ISSN 1000-0054 清华大学学报 (自然科学版) 2003 年 第 43 卷 第 7 期 31/37
CN 11-2223/N J Tsinghua Univ (Sci & Tech), 2003, Vol 43, No. 7 981-984

语音识别中基于两层词法树的跨词搜索算法

张国亮, 徐明星, 李 净, 郑 方, 吴文虎

(清华大学 计算机科学与技术系, 北京 100084)

摘 要: 为了在连续语音识别过程中充分并且高效地使用上下文相关声学模型, 提出了一种新颖的基于两层词法树的跨词搜索算法。采用两层词法树来表示搜索空间, 解决了现有单层词法树的规模爆炸问题, 使其有能力在词边界搜索中高效地使用上下文相关声学模型进行匹配, 充分发挥上下文相关声学模型较好地描述协同发音现象的能力。实验结果表明, 与词内搜索算法相比误识率平均下降 60%, 搜索时间达到实时, 证明基于两层词法树的跨词搜索算法具有很好的识别性能。

关键词: 语音识别; 搜索算法; 跨词搜索; 两层词法树

中图分类号: TP 391. 42 **文献标识码**: A

文章编号: 1000-0054(2003)07-0981-04

Cross-word search algorithm based on two-layer lexical tree for speech recognition

ZHANG Guoliang, XU Mingxing, LI Jing,
ZHENG Fang, WU Wenhu

(Department of Computer Science and Technology,
Tsinghua University, Beijing 100084, China)

Abstract: A cross-word search algorithm was developed based on a two-layer lexical tree using context-dependent acoustic models for efficient continuous speech recognition. The algorithm uses the two-layer lexcial tree to maintain the search network to keep the search space from becoming more complicated during the cross-word search. The search algorithm can then use context-dependent models on the word boundary for good performance. Experimental results show that the search algorithm can work in real-time with 60% less word errors compared with the intra-word search algorithm, which illustrates the efficiency of the cross-word search algorithm based on the two-layer lexical tree.

Key words: speech recognition; search algorithm; cross-word search; two-layer lexical tree

近年来, 上下文相关的声学模型因为能够较好地对协同发音现象进行建模而在语音识别系统中得到广泛应用[1]。共有两种策略来使用上下文相关声学模型: 第一, 词内搜索策略, 即只在一个词的内部进行搜索时使用上下文相关的声学模型, 而在词间搜索使用相对简单的声学模型; 第二, 跨词搜索策略, 即在词的内部和词之间进行搜索时均使用上下文相关声学模型。因为跨词搜索策略充分发挥了上下文相关声学模型的优秀性能, 所以识别性能较好, 成为主要的研究方向。

但是, 跨词搜索策略也有其搜索效率低的缺点。在跨词搜索策略的搜索网络中, 使用经典的静态词法树[2]时, 会使词法树的规模急剧膨胀。而且, 当将语言模型也进一步集成进词法树后, 词法树会变得更加复杂。庞大的规模导致识别效率低下, 最终也影响了识别性能。

本文提出一种新颖的跨词搜索算法来解决以上的问题。采用两层词法树来描述搜索网络, 即将上层知识源和动态规划时的具体知识信息分开描述。与传统的单层词法树描述的搜索网络相比, 所需要的静态和动态内存资源都非常小, 从而提高了搜索效率和搜索性能。

1 搜索网络的组织

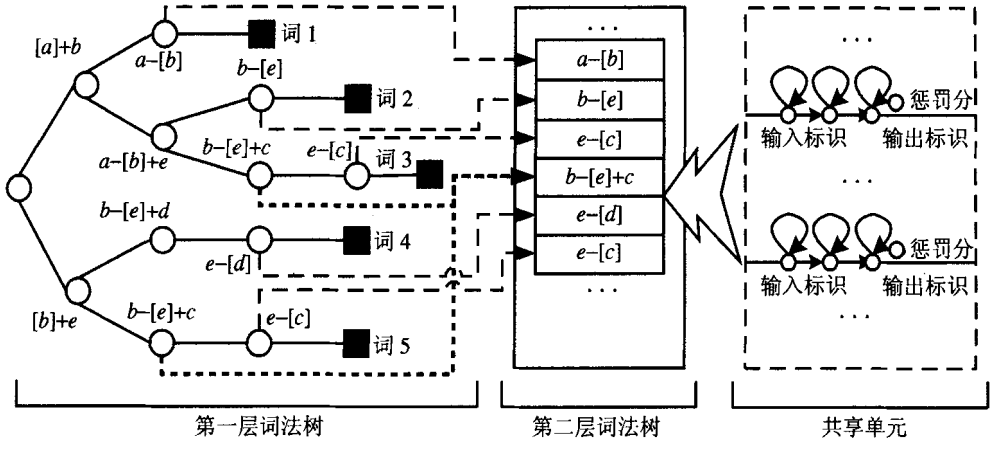
在大词表连续语音识别中, 根据所表述的知识源的不同, 搜索网络可划分为 3 个层次: 词关系层、基元关系层和动态规划层[3], 分别描述词一级、基元一级和实际动态规划运行时的的知识关系。本文采用两层词法树即是在保留层次性的前提下来描述整个搜索空间, 其中第一层反映词关系层和基元关系层, 第二层描述动态规划层。

收稿日期: 2002-07-10
基金项目: 国家自然科学基金资助项目 (19871045)
作者简介: 张国亮(1977-), 男(汉), 河北, 博士研究生。
E-mail: liang@sp. cs tsinghua edu. cn
通讯联系人: 郑方, 副教授,
E-mail: fzheng@sp. cs tsinghua edu. cn

1.1 两层词法树

在两层词法树中, 第一层用于描述搜索网络中的词关系网络和基元关系网络, 维护整体框架。语言模型以及识别基元词法图都体现在这一层。在连续语音识别中, 第一层词法树通常表示为前缀词法树的形式[2]。第二层用于描述动态规划信息, 是一系列共享单元的数组, 所有第一层中没有涉及到的信息都将出现在第二层中。共享单元之间相互独立, 且与第一层词法树中出现的识别基元一一对应。对应识别基元的各种信息如模型信息、上下文信息, 都存储在共享单元中。第一层词法树中所有相同的基元模型都指向同一个共享单元, 这样大量的信息都共享在一起, 大大减少了所需的存储空间。

每个共享单元由一条或多条独立的平行弧组成, 每条弧都代表路径扩展的一个可能路线, 在搜索算法的令牌中弧和路线一一对应。弧被赋予了 4 个域: 声学模型指针、弧惩罚分、进入标识和输出标识。其中: 声学模型指针记录了声学模型信息; 弧惩罚分是为了实现对每一条弧加权的策略; 后两个域为进入标识和输出标识, 主要用于表述弧的左相关属性和右相关属性。从图 1 中可以清楚地看出两层词法树的数据结构和相互关系。



上下文相关识别基元表示为 $a-[b]+c$, 即左相关基元为 a, 中心基元为 b, 右相关基元为 c。同理, 右相关识别基元表示为 $[a]+b$, 左相关识别基元表示为 $a-[b]$

图 1 两层词法树示意图

1.2 跨词搜索网络生成

声韵母结构是汉语的一种特性, 绝大多数汉语音节都由一个声母和一个韵母构成。在本文所进行的实验中, 上下文相关的扩展声韵母(XIF)[4]集被选作基本的识别基元。在扩展声韵母集中, 在原有 21 个声母和 38 个韵母的基础上, 增加了 6 个零声母, 从而保证每个音节都严格由一个声母和一个韵母组成。

两层词法树的构造分为两步: 第一层词法树的构造和第二层词法树的构造。从简单易行的角度出发, 第一层词法树可以从传统词内搜索策略使用的词法树中继承而来。因为传统的词内搜索词法树的框架就是描述词和基元模型的相互关系, 唯一所作的修改是将其描述声学模型信息的域替换为指向第二层词法树中相对应共享单元的指针, 这种方法使得第一层词法树很容易生成, 且在搜索过程中可以通过指针直接从第二层词法树中获取所需要的声学模型信息和时间信息。

第一层词法树维持着搜索网络的整体框架, 在它上面的每个节点都代表着一个上下文相关的 XIF 基元。第二层词法树是一个共享单元数组。在本文所描述的系统中, 词表中的一个词通常是由 3 部分组成: 一个右相关 XIF、几个上下文相关 XIF 和一个左相关 XIF, 部分单字词只包括第 1 部分和第 3 部分。为了在第二层中描述上下文相关模型的跨词搜索信息, 在对应于右相关 XIF 的共享单元中, 包括多条弧, 可以覆盖由这个右相关 XIF 所有可能扩展出的上下文相关的 XIF; 同理, 在对应于左相关 XIF 的共享单元中, 包括多条弧以覆盖由这个左相关 XIF 所有可能扩展出的上下文相关的 XIF; 而对应于上下文相关 XIF 的共享单元只需要一条弧就可以了。在这里左相关属性由左相关基元标识和中心基元标识共同组成, 右相关属性由中心基元标识和右相关基元标识共同组成。具体的例子可见图 2。

最终, 一个已经生成的共享单元包括多条弧, 这些弧覆盖当前基元所有可能的发音变化。在这里需

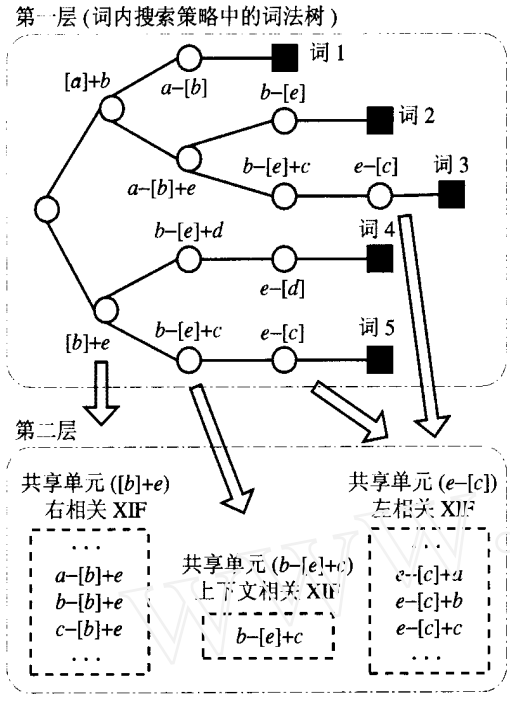


图 2 跨词搜索的具体实现

要阐明, 搜索过程中, 本文采用一条路径来记录在一条弧中扩展的当前状态。

2 基于两层词法树的搜索算法

在两层词法树的框架中, 由于第一层和第二层被分离开了, 没有直接集成在一个搜索网络中, 所以搜索算法需要加以改动以适应新的词法树结构。基于时间同步的 Token Passing 算法[5]被采用作为基准算法。每一个搜索令牌记录识别得分并且包括回溯信息。但是, 和原来算法中只有一条路线可供扩展不同, 搜索令牌中可能要处理多条路线, 路线的个数等于搜索过程所处的共享单元中弧的个数。一个搜索令牌可能具有多个入口和出口, 搜索令牌中的所有输入标识相同的路线都起始于同一入口, 所有输出标识相同的路线都终止于同一出口。每一条路线都有独立的模型信息和惩罚分值, 这些都可以从与该路线相对应的弧上得到。

根据搜索过程所处在搜索网络中的位置, 整个搜索算法被分为 3 个部分。先定义下面两个变量:

$Q_v^p(t, s_y^x)$:= 时刻 t, 历史词为 v, 处于第一层词法树的节点 p, 沿着进入标识为 x、输出标识为 y 的路线扩展到状态 s 的路径的整体识别得分;

$B_v^p(t, s_y^x)$:= 路径 $Q_v^p(t, s_y^x)$ 的起始时刻, 用于回溯得到识别结果;

第 1 步处理在基元模型内部的路径扩展。每一条路径扩展时都沿着共享单元中的一条路线进行, 不同路线上的路径互相不影响, 独立进行路径扩展。可用公式表示为:

$$Q_v^p(t, s_y^x) = \max_{\sigma} \left[q(x_t, s_y^x \mid \sigma_y^x) \cdot Q_v^p(t, \sigma_y^x) \right], \quad (1)$$

$$B_v^p(t, s_y^x) = B_v^p[t, \max(\sigma_y^x)], \quad (2)$$

其中 $q(x_t, s_y^x \mid \sigma_y^x)$ 表示从声学模型中计算得到的转移概率和输出概率的乘积。

第 2 步处理在基元模型间, 但仍然在一个词内的路径扩展, 这种路径扩展就是第一层词法树中节点之间的路径扩展。这时处在基元边界, 搜索令牌将沿着下一个节点对应的共享单元中的每一条路线进行繁殖。每一条路线都选择搜索令牌的一个出口, 该出口的输出标识必须和路线的输入标识相同。

$$Q_v^p(t-1, s_*^x = 0) = \max_{f \in \mathrm{parent}(p)}^{z,f} [Q_v^f(t-1, \sigma_x^z = \mathrm{end})], \quad (3)$$

$$B_v^p(t-1, s_*^x = 0) = B_v^{f'}(t-1, \sigma_x^{z'} = \mathrm{end}), \quad (4)$$

其中:

$$(f', z') = \arg\max_{f,z} [Q_v^f(t-1, \sigma_x^z = \mathrm{end})],$$

s_*^x 表示输入标识是 x, 处于状态 s 的所有路线, * 代表着不关心输出标识。

第 3 步处理词之间的路径扩展。除了要将语言模型的得分累计到路径的总体得分上, 其余的流程和前一步基本一致。因为到达了词的边界, 回溯指针的时刻索引也要重新赋值为当前时刻。整个过程见下面公式:

$$Q_w^{\mathrm{initial}}(t-1, s_*^x = 0) = \max_{v,z} \left[p(w \mid v) Q_v^{\mathrm{final}}(t-1, \sigma_x^z = \mathrm{end}) \right], \quad (5)$$

$$B_w^{\mathrm{initial}}(t-1, s_*^x = 0) = t-1, \quad (6)$$

这里仅仅使用二元语言模型来描述整个搜索算法, 但是三元语言模型可以很容易地被结合进算法中。所以基于两层词法树的跨词搜索算法既可以在一遍集成搜索算法中使用, 也可以在两遍词图搜索算法中使用[6]。

3 实 验

声学模型的训练集和测试集都从“八六三”数据库中得到, “八六三”数据库共有 80 人的男声数据。所有的语音都是在低噪音的环境下录制而成, 采样率为 16 kHz。上下文相关的 XIF 被选作语音识别基元, 每一个基元都用 HTK 训练成一个 3 个状态的 HMM 模型进行描述[4]。

表 1 扩展声韵母集

类型	基元列表
声母(27)	b, p, m, f, d, t, n, l, g, k, h, j, q, x, zh, ch, sh, z, c, s, r, –a, –o, –e, –I, –u, –v
韵母(38)	a, ai, an, ang, ao, e, ei, en, eng, er, o, ong, ou, i, i1, i2, ia, ian, iang, iao, ie, in, ing, iong, iou, u, ua, uai, uan, uang, uei, uen, ueng, uo, v, van, ve, vn

训练数据库包括 70 个男声数据, 约 36 400 句语音。测试数据库共有两个, 测试集 I 包括 2 个男声数据 1 000 句、约 75m in 的语音样本, 而测试集 II 包括 3 个男声数据 240 句、约 15m in 的语音样本。词表大小约 50 000 词, 在此基础上生成了两层词法数结构。目的是评测基于两层词法树的上下文相关模型的跨词搜索策略的性能和效率, 选取上下文相关模型的词内搜索策略作为对比实验, 均使用二元语言模型。下表中的识别率均是语音样本的汉字正确率。

从表 2 中给出的实验结果可以看出跨词搜索策略的误识率比词内搜索策略平均下降接近 60%, 在识别性能上有显著提高。这主要因为在跨词搜索算法当中, 在进行词与词之间匹配时使用的是上下文相关的声学模型, 可以精确地进行模型匹配; 而此时在词内搜索时使用的是粗糙的无相关声学模型, 所以识别性能差别很大。

表 2 识别性能比较

	识别性能/%		
	词内搜索	跨词搜索	误识率下降
测试集 I	78. 1	92. 3	64. 5
测试集 II	74. 5	88. 5	54. 9

从表 3 中的实验数据可看出, 静态两层词法树的内存消耗非常小, 而且在搜索时所需要动态内存的峰值也不是很大, 对于一个 50 000 词的连续语音识别系统来说也是很小的。从实验数据中还可以看出相比第一层词法树, 第二层词法树的内存消耗可以忽略不计。

表 3 内存消耗比较

	词内搜索策略/MB	跨词搜索策略/MB
第一层词法树	1. 86	1. 86
第二层词法树	0. 31	0. 46
搜索动态内存	4	12

表 4 是在一台 Pentium II 450MHz 的计算机上, 分别用两种算法对大约 90m in 的测试语音样本进行识别所需的时间。跨词搜索算法的时间消耗大约是词内搜索算法的 1. 5 倍。虽然需要的时间多出 50%, 但是仍然小于测试语料的实际时间, 可见该方法可以达到实时计算。

表 4 识别算法所需时间比较

	语音段时间总长度/s	词内搜索所需时间/s	跨词搜索所需时间/s
测试集 I	约 4 500	2 671	4 195
测试集 II	约 1 000	623	971
总和	约 5 500	3 294	5 166

4 结 论

本文为大词表连续语音识别提出了一个新颖的上下文相关模型跨词搜索算法。因为采用两层词法树描述着整个搜索网络。所以词法树的内存消耗很小, 解决了现有单层词法树的规模爆炸问题。实验结果证明, 与词内搜索算法相比, 误识率平均下降 60%, 所需内存资源较少, 搜索时间达到实时, 证明基于两层词法树的跨词搜索算法具有很好的识别性能和实时的识别效率。充分发挥了上下文相关声学模型较好地描述协同发音现象的能力。

参考文献 (References)

[1] Hwang M Y, Hon H W, Lee K F. Modeling between-word coarticulation in continuous speech recognition [A]. Proceedings of the ISCA European Conference on Speech Communication and Technology [C]. Paris, France: ISCA (International Speech Communication Association), 1989. 5-8.

[2] Ney H, Haeb-Umbach R. Improvements in beam search for 10000-word continuous speech recognition [A]. Proceedings of ICASSP1992 [C]. San Francisco, USA: IEEE Press, 1992. 9-12.

[3] Zhou Q, Wu C. An approach to continuous speech recognition based on layered self-adjusting decoding graph [A]. Proceedings of ICASSP97 [C]. Munich, Germany: IEEE Press, 1997. 1779-1782.

[4] Zhang J Y, Zheng F, Li J, et al. Improved context-dependent acoustic modeling for continuous Chinese speech recognition [A]. Proceedings of EuroSpeech'2001 [C]. Aalborg, Denmark: ISCA, 2001. 1617-1620.

[5] Odell J J. The Use of Context in Large Vocabulary Speech Recognition [D]. Cambridge: University of Cambridge, UK, 1995.

[6] Ortmanns S, Ney H. A word graph algorithm for large vocabulary continuous speech recognition [J]. *Computer Speech and Language*, 1997, **11**: 43-72.