



(12) 发明专利

(10) 授权公告号 CN 1748245 B

(45) 授权公告日 2011.03.09

(21) 申请号 200480004094.4

(22) 申请日 2004.01.19

(30) 优先权数据

10306022.7 2003.02.13 DE

(85) PCT申请进入国家阶段日

2005.08.12

(86) PCT申请的申请数据

PCT/EP2004/000351 2004.01.19

(87) PCT申请的公布数据

W02004/072954 DE 2004.08.26

(73) 专利权人 西门子公司

地址 德国慕尼黑

(72) 发明人 H·-U·布洛克 S·沙赫特尔

(74) 专利代理机构 中国专利代理(香港)有限公司

司 72001

代理人 吴立明 张志醒

(51) Int. Cl.

G10L 15/18(2006.01)

G10L 15/28(2006.01)

(56) 对比文件

EP 1162602 A1, 2001.12.12, 全文.

US 5873056 A, 1999.02.16, 全文.

CN 1315809 A, 2001.10.03, 说明书第4页最后1段至第6页第2段, 说明书附图1、2.

审查员 邢雲峰

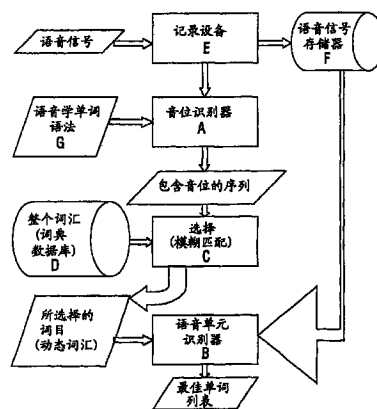
权利要求书 2 页 说明书 6 页 附图 2 页

(54) 发明名称

三级单个单词识别

(57) 摘要

在三级语音识别的情况下, 首先给语音单元分配音位序列, 然后从选择词汇中挑选那些最类似于该音位序列的词目并且最后利用语音单元识别器来识别该语音单元, 该语音单元识别器将所选择的、最类似于该音位序列的词目用作词汇。



1. 用于语音识别的方法，
 - 其中，通过音位识别器将包含音位的序列分配给语音单元，
 - 其中，从该包含音位的序列的选择词汇中选择类似的词目，
 - 其中，该语音单元通过语音单元识别器来识别，
 - 其中，该语音单元识别器的识别词汇包含所选择的、类似于该包含音位的序列的词目，其中，作为音位识别器的识别词汇除了音位以外较大的单元也被用作音位并被分配给语音单元。
2. 根据权利要求 1 所述的方法，
其中，所述较大的单元是辅音群、由元音和辅音构成的组合和 / 或最常出现的音节。
3. 根据权利要求 1 或 2 所述的方法，
其中，音位识别器的词目的数量小于选择词汇的词目的 10%。
4. 根据权利要求 3 所述的方法，
其中，音位识别器的词目的数量小于选择词汇的词目的 1%。
5. 根据上述权利要求 1-3 之一所述的方法，
其中，音位序列中的次序通过语音模型和 / 或通过说明语音学的语音单元构成规则的语音单元语法来限制。
6. 根据权利要求 1 所述的方法，
其中，较大的单元在选择类似于所述包含音位的序列的词目之前被拆成音位。
7. 根据上述权利要求 1-3 之一所述的方法，
其中，选择类似于所述包含音位的序列的词目借助索引搜索来实现。
8. 根据上述权利要求 1-3 之一所述的方法，
其中，为了选择类似于所述包含音位的序列的词目，构成对应于该包含音位的序列的音位双字母组。
9. 根据上述权利要求 1-3 之一所述的方法，
其中，所选择的、类似于所述包含音位的序列的词目的数量小于选择词汇的词目的 10%。
10. 根据权利要求 9 所述的方法，
其中，所选择的、类似于所述包含音位的序列的词目的数量小于选择词汇的词目的 1%。
11. 根据上述权利要求 1-3 之一所述的方法，
其中，所述语音单元识别器具有 HMM 识别器。
12. 根据上述权利要求 1-3 之一所述的方法，
其中，所述语音单元是一个单词。
13. 用于语音识别的装置，其具有：
 - 用于通过音位识别器将包含音位的序列分配给语音单元的装置，
 - 用于从该包含音位的序列的选择词汇中选择类似的词目的装置，其中，该语音单元通过语音单元识别器来识别，
其中，该语音单元识别器的识别词汇包含所选择的、类似于该包含音位的序列的词目，

其中,作为音位识别器的识别词汇除了音位以外较大的单元也被用作音位并被分配给语音单元。

三级单个单词识别

[0001] 在诸如移动电话、无绳电话、PDA、Pad 或导航系统的所谓嵌入式硬件上应实现从大的词汇中语音识别单个单词，该嵌入式硬件当前以在主存储器和小的计算能力方面的限制突出。大的词汇在这里理解为多于 60000 个单词的词汇。例如应实现，通过语音命令将城市名输入导航系统中或者提供针对 SMS 的口授可能性。

[0002] 此外，在市场上可买到的 PC 硬件上应实现从十分大的词汇中语音识别单个单词。十分大的词汇在这里理解为多于 200000 个单词的词汇。由此，例如应实现，通过语音命令挑选公共电话簿或地址寄存器中的条目。

[0003] 迄今应用的方法针对嵌入式硬件必须过多的计算能力或过多的存储位置。针对市场上可买到的 PC 硬件，至少针对大的词汇的方法可供使用。对此，具有所谓的“树搜索 (tree search)”或具有所谓的“FET 搜索 (FST search)”的语音识别器被采用。

[0004] 令人感兴趣的技术解决路径在 US6, 256, 630B1 中被说明。此处，建议一个利用相似程度工作的系统。

[0005] US5, 873, 056 同样说明了一种用于语音识别的方法。

[0006] US5, 638, 425 公开了在 PC 硬件上具有十分大的词汇的语音识别，其中采用单词识别器和音位识别器而且那些具有最好的相符程度的识别结果被用作数据库中的搜索词。

[0007] 由此出发本发明的任务在于，在具有少量存储器和计算能力的大的或十分大的词汇中实现语音识别、特别是单个单词识别。

[0008] 这些任务通过独立权利要求中给出的本发明来解决。由从属权利要求得出有利的改进方案。

[0009] 在用于语音识别的方法中，给语音单元、特别是一个单词通过音位识别器、也即用于识别音位的装置分配包含音位的序列。这通过以下方式执行，即音位识别器在语音单元中识别存在的音位。然后这样产生的、包含音位的序列与选择词汇的词目比较，该选择词汇大或十分大并相应地具有很多词目。从很多词目中选择大量的类似于包含音位的序列的词目。最类似于包含音位的序列的词目是优先的而且其数量显著位于选择词汇的词目的数量之下。这样选择的、类似于包含音位的序列的词目然后作为语音单元识别器的识别词汇、也即用于识别语音单元的装置被装载而语音单元由语音单元识别器从这些识别词汇中被识别出。

[0010] 由此可得到优点，不仅在识别时通过音位识别器而且在识别时通过语音单元识别器可执行相对小的词汇中的搜索，这意味着存储器和计算资源上的显著节约。在该选择词汇上，相反只执行相似性搜索，该相似性搜索实质上计算强度更小。

[0011] 通常的音位识别器可惜常常提供相对差的识别能力。因此有意义的是，在第一级中应用的音位识别器的识别能力通过一定的措施来改善。

[0012] 为此，作为音位识别器的识别词汇不仅应用德语中的例如 36 个音位，而且还应用其他单元，该其他单元作为伪音位被嵌入该音位识别器的词汇中。由于较大的单元通常被更好地识别出，所以较大的单元可被用作伪音位、尤其是待识别的语音单元的语音的辅音群、由元音和辅音构成的组合和 / 或最常出现的音节，该较大的单元还没有单独构成单词。

该音位识别器也即是尤其是识别、但是不仅仅识别音位的识别器。

[0013] 但是音位识别器的词目的数量不应如此大,以致嵌入式或者 PC 语音识别器的资源负担过重,由此本发明的优点丢失。当音位识别器的词目的数量总计为选择词汇量的词目的约 0.01%至 10%、尤其是大约 0.1%至 1%时,实践上已被证明为有利的。按照所嵌入的解决方案,音位识别器的词目的数量也即应总计为大约 300 或也总计为大约 2000。

[0014] 在音位识别器的相对大量的词目、也即约 2000 个词目中,除了 由一个元音和一个辅音构成的组合之外,由多个元音和辅音构成的组合也被考虑为伪音位。

[0015] 改善音位识别器的性能的另一可能性在于,在音位序列中仅允许一定次序的音位或音位和较大单元。该语音学上正确的次序尤其是这样。这在识别时可通过应用语音模型和 / 或通过说明各自语音的语音学的语音单元构成规则的语音单元语法来实现。

[0016] 在第二级,从选择词汇中选择类似于包含音位的序列的词目可通过一定的措施实现显著的效率提高。

[0017] 由于单个单词通常应被识别为语音单元,所以一个错误识别的较大的单元、特别是一个错误识别的音节在确定相似性和因此在选择时导致相对大的错误,因为该单元已经单独构成该单词的大部分。但是这些错误识别的较大的单元经常包含至少一个或多个正确识别的音位。这可以这种方式来利用,即在选择类似于包含音位的序列的词目之前将较大的单元再次拆成音位,以致包含音位的序列作为音位序列唯一或近似唯一地包含音位。由此,在确定相似性时,包含在错误识别的较大单元中的正确的音位做出正确的贡献。

[0018] 在选择类似于包含音位的序列的词目时,原则上可应用任意的相似程度,以便确定包含音位的序列和分别待比较的词目之间的相似性。选择词汇的词目随后根据其相似程度的大小来排序并且最类似于包含音位的序列的词目被选择。

[0019] 但是在大的或十分大的选择词汇中,确定所有词目的相似程度将意味着不可认同的时间花费。因此,当借助模糊匹配和 / 或借助索引搜索实现选择类似于包含音位的序列的词目时,这是有利的。为此,包含音位的序列被拆成大量其音位双字母组 (Phonembigramme)。

[0020] 针对语音单元识别器的识别词汇的大小的要求如针对音位识别器的识别词汇的大小的要求那样大约同样得出。相应地,语音单元识别器的词汇和音位识别器的词汇优选地约同样丰富。由于所选择的、类似于包含音位的序列的词目用作语音单元识别器的识别词汇或者由此构成至少重要的部分,也就是说,所选择的、类似于包含音位的序列的词目的数量总计为选择词汇的词目的大约 0.01%至 10%、特别是大约 0.1%至 1%。由此得出,在如今的解决方案中大约 300 的数量或也是大约 2000 的数量。

[0021] 针对独立于说话者的语音识别,HMM 识别器优选地作为语音单元识别器被采用。但是可替换的或补充的,采用 DTW 识别器也是可能的。语音单元识别器特别是利用维特比搜索来工作。

[0022] 在大多数采用情况下,语音单元是一个单词、特别是大约城市名、街道名和 / 或人名形式的名字。该语音单元识别器相应地通常是单词识别器。

[0023] 一种装置例如可通过数据处理设备的相应的设立和编程来实现,该装置被设立和具有装置,以便执行上述方法。由此特别是可提供音位识别器、选择装置和语音单元识别器来使用。由于语音单元不仅针对音位识别器的识别过程而且针对语音单元识别器的识别过

程是必需的,所以该装置优选地包含装置,通过该装置语音单元不仅可输送给音位识别器而且稍后可输送给语音单元识别器。这种装置例如可以具有语音信号存储器。

[0024] 该装置的其他有利的改进方案类似于该方法的有利的改进方案得出。

[0025] 包含代码段的针对数据处理设备的程序产品可通过以编程语音适当实施该方法并翻译为由数据处理设备可执行的代码来执行,利用该代码段之一可在数据处理设备上执行所描述的方法。该代码段为此被存储。在此,程序产品被理解为作为可处理的产品程序。该程序产品可以任意形式存在,例如分布在纸、计算机可读取的数据载体或网络上。

[0026] 本发明的其他重要优点和特征从根据附图对实施例的说明中得到。在此

[0027] 图 1 示出用于语音识别的方法的流程图;

[0028] 图 2 示出单词语法。

[0029] 针对大的词汇的语音识别系统形式的用于语音识别的装置例如包含以下部件:

[0030] - 音位识别器 A 形式的第一语音识别系统,该第一语音识别系统在时间平台上从连续说出的至少 300 个单词的所确定的词汇中可识别出单元,这是很小的要求,并且该第一语音识别系统允许给出有限的自动装置形式的音位模型;

[0031] - 语音单元识别器 B 形式的第二语音识别系统,该第二语音识别系统在时间平台上从动态可加载的至少 300 个单词的词汇中可识别出单个单词,这同样是很小的要求;

[0032] - 模糊匹配索引系统 C 形式的用于从选择词汇中选择类似于包含音位的序列的词目的装置,如大约从翻译记忆的技术中已知的和例如在 US5,724,593 中说明了其应用的那样;

[0033] - 数据库 D,其中待识别的词汇作为选择词汇以其字位学和语音学的形式来存放;

[0034] - 记录设备 E,利用该记录设备记录由用户表达的语音信号;

[0035] - 语音信号存储器 F,其中利用记录设备 E 记录的语音信号被存储;

[0036] - 可选的语法 G 形式的语音单元语法,该语音单元语法说明语音的语音学规则。

[0037] 在用于语音识别的方法的第一级中执行音位识别。为此用户的表达利用记录设备 E 来记录并存储在语音信号存储器 F 中。借助音位识别器 A 形式的第一语音识别系统,包含音位的序列被分配给语音信号。

[0038] 在此,音位识别器的识别词汇的选择是特别重要的。目前针对音位识别的方法将语音的音位定义、也即例如针对德语为 a、a:、ae、ae:、e、... 为音位识别器的单元,该音位不用语音模型被识别出。这样建立的系统是很易受干扰的并通常提供十分差的识别结果,该识别结果仅仅十分有条件地被利用。特别是在输入同样的单词时,所识别的音位序列因说话者的不同而严重变化。

[0039] 诸如单词的较大的相互关系的识别运行地更好,因为辅音的发音质量也还反映在相邻发音上。

[0040] 因此音位识别器优选地这样形成,使得尽可能较大的单元被识别为音位,但是尽管如此词汇被保持如此小,以致音位识别器在嵌入式硬件上能够运行。语音识别系统的识别词汇的大小也就是确定存储器和 CPU 需求。

[0041] 因而作为音位识别器的识别词汇的单元选择:语音的音位(发音)(如在传统的音位识别器中,在德语中为 36 个)+所有在词汇中出现的辅音群(在德语中例如 lchst、mpft 等等,总共大约 180 个)+120 个常用的音节。这总共大约 300 个词目,这对应于嵌入式硬件

的通常的效率。在有效率的硬件中,也可应用大约 2000 个词目,其中此处外加由一个或多个元音与一个或多个辅音和其他常用音节构成的组合。

[0042] 在一个单词中识别词汇的单元(词汇元素)的次序或者通过统计学的、在词汇上训练的、双字母组或三字母组(trigramm)-语音模型或者通过说明各自语音的语音学单词构成规则的语法 G 来限制。

[0043] 这样的语法 G 例如将一个单词描述为一个或多个音节的序列并接着将这些音节再度分别描述为可选的开头音、元音和可选的尾音的序列,该语法 G 可以有限的自动装置的形式和强大性(Maechtigkeit)来具体规定。该语法 G 确定,由哪些辅音和辅音群可以组成所观察的语音的开头音和尾音并且哪些尾音组可以跟在哪些元音的后面。

[0044] 此外,语法同样也说明辅音音节、如专有名词 Brand1 中的 d1。典型的语法在 2 中示出。

[0045] 音位识别器 A 形式的第一语音识别系统现在将词汇的这种单元的序列作为结果返回。针对单词 Moenchengladbach,该结果例如是序列 m-oe-nch-en-gl-a-t-b-a-x 或者在较差识别时也为序列 b-oe-h-mch-nen-k-au-b-a-k。

[0046] 当针对稍后紧跟的语音单元识别器 B 形式的选择第二语音识别系统的识别器词汇音位和较大的单元的所识别的序列现在再次被拆成音位序列时,已证明为完全特别有利。针对上面列出的例子得到音位序列 m-oe-n-ch-e-n-g-l-a-t-b-a-x 或 b-oe-h-m-ch-n-e-n-k-au-b-a-k,利用该音位序列继续处理。

[0047] 在用于语音识别的方法的第二级中,借助模糊匹配实现从选择词汇中选择类似于包含音位的序列的词目。利用模糊匹配索引系统 C,为此在数据库 D 中搜索那些在模糊匹配意义上与所识别的音位序列最相似的 300 个条目。

[0048] 为了使得模糊搜索成为可能,根据下面的方法建立索引。数据库的 n 个条目分别装备来自 {1...n} 的明确的索引号。为了包含每个记录 i 的开始和结束标记(#)在内的语音学表示,构成大量包含在其中的双字母组。针对 Moenchengladbach 和音位序列 m-oe-n-ch-e-n-g-l-a-t-b-a-x 相应地构成音位双字母组 {#-m, m-oe, oe-n, n-ch, ch-e, e-n, n-g, g-l, l-a, a-t, t-b, b-a, a-x, x-#}。

[0049] 针对来自该大量音位双字母组的每个双字母组 b,存储索引条目 b-{i}。如果在其它条目 j 中出现双字母组 b,则该索引条目相应扩展为 b{i, j}。这针对所有条目和所有双字母组继续进行。

[0050] 该结果是在该条目的各自的索引号上的双字母组(类型)的很少出现的矩阵,其中出现该双字母组。

[0051] 模糊匹配现在这样来实现,使得所识别的音位序列(例如 b-oe-h-m-ch-n-e-n-k-au-b-a-k 被拆成大量该音位序列的双字母组 BM {#-b, b-oe-h, oe-h-m, m-ch, ch-n, n-e, e-n, n-k, k-au, au-b, b-a, a-k, k-#}。

[0052] 对于每个双字母组,所分配的索引号的向量从存储器中被读出并构成那些通过 BM 中的双字母组参阅的索引号的不严格的并集 VM。在此,给并集 VM 的每个元素 e 装备频率数据,该频率数据给出有多少来自 BM 的双字母组参阅 e。

[0053] 针对条目 i 的所给的例子(Moenchengladbach)中的频率数据例如是 2,因为只有 e-n 和 b-a 参阅 i。

[0054] 索引号的所计算的频率 h 紧接着被转换为通过样本的长度 lm 和条目的语音学形式的长度 le 标准化的权重 g 。这按照公式 $g = -\log(h/lm) * 0.5 + -\log(h/le) * 0.5$ 或者按照其它合适的公式执行。

[0055] 不严格的并集 VM 的元素现在按照其权重上升式地分类。

[0056] 针对第三级（通过语音单元识别器的语音单元识别），例如分配给所分类的大量 VM 的第一个 300 个元素的条目作为词汇被载入语音单元识别器 B 形式的第二语音识别系统中。利用语音单元识别器现在还分析一次存储在语音存储器 F 中的语音信号，该语音单元识别器特别是被实现为单词识别器。典型地，语音识别系统应输出 m 个最佳识别的单词。

[0057] 针对音位识别器 A 形式的第一语音识别系统的所列出的词汇大小 300 是任意大小。通常适用的是，在词典中包含的单元越大，识别结果就越好。当硬件的计算能力许可时，例如有利的是，代替所列出的 300 个条目应用由语音的音节和所有在数据库中出现的音节组成的词汇。在德语中，此处估计大约 7000 个条目。语音学单词语法的表达可以限制在普通的规则单词 = 音节 * (一个单词由一个或多个音节的序列组成) 上。

[0058] 当众所周知容易混淆的发音如 m 和 n 不仅在数据库中而且在所识别的音位序列中被标准化为一个发音（例如 n ）时，模糊匹配系统同样可以被改善。在上述例子中，针对 Moechengladbach 的条目在 $m-n$ 标准化时将得到超过两个点。但是第一个经验测试目前已经显示，采用这种标准化也非常有助于选择错误条目。

[0059] 此外，模糊匹配系统必要时可由此来改善，即如此形成频率计算，以致诸如包含不发音的爆破音 p 、 t 、 k 的双字母组（众所周知的识别不理想的发音）仅仅作为 l 、 o 的较小部分有助于相应的索引号的频率的。

[0060] 模糊匹配系统可以由此来改善，即除了双字母组的频率以外数据库条目中的词汇的数量 ve 和搜索样本中的词汇的数量 vs 的绝对差以及数据库条目的长度 le 和搜索样本的长度 lm 的绝对差计入标准化的权重中。这可例如通过根据以下公式计算权重 g 来执行。

[0061] $g = -\log(h/lm) * f_1 + -\log(h/le) * f_2 + -\log(1/(abs(vm-ve)+1)) * f_3$

[0062] $+ -\log(1/(abs(lm-le)+1)) * f_4$

[0063] 其中相关系数 f_n 有意义地被采用。

[0064] 针对语音单元识别器 B 形式的第二语音识别系统的所列出的词汇大小 300 也是任意大小。当硬件的计算能力许可时，保证有利的是，代替所列出的 300 个条目应用大小为 v 的更大的动态词汇。在模糊匹配时，相应地从所分类的大量 VM 中选择第一个 v 个元素，以继续观察。

[0065] 通过本发明，在小的硬件要求的情况下通过分布在三个简单的操作中实现从大的词汇中识别单个单词。

[0066] 在 PDA 或者汽车电子（导航系统）上从直至 300 个单词的词汇中识别语音今天已经是现有技术并是可行的。

[0067] 针对模糊匹配的存储位置和 CPU 需求可证明是很小的。以下例子用于证明。

[0068] 存储器需求：在粗略估计平均单词长度为 10 个音位 / 单词时由 65000 个单词构成的词汇包含 650000 个音位出现（标志），因为德语中的平均单词长度大约是 8（加上单词开始和结束标记得出 10）。该值也在 65000 个城市名上被证实。由于双字母组出现的数量通

常等于音位出现的数量,所以得到大约 650000 个双字母组出现。不同的双字母组类型的数量当然很小。所测量的经验数据得到大约 1200 个双字母组类型的数量。通常很少出现的索引矩阵中的条目的数量等于双字母组出现的数量、也即大约 650000。由于针对 65000 (> 64k) 个索引号的编码每个索引号需要 3 字节,所以针对很少出现的索引矩阵的存储器需求总计为 1950000 字节或约 2MB。添加针对条目的字位学表示(约 700KB) 分别加上针对音位链的长度的存储的一个字节和词汇的数量(总共 130000 字节)的存储器需求以及最大为 65000 个元素的不严格的并集的构成的工作存储器。针对每个元素的存储器需求位于 7 字节处:3 字节针对索引号而 4 字节针对权重的实数描述。针对工作存储器因此应保留 455000 字节。针对模糊匹配的存储器需求因此对于 65000 个条目的词汇总共最大总计为 3.2MB。

[0069] CPU 需求:CPU 需求取决于不严格的并集的构成。在 650000 个双字母组出现和 1200 个不同双字母组类型的情况下,平均得到每个双字母组约 540 个索引号。用于计算频率的必要的加法的数量因此在长度为 n 的所识别的音位链的情况下平均总计为 $n*540$ 。添加权重的计算和工作存储器的相应的分类。

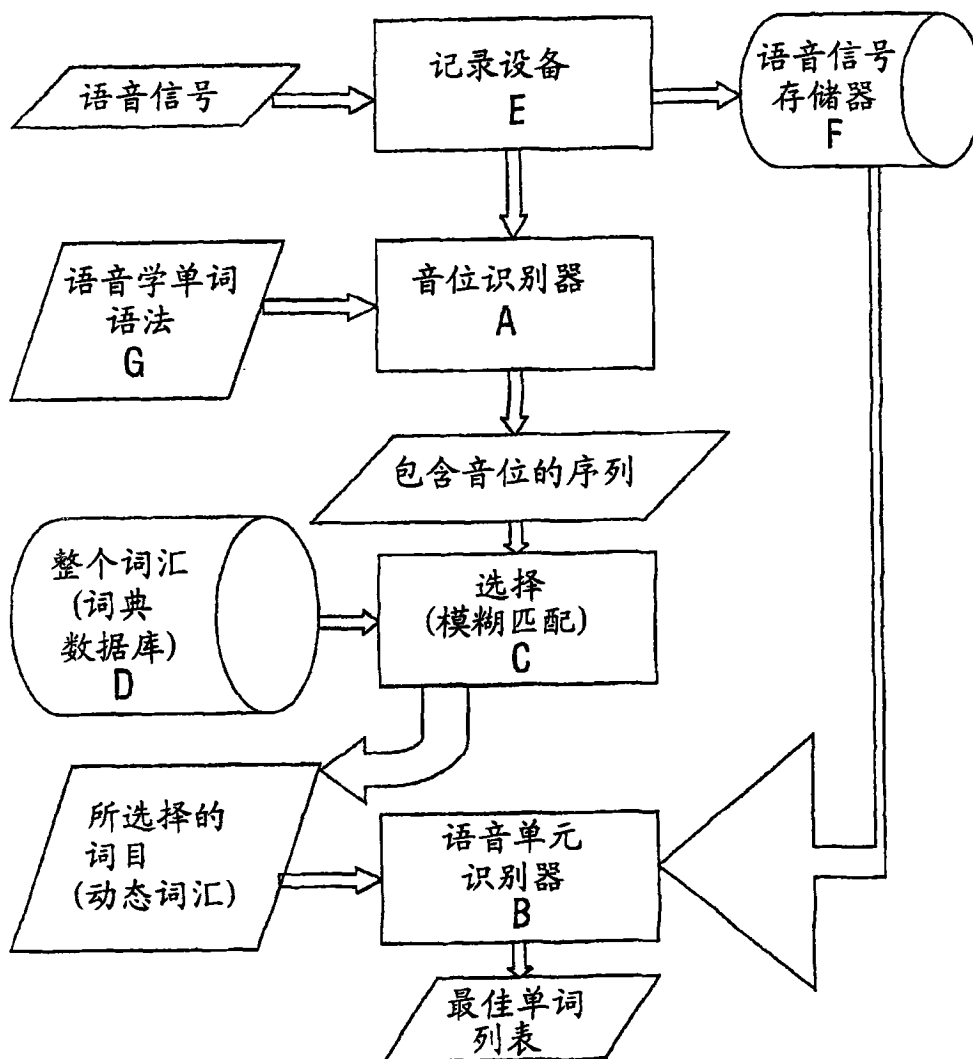


图 1

```

wort(W) --> silbe(t(W,B)), silben(t(B,[])).
wort(W) --> silbe(t(W,[])).

silben(t(A,B)) --> silbe_2(t(A,C)), silben(t(C,B)).
silben(t(A,B)) --> silbe_2(t(A,B)).

silbe(t(A,B)) --> anlaut(t(A,C)), silbe_2(t(C,B)).
silbe(t(A,B)) --> silbe_2(t(A,B)).

silbe_2(t(A,B)) --> ( vokal_1(t(A,D)),
                    ( auslaut_1(t(D,B))
                      ; auslaut_2(t(D,B))
                    )
                    ; vokal_2(t(A,D)), auslaut_2(t(D,B))
                  ).
silbe_2(t(A,B)) --> vokal(t(A,B)).

vokal(T) --> vokal_1(T).
vokal(T) --> vokal_2(T).

vokal_1(t([a|A],A)) --> [a].
vokal_1(t([ae|A],A)) --> [ae].
...
vokal_1(t([uh|A],A)) --> [uh].

vokal_2(t([ai|A],A)) --> [ai].
...
vokal_2(t([au|A],A)) --> [au].

anlaut(t([b|A],A)) --> [b].
anlaut(t([d|A],A)) --> [d].
...
anlaut(t([z|A],A)) --> [z].

anlaut(t([b,l|A],A)) --> [bl].
anlaut(t([b,r|A],A)) --> [br].
...
anlaut(t([t,s|A],A)) --> [ts].

anlaut(t([sh,p,l|A],A)) --> [shpl].
anlaut(t([sh,p,r|A],A)) --> [shpr].
anlaut(t([sh,t,r|A],A)) --> [shtr].

auslaut_1(t([l,t,s|A],A)) --> [lts].
...
auslaut_1(t([x|A],A)) --> [x].
...
auslaut_2(t([r,l,t|A],A)) --> [rlt].
auslaut_2(t([r,m,t|A],A)) --> [rmt].
...
auslaut_2(t([r,n|A],A)) --> [rn].
auslaut_2(t([r,t,s|A],A)) --> [rts].
auslaut_2(t([ng|A],A)) --> [ng].

```

图 2