[19] 中华人民共和国国家知识产权局

[51] Int. Cl.
G06F 17/27 (2006.01)
G06F 17/28 (2006.01)

[12] 发明专利申请公开说明书

[21] 申请号 03826551.6

[43] 公开日 2006 年 5 月 10 日

[11] 公开号 CN 1771494A

[22] 申请日 2003.5.28 [21] 申请号 03826551.6
[86] 国际申请 PCT/EP2003/005627 2003.5.28
[87] 国际公布 WO2004/107202 英 2004.12.9
[85] 进入国家阶段日期 2005.11.28
[71] 申请人 洛昆多股份公司
地址 意大利托里诺
[72] 发明人 莱昂纳多·巴迪诺
[74] 专利代理机构 中国国际贸易促进委员会专利商标事务所
代理人 李春晖
权利要求书 5 页 说明书 12 页 附图 4 页

[54] 发明名称

包括无分隔符的块的文本的自动分块

[57] 摘要

包括无分隔符书写的备个元素的文本的语段被分块为块，块由包括各个元素中的至少一个的串组成，元素例如是普通话汉语的表意符号。 定义包括一组串的词典(LEX，SLEX)，每个串由各个元素中的至少一个组成。 通过在词典中搜索对应于块中的任何一个的串，从而逐元素(INDX)搜索正被分块的语段。 如果得到肯定搜索结果，则一起存储所定位出的对应的块和关联的成本。 检查所定位出的块是否已出现在词典中。 如果所定位的块已出现，则降低关联到其的成本。 从而生成多个候选分块序列，每个对应于各自的分块模式，并且具有关联的对应累积成本。 具有最低关联累积成本的候选序列被选择为最终分块结果。

1. 一种用于将包括无分隔符书写的各个元素的文本的语段分块为块的方法，所述块由包括所述各个元素中的至少一个的串组成，所述方法包括下述步骤：

定义词典（LEX，SLEX），所述词典包括一组串，每个串由所述各个元素中的至少一个组成，其中所述词典中的串至少部分代表所述块，

通过在所述词典中搜索对应于所述块中的任何一个的串，从而逐元素（INDX）有序地搜索正被分块的语段，其中，如果得到肯定搜索结果（312），则一起存储（C）所定位的对应的块和关联的成本（CM），

检查所定位的块是否已出现在词典中（SLEX），如果所定位的块已出现，则降低与其相关联的成本，

将多个候选分块序列存储为所述有序搜索的结果，每个候选分块序列对应于各自的分块模式，并且具有关联的对应的累积成本，以及

选择具有最低关联累积成本的候选序列作为最终分块结果。

2. 如权利要求 1 所述的方法，其特征在于，在出现两个具有相同关联成本的候选序列时，该方法包括选择从下述组中选择出的候选序列作为分块结果的步骤：

具有较长第一块的序列，和

具有较低长度变动的序列。

3. 如权利要求 1 所述的方法，其中所述文本中的至少一个语段先前已被分块，其特征在于该方法包括确定下述至少一个的步骤：

在当前语段中被定位的、已在所述至少一个先前已分块的语段中出现的块的数目（NOL），和

在分块过程期间已发现的块的计数（NW）。

4. 如权利要求 3 所述的方法，其特征在于基于成本函数选出具有最小关联成本的所述序列，所述成本函数包括所述块数目（NOL）和所述计数（NW）中的至少一个。

5. 如权利要求 3 所述的方法，其特征在于基于成本函数选出具有最小关联成本的所述序列，所述成本函数包括所述块数目（NOL）和所述计数（NW）的比例。

6. 如权利要求 1 所述的方法，其特征在于该方法包括在所述逐元素搜索中的每个新步进（INDX）将所述关联成本（CM）增加常量值的步骤。

7. 如权利要求 6 所述的方法，其特征在于该方法包括在增加所述关联成本（CM）后消除成本比给定阈值（CM）高的那些块的步骤。

8. 如权利要求 1 所述的方法，其特征在于该方法如果得到肯定搜索结果（312），则包括通过去除搜索到的串的一个末端元素来缩短该串然后在所述缩短后的串基础上重复搜索的步骤。

9. 如权利要求 8 所述的方法，其特征在于该方法包括通过去除所述串的最右元素从而缩短所述串的步骤。

10. 如权利要求 1 所述的方法，其特征在于该方法包括将所述词典的至少一部分作为动态词典（SLEX）管理的步骤，所述管理包括下述步骤：

如果所定位出的块已出现在所述动态词典（SLEX）中，则将与其相关联的成本降低常量值（DC），

如果所定位的块先前未出现在动态词典中，则检查（440）所述动态词典是否为满，并且

i） 如果所述动态词典未满，则将所定位的块和各自的降低了常量值（DCI）的成本（CM，CF）一起存储到动态词典中，

ii） 如果动态词典为满，则搜索所存储的具有比给定的成本阈值高的关联成本的任何块，并且如果发现这种块，则用新块替换该块（450）。

11. 如权利要求 1 所述的方法，其特征在于该方法还包括以下步骤：

至少将所述词典（LEX）中的所述串的集合的一部分定义为代表对应于所定义的规则的特定块，

通过在所述词典中搜索下述至少一个来逐元素（INDX）搜索正被分块的语段:

（A） 对应于任何所述特定块的最长串，其中，如果得到肯定搜索结果（312），则所定位的对应的块与关联的第一成本（CF）一起被存储（C），

（B） 对应于所述词典中的任何其他串的最长串，其中，如果得到肯定搜索结果（324），则所定位出的对应的块与关联的第二成本（CM）一起被存储（C），所述第二成本（CM）比所述第一成本（CF）高。。

其中，如果（A）和（B）情况下的搜索都未得到肯定结果，则用作搜索开始元素的单个元素与关联的第三成本（CS）一起被存储（C），所述第三成本（CS）比所述第二成本（CM）高。

12. 如权利要求 11 所述的方法，其特征在于该方法包括在所述逐元素搜索(A，B)中的至少一个中的每个新步进（INDX），将所述第一成本（CF）、第二成本（CM）和第三成本（CS）增加常量值的步骤。

13. 如权利要求 12 所述的方法，其特征在于该方法包括当所述成本（CF，CM，CS）被增加时，消除成本比给定阈值（CM）高的那些块的步骤。

14. 如权利要求 13 所述的方法， 其特征在于所述给定阈值被选择为等于所述第二成本（CM）。

15. 如权利要求 11 所述的方法，其中所述文本中的至少一个语段先前已被分块，其特征在于该方法包括确定下述步骤:

确定在当前语段中定位的、已在所述至少一个先前已分块的语段中出现的块的数目（NOL）和在分块过程期间已发现的块的计数（NW），

基于成本函数选出具有最小关联成本的所述序列，所述成本函数定义如下:

i）如果所定位的块先前未包括在所述词典中，则

$CostW_{i,j} = CSLEX$

ii）否则

$CostW_{i,j} = CSLEX + （Cfs-CSLEX）*（1-NOL/NW）/K$

其中，Cfs 等于所述第二成本（CM）或所述第一成本（CF），这取决于所考虑的词是通过所述第二搜索（B）还是所述第一搜索（A）定位的，K 为常量值，CSLEX 为与所述词典中的块 $W_{i,j}$ 相关联的成本，并且 NOL 和 NW 分别是所述数目和所述计数。

16. 如权利要求 1 所述的方法，其特征在于该方法还包括使用下述编码技术中的至少一个将所述各个元素编码为位串的步骤：ISO 标准，Unicode，GB 或 BIG5 编码技术。

17. 如权利要求 1 所述的方法，其特征在于所述各个元素对应于表意符号。

18. 如权利要求 17 所述的方法，其特征在于所述表意符号是普通话汉语表意符号。

19. 如权利要求 18 所述的方法，其特征在于该方法包括在所述语段被分块前将所述表意符号音译为拼音语音音译。

20. 如权利要求 11 所述的方法，其特征在于所述特定块从包括日期、小时和数字的组中选出。

21. 一种用于将包括无分隔符书写的各个元素的文本的语段分块为块的分块器（10），所述块由包括所述各个元素中的至少一个的串组成，所述分块器包括配置为执行权利要求 1 到 20 中的任何一个的方法的数据处理结构（10；A、B、C、RET）。

22. 一种文本到语言合成系统（20），包括：

文本源（30），用于生成至少一个要被分块为块的文本语段，所述语段包括无分隔符书写的各个元素，所述块由包括所述各个元素中的至少一个的串组成，

分块器（10），用于接收所述至少一个文本语段，所述分块器包括数据处理结构（10；A、B、C、RET），该数据处理结构配置为执行权利要求 1 到 20 中的任何一个的方法，从而作为最终分块结果生成

所述具有最低关联成本的候选序列，以及

语音信号发生器（40，50），用于将分块得到的所述序列转换为对应的音频语音信号。

23. 一种计算机程序产品，可加载到计算机存储器中，并且包括软件代码部分，所述软件代码部分用于执行权利要求 1 到 17 中的任何一个的方法的步骤。

包括无分隔符的块的文本的自动分块

技术领域

本发明涉及对多种语言的文本进行分块，这种文本包括所撰写的没有诸如空格、连字符号等分隔符的块（chunk）。这种语言的示例如普通话汉语拼音，在这种语言中，块一般由表意符号代表。

对于语音合成领域的技术人员公知的是，语音元素的“分块”通常对应于一个词。除了普通话汉语，还存在其他语言，但是，单个词可能实际上包括数个块：这种语言的典型示例是德语，在德语中，存在例如“Patentubereinkommen”这样的复杂的词，甚至包括两个不同的块，即“Patent”和“Ubereinkommen”写在单个词，而没有分隔符。

但是，将参考普通话汉语描述本说明书的其余部分（但是不应当将此解释为限制本发明的应用范围），因为本发明可以最有效地应用于该语言。

背景技术

汉语的书面形式是希望学习该语言的外行的一个基本难题。实际上，汉语“字符”集包括大约 45000 个表意符号（汉语中的“汉字”）。许多这种表意符号是涉及不再存在的对象并且因此已变得实际无用的词（由单个字符组成的词）。当前的估计是要能够阅读汉语报纸约 4000 个表意符号就足够了。

不管是 4000 或是 40,000 个表意符号，在任何情形中这一数量级都远大过印欧语系的字符集。

因此，在开发汉语的文本到语音合成的系统时就遇到了基本困难。实际上，根据 ISO 标准，利用包括 8 位（即，一个字节）的二进制数字编码印欧语系的单个字符一般就可行了。相反，对于汉字编码每个单个表意符号要求至少两个字节。

ISO 标准未提供这种编码，但是存在可以解决该问题的替换编码技术，例如，利用已知的 Unicode、GB 和 BIG5 编码技术。

借助“拼音”可以在一定程度上减轻编码问题。拼音是一种基于拉丁字母的标音/音译形式，示出汉字如何发音。在教导汉语基础的课本和汉语词典中提供了拼音标音，同样许多讲汉语的人也认识拼音标音。

汉语普通话的另一个基本特征是在没有分隔符的情况下书写表意符号（即，组成语言的块）。因此，标识语句中的每个单个的词很不容易，因为每个词可能实际上由一个或多个汉字组成。

有人可能错误地相信通过仅仅每次翻译一个字符而不考虑某个词的结束和新的词的开始就可以轻易地绕过这个问题。

实际上，为了实现可接受的语音合成质量，有必要将文本分解成单个的词（即使表意符号以拼音形式转写）。

这种需要是由于多个因素，

- 取决于单个表意符号所属的词，每个单个的表意符号可能具有不同形式的发音；

- 某些音位和语音规则取决于正确的分词：例如，所谓的音调连音音位（tonal syllables phonologic）规则提供了这样的规则，在出现每个都表达第三音调的两个音节时，如果这两个音节属于同一个词，则前一个音节将改变其音调；以及

- 涉及每个词的信息是必需的，以便准许进行正确的语法和韵律句法（syntactic－prosodic）分析。

总而言之，将文本分块为块的高效安排是真正满足汉语普通话文本到语言合成的基本要求。

将普通话汉语文本分块为块的已知解决方案本质上可以划分为三类，即：

- 纯统计算法，例如利用所谓的分类与回归树（CART）的那些实现，

- 基于词汇规则的算法，以及

- 组合前述两种解决方案的算法。

第一种方法（常常被称作最大匹配分块或 MMS）用于基于给定的词典将语句分块成词，这是通过试图以数种方法尽可能最佳地解决涉及给定的适于分解的语句的歧义实现的，从而抽取出不同的词。

为了解决这种歧义，常常使用启发式解决方案，例如最大匹配标准，可能还用其他标准进一步细化该标准。最大匹配基于对这样的实事的认识：作为一般的规则，给定表意符号序列属于词典中的单个词的概率比该序列对应于文本中的连结的多个更短的词的概率高。

在最简单的版本中，算法从语句的开始借助于其自己的词典搜索由最多数目的表意符号组成的词。在定位这种词后，算法立即对紧接刚定位的词的表意符号进行分析，从而开始再次搜索。

混合方法提供与每个词关联的固定成本。该成本由下述因素分配：可以与在给定的语言中词出现的频率相关的度量，或者词所属的语法类别可以在语句的句法上下文中出现的概率。

在对给定的语句定义的各种类型的分块中，选出具有最小成本的一种。

这些现有方法的示例例如 R. Sproat 等的文章“A Stochastic Finite-State Word-Segmentation Algorithm for Chinese”, Computational Linguistics, Volume 22, Number 3, 1997 的第 378～402 页，以及 US－A－6 173 252。

具体地说，在 Sproat 等的文章中公开的安排提供了所应用的成本函数，该成本函数与各自的词汇中的给定的词的出现频率成反比。

相反，US－A－6173252 中的安排本质上是基于句法类型的，即是这样的类型的，在该类型中，所使用的成本/权重函数与下述内容相关，所述内容是与常规字典中的词一起作为普通分块单元的通常所犯的错误串、人名、地点和组织、数字、以及数字的组合和测量词汇。

发明内容

因此，需要改进用于包括无分隔符的块的语言（例如普通话汉语）的文本到语言合成的解决方案。

本发明的目的是提供一种改进的布置。

根据本发明，利用具有在下述各个方面中要求的特征的方法实现了该目的。

本发明还涉及根据这种方法工作的分块器，该分块器优选采用适当编程的通用计算机形式。因此，本发明还涉及可加载到计算机存储器中的计算机程序产品，该计算机程序产品包括软件代码部分，用于该产品在计算机上运行时执行本发明的方法。另外，本发明覆盖包括前述分块器的文本到语言合成系统。

本发明的一个显著特征在于使用不同于现有技术的度量。具体地说，本发明考虑每个单个的词的语义上下文。这样，对文本中的语句进行分块取决于前面的语句（提供语义相关性存在），并且分配给每个词的成本作为在前面的分块中发现的词的函数而变化。

这样获得的所有分解可以从而被映射到网格或矩阵，其中每个元素由词加该词的成本组成。然后，例如使用动态编程挑选出具有最低成本的分块。

附图说明

下面将参考附图描述本发明。

图 1 到图 4 每个都由包含在这里公开的安排中执行的步骤序列的流程图组成。以及

图 5 是对应系统的示意性基本框图。

具体实施方式

通过介绍，将提供这里所公开的安排的基本概念的一般描述。

简言之，这里所公开的文本到语言合成安排是基于词典方法的，该词典方法基本上与最大匹配方法相关。

作为第一步骤，根据某些基本规则输入文本被细分为语段（syntagm），其中语段是文本的一部分，例如由标点符号定界的语句。此后每个语段按顺序被发送到分块模块。

更具体地说，从语段中的第一表意符号（即，块）开始，搜索对应于所定义的规则（例如，日期、小时等）的“特定”序列。如果这种序列被定位出，则给这种序列分配明确的成本。

还搜索以该表意符号开始的词典中的最长的词，然后搜索第二长的词，等等，直至以该表意符号自身结束的词。

在词典中发现的那些词都具有相同的成本（例如等于 5 的成本），比分配给特定序列的成本（例如等于 3 的成本）高。对于根据基本规则搜索或在词典中搜索都未定位（即，发现）的那些词，分配相对于前述成本较高的成本。

以此方法，创建一种具有和语段中的表意符号一样多的列的网格或者说矩阵，从而表意符号可以与每列相关联。行数随列而变，并且对应于在词典中定位出的以对应于该列的表意符号作为第一表意符号的词的数目。

如果未发现从给定列开始的词，则行数固定（存在某些例外），并且包括单一长度的词，然后是具有后续表意符号的词，等等，直到给定的长度。

在所提供的下面的示例中，在汉语表意符号的位置，拉丁字符例如 A、B、C、D 等被用作包括要分块的语段的各个元素的代表。

假定包括大量虚构的词的词典 Lex 可用：

Lex =（A，ABC，BC，CD，CDAC，D）

并且考虑语句 ABCDACEFD。

网格或矩阵将如下布置：

列	0	1	2	3	4	5	6	7	8
行	ABC	BC	CDAC	D	A	CD	E	F	D
	A		CD				EF	FD	
							EFD		

在指定的位置 6 和 7 处未发现词，从而给在列 6 和 7 中的那些词分配随着长度增加而增加并且比分配给在词典中发现的具有相同长度

的词的成本高的成本。

在该示例中，在几乎所有汉语的语句中都是这种情形，各种可能的分块例如是：ABC－D－A－EF－D 或 AB－CDAC－E－F－D。

这里公开的布置查找具有最低成本的序列。这优选是通过动态编程实现的，一旦创建了网格或矩阵，就可以轻易地借助这种动态编程。与确定所有可能的序列和各个成本的“蛮力（brute force）”方法相比，动态编程节省大量计算。

从语句/语段的最后位置（例如位置 8）开始，针对列中的每个词搜索具有最低成本的序列。参考前述示例，这是从 D 开始。

给定由行 j 和列 i 标识的词（下文简称 Wi,j），从 Wi,j 开始的最低成本序列由下述公式给定：

$$MincostW_{i,j} = Min_{(k)}\{CostW_{i,j} + MinCostW_{(i+lengthWi,j),k}\}$$

可能存在这样的情形，在这些情形中，可能存在从词 Wi,j 开始的数个具有相同成本的情形，尤其是如果该词在语段的末尾。

在这种情形中，至少两种启发式方法可用来选择序列。第一种方法是选择具有较长第一词的序列。替换方法是选择具有较低长度变化的序列。

现在将进一步通过解释把刚描述的布置与基于（纯）词典方法工作的解决方案相比较。

作为示例，将参考前面所用的相同词典，使用最大匹配法对语句/语段 ABCDAC 分块。

实际上，所讨论的序列仅可以以一种方式被分块，即 AB－CDAC。但是，最大匹配解决方案一般会定位不完整的序列 ABC－D－A，并且此后在未定位出正确序列的情况下停止。

当然，借助于回溯（backtrack）步骤可以消除这种缺陷，但是这会带来极大的计算复杂性负担，从而将给当前认为是最大匹配方法的强项的内容带来不利影响。

已知的 MMS 布置本质上是利用最大匹配启发式概念的基本算法。

这种方法的示例是所谓的 MMSEG（关于 MMSEG 的一般信息参考例如 Chih－Hao Tsai 的文章："MMSEG: A Word Identification System for Mandarin Chinese Text Based on Two Variations of the Maximum Matching Algorithm"，该文章在下述网址处可获得：http://input.cpatch.org/cutphase/mmseg.htm）。

MMSEG 的确是最有效的利用最大匹配概念的分块器之一。但是，类似于 MMS（尽管具有低的多的概率），该方法也可能无法定位出正确的序列，即使存在一个序列。在这种情形中，回溯也可以代表该问题的解决方案。

具体地说，MMSEG 从语段的开头开始挑选具有最大长度的三个块的序列中的第一个词。例如，假定词典 Lex＝（A，B，AB，CD，E，EF），并且语句为 ABCDEFABCD，MMSEG 搜索所有可能的由三个块的窗口组成的序列，即：

（1） A－B－CD

（2） AB－CD－E

（3） AB－CD－EF

然后，该方法选择最长序列（序列 3）的第一个词，该词对应于 AB。

从而，MMSEG 实现了良好的结果。但是，除了具有可观的计算负载之外，它还具有不能考虑所有可能的序列的限制，同时冒着不能以一致的方式应用最大匹配启发式标准的风险。

进一步的示例可以参考词典 Lex＝（A，AB，BC，CD，DC，EF，GH，I，FGHI）和语句 ABCDEFGHI。MMSEG 方法将不能定位出词 FGHI，即使该词包括在可接受的序列（A－CD－DE－FGHI）中。

这里所公开的方法可以消除该缺陷，因为其可以考虑所有可能的序列，而不排除任何序列。在该方法中，避免了不能检测出根据最大匹配标准很可能为正确的词的词。

所谓的统计算法稍稍不同于具有词典基础的那些算法，这是由于它们在对未知的词（即未包括在训练集中的词，例如人名）进行分块

时的改进的特性。这里所公开的方法部分程度上具有相同的缺点，但是可以利用使识别特定记号（例如日期，小时等）更容易的规则得到补充。

再次说明，这里所公开的方法的一个显著特征在于所使用的不同于现有技术的度量。

具体地说，这里所公开的方法考虑每个单个词的语义上下文。从而使得对文本中的语句进行分块依赖于前面的语句（假设存在语义相关性），并且分配给每个词的成本作为在前面的分块中发现的词的函数而变化。

这样获得的所有分解可以从而被映射到网格或矩阵，其中每个元素由词和该词的成本组成。然后，例如通过动态编程挑选出具有最低成本的分块。

现在转到图 1 到图 4 的流程图，假定这里所公开的分块器接受用 Unicode 系统（或者类似的系统）编码的文本作为输入，这种文本被细分为段落，段落然后被细分为“语段”，语段是用特定字符序列（例如，后接空白或新行的句号或逗号、惊叹号或问号、两个表意符号之间的空白等）定界的文本串。

在图 1 中，步骤 100 一般指示文本被输入到系统的步骤，而步骤 110 是检查正处理的文本是否为空的步骤。如果为空则过程在步骤 160 中结束。

否则，从文本中抽取出段落，并且加载到缓冲器 A（图 5）中。这发生在步骤 120 中。

在步骤 130 中，检查缓冲器 A 确认其是否为空。

如果缓冲器 A 不空，则抽取出语段并将其插入到缓冲器 B 中。这发生在步骤 140 中，此后系统再次上行到步骤 110。

如果缓冲器 A 为空，则系统前进到步骤 150，然后上行返回到步骤 130。

一旦语段被插入到缓冲器 B，系统返回到步骤 130，就意味着发生了步骤 140，步骤 140 是等待步骤，其是要确定缓冲器 B 中的所有

语段都已被分块器处理，在清空缓冲器 B 后，返回到步骤 110。

本领域的技术人员将马上理解，将文本细分为段落并非严格必需的。实际上，整个输入文本可以看作单个段落。

一旦缓冲器 B 填满了当前段落的语段，在步骤 200（图 2）中抽取每个单个语段，此后，在步骤 210 中，检查缓冲器 B 以确认其是否为空。如果为空，则在步骤 220 中清空动态词典（见下面），然后返回到步骤 160。如果步骤 210 出现否定结果，则如步骤 230 所示例，系统进行适当地分解，以分解成词。

图 3 的流程图的输入是单个语段，由 300 来表示。在步骤 304 中，设置指向语段的第一个字符的指针（INDX）（设置为 0 的指针）。

在步骤 308 中，搜索从指针 INDX 指定的位置中的表意符号开始的最长可能串。

在这种搜索中，所谓的“特定”块被搜索：这些块例如包括日期、小时、数字（作为表意符号的和作为拉丁字符的）以及不同于表意符号的那些字符序列。

在步骤 312 中，如果该搜索结果为肯定，则新的块被添加到缓冲器 C（再次参见图 5），该新块具有关联的对应固定成本 CF。这发生在步骤 316 中。

相反，如果搜索得到否定结果（步骤 312 的否定输出），则系统直接前进到步骤 320，在该步骤中执行新搜索。

在该阶段中，如果给定的表意符号不是最后的表意符号的话，则从文本抽取出的串包括在指针 INDX 指定的位置中的表意符号直到该给定的表意符号（例如第十一个表意符号）之间。如果相反，则该串是 INDX 到该语段末尾之间的一个。

在包括在静态词典中的词中搜索这样获得的串。

如果搜索得到肯定结果，则所定位出的词与其自己的成本一起被写到缓冲器 C 中，其中该成本等于 CM 表示的常量值（该值一般比 CF 高）。随后，通过去除右端的最后一个表意符号来缩短串，然后重复该搜索。

一旦该搜索完成，则通过插入在此期间定位出的所有的词与它们的成本（即，CM），从而更新缓冲器 C。这发生在步骤 324 中。

然后，在步骤 328 中，如果这两个搜索中的至少一个得到了肯定结果，则系统前进到步骤 332。否则，系统直接进行到步骤 344。如果出现在缓冲器 C 中的每个词出现在 SLEX 中并且该词的长度至少为两个字符，则该词的成本被更新为 SLEX 中的对应的成本。

在步骤 332 中，在前面的语段中出现的已定位的词的数目（NOL）以及所有已定位的词的计数（NW）的值被更新。

步骤 336 对应于更新动态词典（SLEX）的步骤，该步骤将在下面参考图 4 的流程图更详细地解释。

随后，在步骤 340 中，如果搜索都未得到结果，由在指针 INDX 指定的位置中的单个表意符号组成的词与成本 CS 一起被加载到缓冲器 C 中，其中成本 CS 比 CM 高。仍旧在步骤 340 中，缓冲器 C 的中的所有词被传送到网格或矩阵 RET（其对应于前面所述的表）的由指针 INDX 指定的列。

此后，在步骤 344 中，指针 INDX 被加 1，并且在步骤 348 中检查结果值是否超出语段的最后的表意符号。

如果未超出，则刷新动态词典 SLEX，其中每个条目的所有成本都被增加一个常量值，同时消除具有比 CM 高的成本的块。这发生在步骤 352 中。

相反，如果更新后的 INDX 值超出语段中的最后的表意符号，则在步骤 356 中，刷新动态词典，同时将 NOL、NW 和 INDX 的值重置为 0。此时，系统返回到步骤 200。

图 4 的图详细示出了动态词典 SLEX 的更新过程。

在步骤 410 中，在动态词典中搜索缓冲器 C 中包含的每个单个的词（在步骤 400 中定位出的），在开始处理新段落时动态词典被完全清空（步骤 420）。

如果该词已出现在动态词典中，则在步骤 430 中将相对成本减少常量值 DC。如果该词未出现在动态词典中，则在步骤 440 中检查动

态词典是否为满。

如果未满，则在步骤 450 中将该词与减少了值 DCI 的相对成本（CM 或 CF）一起插入。

相反，如果动态词典 SLEX 为满，则在步骤 460 中检查是否存在具有比 CM 高的成本的任何词。

如果存在，则在步骤 470 中，用具有在前面的步骤 450 中定义的成本的新词替换该词。

如果不存在这种具有比 CM 高的成本的词，则系统直接前进到步骤 480。该步骤实质上检查以确认是否已检查了缓冲器 C 中的所有词。

如果尚未，则系统返回步骤 400。相反，如果已检查了缓冲器 C 中的所有词，则系统前进到最后步骤 490。

应当理解，动态词典中的每个词的成本永远不会小于 0。

一旦网格或矩阵 RET 已完成，则要定位最小成本序列。优选借助于动态编程实现该过程。

具体地说，对于网格中的每个词 $W_{i,j}$，基于下述公式计算从 $W_{i,j}$ 开始的序列的最小成本：

$$MincostW_{i,j} = Min\ (over\ k)\ \{CostW_{i,j} + MinCostW_{(i+lengthWi,j),k}\}$$

其中，Mincost 表示最小成本，Min 表示遍历 K（over k）的最小函数，并且所考虑的长度为词 $W_{i,j}$ 的长度。

如果正处理的词包含多于两个表意符号，则 $CostW_{i,j}$ 所表示的成本因子是 NOL 对 NW 的比例的函数，其给出了当前语段与前面的语段的语义相关性的定量的含义。另外，该比例取决于该词是否已出现在动态词典 SLEX 中而变化。

优选地，该函数定义如下：

- 如果先前该词未包括在动态词典中，则

$$CostW_{i,j} = CSLEX$$

- 否则

$$CostW_{i,j} = CSLEX + (Cfs - CSLEX) * (1 - NOL/NW) /K$$

在这两个公式中，CSLEX 代表动态词典（SLEX）中的词的成本，

而取决于该词是通过第二搜索（B）还是第一搜索（A）定位出，Cfs 等于 CM*CF，而 K 是常数值。

这些成本涉及每个字符。

本领域的技术人员将立即认识到，图 1 到图 4 的流程图直接映射到适于借助于计算机基于图 5 示意性示出的体系结构实现的各个分块器 10 的对应功能块，所述计算机例如是专用处理器、或者适于编程的通用计算机/处理器、或者任何等价的数据处理结构。

分块器 10 适于构成包括由 30 和 40 一般示出的其他子系统组件的文本到语言合成系统的基本构造块。

在这些子系统中（它们本质上是本领域已知的，所以在这里没必要提供详细的描述），子系统 30 包括文本输入设备，例如 OCR 阅读仪、键盘/小键盘、或者适于将文本例如普通话汉语输入分块器 10 的任何其他文本源。

这种输入设备可以包括（如果未包括在分块器 10 中的话）例如适于使用编码技术将组成文本的各个元素（即，表意符号）编码为位串的处理模块（未示出，但是本领域已知），所述编码技术例如是 ISO 标准、或者 Unicode、GB 或 BIG5 编码技术。考虑到分块器 10 中的分块，编码技术的选择可能取决于已经过拼音语音音译（pinyin phonetic transliteration）的表意符号。

标号 40 表示本质上也是已知类型的整个语音合成系统，它适于将在分块器 10 内的分块产生的序列转换为发音合成数据，该数据适于生成例如利用扬声器 50 发声的对应音频语音信号。

当然，在不对本发明的内在原理存在偏见的情况下，相对于仅作为示例已描述的内容，细节和实施例可以显著改变，而不脱离所附权利要求限定的本发明的范围。

图1

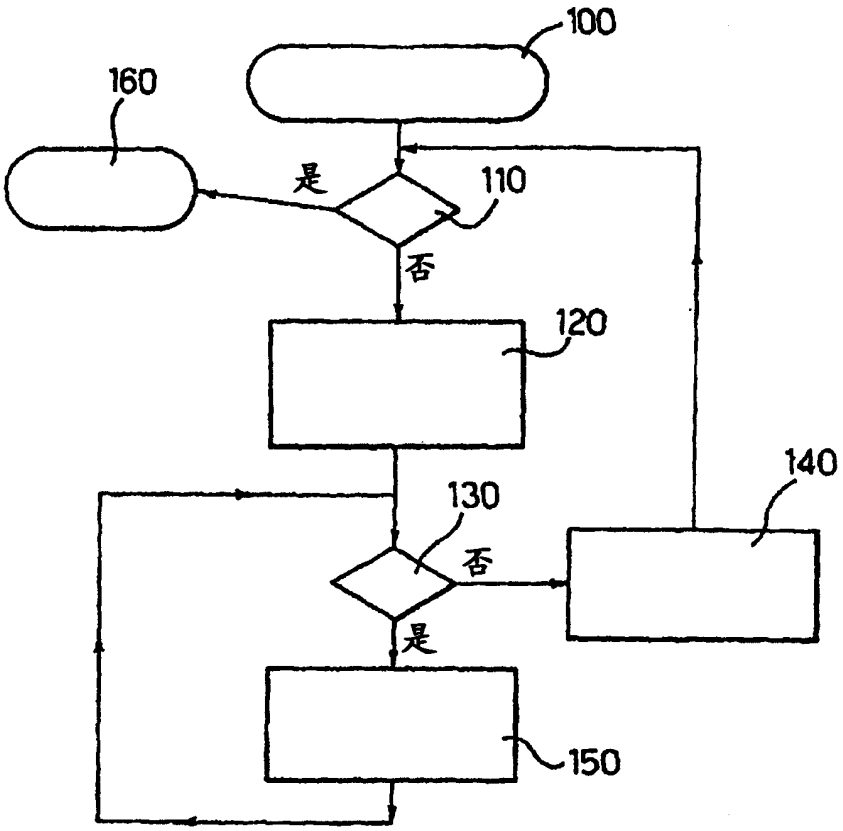


图2

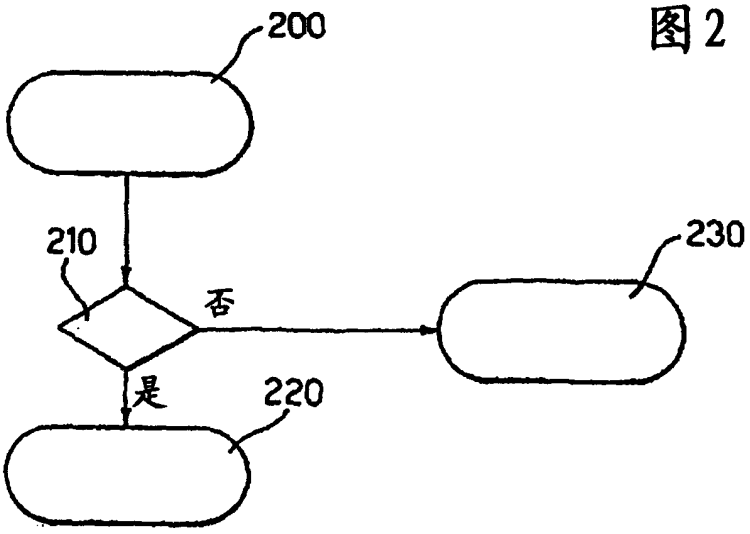


图 3

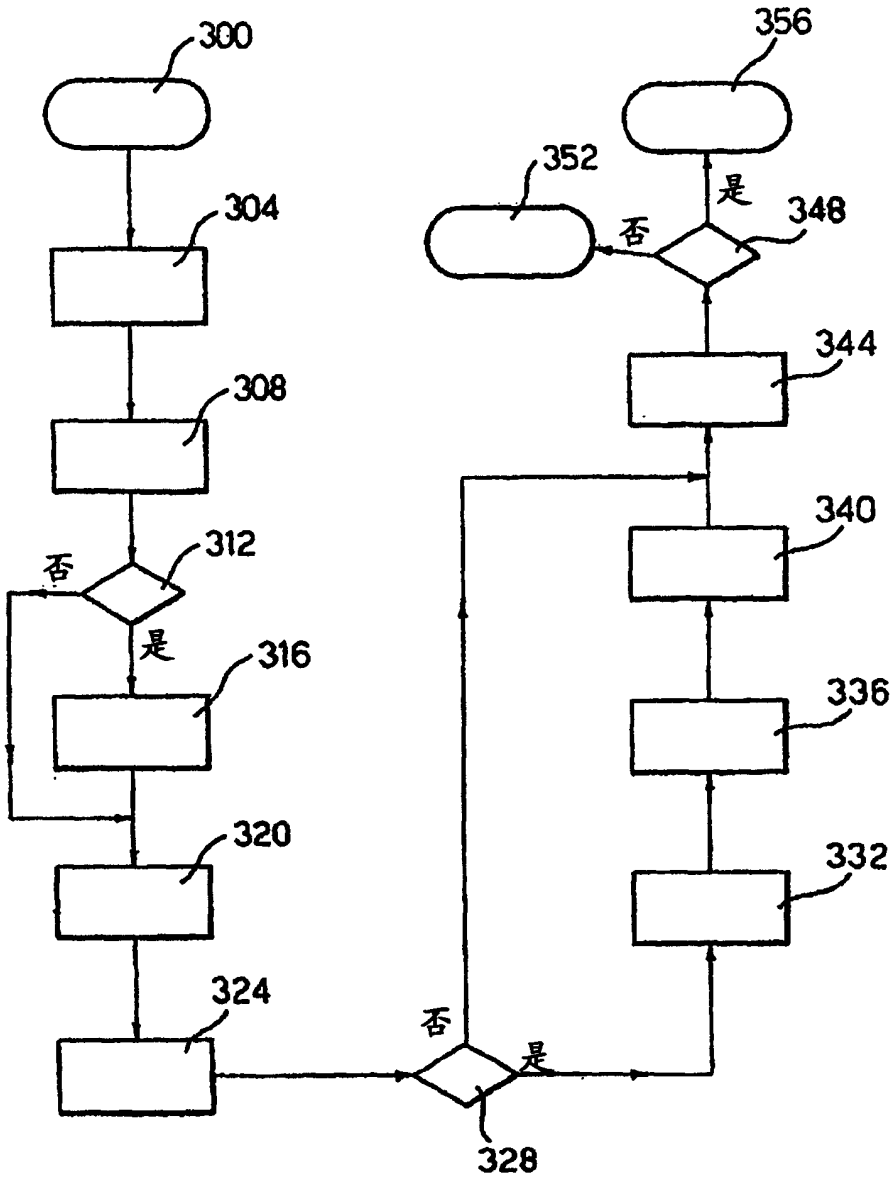
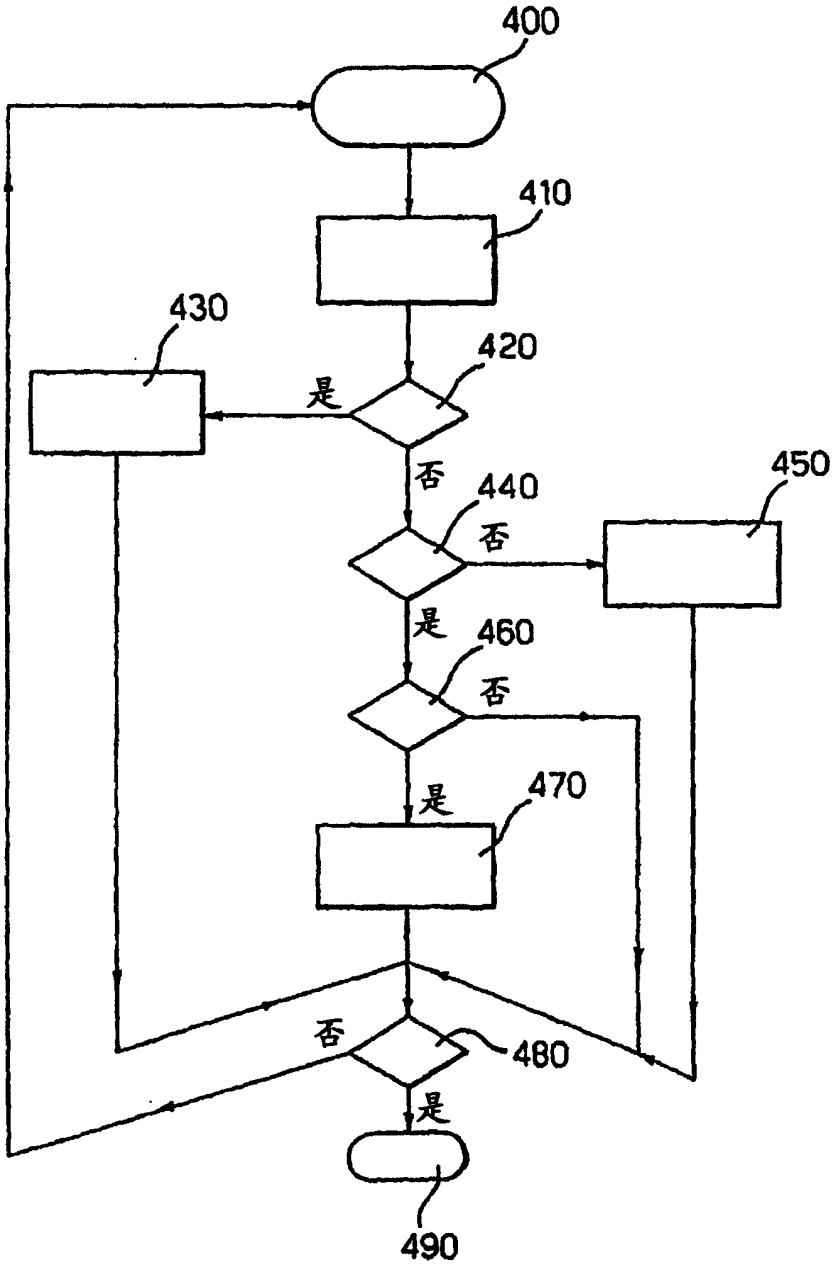

300
304
308
312
否
是
316
320
324
否
是
328
356
352
是
否
348
344
340
336
332

图4



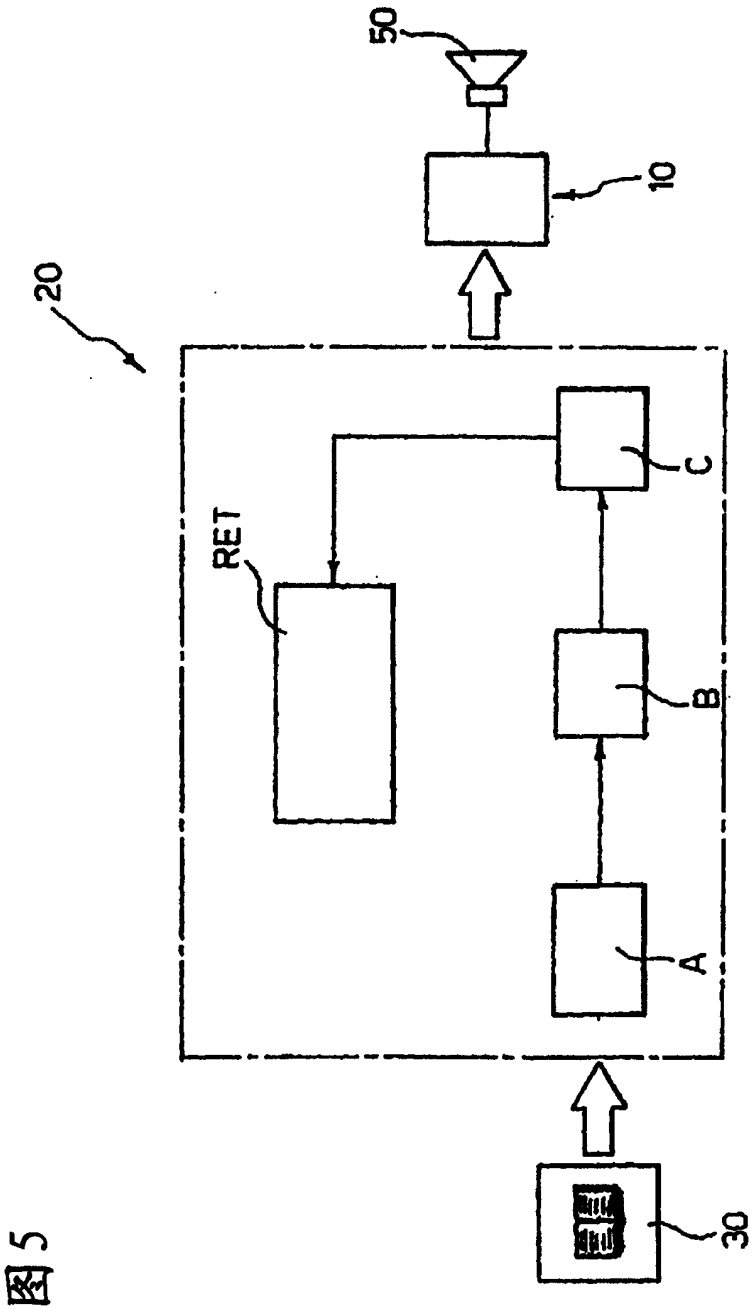


图5