[19] 中华人民共和国国家知识产权局

[51] Int. Cl[7]
G06F 17/28

[12] 发明专利说明书

[21] ZL 专利号 99804423.7

[45] 授权公告日 2004 年 3 月 24 日

[11] 授权公告号 CN 1143232C

[22] 申请日 1999.11.18 [21] 申请号 99804423.7
[30] 优先权
[32] 1998.11.30 [33] EP [31] 98204038.8
[86] 国际申请 PCT/EP99/08942 1999.11.18
[87] 国际公布 WO00/33211 英 2000.6.8
[85] 进入国家阶段日期 2000.9.25
[71] 专利权人 皇家菲利浦电子有限公司
地址 荷兰艾恩德霍芬
[72] 发明人 朱亚成
审查员 张江峰

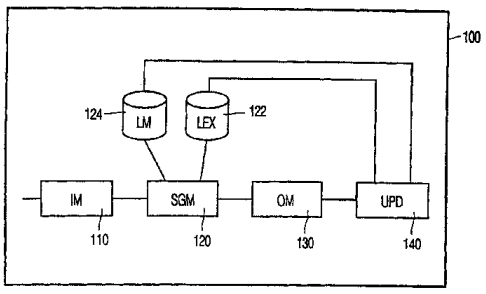
[74] 专利代理机构 中国专利代理(香港)有限公司
代理人 马铁良 陈景峻

权利要求书 3 页 说明书 14 页 附图 5 页

[54] 发明名称 正文的自动分割

[57] 摘要

一种系统(100)能够把诸如中文、日文句子的连贯正文分割为单词。该系统包括用于读取代表连贯正文的输入字符串的装置(110)，分割装置(120)通过以迭代方式建立代表输入字符串中的单词序列的树形结构来识别所述连贯正文中的至少一个单词序列。开始将输入字符串当作工作字符串。将词典中的每个单词与所述工作字符串的首部进行比较。利用树中的节点代表匹配，并且对于输入字符串的剩余部分，继续此处理。该系统还包括装置(130)，用于输出至少一个识别的单词序列。语言模型可以用于在候选序列之间进行选择。优选地，此系统用于语音识别系统中，以便根据提供的文本更新辞典。



ISSN1008-4274

1. 一种将连贯正文分割成单词的方法，包括如下步骤：

-读入代表所述连贯正文的输入字符串；

-通过将所述输入字符串与一词典中的单词进行比较，识别所述输入字符串中至少一个独立单词序列；和

-输出至少一个所述已识别的单词序列；

其特征在于，所述识别至少一个单词序列的步骤包括利用以下步骤以迭代方式建立代表所述输入字符串中的单词序列的树形结构：

将所述输入字符串当作工作字符串；

对于词典中的每一个单词：

比较所述单词与所述工作字符串的开头；并且

如果所述单词与所述工作字符串的开头相匹配：

则在代表该单词的树中建立一个代表节点；

将所述节点和开始于与该单词的结束位置紧紧相邻的位置上的所述输入字符串的一部分相关联；和

通过将所述相关部分用作工作字符串来形成链接到该节点的子树，工作字符串所述子树代表与所述节点相关联的那部分输入字符串中的单词序列；

其中根据预定规则决定是否要将新单词增加到所述树结构中去；

如果将增加新单词：

在其相关单词之后将跟随新单词的树中选择至少一个节点；

形成多个新单词；每个新单词和与所述选择节点相关联的所述输入字符串部分的开头相匹配并利用不同数量的字符构成；

对于每个形成的新单词，形成链接到所述选择节点的相应子树；每一个子树代表以与选择节点相关联的输入字符串部分中的相应新单词开始的单词序列。

2. 根据权利要求 1 的方法，其特征在于，所述方法包括检查遍历树结构的至少一条路径是否代表匹配整个输入字符串的单词序列，其中所述单词序列只包括词典中的单词；并且所述方法还包括在结果不好时决定添加新单词。

3. 根据权利要求 2 的方法，其中在其相关单词之后将跟随新单词的树中选择至少一个节点的步骤包括识别代表不匹配整个输入字符

串的单词序列的遍历树的至少一条路径，并且将识别路径的端节点用作所述选择节点。

4. 根据权利要求 1 的方法，其中所述方法包括：在建立树形结构的同时，对于每个单词序列计算该序列中新单词的数量，并且在所计算的新单词数量超过一预定阈值时，终止沿代表该单词序列的路径延伸树形结构。

5. 根据权利要求 1 的方法，其中所述方法包括：在建立树形结构的同时，计算每个单词序列的似然性，并且在相应单词序列的似然性低于预定阈值时，终止沿代表该单词序列的路径延伸树形结构。

6. 根据权利要求 5 的方法，其中单词序列的所述似然性作为此单词序列中新单词数量的函数而减小。

7. 根据权利要求 1 的方法，其特征在于，所述形成新单词的步骤包括形成最大到 K 个单词的步骤，K>1，每个单词都从工作字符串的起始字符开始，并且各自包含工作字符串的 1 到 K 个起始字符。

8. 根据权利要求 1 的方法，其特征在于，输出至少一个利用树代表的单词序列的步骤包括选择遍历此树的路径之一，其中只考虑利用路径的末尾节点代表的单词匹配输入字符串的末尾的路径。

9. 根据权利要求 1 的方法，其中对于每个工作字符串，所述方法包括：

确定词典中的多少单词匹配工作字符串的开头；

如果词典中匹配工作字符串开头的单词的数量比预定阈值低，则决定添加新单词；并且

选择与所述工作字符串相关联的节点作为其相关联单词之后将跟随新单词的树中的节点。

10. 根据权利要求 9 的方法，其中所述阈值是 1。

11. 根据权利要求 9 的方法，其特征在于，选择遍历树的路径之一的步骤包括：根据统计 N 型语言模式计算每个候选路径的似然性并且选择一个最大可能的路径，其中 N>=2。

12. 一种将连贯正文分割成单词的系统，该系统包括：

-用于读入代表所述连贯正文的输入字符串的装置；

-用于通过比较输入字符串与词典中的单词来识别输入字符串中至少一个独立单词序列的装置；和

-用于输出至少一个识别单词序列的装置；

其特征在于，

用于识别至少一个单词序列的所述装置用于通过以下步骤以迭代方式建立代表所述输入字符串中的单词序列的树形结构：

将输入字符串当作工作字符串；

对于词典中的每一个单词：

比较所述单词与所述工作字符串的开头；并且

如果所述单词与所述工作字符串的开头相匹配：

则在代表该单词的树中建立节点；

将以与该单词的结尾位置紧紧相邻的位置开始的输入字符串的一部分和所述节点相关联；并且

通过将相关部分用作工作字符串来形成链接到该节点的子树，该子树代表与所述节点相关联的那部分输入字符串中的单词序列；

其中根据预定规则决定是否要将新单词增加到所述树结构中去；

如果将增加新单词：

在其相关单词之后将跟随新单词的树中选择至少一个节点；

形成多个新单词；每个新单词和与所述选择节点相关联的所述输入字符串部分的开头部分相匹配并利用不同数量的字符构成；

对于每个形成的新单词，形成链接到所述选择节点的相应子树；每一个子树代表从与选择节点相关联的输入字符串部分中的相应新单词开始的单词序列。

正文的自动分割

技术领域

本发明涉及一种可将连贯的正文分割成单词的方法，该方法包括如下步骤：读入一个代表连贯正文的输入字符串；通过将输入字符串与一词典中的单词进行比较，识别所述输入字符串中孤立单词的至少一个序列；并输出至少一个所述已识别的单词序列。

本发明还涉及一种可将连贯的正文分割成单词的系统，该系统包括：读入一个代表连贯正文的输入字符串的装置；通过将输入字符串与一词典中的单词进行比较，识别所述输入字符串中孤立单词的至少一个序列的装置；输出至少一个所述已识别的单词序列的装置。

日益高级的自然语言处理技术被用于数据处理系统中，例如语音处理系统、手写/光学字符识别系统、自动翻译系统或在单词处理系统中用于拼写/语法校验。这样的系统经常使用涉及单个单词或单词序列的统计信息。统计信息是通过分析大型正文全集来获得的。为了分析，单个单词需要在正文中被识别。在许多语言中，包括西方语言，单词是通过使用分界标志来分割的，例如一个空格或其他标点符号，使得识别工作变得容易。可是，许多其他的语言在单词与单词之间没有分界标志。此类语言的例子是许多亚洲语言，例如中文、日文、以及韩国语（Hangul）。这些语言有时称为胶合性语言。代表性地，这些语言是使用特殊的字符（象形文字）书写的，每一个这种字符都代表了单个或多个音节并且通常还代表一个概念或有意义的单元。一个词由一个或多个这种字符组成。读者在阅读此类语言的正文时必须识别这些单词的分界以了解正文的意义。在许多应用中只需用识别一个序列的单词。

背景技术

从US 5,448,474中我们了解了一种从连贯正文中隔离出中文单词的方法和系统。在此系统中，执行一个查找词典处理，将正文中所有子字符串单元进行识别。对于正文中的每一个字符，都对词典中的每一个单词进行查看，是否该单词与从那个位置的正文开始相匹配。例如，对于正文“software”，在位置0（正文中的第一个字符）找到单词“so”，“soft”以及“software” 的一个匹配；在位置1找到单词

“of”和“oft”的匹配；在位置4找到“war”和“ware”的匹配；在位置5找到“a”和“are”的匹配；还有在位置6找到“re”的匹配。对于每一个匹配都在一张表中建立一个入口。所述入口包括匹配单词、该匹配在正文中起始的位置以及该单词的长度。如果在某一位置没有找到可匹配的单词，就在表中构造一个只包含此单个字符的入口。用这种方法所有可能的单词和无匹配的字符就都被添加到表中了。然后，根据规则，例如一个单词的开始必须邻接前面一个单词末尾且该单词的末尾必须邻接于下一个单词的开始，减少表中入口的数量。由于在这一过程中重叠词（不相邻单词）将被除去，所以正文的各部分就不能被已识别的单词覆盖。根据保持最长匹配重叠词的规则，执行一个独立的恢复处理用来纠正那些对于重叠词的不正确删除。最后，再把那些不与正文的头尾相邻接或不与另一个必留单词相邻的单词都除去。最终的结果将包含几个可能的单词序列。根据单词在常规正文中的出现频率的信息来选择其中的一个序列。例如，一个具有两个字符的中文单词的序列可在拥有由两个单字符单词所表示的两个字符序列之上被选择，因为双字符单词比单字符单词更普遍。

已知的分离程序是复杂的并且还需要一个恢复处理来纠正删除错误。

发明内容

本发明的目的是要提供更有效的方法和系统。

为了满足本发明的这个目的，本发明方法通过将所述输入字符串与一词典中的单词进行比较来识别所述输入字符串中至少一个独立单词序列，并且输出至少一个所述已识别的单词序列，其特征在于，识别至少一个单词序列的步骤包括利用以下步骤以迭代方式建立代表所述输入字符串中的单词序列的树形结构：

将所述输入字符串当作工作字符串；

对于词典中的每一个单词：

比较所述单词与所述工作字符串的开头；并且

如果所述单词与所述工作字符串的开头相匹配：

则在代表该单词的树中建立一个代表节点；

将所述节点和开始于与该单词的结束位置紧紧相邻的位置上的所述输入字符串的一部分相关联；和

通过将所述相关部分用作工作字符串来形成链接到该节点的子树，工作字符串所述子树代表与所述节点相关联的那部分输入字符串中的单词序列；

其中根据预定规则决定是否要将新单词增加到所述树结构中去；

如果将增加新单词：

在其相关单词之后将跟随新单词的树中选择至少一个节点；

形成多个新单词；每个新单词和与所述选择节点相关联的所述输入字符串部分的开头部分相匹配并利用不同数量的字符构成；

对于每个形成的新单词，形成链接到所述选择节点的相应子树；每一个子树代表从与选择节点相关联的输入字符串部分中的相应新单词开始的单词序列。

通过建立树形结构，自动地分析输入字符串将导致只有那些与前面的单词相邻的单词才可被识别。所有被识别的单词序列中，最后一个单词在输入字符串末尾处结束的情况在原则上是可能的。这样，那些不可能的单词（从前面的单词来看）就不被当作候选者了。这减少了要被处理的数据的数量。而且，也不需要复杂的删除单词和再引入重叠的程序。根据本发明对取样串“software”的分割产生一个有两个主分支的逻辑树形结构，一个分支由单个节点代表单词“software”，另一个分支由两个链接的节点分别代表单词“soft”和“ware”。从而只需要三个入口而不是先有技术系统中的10个。

根据本发明的一个实施例，如果满足预定规则，多个具有不同长度的新单词将被添加。通过把不同于单字符单词的未知单词序列添加到数据结构中，用一种简单的方法识别多字符新单词就变得可能了。这使得程序适合如日语等的语言，因为此类语言中许多单个字符并不代表一个单词。此外，它允许将多字符单词作为首选新单词进行识别，在这种情况下，单字符单词不必被添加到词典中。这样词典被单字符单词“污染”的情况就被避免了。词典中有许多单字符入口会降低正文正确地分割为单词的可能性。例如，如果词典中有单字符“t”，则正文“thisbook”可能被分割为单词序列“t”、“his”和“book”。

根据本发明的一个实施例，其规则是一个全局决策，基于是否可使用现存词典对一个单词序列进行识别。如果没有序列可被识别，则添加新单词。执行这个测试可先建立一个只使用现存词典中的已知单

词的树形结构，然后该树被建立来检测是否至少有一个路径代表匹配整个输入字符串的单词序列。只要遍历树的第一条路径达到输入字符串的末尾，在建立树形结构过程中设置参数（可达串尾），可使确认变得十分简单。

根据本发明的一个实施例，当路径相应的单词序列与整个输入字符串不匹配时，新单词被添加到路径的一个或多个端末节点。此类节点的定位可简单地通过跟随遍历树的路径和检验路径的端末节点是否与输入字符串的结尾相符合来完成。（即，单词匹配及在字符串中的位置。这个过程能以一种简单的方法核对，通过检验是否与端末节点相接的输入字符串部分是空的，表明匹配单词已沿整个字符串被找到了）。在优选的实施例中，是否添加新单词（如上描述）作为一个全球决策被接受。如果要添加新单词，则树形结构要被重建。在重建树的过程中，被加新词的节点处于树的这些位置，即在词典中没有词与输入字符串的剩余部分相匹配（整个串还没有被处理过）。

根据本发明的一个实施例，与工作字符串开始处相匹配的单词的数量可被计算出来。如果这个数字少于一个阈值，则要添加新单词。添加单词的数量取决于查找到的匹配单词的数量，如果查找到的匹配极少，那么更适宜添加较多新单词。这样，所希望的单词序列中替换数目可被建立起来。在一个实施例中，作为一个极端，这个阈值可以是一，如果现存词典中没有一个单词与工作字符串的开始处相匹配，则要添加新单词。在这种对于树中的每一个分支决定是否添加（更多）新单词的意义上来说，本发明的实施例更适宜用来取树中的局部决策。

在取局部决策的另一实施例中，已在路径中的新单词的数量被视为决定新单词是否需要被添加的量度。在一个“理想的”方案中，如果第一次在一个路径中需要一个新单词，而只有一个或少数新单词被添加，此时确实所有情况都是可能的（从读者的观点来看）。实际上许多候选词每个都有不同的长度，是需要被测试的。一些错误的候选单词可导致识别剩余的字符串中的单词时的未对准现象。如果没有特定的办法，这种未对准将导致添加更多的新单词（可能被其它新单词跟随等等）。例如，通过在路径中允许两个或三个新单词，避免了树的迅速扩展，而这种扩展是由错误的新单词所导致了许多序列造成的。

根据本发明的取局部决策的另一实施例中，计算出单词序列（和遍历树的相应路径）的似然性。如果似然性降到太低，则该路径不再延伸。这样，可能包含新单词的那些不切实际的分割将不再进一步考虑。较好地，阈值应动态的建立，以保证相关秩序。如果一个或多个序列已经（用计算的似然性）被识别，则只处理那些具有较高似然性的其它序列。优选地，一个新单词具有较低的似然性，这里，似然性可依赖于该新词的长度。这样，词序列的似然性随该序列中新单词的数量而降低。这种方式就可以避免由于错误的选择新单词（这导致该串的剩余部分也失对准且要求进一步的新单词）导致因很多新单词使树继续膨胀。

根据本发明的一个实施例，新单词长度被限制到 K 个字符，K>1。K 最好等于 5，保证大多数单词，特别是对于主要是短单词的亚洲语言，不必建立一棵非常大的树就可被识别。

根据本发明的一个实施例，如果树中路径尾端的最后一个单词与输入字符串的尾端对准，则该树中的路径只被认为是代表一个有效的分割。它允许通过只从树中的相关单词与输入字符串末端对准的那些端末节点（叶子）开始回溯而识别有效序列。

根据本发明的一个实施例，一个统计学的 N 形（N－gram）语言模式用来确定通过遍历树的最可能的路径。这样一个基础的决策将被接受来从几个可能的序列中选择最可能的序列。这个序列的单词作为分割正文被输出。特别地，如果该方法用来建立语音识别系统的词典（词汇和/或语言模式），最好使用现存默认的具有 N 形语言模式的词典。如果词汇量很大（如大于 10,000 个入口）最好是使用 2 形或 3 形语言。

为了满足本发明的目的，本发明的系统包括：

用于读入代表所述连贯正文的输入字符串的装置；

用于通过比较输入字符串与词典中的单词来识别输入字符串中至少一个独立单词序列的装置；和

用于输出至少一个识别单词序列的装置；

其特征在于，

用于识别至少一个单词序列的所述装置用于通过以下步骤以迭代方式建立代表所述输入字符串中的单词序列的树形结构：

将输入字符串当作工作字符串；

对于词典中的每一个单词：

比较所述单词与所述工作字符串的开头；并且

如果所述单词与所述工作字符串的开头相匹配：

则在代表该单词的树中建立节点；

将以与该单词的结尾位置紧紧相邻的位置开始的输入字符串的一部分和所述节点相关联；并且

通过将所述相关部分用作工作字符串来形成链接到该节点的子树，工作字符串所述子树代表与所述节点相关联的那部分输入字符串中的单词序列；

其中根据预定规则决定是否要将新单词增加到所述树结构中去；

如果将增加新单词：

在其相关单词之后将跟随新单词的树中选择至少一个节点；

形成多个新单词；每个新单词和与所述选择节点相关联的所述输入字符串部分的开头部分相匹配并利用不同数量的字符构成；

对于每个形成的新单词，形成链接到所述选择节点的相应子树；每一个子树代表从与选择节点相关联的输入字符串部分中的相应新单词开始的单词序列。

附图说明

参阅附图实施例，本发明的如上和其它方面是显而易见的。

图 1 示出本发明系统的方框图。

图 2 示出一语音识别系统的方框图。

图 3 示出用于模化词或子词的 Hidden Markov Model.

图 4 示出只采用已知单词和新词进行分割的二步方式流程图。

图 5 示出对已知单词基于树的分割的流程图。以及

图 6 示出对新单词的基于树的分割的流程图。

具体实施例

为了便于说明，对用拉丁字符表示的正文给出了许多将正文分割成单词的例子。实际上语言包括使用不同的字符标，如 Katakana(日假名)或 Hiragana(平假名)。

图 1 展示了本发明系统 100 的方块图。系统 100 包含一个输入装置 110 用来接收一代表连贯正文的输入字符串。该串可代表一个词组、一个句子或由若干句子组成的大型正文。在亚洲语言中，如日语或中

文，句子由分界符来分割。对于此类语言，大型正文更适宜在句子的基础上进行分割。为此，通过使用句分隔符识别句子，将大型正文首先分割成句子，然后根据发明方法对每个句子进行分割。代表性地，输入字符串将从正文文件中读入。如果需要，使用内置或外置转换器可将文件转换为普通格式。正文也可从一个硬拷贝文档中被检索，例如通过扫描文档并且使用 OCR（光学字符识别）技术来识别字符串。

该系统还包含识别装置 120 用于将输入字符串分割为一个或多个单词序列。代表性地，识别装置 120 是以软件方式在适宜的处理器上如 PC 或工作站处理器上被执行。识别装置 120 使用词典（词典）122，并可选择性地使用语言分割模块 124。我们假定词典 122 与语言模块 124 中的词汇是基于一种具体语言中的孤立的单词。此系统可支持不同语言的不同词汇。词汇的规模是随系统的规模和复杂度而变化的。输出装置 130 用来输出至少一个序列的已识别单词序列。在许多情况下，它更适于仅仅输出一个（或少量）单词序列。我们十分欣赏的是本发明的方法和系统也可用于应用，在其中，期望分析几个或所有可能的单词后选，例如用来产生一自动指标。

该方法和系统最好被用于图形识别，如大型词汇的连续语音识别或手写识别系统，这里，使用一词表来识别单词并且使用一语言模式来改进基本识别结果。因为用于图形识别的技术也可被方便地用于本发明的分割，首先，描述图形识别系统。图 2 示出一个连续语音识别系统 200，其包含一频普分析子系统 210 与和一单元匹配子系统 220[见 L. Rabiner 和 B-H. Juang，“语音识别基础”，Prentice Hall 1993，第 434 至 454 页]。在频普分析子系统 210 中，语音输入信号（SIS）被频普地和/或暂时地分解以便计算出有代表性的特征矢量（观测矢量，OV）。典型地，语音信号被数字化（例如以 6.67kHz 的速率取样）并且被预处理，例如通过应用预加重（applying pre-emphasis）。连续的取样被分组（分块）为帧，相应于如 32msec. 语音信号。相继的帧是部分重叠的，如 16msec.。经常采用 Linear Predictive Coding (LPC) 频普分析方法来计算每一帧的代表性特征矢量（观测矢量）。特征矢量可能，举例来说，有 24、32 或 63 个分量。在单元匹配子系统 220 中，观测矢量相对于语音识别单元列表被匹配。一个语音识别单元代表一个声音参考序列。可采用各种形式的语音识别单元。

举例来说，一个完整的单词或甚至是一组单词都可以一个语音识别单元表示。一个单词模式（WM）对特定词表中的每个单词提供在声音参考序列中的录音。对于某些系统，其中一个完整的单词由一个语音识别单元所表示，则在单词模式与语音识别单元之间存在着直接关联。其他系统，特别是大型词汇系统，可对语音识别单元采用语言上基于子单词的单元，如单音、双音或音节，以及派生出的单元，如 fenenes 和 fenones。对于这些系统，由词典 234 给出了一描述涉及词表中的单词的子单词单元序列的单词模式，以及一描述包括语音识别单元的声音参考序列的子单词模式 232。一个单词模式合成器 236 根据子单词模式 232 和词典 234 合成单词模式。图 3A 示出一系统的单词模式 300，该系统是基于整个单词的语音识别单元的，这里所示单词的语音识别单元采用十个声音参考序列（301 到 310）被模仿。图 3B 示出基于子单词单元系统的单词模式 320，这里示出的单词被三个子单词模式（350，360 和 370）的序列模仿，每个单词都有一个四声音参考序列（351，352，353，354；361 到 364；371 到 374）。图 3 中示出的单词序列是基于 Hidden Markov 模式的，其被广泛应用于随机模仿语音信号和手写信号。使用这种模式，每个识别单元（单词模式或子单词模式）都被 HMM 典型地表征，它的参数从训练用数据组进行评估。对于大型词汇语音识别系统包含，如 10，000 到 60，000 个单词，通常仅使用有限的一组子系统单元，如 40 个，因为它需要大量训练用数据来充分地为更大单元训练 HMM。HMM 状态对应于声音参考（用于语音识别）或代写参考（用于手写识别）。各种技术都可用于模仿参考，包括不连续的或连续的几率密度。

图 2 所示的单词水平匹配系统 230 相对于所有语音识别单元序列与观测矢量匹配并且提供矢量与序列之间的匹配似然性。如果子单词单元被使用，通过使用词典 234 在匹配上放置限制条件来限制可能的子单词单元序列到词典 234 中的序列。这把输出减小到可能的单词序列。句子水平匹配系统 240 的采用一语言模式（LM）把更多的限制条件加到匹配，这样被查实的路径是相应于语言模式中规定的那些合适的单词序列的路径。这样，单元匹配子系统 220 的结果就是一个已识别的句子(RS)。在图形识别中的语言模式可包括语言的依照句法的和/或语义的限制条件 242 以及识别任务。基于句法限制条件的语言模式

通常是指语法 244。

相似的系统还用于识别手写。用于手写识别系统的语言模式除确定单词序列外或做为替代方案还可确定字符序列。

语言模式所用的语法 244 提供了单词序列 W=w1w2w3...wq 的概率，其在原则上由

$$P(W)=P(w1)P(w2|w1).P(w3|w1w2)...P(wq|w1w2w3...wq-1)$$

给出。由于实际上在给定语言中可靠地对所有单词和所有序列长度估算条件单词概率是不实际的，所以广泛地采用了 N 形单词模式。在 N 形模式中，项 P(wj|w1w2w3...wj-1)被 P(wj|wj-N+1...wj-1)逼近。实际上,使用二形语言(bigrams)或三形语言(trigrams).在 trigrams 中，项 P(wj|w1w2w3...wj-1)被 P(wj|wj-2wj-1)逼近。自动建立 N 型语言模式的方法就是通过一个简单的相对频率：F(wj-N+1...wj-1wj)/F(wj-N+1...wj-1)来评估条件概率 P(wj|wj-N+1...wj-1)，在其中，F 是在给定正文训练全集中以自身为变量时串出现的次数。为使评估更可靠，F(wj-N+1...wj-1wj)在已知全集中必须是具体的。

对于图形识别，我们希望词典和辞典都以代表要被识别的正文的单词为基础。这可通过分析典型正文，从正文中抽取单词以及建立基于单词或单词序列频率的语言模式来实现。本发明的分割方法可方便地用于从连贯正文中抽取单词。为了训练图形识别系统的辞典或词典，进行分割导致只有一个输出单词序列中就足够了。如果分割系统 100 在一图形识别系统中使用，最好还包含用于将输出单词序列中的新单词（也就是还不在词典 122 中的单词）合并到词典 122 中的更新装置 140。语言模式最好也被更新，例如，以反映新单词或单词序列似然性，包括新单词与已知单词或单词序列的似然性。

根据本发明，完成分割要通过建立代表输入字符串中的独立单词序列的树形结构。如果词典已经包含正文中所有要被分割的单词，在原则上就不必添加新单词了。因而，一个单词序列可能包括也可能不包含一个或多个新单词。由于添加新单词会使分割变得更困难和精细，所以最好先决定该正文是否可以只使用已知单词进行分割。整个过程由图 4 示出。在步骤 410，输入字符串在正文全集 420 中被检索。如前描述，字符串可表示一个词组、一个句子或多个句子正文。在步骤 430，核实整个全集是否已被分割。如果是（字符串空），过程在步

骤 440 退出。否则，在步骤 450 中使用已知词典（只有已知单词）对正文进行分割。这最好通过建立树形结构来实现，该树的每个节点代表一个已知单词。不可完成的（没有已知单词与串的剩余部分匹配）遍历树的路径（表示单词序列）将被终止。步骤 450 的更多的细节将参阅图 5 在下面进一步阐明。在步骤 460 中检查正文是否可以只被已知单词分割。这可通过检验是否至少有一个路径遍历该建立的树是完整的（也就是符合路径的端末节点的单词匹配于字符串的端尾字符并且被放置于字符串的末尾）来测试。为此目的，遍历树的路径可被跟随直至符合整个字符串的路径被找到。更适宜地，在建树过程中，当达到字符串的末尾时，此事实将被存储如作为一个“已达串端末的”参数。这样，检验路径是否完整就仅仅涉及检查存储的信息了。如果步骤 460 的检测表明正文可被分割，则已识别的单词序列在步骤 470 中被输出并且该过程在步骤 410 继续。如果不可分割，则过程在步骤 480 继续且重新分割字符串，但此次允许添加新单词。步骤 480 的更多的细节将参考图 6 在下面阐明。我们将知道，在步骤 460 中识别的字符串与用已知单词可分割一样实际上也可用新单词分割。正文的读者也许更优先采用包括新单词的分割，即具有新单词的单词序列出现的可能性比具有已知单词的已识别序列更大。这种情况将极少发生。然而，为了处理这种情况而不经常采用可能的新单词分割字符串，最好是作为步骤 460 的部分（选择性地）一个已知单词序列的似然性被确定下来（举例来说使用 N 型语言模式），如果似然性大于给定阈值，则正文按已知单词分割来识别，否则，启动采用新单词的分割。

图 5 示出了只采用已知单词分割字符串的流程图。根据本发明，建立一个树形结构。原则上，任何建立和表示树形结构的合适的技术都可被应用。在图 5 的例子中，用元素列表（表示树节点）以及元素之间的连接（指针）（表示节点间的路径）来表示树。在本例中，使用了两个列表。一个端末列表包含相应于匹配于并且对准输入字符串的结尾的单词的那些元素。一个等候列表包含相应于与字符串匹配但不与输入字符串的端末对准的单词的那些元素。每个元素与词典中的一个现存单词相关。这种相关可以任何合适的方式完成（例如，复制单词并且储存于元素的数据结构中或把参考（指针或数字）储入词典入口）。此外，每个元素是与某种输入字符串部分相关的，该种输入字符

串跟随着与该元素相关联的单词。根元素是特殊的元素，其与整个输入字符串相关联，但不与任何单词相关联。根元素的作用是联合所有可能的单词序列。实际上，没有必要设置一个分离的根元素。替之，对于词典中的每一个匹配于输入字符串开头的单词可建立一个新元素。如此建立的元素是作为单词序列的首元素。步骤 510 与 511 包含过程环的初始化。在步骤 510，建立根元素，并且输入字符串与根元素相关联。在步骤 511，根元素被放入等候列表。在步骤 512，等候列表中的一个元素被选中作为激活元素（从而，该根元素就选作初始激活元素）。在步骤 512，工作字符串被装入。与等候列表中的当前激活元素相关联的串用作工作字符串。所以，由于最初输入字符串与根元素相关联，该根元素最初是等候列表的激活元素，则最初整个输入字符串作为工作字符串使用。在循环中，在步骤 514 和 516，词典中的所有单词都成功地从词典中被检索出。正文单词的检索是发生于步骤 514。在步骤 516，将检测是否仍有单词被检索（并非所有单词都已检测）。如果是，在步骤 518，检验单词是否匹配于输入字符串的开头。如果不是，在步骤 514 中检索正文单词。如果发现了一个匹配，在步骤 520 创建一个新元素。该元素耦合于单词（举例来说，单词与元素一起相关存储），耦合于剩余的工作字符串（将匹配单词从工作字符串的开头移去之后）并与父元素连接（也就是该元素与输入字符串中前面的单词关联）。对于匹配于输入字符串开头的单词，因为元素与开始单词关联，根元素就作为父元素了。在步骤 522，检查输入字符串的端末尾是否已到达（也就是，剩余工作字符串是否为空）。如果是，遍历树的路径就结束了，则一个单词序列被建立起来。为保证此序列可很容易地被检索，在步骤 524 中该元素被存储于端末列表中。对于只识别一个单词序列就足够充分（不需要最大的可能性）的系统，一旦达到串的结尾，该程序就退出。如果串的结尾还未被检索，则元素就被存储于等候列表中（步骤 526）。剩余串稍后将被分割。这两种情形单词都被处理了（与工作字符串的开始相比较），并且在步骤 514 中下一个单词也被检索了。如果对于一个工作字符串，词典中的所有单词都已经与串的开头相比较了，循环在步骤 516 退出。在步骤 528，等候列表中的当前选定元素从等候列表中被除去，因为该元素被完全地处置了。在一个封闭循环中，还未完全被处理的所有的工作字符串都被处置了。

每个此类串都由等候列表中的一个元素表示。因此，在步骤 530 中检查等候列表是否为空。如果不空，在步骤 512，等候列表的下一个单词作为当前激活元素被选定。如果等候列表为空，原始输入字符串就被完全地分割（尽可能只采用已知单词）。在步骤 534，检查端末列表是否包含任何入口。如果不，在步骤 536 中将返回只用已知单词分割不成功的消息。如果端末列表不空，其上的每一个元素代表一个单词序列。实际上，这些元素是与单词序列的最后一个单词相关联的并且与该序列中前面的单词连接。这使得在步骤 540 回溯从端末列表上的一个元素开始的连接的元素的方法来检索单词序列。若不返回所有已识别单词序列，选择性地，在步骤 538 选择一个或多个单词序列并在步骤 542 返回。选择最好是基于路径的似然性。为了这个目的，使用一个统计 N 型语言模式来确定最相似的单词序列是有利的。特别是如果分割改善了词典和/或图形识别系统的语言模式，就可使用现存语言模式。Bigram 或 trigram 语言模式最好用于大型词汇图形识别系统（举例来说超过 10，000 个入口）。

根据本发明，如果满足一预定规则，长度不同的多个新单词会被添加到树形结构中。在一个实施例中，新单词被添加到一个或多个路径的端末节点，这些路径的相应单词序列不匹配于整个输入字符串。在图 5 中阐明的方法可用于建立一基本树形结构。举例来说，如果后来表现出使用已知单词不可建立合适的分割，则需要添加新单词的节点可简单地通过跟随遍历树的路径和检验路径端末节点是否相应于输入字符串尾部（即单词匹配并且分割已达输入字符串尾部）而找到。采用图 5 的技术，元素之间可保持双链接，一个链路将子节点链接到父节点（如前），另一链路将父节点链接到子节点。这样路径从根开始可遍历棵树而被跟随。对于路径的端末节点，则检查路径的端末节点是否在端末列表上。如果不在，新单词将被添加到端末节点。做为替代，不循迹路径遍历树，这里可引入第三个列表，当相应于端末节点的串是非空时其表示路径的端末节点（即没有已知单词匹配于输入字符串剩余部分的开头）。这一点可通过在步骤 528 中检验是否至少可找到一个匹配来实现。如果没有，元素将从等候列表中放置到第三个列表上表示不完全分割。一旦节点被定位，新单词可被创建并作为树中元素被表示，正如将在更多的细节与图 6 的参考被描述那样。通过将

元素放置于等候列表上，可用图 5 中描述的用于已知单词分割的同样的方法建立树的剩余部分。

图 6 示出了一分割输入字符串的优选方法。在此实施例中，已知单词的识别和新单词的添加都发生在一个完整的方法中。相应于图 5 的相同项目用图 5 中相同的数字表示。这些项目不再详细描述。如果在步骤 516 后所有单词相对于工作字符串的开头已匹配，在步骤 610 就使用一个预定规则来决定是否需要添加新单词。如果需要新单词新单词，在步骤 612，就初始化此新单词的长度（在本例中到 1）。在步骤 614，通过从串开头复制字符数，建立该长度的单词。以在步骤 520、522、524 和 526 中描述的同样的方式，在步骤 616 建立相应的元素并与单词、父节点和该串的剩余部分相关。元素被置于等候列表（步骤 622）还是端末列表（步骤 620）依赖于串的结尾是否已到达。在步骤 624，检查是否直到最大长度 K 的全部所须单词都被建立了，K 至少是 2。如果没有，在步骤 626，增加长度并在步骤 614 建立新单词。如果所有的新单词都已被建立，如图 5 描述该程序从步骤 528 继续。对于亚洲语言，K 最好选 3 至 6 之间。如果这不能导致成功的分割，K 可增加。

根据本发明的一个实施例中，步骤 610 的检测规则是有多少匹配于工作字符串开头的单词。这一点的实现可通过在步骤 520 增加一计数器，它的复位作为步骤 512 或 513 的一部分。如果匹配单词数少于一个阈值，则添加新单词。添加多少新单词取决于找到的匹配单词数，如果只找到很少的匹配则最好添加更多的新单词。我们十分欣赏的是作为一个极端阈值这个极限可以是 1，导致如果现存词典中没有一个单词匹配于与那个节点的工作字符串的开头，则新单词附属于该树的节点。

在另一实施例中，规则是基于已在一路径中的新单词的数量。这一点可通过每次在一路径中插入一新单词就步进一个计数，且将该计数与路径中端末元素相关来实现。如果一路径已包括两个或三个新单词了，最好是不再添加更多的新单词到该路径中。然而，如果这样也不能实现成功的分割，则可增加路径中允许的新单词数。

在另一实施例中，规则是基于单词序列的似然性（以及遍历树的相应路径）。更适宜地，在建立相应路径时计算每个单词序列的似然

性。累积分数可与路径的端末元素相关存储。作为步骤 610 的部分测试，如果似然性低于一个阈值，路径将不再延伸：也不再有新单词添加到该路径。方便地，似然性阈值动态地建立来确保一个相对的队列。如果已经有一个或多个序列被识别（用已算出的可能性），则其他序列只在序列有更高或相似的似然性时才被处理。更适宜地，新单词的似然性相对较低，在此，似然性取决于新单词的长度。这样，单词序列的似然性将随着序列中的新单词数而减少。可以采用任何适当的似然性计算。最好采用下面的新单词似然性记分：

Unknown_word_score=penalty+weight*[min_unigram*char_no*unigram_weight+Length_prob*length_weight]，

这里

-penalty 是对每个单词的固定的补偿值，

-weight 是每个新单词分数的全局加权因子，

-min_unigram 是所有已知单词的最小发生频率（unigram 模式），

-char_no 是新单词的字符数，

-unigram_weight 是 unigram 得分的局部加权因子，

-length_prob 是有这个长度的单词的概率（长度分布概率），

-length_weight 是该长度概率的局部加权因子。

补偿和加权参数确保新单词比已知单词得到更低的分数。

在根据本发明的另一实施例，如果在步骤 610 确定了没有单词匹配于工作字符串的开头，则这就被看作某种指示，即在较早的位置可能已引发错误的分割。例如，存在一个已知单词匹配，然而实际上字符是新单词的一部分。为了这个目的，树被回溯，最好是只一步，并且将一个或多个新单词添加到在回溯中被定位的节点。显然如果这些单词已被添加了，则没有新单词被添加。如果在那个位置已找到几个已知单词的匹配那么也没有必要添加新单词。在后一种情形中，假定这些单词中至少有一个单词将引导致成功的序列。

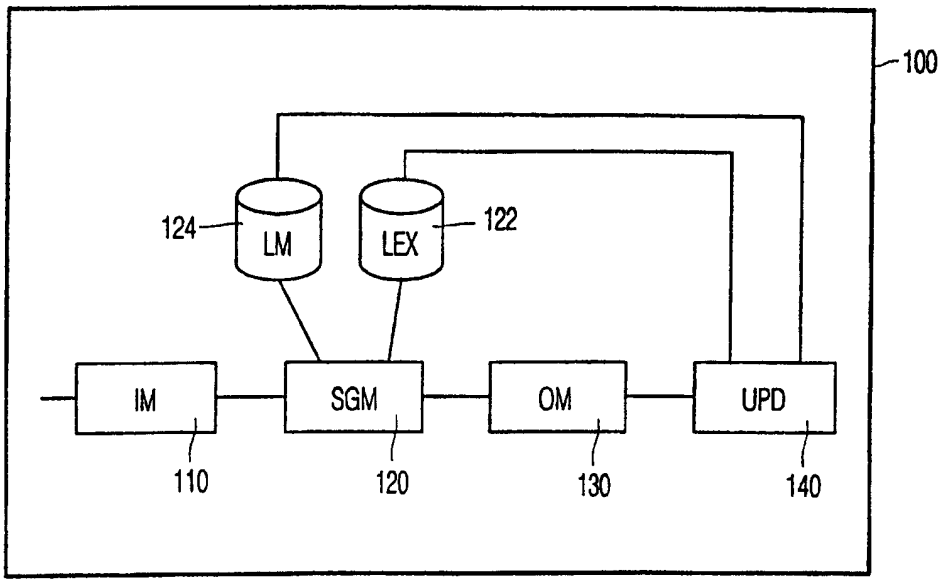

100
124
LM
LEX
122
IM
SGM
OM
UPD
110
120
130
140

图 1

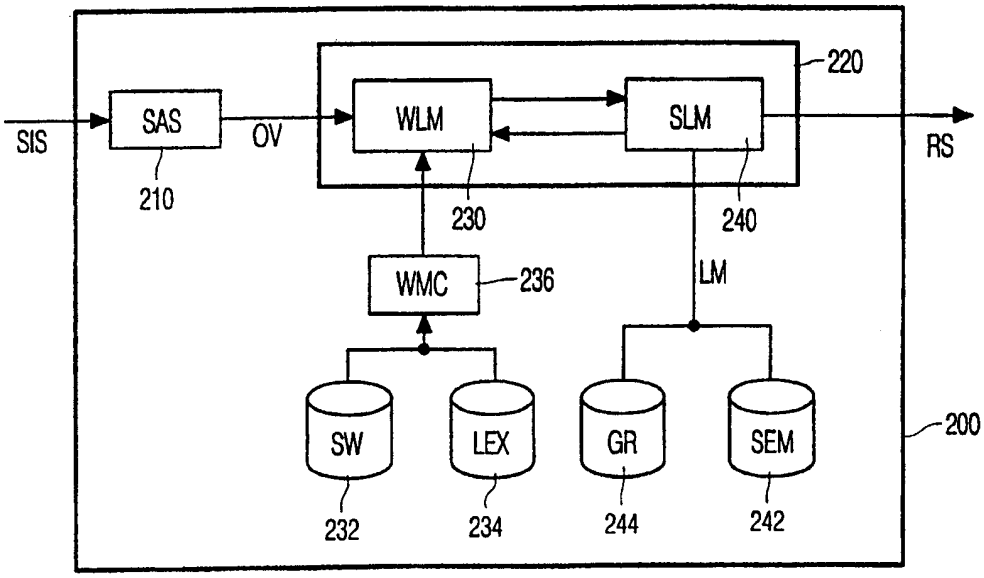

220
SIS
SAS
OV
WLM
SLM
RS
210
230
240
WMC
236
LM
SW
LEX
GR
SEM
200
232
234
244
242

图 2

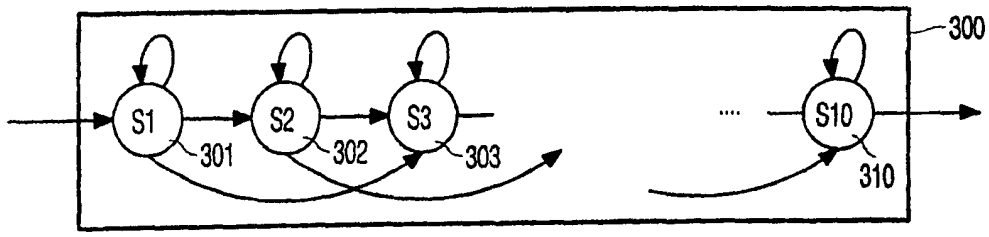

300
S1
S2
S3
S10
301
302
303
310

图 3a

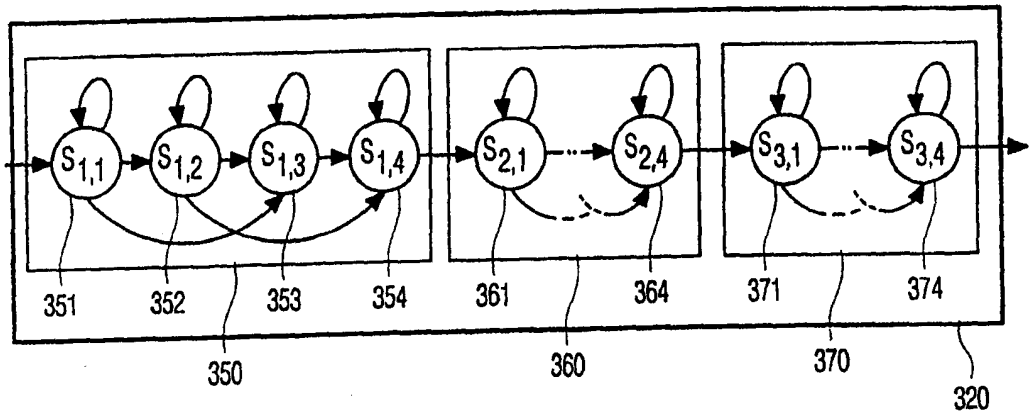

S1,1
S1,2
S1,3
S1,4
S2,1
S2,4
S3,1
S3,4
351
352
353
354
361
364
371
374
350
360
370
320

图 3b

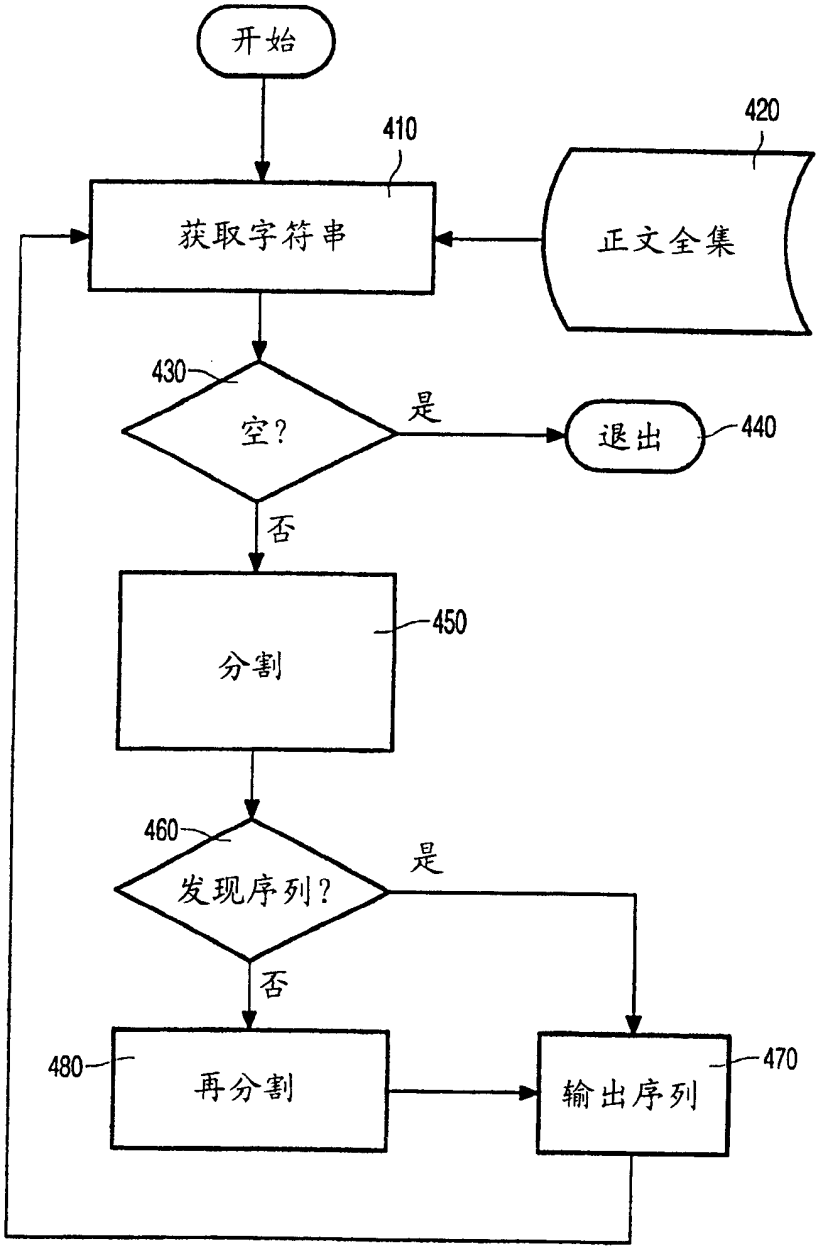

开始
410
420
获取字符串
正文全集
430
空?
是
退出
440
否
分割
450
460
发现序列?
是
否
480
再分割
输出序列
470

图 4

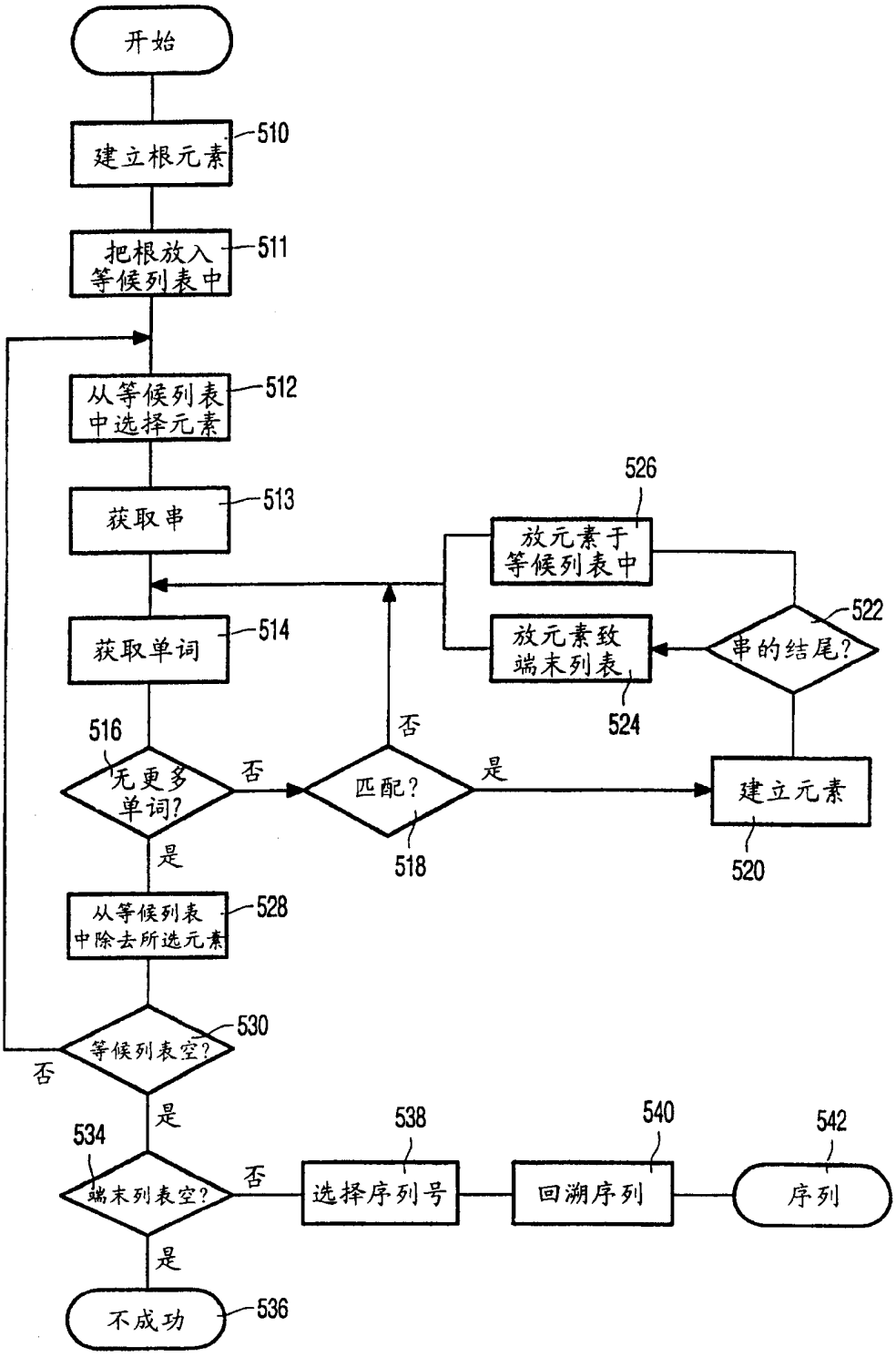

开始
建立根元素
510
把根放入等候列表中
511
从等候列表中选择元素
512
获取串
513
获取单词
514
516
无更多单词?
否
是
匹配?
否
是
518
526
放元素于等候列表中
放元素致端末列表
524
串的结尾?
522
建立元素
520
从等候列表中除去所选元素
528
等候列表空?
530
否
是
534
端末列表空?
否
是
538
选择序列号
540
回溯序列
542
序列
不成功
536

图 5

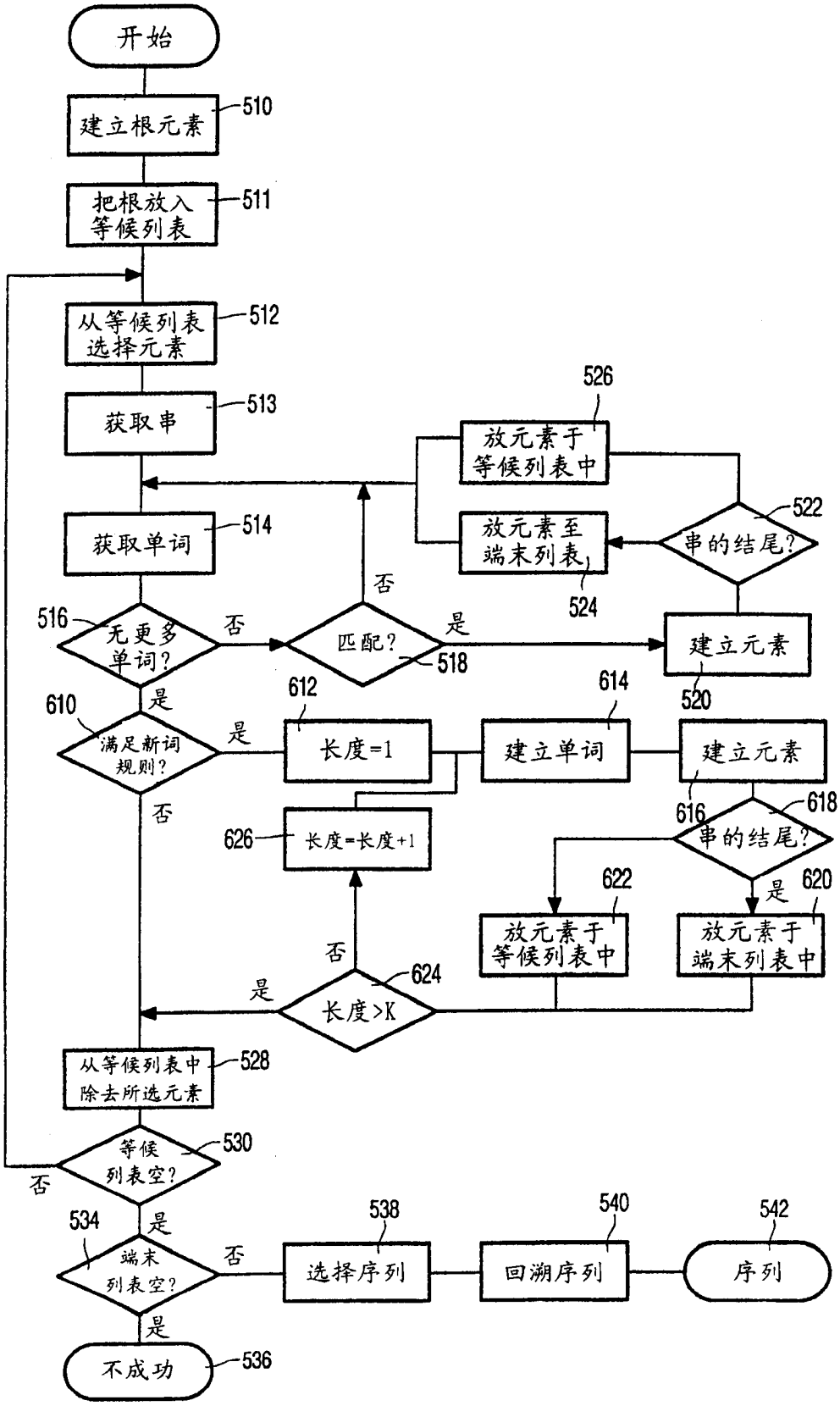

开始
建立根元素
510
把根放入等候列表
511
从等候列表选择元素
512
获取串
513
获取单词
514
无更多单词?
516
否
是
匹配?
518
否
是
满足新词规则?
610
是
否
长度=1
612
建立单词
614
建立元素
520
串的结尾?
522
放元素至端末列表
524
放元素于等候列表中
526
长度=长度+1
626
串的结尾?
616
618
是
放元素于等候列表中
622
放元素于端末列表中
620
否
是
长度>K
624
从等候列表中除去所选元素
528
等候列表空?
530
否
是
端末列表空?
534
否
是
选择序列
538
回溯序列
540
序列
542
不成功
536

图 6