是不完整的，并且从好的短语列表 208 中移除，并且被加入不完整短语列表 216。因此，如果存在至少一个不是 g_j 的扩展的短语 g_k，那么 g，是完整的，并且保留在好的短语列表 208 中。例如，“President of the United”是不完整短语，这是因为它预测的唯一其他短语是它的扩展“President of the United States”。

[0084] 不完整短语列表 216 本身在实际搜索期间非常有用。当接收敌搜索查询，针对不完整短语列表 216，比较该搜索查询。如果该查询（或者它的部分）匹配列表中的一个条目，那么搜索系统 120 可以查找不完整短语的最可能短语扩展（相对于不完整短语具有最高信息增益的短语扩展），并且将此短语扩展提示给用户，或者自动对短语扩展进行搜索。例如，如果搜索查询是“President of the United”，搜索系统 120 可以自动向用户提示“President of the United States”作为搜索查询。

[0085] 在索引过程的最后阶段完成之后，好的短语列表 208 将包含大量已经在资料库中发现的好的短语。这些好的短语的每个将预测至少一个不是它的短语扩展的其他短语。即，使用每个好的短语以足够的频率和独立性来表示在资料库中表达的有意义的概念或者思想。不像使用预定或者手选短语的现存系统，好的短语列表反映资料库中实际被使用的短语。而且，由于当新文档被加入文档集合时周期性地重复上述爬过和索引过程，所以当新的短语进入词典时，索引系统 110 自动检测新的短语。

[0086] 2. 相关短语的识别和相关短语的集群

[0087] 参考图 4，相关短语识别过程包括下列功能操作。

[0088] 400：识别具有高信息增益值的相关短语。

[0089] 402 :识别相关短语的集群。

[0090] 404 :存储集群比特向量和集群号。

[0091] 现在,详细描述这些操作的每一个。

[0092] 首先,回想同现矩阵 212 包含好的短语 g_j,其中每一个使用高于信息增益阈值的信息增益预测至少一个其他好的短语 g_k。为了识别 400 相关短语,那么,对于每对好的短语 (g_j, g_k),将信息增益与相关短语阈值 (例如,100) 进行比较。即,在 $I(g_j, g_k)>100$ 的情况下,g_j 和 g_k 是相关短语。

[0093] 此高阈值被用于识别很好超越统计期望率的好的短语的同现。统计地,它意味着短语 g_j 和 g_k 同现超过期望同现率 100 倍。例如,考虑在文档中的短语“monica Lewinsky(莫妮卡莱温斯基)”,短语“BillClinton(克林顿)”在同样文档中很可能出现 100 次,那么短语“克林顿”很可能出现在任何随机选择的文档上。解释这个另一个方式是 :因为同现率是 100 ： 1,则预测的准确率是 99.999％。

[0094] 因而,任何小于相关短语阈值的条目 (g_j, g_k) 被清零,表示短语 g_j、g_k 不相关。同现矩阵 212 中的任何保留条目现在表示所有相关短语。

[0095] 然后通过信息增益值 $I(g_j,g_k)$ 排序同现矩阵 212 的每行 g_j 中的列 g_k,使得带有最高信息增益的相关短语 g_k 被最先列出。因此,针对给定的短语 g_j,此排序识别根据信息增益识别哪个其他短语最可能相关。

[0096] 下一步是确定 402 哪些相关短语一起组成相关短语的集群。集群是其中每个短语关于至少一个其他短语具有高信息增益的相关短语的集合。在一个实施例中,集群被如下识别。

[0097] 在矩阵的每行 g_j 中,将有一个或多个相关于短语 g_j 的其他短语。此集合是相关短语集合 R_j,其中 $R = \{g_k, g_1, \ldots g_m\}$。

[0098] 对于 R_j 中的每个相关短语 m,索引系统 110 确定是否 R 中的每个其他相关短语也相关于 g_j。因此,如果 $I(g_k,g_1)$ 也是非零,那么 g_j,g_k 和 g_1 是集群的一部分。针对 R 中的每对 (g_1, g_m),重复此集群测试。

[0099] 例如,假设好的短语“比尔克林顿”相关于短语“总统”、“莫妮卡莱温斯基”,因为这些短语的每一个关于“比尔克林顿”的信息增益超过了相关短语阈值。进一步假设短语“莫妮卡莱温斯基”相关于短语“皮包设计师”。那么,这些短语组成集合 R。为了确定该集群,索引系统 110 通过确定它们的相应信息增益来评估这些短语的每个到其他短语的信息增益。因此,对于 R 中的所有对,索引系统 110 确定信息增益 I(“总统”,“莫妮卡莱温斯基”)、I(“总统”,“皮包设计师”) 等等。在此例子中,“比尔克林顿”、“总统”和“莫妮卡莱温斯基”组成一个集群,“比尔克林顿”和“总统”组成第二集群,并且“莫妮卡莱温斯基”和“皮包设计师”组成第三集群,并且“莫妮卡莱温斯基”、“比尔克林顿”和“皮包设计师”组成第四集群。这是因为虽然“比尔克林顿”没有使用足够的信息增益预测“皮包设计师”,但是“莫妮卡莱温斯基”预测了这两个短语。

[0100] 为了记录 404 集群信息,分配给每个集群唯一集群号 (集群 ID)。然后,记录此信息连同每个好的短语 g_j。

[0101] 在一个实施例中,集群号通过也表示短语间正交关系的集群比特向量确定。集群比特向量是长度为 n 的比特序列,好的短语列表 208 中好的短语的数目。对于给定的好的

短语 g_j，比特位置对应于 g_j 的排序相关短语 R。如果 R 中的相关短语 g_k 和短语 g_j 在相同的集群中，则比特被设定。更通常地，如果在 g_j 和 g_k 间的任一方向存在信息增益，这意味着集群比特向量中的相应比特被设定。

[0102] 那么，集群号是所产生的比特串的值。此实现具有在相同集群中出现拥有多路或者单路径信息增益的相关短语的性质。

[0103] 集群比特向量的例子如下，使用上面的短语：

[0104]

	比尔克林顿	总统	莫妮卡莱温斯基	皮包设计师	集群 ID
比尔克林顿	1	1	1	0	14
总统	1	1	0	0	12
莫妮卡莱温斯基	1	0	1	1	11
皮包设计师	0	0	1	1	3

[0105] 总之，此过程之后，对于每个好的短语 g_j、识别相关短语 R 的集合，其中按照信息增益 $I(g_j, g_k)$ 从高到低排序相关短语 R。另外，对于每个好的短语 g_j，存在集群比特向量和正交值，其中集群比特向量的值是识别短语 R_j 是成员的主集群的集群号，并且正交值（用于每个比特位置的 1 或者 0）表示 R 中的哪些相关短语与 R_j 在共同的集群中。因此在上面的例子中，基于短语“比尔克林顿”的行中的比特值，“比尔克林顿”、“总统”和“莫妮卡莱温斯基”位于集群 14 中。

[0106] 为了存储该信息，两个基本表示是可用的。第一，如上所示，信息可以存储在同现矩阵 212 中，其中：

[0107] 条目 G[rowj, col.k] = (I(j,k)，集群号，集群比特向量)。

[0108] 可选地，可以避免矩阵表示，并且所有信息存储在好的短语列表 208 中，其中每行代表好的短语 g_j：

[0109] 短语 row_j = 列表 [短语 g_k(I(j,k)，集群号，集群比特向量)]。

[0110] 此方法对于集群提供有用的组织。首先，不是严格而经常是任意限定主题和概念的层级，此方法认识到：由相关短语表示的主题组成复杂的关系图，其中一些短语和很多其他短语相关，并且一些短语具有更有限的范围，并且其中关系可以是相互的（每个短语预测其他短语）或者单向的（一个短语预测其他，但不是反之亦然）。结果是集群对于每个好的短语可以被赋予特征“局部”，并且一些集群将通过具有一个或多个共有相关短语而重叠。

[0111] 然后，对于给定的好的短语 g_j，相关短语的通过信息增益的排序提供用于命名该短语的集群的分类法：集群名是在集群中具有最高信息增益的相关短语的名称。

[0112] 上面过程提供一种识别出现在文档集合中重要短语的非常鲁棒的方法，以及有益地，提供一种在实践中这些相关短语被一起使用于自然“集群”中的方法。结果，此相关短语的数据驱动集群避免了在相关词和概念的任何人工直接“编辑”选择中固有的偏差，正如许多系统中所共有的偏差。

[0113] 3. 使用短语和相关短语索引文档

[0114] 考虑好的短语列表 208，包括关于相关短语和集群的信息，索引系统 110 的下一个功能操作是关于好的短语和集群索引文档集合中的文档，并且存储将更新信息存储在主索引 150 和次索引 152 中。图 5 说明了这个过程，其中存在用于索引文档的下列功能阶段：

[0115] 500：将文档置入在文档中找到的好的短语的置入列表。

[0116] 502：针对相关短语和次相关短语，更新实例计数和相关短语比特向量。

[0117] 504：根据置入列表尺寸重新排列索引条目。

[0118] 506：通过信息检索分值或者特征值对每个置入列表中的索引条目排序。

[0119] 508：在主服务器 150 和次服务器 152 之间划分每个置入列表。

[0120] 现在详细描述这些阶段。

[0121] 如前，遍历或者爬过的文档集合；这可以是相同或者不同的文档集合。对于给定的文档 d，使用长度为 n 的序列窗口 302 从位置 i 以上述该方式逐个单词地遍历 500 文档。

[0122] 在给定短语窗口 302 中，从位置 i 开始，识别窗口中所有好的短语。每个好的短语以 g_j 代表。因此，g1 是第一个好的短语，g2 将是第二个好的短语，以此类推。

[0123] 对于每个好的短语 g_1（例如 g1“President”和 g4“PresidentofATT”），将文档标识符（例如，URL）置入索引 150 中好的短语 g_j 的置入列表。此更新识别出现在此特定文档中的好的短语 g_j。

[0124] 在一个实施例中，短语 g_j 的置入列表采用下列逻辑形式：

[0125] 短语 g_j：列表（文档 d，[列表：相关短语计数][相关短语信息]）

[0126] 对于每个短语 g_j 存在短语出现在上面的文档 d 的列表。对于每个文档，存在也出现在文档 d 中的短语 g_j 的相关短语 R 的出现次数的计数列表。

[0127] 在一个实施例中，相关短语信息是相关短语比特向量。此比特向量可以被赋予特征作为“双比特”向量，其中对于每个相关短语 g_k，存在两个比特位置，g_k-1、g_k-2。第一个比特位置存储表示是否相关短语 g_k 出现在文档 d 中的标记（即，文档 d 中 g_k 的计数大于 0）。第二比特位置存储表示是否 g_k 的相关短语 g_1 也出现在文档 d 中。短语 g_j 的相关短语 g_k 的相关短语 g_1 这里称作“g_j 的次相关短语”。计数和比特位置相应于 R 中短语的典范次序（以递减的信息增益顺序排序）。此排序顺序具有使相关短语 g_k 是由与相关短语比特向量的最高有效比特相关联的 g_j 最高度地预测的，并且相关短语 g_1 是由与最低有效比特相关联的 g_1 最少预测的效果。

[0128] 注意，对于给定的短语 g，关于包含 g 的所有文档，相关短语比特向量的长度以及相关短语到该向量的各个比特的相关性是相同的。此实现具有允许系统容易比较针对任何（或者所有）包含 g 的文档的相关短语比特向量的性质，以发现哪个文档具有给定的相关短语。这对于便于搜索过程响应于搜索查询来识别文档是有益的。因而，给定的文档将出现在很多不同短语的置入列表中，并且在每个这样的置入列表中，该文档的相关短语向量对于拥有该置入列表的短语将是特定的。这方面保持了关于单独短语和文档的相关短语比特向量的位置。

[0129] 因而，下一阶段 502 包括在文档中遍历当前索引位置的次窗口 304（如前 +/-K 个词的次窗口，例如，30 个词），例如从 i-K 到 i+K。对于出现在次窗口 304 中的每个 g1 的相关短语 g_k，索引系统 110 在相关短语计数中增加关于文档 d 的 g_k 的计数。如果 g_i 稍后出现在文档中，并且再次在后面的次窗口中发现相关短语，则再次增加计数。

[0130] 如提到的，基于计数，设置相关短语比特位图中的相应第一比特 g_k-1，如果 g_k 的计数大于 0，则比特设置为 1，或者如果计数等于 0，则设置为 0。

[0131] 接下来，通过查寻索引 150 中的相关短语 g_k 设置第二比特 g_k-2，在 g_k 的置入列表中识别文档 d 的条目，然后，并且针对 g_k 的任何相关短语，检查 g_k 的 $_k$ 的次相关短语计数（或比特）。如果任何这些次相关短语计数 / 比特被设置，那么这表示 g_j 的次相关短语也出现在文档 d 中。

[0132] 当文档 d 已经以此方式被完全处理过时，索引系统 110 已经识别下列：

[0133] i) 文档 d 中的每个好的短语 g_j；

[0134] ii) 对于每个好的短语 g_1 它的哪些相关短语 g_k 出现在文档 d 中；

[0135] iii) 对于出现在文档 d 中的每个相关短语 g_k，它的哪些相关短语 g_l（g_l 的次相关短语）也出现在文档 d 中。

[0136] a) 划分的索引

[0137] 基于短语在资料库中的出现频率，索引 150 中的每个短语被赋予短语号。越普通的短语，在索引中它接收的短语号越低。然后，索引系统 110 根据列于每个置入列表中的文档号降序排序 504 主索引 150 中的所有置入列表 213，以便最频繁出现的短语具有最低的短语号并且在主索引 150 中首先列出。如上面看到的，主索引 150 被分布在 M1 主服务器上。为了减少磁盘竞争，通过散列函数，例如 phase_number MOD M1，将短语分布在这些机器上。

[0138] 为了显著地增加可以由系统索引的文档数目，主索引 150 被进一步处理以选择性地划分每个置入列表 214。如上面看到的，每个短语的置入列表包含文档的列表。置入列表中的每个文档被给出 506 关于短语的信息检索类型分值。然而，计算该分值，然后置入列表中的文档通过此分值以降序被排序，最高分值的文档在置入列表中列于第一。当响应于查询检索文档时，此文档预排序对于改进性能特别有益。

[0139] 文档预排序的分值算法可以是与使用来在搜索系统 120 中生成相关分值的相同的基本相关分值算法。在一个实施例中，IR 分值是基于网页排序算法，如美国专利号 6,285,999 描述的。可选地或是额外地，也可以存储以及单独或组合使用文档的多个 IR 相关属性的统计，诸如内链接、外链接、文档长度的数目，以便排序文档。例如，根据内链接数目，文档可以以降序被排序。为了进一步便于从主索引 150 进行信息的最快可能检索，通过 IR 类型分值排列次序将每个置入列表 214 中的条目物理地存储在适当的主服务器上。

[0140] 考虑到对于给定短语的最高等分文档现在位于置入列表的开始，置入列表 214 在主索引 150 和次索引 152 间被划分开。对于直到第一 K 个保留文档的置入列表条目存储在主服务器 150 上。在一个实施例中，K 被设置为 32,768(32K)，但是可以使用 K 的较高或者较低的值。具有在主和次索引之间划分开的置入列表的短语被称作‘共同’短语，然而没有划分开的短语被称作‘稀有’短语。存储于主索引 150 中的置入列表的部分被称作主置入列表，并且包含主条目，并且存储于次索引 152 中的置入列表的部分被称作次置入列表，并且包含次条目。对于给定置入列表 214 的次条目根据短语号的另一个哈什函数，例如，短语号 MOD M2，被分配给次服务器。次服务器 ID 存储在主服务器上的置入列表中，以允许搜索系统 120 容易地访问需要的适当的次服务器。对于每一个存储在一个次服务器上的短语置入列表，次条目以它们文档号的顺序，从最低文档号到最高（对比于主索引 150 中的相关性排序）物理地存储。优选地，没有相关性信息存储在次条目中，以便条目包含最小数据量，诸

如文档号以及文档定位符（例如 URL）。排序和划分步骤对于每个短语可以顺序执行；可选地，所有（或者多个）短语可以先被排序，然后划分；算法设计只是一个设计选择并且上述变化可以被同等考虑。在每个索引通过文档集合期间，进行排序和划分步骤，以便索引单程期间任何随新文档被更新的短语可以重排序以及重划分。其他最优化和操作也是可能的。

[0141] 在一个实施例中，对于置入列表 214 中每个短语的主索引 150 中存储的文档属性的选择是可变的，并特别地向着主索引中置入列表 214 的末端减少。换句话说，基于它们的相关性分值（或者其他基于相关性的属性）在置入列表中排序较前的文档将具有所有或者大部分存储在置入列表的文档条目中的文档属性。主索引中的置入列表 214 末端附近的文档将仅具有更有限的存储的这种属性的集合。

[0142] 在一个实施例中，主索引 150 中的每个置入列表 214 具有三分段（或者等级），长度为 m、3m、5m，其中 m 这里是文档条目的数目，在此实施例中，希望每个分段具有长度 K，如上描述，即 m ＝ K，并且条目主索引具有 9K 个条目；那么，次索引将存储次条目，其中 n>9K。

[0143] 在第一分段中（第一 m 个条目），针对给定短语的置入列表中的每个文档条目，存储下列相关性属性：

[0144] 1. 文档相关性分值（例如，网页排序）；

[0145] 2. 文档中该短语的出现总数目；

[0146] 3. 也包含该短语并且指向该文档的多达 10,000 个锚文档的排列次序的列表，并且对于每个锚文档，它的相关性分值（例如，网页排序），以及锚文本本身；并且

[0147] 4. 每个短语出现位置，以及对于每个出现的一组标记，所述标记表示出现是否是题目、粗体、标题、位于 URL 中、位于主体中，位于工具条中、位于页脚中，位于广告中、大写的、或者位于某些类型的 HTML 标记中。

[0148] 在第二分段中（接下来的 3m 个条目），仅存储条 1-3。

[0149] 在第三分段中（最后 5m 个条目），仅存储条 1。

[0150] 系统地减少存储在每个置入列表 214 的后面部分的文档属性是可接受的，因为置入列表末端附近的文档已经被确定与特定短语很少相关（低相关性分值），并且因此不需要全部存储它们的所有相关性特性。

[0151] 前述存储安排使得能够在给定硬盘存储容量的情况下存储显著多于传统技术的条目。第一，对于每个文档中每个短语的词位置信息的消除提供了对于给定文档集合所需存储容量 50％的削减，因而有效加倍了可以存储文档的数目。第二，在主索引和次索引间划分置入列表以及仅在主索引中存储相关性信息提供了进一步实质的分段省。很多短语在他们的置入列表中具有超过 100,000，甚至 1,000,000 个文档。仅在主索引中存储有限数量条目的相关性信息消除了不太可能在搜索中被返回的文档的存储需要。这个方面将可存储文档数量增加了大约 10 倍。最后，通过针对每个置入列表 214 中的很少相关（较低排序的）文档来选择性地在主索引 150 中存储更少的相关性信息，实现了进一步的分段省（大概削减 25％ -50％所需存储容量）。

[0152] b) 确定文档的主题

[0153] 通过短语所引文档以及实现集群信息提供了索引系统 110 的另一个优势，其能够基于相关短语信息确定关于文档的主题。

[0154] 假设对于给定好的短语 g_j 并给定文档 d，置入列表条目如下：

[0155] gj :文档 d :相关短语计数 := {3,4,3,0,0,2,1,1,0}

[0156] 相关短语比特向量 := {1111100000101010 01}

[0157] 其中,相关短语比特向量示出于比特 - 比特对中。

[0158] 从相关短语比特向量,可以确定文档 d 的主和次主题。主主题由比特对 (1,1) 表示,并且次主题由比特对 (1,0) 表示。相关短语比特对 (1,1) 表示针对比特对的相关短语 g_j 出现在文档 d 中,次相关短语 g_l 也是如此。这可以解释为文档 d 的作者在撰写文档中一起使用几个相关短语 g_j、g_k 和 g_l。比特对 (1,0) 表示 g_j 和 g_k 都出现,但是没有来自 g_k 的其他次相关短语出现,并且因此这是不大重要的主题。

[0159] c) 针对档案检索来索引文档实例

[0160] 本发明的另一个实施例支持在索引中存储和保留历史文档的能力,并且因而支持各个文档或者网页的日前特定实例 (版本) 的档案检索。这个能力具有不同的有益用途,包括使用户可以搜索在特定日期范围的文档,使搜索系统 120 可以在响应于搜索查询来评估文档以及组织搜索结果时,使用日期或者版本相关的相关性信息。

[0161] 在此实施例中,文档标识符针对日期间隔编码文档标识。在索引系统 110 第一次爬过文档时,文档标识符被存储作为文档 URL 和文档的日期戳的散列,例如,MD5(URL,第一日期)。日期范围字段与文档的特定实例相关联,其包括对于被认为是有效的文档实例的日期范围。日期范围可以被指定作为日期对,包括文档被认为有效的第一日期 (索引日期) 和文档被认为有效的最后日期 (例如,11-01-04 ;12-15-04)。可选地,日期范围可以被指定作为第一日期,以及表示在第一日期之后的天数的数目 (例如,11-01-04,45)。日期可以被指定为任何有效的格式,包括日期字符串或者天数。在文档是当前有效文档周期期间,第二值是表示该状态的状态标记或者令牌 (包括 NULL 值) ;这被称作当前间隔。例如,(11-01-04,“打开”) 表示文档当前是有效的。这表示文档将满足包括第一日期后的日期限制的搜索。不论特定的实现,针对给定日期间隔的第一日期可以被称作“打开日期”,并且针对给定间隔的最后日期可以被称作“关闭日期”。

[0162] 在由索引系统 110 随后的索引单程期间,索引系统 110 确定是否文档已经改变。如果文档中不存在改变,那么索引系统 110 不对文档采取进一步的行动。如果文档已经改变 (文档的新实例或者版本),那么索引系统 110 重索引文档。在重索引时,索引系统 110 通过改变打开状态标记为当前日期减一天,来关闭当前间隔。例如,如果索引系统 110 在 12/16/04 索引文档并且确定文档已经改变,那么关闭当前间隔如下 :(11-01-04,12-15-04),并且创建新的当前间隔,例如,(12-16-04,“打开”)。索引系统 110 保持文档的每个日期范围,连同日期范围的相应索引相关性数据 (例如,短语,相关性统计、文档内链接等等)。因此,每个日期范围和相关性数据集合与文档的特定实例或版本相关联。对于用于给定文档的每个日期间隔,索引系统保持唯一文档标识符,例如 MD5(URL,第一日期),以便能够检索适当缓存的文档实例。在使用主和次索引的实施例中,当完成索引单程时,主索引中的置入列表 214 被重评分、重排序、并且重划分。

[0163] 给定文档是否自上次索引单程已经改变的确定可以以任何方式做出,包括使用统计规则,语法规则或者类似启发式规则。在一个实施例中,索引系统 110 使用文档短语确定是否文档已经改变。每当索引文档时,前 N 个主题被识别和保持作为与日期范围信息相关联的列表,例如,日期范围 (11-04-04,12-15-04) 的前 20 个主题。被索引的实例主题列表

然后与现有文档实例的主题列表（优选地最近关闭日期范围）进行比较。如果超过 M％的主题已经改变（例如，5％），那么文档被认为改变，并且针对所有短语进行重索引。应该注意：确定文档是否已经改变的其他方法也可以使用，以及基于短语的索引使用不是必需的。例如，可以基于文档长度的改变、最频繁词的改变、词频的改变、HTML 标记类型数量的改变、或者文档结构和内容的其他方法，来使用统计规则集合。

[0164] III. 搜索系统

[0165] 搜索系统 120 操作以接收查询并且搜索与查询相关的文档，并且在搜索结果集合中提供这些文档（使用到文档的链接）的列表。图 6 说明了搜索系统 120 的主要功能操作：

[0166] 600：识别查询中的短语。

[0167] 602：检索与查询短语相关的文档。

[0168] 604：根据短语对搜索结果中的文档进行排序。

[0169] 这些阶段的每个的细分段如下。

[0170] 1. 识别查询和查询扩展中的短语

[0171] 搜索系统 120 的第一阶段 600 是识别在查询中出现的任何短语，以效地搜索索引，在该部分中使用了以下术语：

[0172] q：作为输入并且由搜索系统 120 接收的查询。

[0173] Qp：查询中出现的短语。

[0174] Qr：Qp 的相关短语。

[0175] Qe：Qp 的短语扩展。

[0176] Q：Qp 和 Qr 的联合。

[0177] 从客户端 190 接收查询 q，具有多达特定最大数量的字符或单词。

[0178] 由搜索系统 120 使用的尺寸为 N（例如，5）的短语窗口，以遍历查询 q 的词。短语窗口开始于查询的第一词，向右扩展 N 个词。然后，此窗口向右移动 M-N 次，其中 M 是查询中词的数目。

[0179] 在每个窗口位置，窗口中存在 N 个词（或更少）。这些词组成可能的查询短语。在好的短语列表 208 中查找可能的短语，以确定它是否是好的短语。如果可能短语出现在好的短语列表 208 中，那么返回针对短语的短语号；现在可能的短语是候选短语。

[0180] 在每个窗口中的所有可能短语已经被检验以确定它们是否是好的候选短语后，搜索系统 120 将具有针对查询中相应短语的短语号集合。然后，然口对这些短语号进行排序（降序）。

[0181] 开始于最高短语号作为第一候选短语，搜索系统 120 确定排序列表内的固定数字距离中是否存在另一个候选短语，即，短语号间的差值在阈值量内，例如，20,000。如果这样，那么查询中最左端的短语被选作有效查询短语 Qp。此查询短语和所有它的子短语被从候选者列表中移除，并且列表被重排序并且重复处理。此处理结果是有效查询短语 Qp 的集合。

[0182] 例如，假设搜索查询是“Hillary Rodham Clinton Bill on the SenateFloor”。搜索系统120将识别下列候选短语，“Hillary Rodham ClintonBill on”、“Hillary Rodham Clinton Bill”和“Hillary Rodham Clinton”。前两个被放弃，并且最后一个作为有效查询

短语被保留。接下来搜索系统 120 将识别“Bill on the Senate Floor”以及子短语“Bill on theSenate”、“Bill on the”、“Bill on”、Bill “ ”并且将选择“Bill”作为有效查询短语 Qp。最后，搜索系统 120 将解析短语“on the senate floor”并且确定“Senate Floor”作为有效查询短语。

[0183] 接下来，搜索系统 120 调整有效短语 Qp 用于首字母大写。当解析该查询时，搜索系统 120 在每个有效短语中识别潜在的首字母大写。使用已知的首字母大写表，诸如首字母大写为“United States”的“united states”，或者通过使用基于语法的首字母大写算法，可以完成这个。这就生成了适当的首字母大写的查询短语集合。

[0184] 然后，搜索系统 120 使第二单程通过首字母的大写短语，并且仅选择最左端并首字母大写的短语，其中短语和它的子短语出现在集合中。例如，关于“president of the united states”的搜索将被首字母大写作“President of the United States”。

[0185] 在下一个阶段，搜索系统 120 识别 602 与查询短语 Q 相关的文档。搜索系统 120 然后检索查询短语 Q 的置入列表，并且如果需要，对这些列表取交集 (intersect)，以确定哪些文档出现在针对查询短语的所有（或一些）置入列表上。如果查询中的短语 Q 具有短语扩展 Qe 集合（下面进一步描述），那么搜索系统 120 在作对置入列表取交集之前，首先组成短语扩展置入列表的联合。如上所述，搜索系统 120 通过在不完整短语列表 216 中查找每个查询短语 Q 来识别短语扩展。

[0186] 使用主索引 150 和次索引 152，搜索系统 120 可以进一步优化取交集操作。基于查询短语是否是普通的或者稀有的，存在搜索系统 120 不得不处理的四种常用的取交集分析情况。

[0187] 第一种情况是针对单查询短语，其既可以是普通的也可以是稀有的。在此情况中，搜索系统 120 将来自于主索引 150 的短语的置入列表中选择数量的（例如，100 或者 1000）第一条目传递到排序阶段 604，用于最终排序。由于文档已经按照排序次序，排序阶段可以优化排序操作。可选地，由于通过它们与短语的相关性已经预排序了这些，所以可以直接将文档集合作为搜索结果提供，基本上提供瞬时结果给用户。

[0188] 第二种情况是其中存在两个普通查询短语。这里，搜索系统 120 访问主索引 150 中每个短语的置入列表 214 并且对这些列表取交集，以形成最终文档列表，然后该最终文档列表被传送到排序阶段 604，用于基于与文档相关联的相关性属性集合进行相关性评分。因为在每个置入列表中存在至少 K 个文档，所以存在包含两个短语的足够数量的文档的极高可能性，并且因此次索引 152 中的次条目的取交集不是必需的。这进一步减少了检索所需的时间量。

[0189] 第三种情况是其中存在两个稀有查询短语。此情况以与第二种情况相同的方式处理，因为在这里每个短语的全部置入列表存储在主索引中。

[0190] 最后一种情况是其中有效查询短语包括普通短语和稀有短语。在此情况中，搜索系统 120 首先对来自于用于两个短语的主索引 150 的置入列表 214 取交集，以形成第一集合或者普通文档。接下来，搜索系统 120 对稀有短语的置入列表与普通短语的次条目（其已经以文档号顺序排序）取交集，以形成普通文档的第二集合。将两个集合结合并且然后传递到排序阶段。

[0191] 其中可以由使用上述方法的一个连续的取交集来处理存在三个或更多查询短语

的所有实例。

[0192] 2. 排序

[0193] a) 基于包含的短语对文档排序

[0194] 搜索系统 120 提供排序阶段 604，其中使用相关性信息和文档属性连同每个文档的相关短语比特向量中的短语信息，以及用于查询短语的集群比特向量，对搜索结果中的文档排序。此方法根据包含在文档中的短语，或者非正式地“主体索引项 (body hits)”，来排序文档。

[0195] 如上所述，对于任何给定短语 g_j，g_j 的置入列表中的每个文档 d 具有关联的相关短语比特向量，其识别哪个相关短语 g_k 以及哪个次相关短语 g_l 出现在文档 d 中。出现在给定文档中的相关短语和次相关短语越多，将在针对给定短语的文档的相关短语比特向量中设置的比特越多。被设置的比特越多，相关短语比特向量的数值越大。

[0196] 因而，在一个实施例中，搜索系统 120 根据它们的相关短语比特向量值对搜索结果中的文档排序。包含与查询短语 Q 最相关短语的文档将具有最高值的相关短语比特向量，并且这些文档将成为搜索结果中排序最高的文档。

[0197] 此方法是希望的，这是因为语义上这些文档都与查询短语在主题上最相关。请注意，因为相关短语信息被用于识别相关文档并且然后对这些文档排序，所以尽管文档不包含输入查询词 q 的高频率，此方法也提供了高度相关的文档。带有输入查询词的低频率的文档仍旧可以具有大量的与查询词和短语相关的短语，并且因此比具有仅查询词和短语的高频率但是没有相关短语的文档更相关。

[0198] 在第二实施例中，根据文档包含的查询短语 Q 的相关短语，搜索系统 120 对结果集合中的每个文档评分。如下完成这个过程：

[0199] 给定每个查询短语 Q，将存在与查询短语相关的 n 个短语，如短语识别过程期间所识别的。如上所述，相关查询短语 Qr 根据来自于查询短语 Q 的它们的信息增益被排序。然后，这些相关短语被分配点数，开始于针对第一相关短语 Qr1 的 N 个点（即，具有来自于 Q 的最高信息增益的相关短语 Qr），然后针对下一个相关短语 Qr2 是 N-1 个点，然后，对于 Qr3 是 N-2 个点，以此类推，以便给最后相关短语 QrN 分配一个点。

[0200] 然后，通过以下方式，给搜索结果中的每个文档评分，即通过确定查询短语 Q 的那个相关短语 Qr 出现，并且将分配给每个这样的相关短语 Qr 的点数提供给文档。然后文档被从最高分到最低分排序。

[0201] 当进一步提炼时，搜索系统 120 可以从结果集合中选出特定文档。在某些情况中，文档可以与很多不同主题有关；特别是对于更长的文档是尤其是这样。在很多情况中，与关于许多不同主题的文档相比，用户更喜欢针对在查询中表达的单个主题的文档。

[0202] 为了精选这些后面文档类型，搜索系统 120 使用查询短语的集群比特向量中的集群信息，并且移除其中存在超过文档中集群阈值数目的任何文档。例如，搜索系统 120 可以移除包含多于两个集群的任何文档。这个集群阈值可以是预定的，或者由用户作为搜索参数设定的。

[0203] b) 基于锚短语对文档排序

[0204] 除了基于查询短语 Q 的主体索引项对搜索结果中的文档进行排序外，在一个实施例中，搜索系统 120 也基于查询短语 Q 和相关查询短语 Qr 在到其他文档的锚点 (anchor)

中的出现，对文档排序。在一个实施例中，搜索系统 120 对于每个文档计算分值，其是两个分值的函数，主体项索引分值和锚索引项分值。

[0205] 例如，对于给定文档的文档分值可以计算如下：

[0206] 分值＝.30*（主体索引项分值）+.70（锚索引项分值）。

[0207] .30 和 .70 的权重可以根据希望调整。按照上述方式，考虑查询短语 Q，文档的主体索引项分值是文档的最高值相关短语比特向量的数值。可选地，此值可以由搜索系统 120 通过在索引 150 中查找每个查询短语 Q，从查询短语 Q 的置入列表中访问文档，并且然后访问相关短语比特向量来直接获得。

[0208] 文档 d 的锚索引项分值是查询短语 Q 的相关短语比特向量的函数，其中 Q 是参考文档 d 的文档中的锚项。当索引系统 110 索引文档集合中的文档时，它为每个短语保持一个文档列表，其中在所述文档中短语是外链接中的锚文本，以及也为每个文档保持来自其他文档的内链接（以及关联的锚文本）列表。文档的内链接是从其他文档（参考文档）到给定文档的参考（例如，超链接）。

[0209] 为了确定给定文档的锚索引项分值，搜索系统 12 通过锚短语 Q 反复通过索引中列出的参考文档 R(i ＝ 1 到参考文档的数目）的集合，病得出以下成绩：

[0210] R.Q. 相关短语比特向量 *D.Q. 相关短语比特向量。

[0211] 这里的乘积值是主题锚短语 Q 与文档 D 相关的分值。这个分值这里被称为“入站 (inbound) 分值组分”。此乘积通过参考文档 R 中的锚短语的相关比特向量有效地权衡了当前文档 D 的相关比特向量。如果参考文档 R 自己与查询短语 Q 相关（并且因此，具有更高值的相关短语比特向量），那么这将增加当前文档 D 分值的重要性。如上所述，主体索引项分值和锚索引项分值被组合以创建文档分值。

[0212] 接下来，对于每个参考文档 R，获得每个锚短语 Q 的相关短语比特向量。这是锚短语 Q 与文档 R 相关的方法。这里乘此值为出站 (outbound) 分值组分。

[0213] 然后，针对锚短语 Q，从索引 150 中抽取所有（参考文档，被参考文档）对。然后，这些对通过它们的关联（出站分值组分，入站分值组分）值被排序。取决于实现，这些组分的任意一个可以是主排序关键字，并且其他可以使次排序关键字。然后，排序结果被呈现给用户。按照出站分值组分对文档进行排序使对作为锚索引项的查询具有许多相关短语的文档排序最高，因此代表这些文档作为“专家”文档。按照入站分值组分对文档进行排序使由锚项频繁参考的文档排序最高。

[0214] c) 基于日期范围相关性对文档排序

[0215] 搜索和排序操作期间，搜索系统 120 可以以若干种方式使用日期范围信息。第一，搜索系统 120 可以使用日期范围作为明确的搜索分界符。例如，可以包括词或者短语以及日期，诸如“美国专利和商标局 12/04/04”。搜索系统 120 可以识别日期项，并且然后选择具有希望的短语并且被针对查询中包括日期项的日期范围索引的文档。然后，从选择的文档中，搜索系统 120 可以使用与日期范围相关联的索引的相关性数据，获得针对每个文档的相关性分值。以此方式，代替当前实例，可以检索文档较旧的或者以前的实例，其中与搜索查询更相关。这对于频繁改变的文档和页面特别有用，诸如包含频繁改变信息的新网站和其他网站的主页。

[0216] 第二，在搜索查询中不包括日期项的情况下，搜索系统 120 可以通过以下方式在

相关性排序期间使用索引中的日期信息，，即通过根据它们有多旧对文档相关性分值进行加权，以便较旧的文档使它们相关性分值权重下降（或者较新的文档权重更高）。可选地，在一些情况中，与主题最相关的是文档较旧的版本，而不是文档最当前版本。例如，在历史性事件时刻同时创建的新闻入口站点很可能比新入口的当前实例与关于事件的特定查询更相关。在此情况中，搜索系统 120 可以提升较旧文档实例的权重，其中例如，针对所有文档实例的文档相关性分值模式示出了围绕一些历史日期的增加，跟随着针对该文档更多当前实例的相关性分值的下降。

[0217] 在一个或者多个日期项包括在搜索查询中的情况中，如上所述，文档可以使它们的权重下降与日期词和文档日期范围间的差距成比例，以便比日前范围（从打开或者关闭日期衡量）更旧或者比希望的日期项更新的文档使它们的相关性分值权重下降。相反地，在文档的日期范围更靠近希望的日期的情况下，相关性分值权重增加，而不是权重下降。

[0218] 第三，搜索系统 120 可以使用日期范围信息作为用于对搜索结果排序的主要或者次要因素。例如，文档可以以逆时间顺序分组（例如，月分组），并且在每组中，文档可以从与搜索查询最相关到最不相关列出。

[0219] 数据范围信息的另一个使用是：基于文档的更新频率来排序文档。搜索系统 120 可以确定超过一时间间隔（索引期间可以保持此计数）的给定文档的实例数目（例如，不连续日期范围的数目）。然后，实例数目被用于提高那些更加频繁更新的文档的权重。

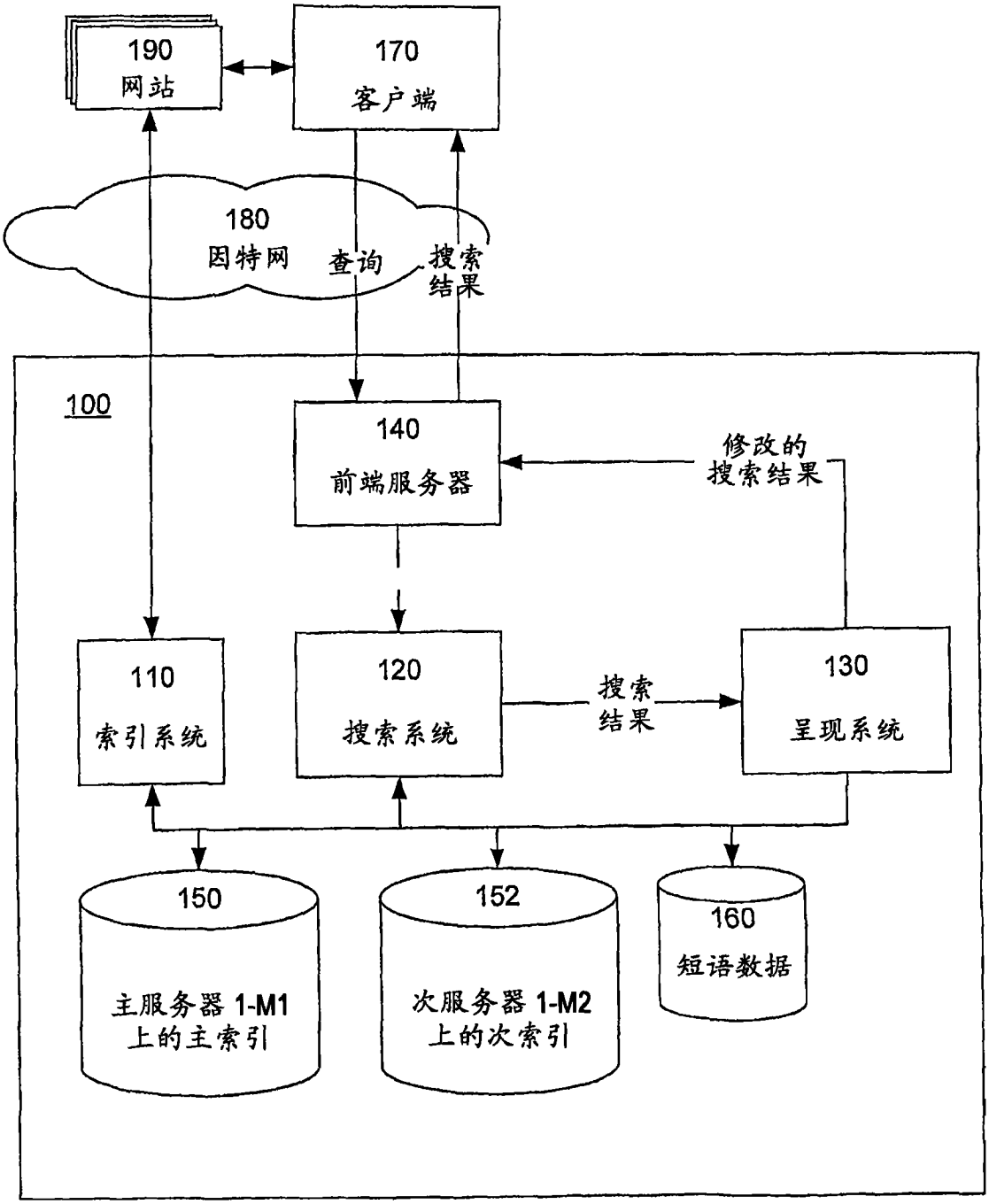

190
网站
170
客户端
180
因特网
查询
搜索结果
100
140
前端服务器
修改的搜索结果
110
索引系统
120
搜索系统
搜索结果
130
呈现系统
150
主服务器 1-M1 上的主索引
152
次服务器 1-M2 上的次索引
160
短语数据

图 1

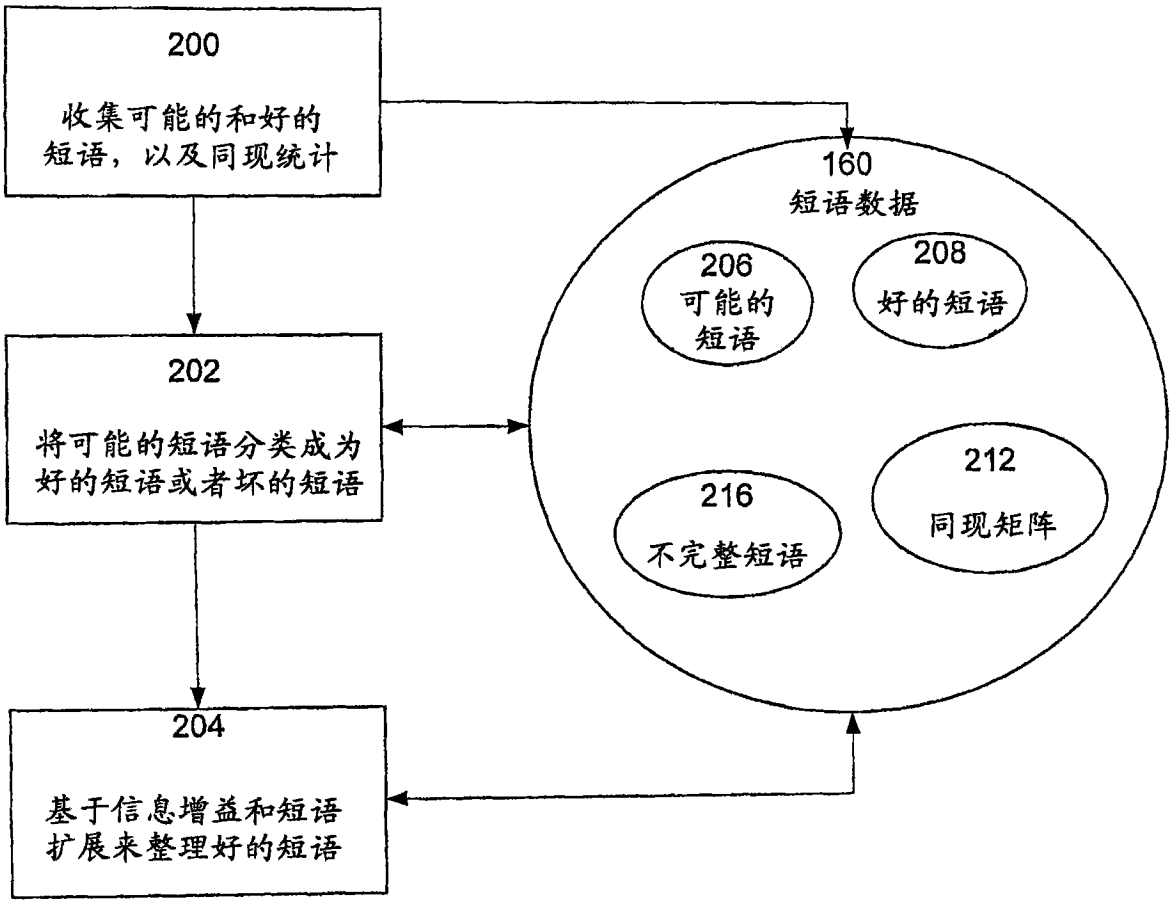

200
收集可能的和好的短语，以及同现统计
160
短语数据
206
可能的短语
208
好的短语
202
将可能的短语分类成为好的短语或者坏的短语
212
同现矩阵
216
不完整短语
204
基于信息增益和短语扩展来整理好的短语

图 2

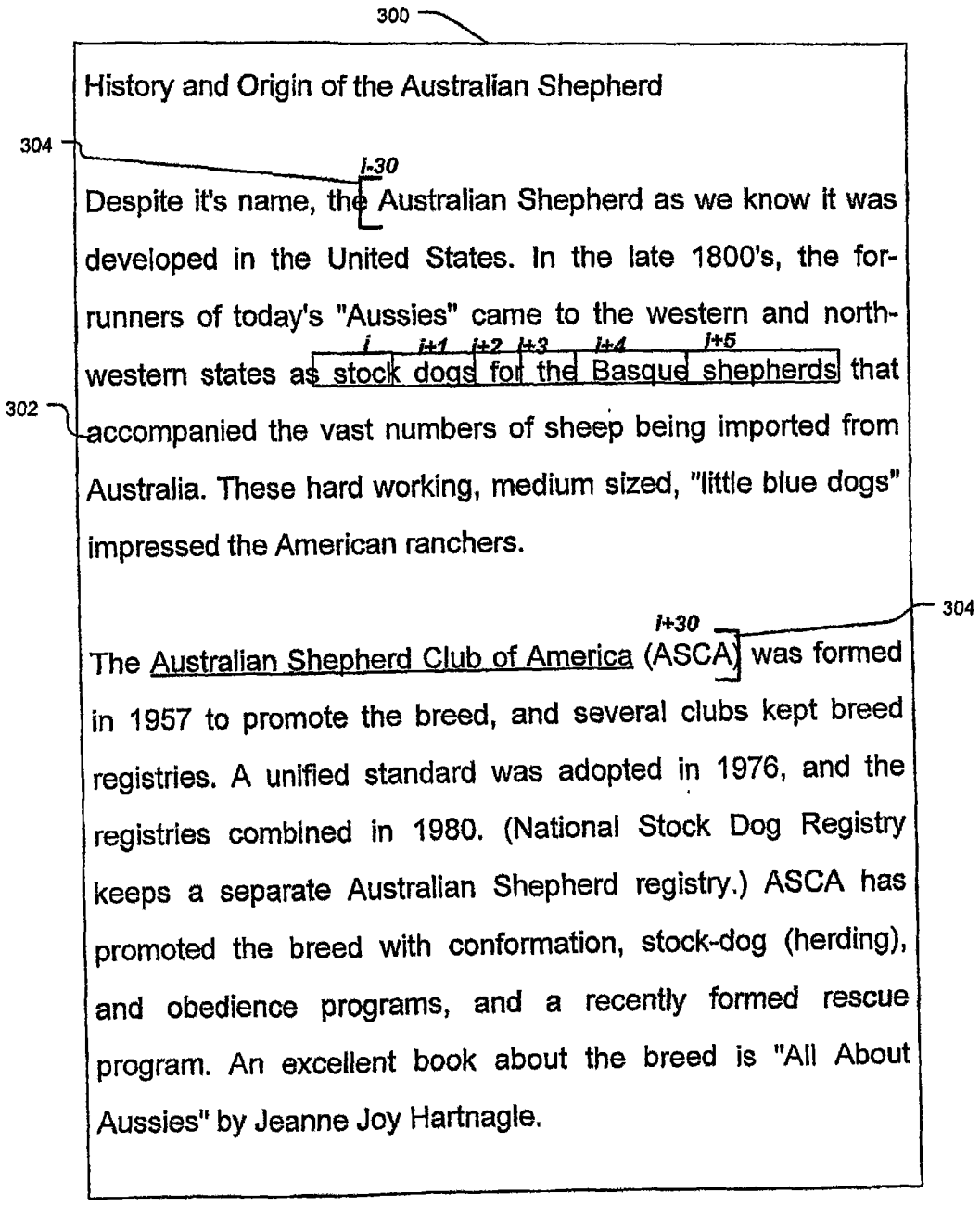

300
History and Origin of the Australian Shepherd
304
i-30
Despite it's name, the Australian Shepherd as we know it was developed in the United States. In the late 1800's, the forrunners of today's "Aussies" came to the western and northwestern states as stock dogs for the Basque shepherds that
i i+1 i+2 i+3 i+4 i+5
302
accompanied the vast numbers of sheep being imported from Australia. These hard working, medium sized, "little blue dogs" impressed the American ranchers.
304
i+30
The Australian Shepherd Club of America (ASCA) was formed in 1957 to promote the breed, and several clubs kept breed registries. A unified standard was adopted in 1976, and the registries combined in 1980. (National Stock Dog Registry keeps a separate Australian Shepherd registry.) ASCA has promoted the breed with conformation, stock-dog (herding), and obedience programs, and a recently formed rescue program. An excellent book about the breed is "All About Aussies" by Jeanne Joy Hartnagle.

图 3

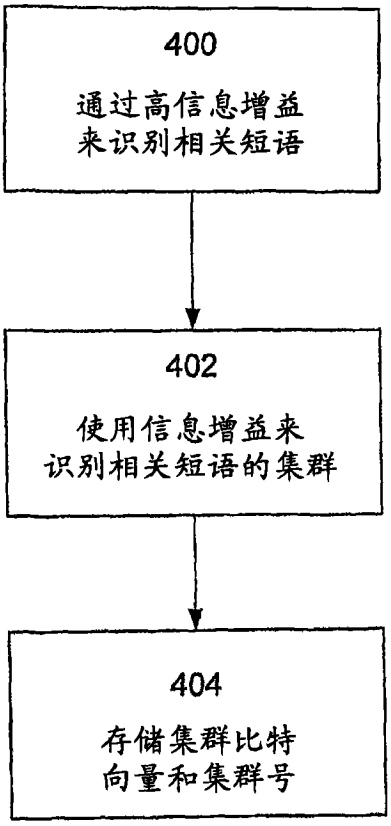

400
通过高信息增益
来识别相关短语
402
使用信息增益来
识别相关短语的集群
404
存储集群比特
向量和集群号

图 4

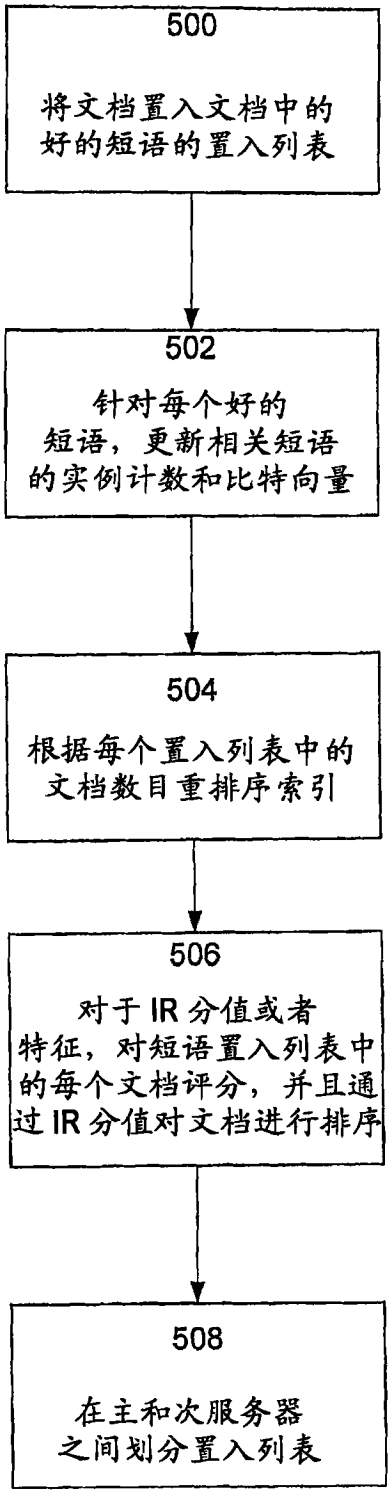

500
将文档置入文档中的
好的短语的置入列表
502
针对每个好的
短语，更新相关短语
的实例计数和比特向量
504
根据每个置入列表中的
文档数目重排序索引
506
对于 IR 分值或者
特征，对短语置入列表中
的每个文档评分，并且通
过 IR 分值对文档进行排序
508
在主和次服务器
之间划分置入列表

图 5

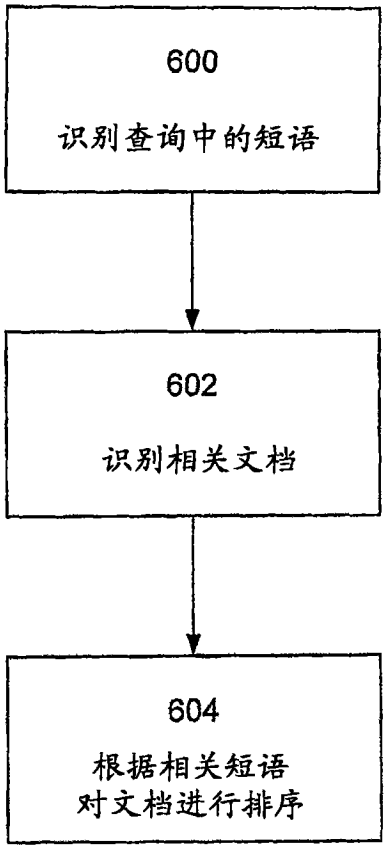


图 6