第六届全国人机语音通讯学术会议，239-242 页，2001 年 11 月 20-22 日，深圳

基于两层词法树的大词表连续语音识别搜索算法

张国亮，郑方，吴文虎

清华大学计算机科学与技术系 智能技术与系统国家重点实验室
语音技术中心 100084 北京
[liang, fzheng, wuwh]@sp.cs.tsinghua.edu.cn, http://sp.cs.tsinghua.edu.cn

摘要

在连续语音识别中，首先要考虑的就是词典的表示问题，它关系到整个搜索空间的规模和搜索算法的效率。在现有的词典表示方法中，最通用的就是树形词典表示方法，也称为词法树(Lexical Tree)[1]。但是在词法树构成中，因为要将所有的搜索空间都体现在词法树上，所以在使用跨词(Cross-Word)模型、多发音词典时就会出现词法树规模超大的问题需要解决[2]我们从搜索空间组成的角度入手，提出了两层词法树的概念，解决了现有的词法树规模问题。其中第一层词法树重点描述词网络信息和基元网络信息，而第二层重点描述实际动态规划搜索中的各种信息。因为声学信息都很好地在第二层中实现了共享，所以词法树的规模很小，使得它有足够的能力来处理很多复杂的问题。另外，本文还给出了完整的基于两层词法树的动态规划搜索算法，从实验结果可以看出，这种搜索算法具有很好的识别性能和效率。

1. 引言

一个基于统计方法的语音识别系统通常由以下几个模块组成：隐含马尔科夫模型(HMM)、发音字典、搜索网络、语言模型和搜索算法。其中，搜索算法处于关键位置，它决定了如何使用其它的各种资源。所以，为了构建一个优秀的识别系统，必须研究出好的搜索算法将各个模块有效并且高效地结合在一起。

近年来，在大多数大词表连续语音识别系统中，词法树都被用来描述搜索网络以减少计算复杂度[1]。而且，词表越大计算复杂度减少的比例越大。但是在一些新提出的应用中，传统的词法树遇到了困难，例如：

第一，目前上下文相关(context dependent)的 triphone 模型因为能够较好地对协同发音现象进行建模而得到了广泛应用。但是，如果用静态词法树来表示跨词的上下文相关 triphone 模型，会使得词法树的规模急剧膨胀，非常庞大。而且，当将语言模型也集成进词法树后，完整的基于词的词法树[3]就会变得更加复杂。在现在的很多系统中，词法树都有不止一个的起始节点和终结节点，在它们之间还有很多循环弧进行连接[4]，这样的词法树规模很庞大而导致系统效率的降低，而且也缺乏可扩展性。

第二，在很多中文连续语音识别系统中都采用了模糊音集以提高系统处理说话人方言的鲁棒性[5]。在模糊音集中，很多音节都直接被映射为另一个音节，例如：“zhi→ji”，“guo→gui”。当将模糊音信息也要在词法树中描述时，就会发现每一个词中的任何一个有模糊音映射关系的音节都要扩展出一条新的分支。这样最终导致词法树的规模随模糊音的个数成几何关系增长，当模糊音比较多时，词法树的规模就会无法忍受，而且这时的词法树有很大的容余性。

第三，发音变化建模因为其较高的鲁棒性而在语音识别领域显得越来越重要[6]。在中文语音识别系统中，发音变化建模的基元一般为三种：音节、半音节和音素。但是无论用哪种基元进行发音变化建模都会遇到和上一个应用相类似的问题：为了描述多发音变化，词法树中每一个词都会扩展出多个分支，静态词法树的内存消耗增长迅速以至于无法接受。

在本文中，将提出一种新颖的两层词法树结构来解决以上的问题。其中第一层直接由传统的单层词法树而来，不用大的修改，其余的各种信息都存储于第二层中。因为在第二层中，在不同词的相同识别基元的信息都存储在一个共享单元中，所以，整个词法树的内存占用非常小，即对搜索空间的组织更加精巧。众所周知，对搜索空间的组织是动态规划中的核心问题，所以基于两层词法树的搜索算法更为高效。两层词法树的另一个优点就是拥有非常好的可扩展性，将两层词法树集成进一个新的语音识别系统所需要做的工作仅仅是构建第二层词法树，这是非常容易的。而且，如果想再集成进其它的应用，也只需要修改第二层词法树，而不必对其它的各个模块进行修改。

本文也给出了基于两层词法树的搜索算法，他非常类似于经典的时间同步的 token-passing 算法[7]。当考虑语言模型后，基于两层词法树的搜索算法在我们的大词表汉语连续语音识别系统中取得了很好的效果。在文章的后部给出的实验结果显示这种算法是一个高效的搜索算法。

文章的其余部分以如下方式组织：在第二节将给出两层词法树的体系结构。在第三节介绍基于两层词法树的搜索算法。在第四节中，我们详细地描述两层词法树在处理上下文相关模型跨词搜索时的具体实现。最后，在第五节中给出一些实验结果，第六节给出结论。

2. 两层词法树结构

在大词表连续语音识别中，因为搜索策略中要用到很多种相关的知识源，所以搜索网络往往非常复杂。这不但使搜索网络的动态扩展很难实现，而且使当搜索过程中需要修改某一局部的搜索网络时无法定位。根据表述的知识源不同，搜索网络被划分为三个层次：词关系网络层、基元关系网络层和动态规划网络层[8]。词关系网络层主要功能为描述词一级的关系，如

语法规则和语言模型限制；基元关系网络层主要描述识别基元一级的知识关系，如音素词法图；动态规划网络层则主要描述实际动态规划运行时的关系，由所有的知识源共同决定。为了保留搜索空间的层次性，本文采用了一个两层词法树来表示整个搜索空间，其中第一层反映词关系网络层和基元关系网络层，第二层仅仅影响动态规划网络层。

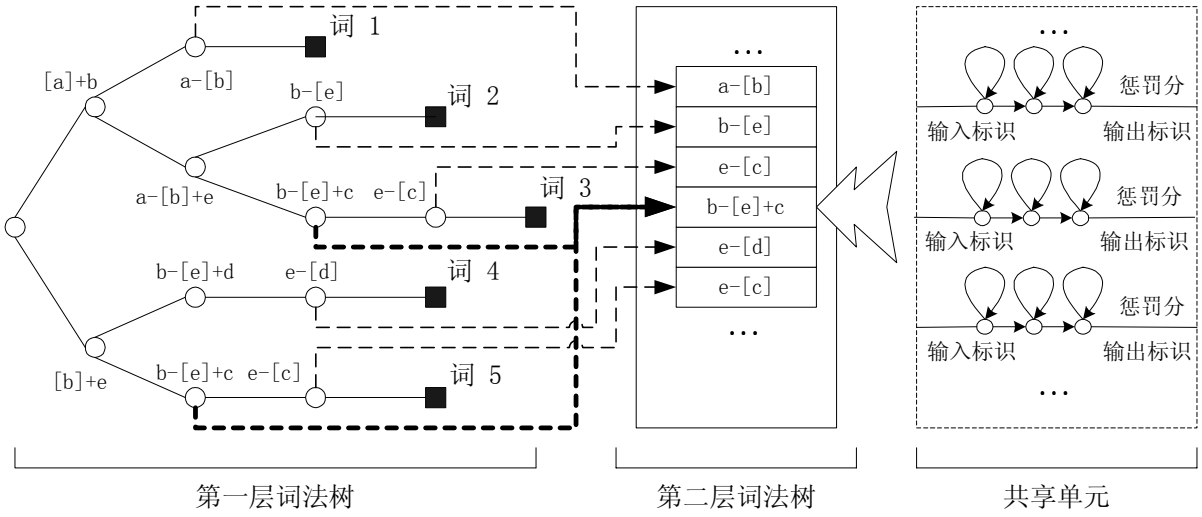

图 1. 两层词法树示意图

（第一层中的每个叶子节点都包含着整个词的信息，如从根节点到其父节点所有所经过识别基元的路径）

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，所以第一层词法树与声学模型独立且不包含任何时间信息。从简单易行的角度出发，第一层词法树可以从传统的词法树中继承而来，因为传统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传统词法树中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的获取所需要的声学模型信息和时间信息。

第二层词法树用于描述动态规划网络，所以第一层中没有涉及到的所有信息都将体现在第二层中。第二层词法树是一系列共享单元的数组，共享单元之间相互独立，没有必然的联系。共享单元的数目和第一层词法树中出现的识别基元的个数一样多，且一一对应。关于这个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，这样使得大量的信息都共享在一起，大大减少了所需的存储空间。这点从后面给出的实验数据中可以体现出来，第二层所需要的存储空间和第一层相比完全可以忽略不计。图 1. 是两层词法树的示意图，从中可以清楚地看出两层词法树的数据结构和相互关系。

一个共享单元由多条相互独立的平行弧组成，每一个弧都代表路径扩展的一个可能路线，在搜索令牌(Token)中弧和路线一一对应。为了描述一条弧上的所有信息，弧被赋予了四个域：声学模型指针、弧惩罚分、进入标识和输出标识。声学模型指针记录了声学模型信息，在识别时可以直接通过指针找到所需要的声学模型；弧惩罚分(penalty)是为了实现对每一条弧加权的策略，这样每一条弧都有着不同的惩罚分，这将在不同的应用中体现不同的实际意义；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。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，同理右相关属性是由中心基元标识和右相关基元标识共同组成。当使用的声学模型是上下文无关模型时，进入标识和输出标识可以忽略不计。

两层词法树的生成非常方便易行。因为传统的词法树可以很简单地转化为第一层词法树，而且第二层词法树是一个共享单元的数组，只需对在第一层词法树中出现的所有识别基元建一个共享单元即可。构建共享单元的步骤如下：

1. 用共享单元所代表的原始的上下文相关模型生成第一条弧；
2. 考虑中心基元的所有可能变化。从第一条弧复制出一条新弧，在新弧的属性中修改与中心基元标识有关的域，并修改惩罚值；
3. 考虑左相关基元的所有可能变化。复制所以已经生成的弧，修改新弧中与左相关基元标识有关的域，并修改每条弧的惩罚值；
4. 考虑右相关基元的所有可能变化。复制所以已经生成的弧，修改新弧中与右相关基元标识有关的域，并修改每条弧的惩罚值；

最终，一个已经生成的共享基元包括很多条弧，这些弧覆盖了当前基元所有可能的发音变化。在这里需要阐明，在搜索过程中，我们将采用一条路经来记录下在每一条弧中扩展的当前状态。

3. 基于两层词法树的搜索算法

在两层词法树的框架中，虽然第一层还是一个发音前缀树，但是第二层被分离开了，即它们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。

我们采用基于时间同步的 Token Passing 算法作为基准算法。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共享单元的弧上得到。

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三个部分。为了更好地说明这种动态规划算法的整个过程，先定义下面两个变量：

- $Q_v^p\left(t, s_y^x\right) :=$ 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；
- $B_v^p\left(t, s_y^x\right) :=$ 路径 $Q_v^p\left(t, s_y^x\right)$ 的起始时刻；

第一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：

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

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

在公式(1)中，$q\left(x_t, s_y^x \mid \sigma_y^x\right)$ 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。

第二步处理在一个词内部，但在基元模型间的路径扩展，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。

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

$$B_v^p\left(t-1, s_*^x = 0\right) = B_v^{\max(f)}\left(t-1, \sigma_x^{\max(z)} = end\right) \quad (4)$$

在公式(3)和(4)中，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。

第三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。整个过程可见下面公式。

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

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

因为词网络关系层和基元网络关系层都体现在第一层词法树中，所以在一个搜索令牌中的所有路线都可以共享相同的语言模型得分。我们这里仅仅使用二元语言模型(Bigram)来描述整个搜索算法，但是三元语言模型(Trigram)可以很容易地被结合进算法中。所以两层词法树即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[9]。

4. 跨词搜索的实现

在本节我们给出用两层词法树来解决上下文相关模型跨词搜索问题的具体实现方法，用两层词法树来解决模糊音问题和发音变化建模问题的实现方法非常类似，区别只限于第二层词法树不同。

4.1. 语音识别基元

声韵母结构是汉语仅有的一种特性，绝大多数汉语音节都由一个声母和一个韵母构成，只有少数音节仅仅由韵母构成。在本文所进行的实验中，上下文相关的扩展声韵母(XIF)集被选作基本的识别基元，在扩展声韵母集中，包含 27 个声母和 38 个韵母，其中有 6 个零声母是新加入的[10]。扩展声韵母集见表 1。

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*

4.2. 两层词法树

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。在我们的系统中，一个词是由三部分组成：一个右相关 XIF、几个上下文相关 XIF 和一个左相关 XIF。为了在第二层中描述上下文相关模型的跨词搜索信息，对应于右相关 XIF 的共享单元中包括多条弧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，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。具体的例子可见图 2。

5. 实验

所有的实验数据都基于一个汉语普通话听写系统“Easytalk”而得到。声学模型的训练集和测试集都从 863 数据库中得到[5]，863 数据库共有 80 人的男生数据，每一个人有 520 句语音样本。所有的语音都是通过有去噪功能的麦克风在低噪音的环境下录制而成，采样率为 16kHz。我们对每帧语音提取 42 维特征参数，由三部分组成：13 维的 MFCC 和 1 维对数能量，MFCC 和对数能量的自回归参数，自回归参书的一阶差分。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[11]。

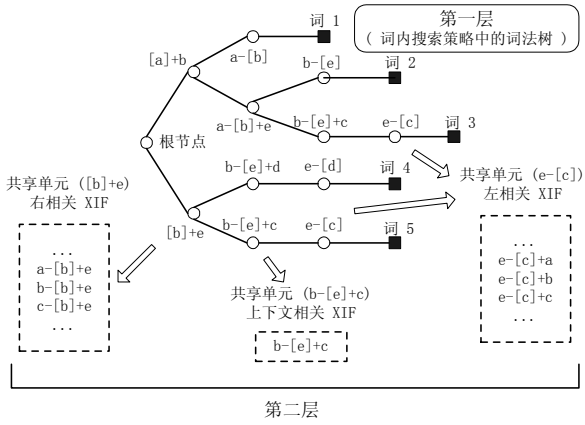

图 2. 跨词搜索的具体实现

训练数据库包括 70 个男声数据，约 36400 句语音。测试数据库共有两个，测试集 I 包括 2 个男声数据约 1000 句语音，而测试集 II 包括 6 个男声数据约 240 句语音。系统的词表大小为 50000 词，平均每个词有 4.8 个 XIF 组成，在此基础上生成了两层词法数结构。在本次实验的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(cross-word context dependent XIF)的性能和效率，同时还选取上下文相关模型的词内搜索策略(intra-word context dependent XIF)作为对比实验。在本实验中，没有考虑发音变化。

表 2: 识别性能比较

	词内搜索策略	跨词搜索策略	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%

首先，我们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，与对比搜索策略相比较。从表 2 中给出的实验结果我们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%，在识别性能上有了巨大的提高。

表 3: 内存消耗比较

	词内搜索策略	跨词搜索策略
第一层词法树	1.86MB	1.86MB
第二层词法树	0.31MB	0.46MB
搜索时的动态内存	4MB	12MB

从表 3 中的实验数据我们可以看出，静态两层词法树的内存消耗是非常小的，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 50000 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完全可以忽略不计。

上下文相关模型跨词搜索的时间消耗大约是上下文相关模型词内搜索的 2 倍。如果再加入比较好的剪枝策略，是可以做到更为高效。

6. 总结

本文中，为大词表连续语音识别提出了一个新颖的两层词法树结构及相应的搜索算法。两层词法树描述着整个搜索网络。词法树的第一层反映的是词网络和基元网络，第二层反映的是动态规划层的信息。因为在不同词中的相同的识别基元的信息都存储在第二层词法树的同一个共享单元中，所以词法树的内存消耗很小以至于可以方便地处理很多复杂的应用，例如上下文相关的跨词搜索、中文模糊音映射和发音变化建模等等。两层词法树还有很好的兼容性，它可以很方便地集成进一个新的系统。最后，本文还给出了用两层词法树来处理上下文相关模型跨词搜索的具体解决实例，实验结果证明基于两层词法树的搜索算法可以取得非常好的性能和高的效率。

7. 参考文献

[1] Ney.H, Haeb-Umbach.R, “Improvements in beam search for 10000-word continuous speech recognition”. *ICASSP1992,* San Francisco, Vol1, pp9-12.

[2] Zhang, G-L, Zheng, F, and Wu, W-H, “A Two-Layer Lexical Tree Based Beam Search in Continuous Chinese Speech Recognition” *EuroSpeech2001,* Aalborg, Vol3, pp1801-1804

[3] Ortmanns.S, Ney.H, “A Comparion of Time Conditioned and Word Conditioned Search Techniques for Large Vocabulary Speech Recognition”. *Proceedings of ICSLP1996*, Philadelphia, pp.2091-2094

[4] Gauvain.J.-L, Lamel.L, “Speaker-independent continuous speech dictation”. *Speech Communication,* 15:21-37, October 1994.

[5] Zheng, F., Song, Z.-J, and Xu, M.-X, “EASYTALK: A large-vocabulary speaker-independent Chinese dictation machine”, *EuroSpeech’99*, Vol.2, pp.819-822, Budapest, Hungary, 1999

[6] Zheng, F., Song, Z.-J., Fung, P., Byrne, W., “Mandarin Pronunciation Modeling Based on CASS Corpus,” *Sino-French Symposium on Speech and Language Processing,* pp. 47-53, Oct. 16, 2000, Beijing

[7] Odell.J.J, “The Use of Context in Large Vocabulary Speech Recognition”. *PhD thesis*, University of Cambridge, U.K., March 1995

[8] Zhou.Q, Wu.C, “An Approach to Continuous Speech Recognition Based on Layered Self-Adjusting Decoding Graph”. *Proceedings of ICASSP97,* V3, p1779-1782

[9] Ortmanns.S, Ney.H, “A Word Graph Algorithm for Large Vocabulary Continuous Speech Recognition”, *Computer Speech and Language,* 1997, 11, p43-72

[10] Zhang, J-Y., Zheng, F., Li, J., Luo, C-H., Zhang, G-L., “Improved Context-Dependent Acoustic Modeling for Continuous Chinese Speech Recognition”, *EuroSpeech’2001*, Vol3, pp1617-1620

[11] Yong, S., Kershaw, D., Odell, J., Ollason, D., Valtchev, V. and Woodland, P., The HTK Book (for HTK Version 2.2), Cambridge University, 1999