摘 要

关键词检出技术就是从连续的语音流中检测并识别出表征预定义关键词表中单词的语音段的一种技术。本文讨论的关键词检出技术基于概率统计方法的语音识别技术。一个完整的关键词检出系统应该包含三大模块，分别是声学模型、识别模块和后处理模块。其中声学模型的训练虽然不在关键词检出系统的研究范围内，但是却是必不可少的部分；识别模块主要研究的是语音段的发现和对齐的问题，若采用连续语音识别的方法，还应该考虑剪枝的策略；后处理模块主要是通过设计一种置信度方法，对识别阶段的输出结果给出置信分数，也是给出衡量一个关键词检出系统性能参数的模块。

本文研究的重点是可定制的中文关键词检出系统，研究分为以下几个方面：

基于上下文相关的扩展声韵母（eXtended Initial/Final）的中文语音基元的建模和利用决策树对模型规模的限制。通过上下文相关的扩展声韵母，有效的解决了可定制词表的关键词识别系统的实现问题；利用中文语音的先验知识，采用决策树方法对模型的状态和参数进行了共享，有效的限制了大词表关键词检出系统中的模型数量膨胀的问题。

提出了 N-Best 的多条路径决策的不匹配帧加权的置信度方法，并作为关键词检出的后处理部分的实现。普通的基于驻留归一化的方法无法利用 N-Best 路径的决策信息，而部分采用 N-Best 的置信度方法又无法详细刻划连续语音中关键词附近的识别效果，采用综合 N-Best 多条路径的信息并利用不匹配帧描述关键词附近的信息以进行路径得分的加权的置信度的方法很好的解决了这个问题。

关键词：语音识别，关键词检出，置信度，基于 N-Best 的不匹配帧加权

Abstract

Keyword spotting (KWS) is the technology extracts and identifies pre-defined keywords from streams of speech signal, and this paper only deal with the keyword spotting system within the framework of statistical speech recognition. An applicable keyword spotting system should include mainly three functional elements: the acoustic models, the recognizer, or said the decoding procedure and the post-process procedure. Although the training of acoustic models is beyond the discussion of building a keyword spotting system, it plays a vital role in the recognition process and is indispensable. The recognizer searches through the trellis in order to find the best alignment between given feature sequence and HMM states sequence then outputs the best path(s) with the highest likelihood scores, while the post-process procedure uses some kind of confidence measure to create a metric for the verification of the result.

This paper mainly focuses on the building of customizable Chinese keyword spotting system, and my work can be divided into mainly two parts described below:

The utilization of decision tree in organizing the modeling of extended Initial/Final not only overcomes the explosive expansion of model space when realizing a large vocabulary Chinese keyword spotting system, but also satisfies the need in keyword customization without any model re-training.

This paper also proposes a confidence measure called N-Best based mismatch frames proportion weighting for the keyword spotting system in which combines the probabilities of N-Best viterbi paths and emphasizes the recognition quality of the keyword among the whole utterance.

Key words: speech recognition, keyword spotting, confidence measure, N-Best based mismatch frames proportion weighting

目录

第一章 绪论

语音识别就是将数字化的语音流转化为有意义的符号的过程。语音识别技术可以应用于许多领域，比如语音拨号，自动语音台，语音听写，语音指令控制，自动监听等。由于目前对人类听觉感知原理的认知尚未得到进一步的发展，语音识别技术无法完全获得仿生学上的理论支撑，相反的，目前比较主流的做法则是寻找一种比较好的声学感知特征来刻画多变的语音形态背后的相对稳定的特征，并利用统计学的办法对语音信号和模型进行匹配。本章首先介绍语音识别的原理，然后简单介绍关键词识别的应用，最后是对本文结构的描述。

1.1 语音识别的基本原理

目前的语音识别研究本质上就是一个基于统计的模式分类的问题。假设有类的集合 $\Omega=\{w_1,w_1,\cdots,w_c\}$，一个模式则由一个特征的集合来描述，唯一的属于某一个类。这个特征的集合通常由特征矢量组成，一个特征矢量的各维由某一类特征组成，可以表示为 $X=\{x_1,x_1,\cdots,x_d\}$。需要注意的是特征矢量与类并不是一一对应，比如，不同的类可能对应于同一个特征矢量，而不同的特征矢量可能属于同一个类。于是特征矢量与类应当看做随机变量。模式分类的过程就是将特征矢量映射到某一个类上。这个映射函数也称为决策函数，可以写成 $g:\Re,d\to\Omega$。于是问题转化为寻找一种最小分类错误的决策函数 g 的过程。

错分率可以定义为将特征矢量错分的概率：

$$L(g)=P\left(g(x)\neq w(x)\right) \tag{1.1}$$

由于特征矢量与类并非一对一映射，所以可以取得分类错误概率的下界，也就是所说的贝叶斯错误率，记为 L^*，而相应的决策函数称为贝叶斯分类器，记为 g^*。于是，根据最大后验概率准则，有：

$$g^*(x)=\arg\max_i P(w_i\mid x) \tag{1.2}$$

然后，根据贝叶斯公式，有：

$$P(w_i\mid x)=\frac{P(x\mid w_i)P(w_i)}{P(x)} \tag{1.3}$$

代入公式 1.2，有：

$$g^*(x) = \arg\max_i \frac{p(x|w_i)p(w_i)}{p(x)} = \arg\max_i p(x|w_i)p(w_i) \tag{1.4}$$

其中 $p(x)$ 相当于一个独立参数被忽略了。语音识别就是通过估计 $p(x|w_i)$（也称为声学概率）和 $p(w_i)$（也称为语言模型）来计算 $g^*(x)$ 的。在语音识别中，声学事件的统计特性就是由声学模型来描述的，在基于 HMM 的语音识别里，我们假设每一个字（这里应该理解为语音建模的最小基元）对应的观察向量序列都是由一个马尔科夫链产生的。由图 1.1 所示，一个 HMM 就是一个离散时域的有限状态自动机，随着每一帧语音而改变状态，并且在 t 时刻到达 j 状态的时候，以 $b_j(x_t)$ 的概率产生观察向量 x_t。而自动机的转移特性由转移概率 $a_{ij} = \Pr(q_t = j | q_{t-1} = i)$ 来描述，其中 q_t 代表 t 时刻的模型的状态。一般来说，每一个 HMM 还包含一头一尾两个非发射（non-emitting）状态，用以描述语音向量产生和结束的时刻。

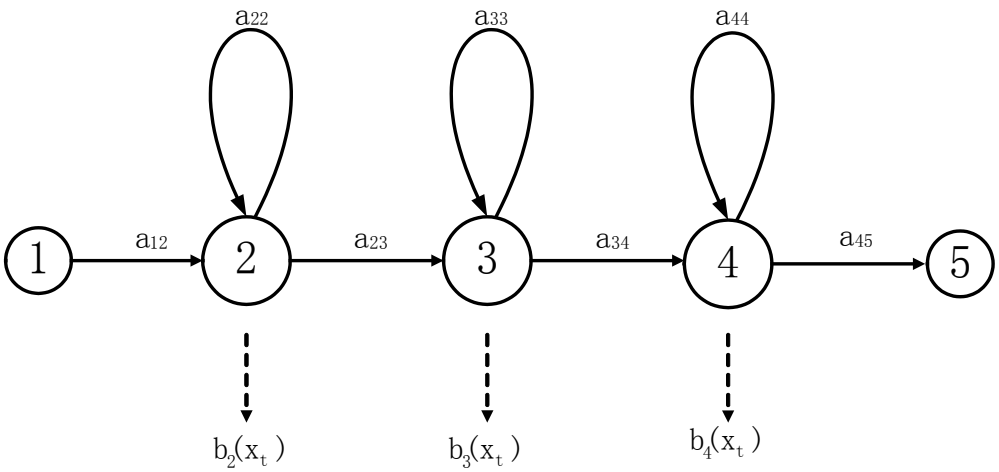

图 1.1 一个 5 状态从左至右 HMM 示意图

图 1.1 描述的就是一个 5 状态从左至右的 HMM 模型。该模型有 5 个状态，其中 2 个非发射状态，3 个发射状态；每一个发射状态仅能够向右边或自己跳转。对某一个状态 i 来说，永远满足：

$$\sum_{j=1}^{N} a_{ij} = 1 \tag{1.5}$$

而发射概率 $b_j(x_t)$ 则用来描绘观察向量在 j 状态的空间分布。在离散 HMM（Discrete HMM, DHMM）模型下，该发射概率是一个多项式分布，而在连续 HMM（Continuous Density HMM, CDHMM）模型下，则是用高斯混合密度（Gaussian Mixture Density, GMD）来表征。所谓高斯混合密度，其实就是多维高斯分布的加权和，也即是：

$$b_j(x_t)=\sum_{m=1}^{M}c_{jm}\cdot N\left(x_t;\mu_{jm},\sum_{jm}\right) \tag{1.6}$$

其中 $N\left(x_t;\mu_{jm},\sum_{jm}\right)$ 就是高斯分布的密度，μ_{jm} 是语音向量的期望，而 $\sum_{jm}$ 是一个用协方差矩阵表示的方差。依照高维高斯分布（也称为高维正态分布）的公式，不难得到：

$$N\left(x_t;\mu_{jm},\sum_{jm}\right)=\frac{1}{(2\pi)^{\frac{D}{2}}\left|\sum_{jm}\right|^{\frac{1}{2}}}e^{-\frac{1}{2}(x-\mu_{jm})^T\sum_{jm}^{-1}(x-\mu_{jm})} \tag{1.7}$$

其中 D 表示语音向量的维度。一般来说，每一个发射概率的密度函数都描述一个声音事件，所以我们希望不同的概率密度要尽可能的具有区分性，以便于系统对声学特性具有相对的鲁棒性。

对于一个模型参数已知的系统来说，相当于给定了 a_{ij} 和 $b_j(x_t)$，其中 0<i<N; 0<j<N，观察向量序列 $X=\{x_1,x_1,\cdots,x_T\}$ 的概率可以计算为：

$$p(X\mid a;b)=\sum_{q}p(X,q\mid a;b) \tag{1.8}$$

其中 $q=\{q_1,q_1,\cdots,q_T\}$ 代表观察向量 X 对应的模型可能的一条状态序列。并且，$p(X,q\mid a;b)$ 可以计算为：

$$p(X,q\mid a;b)=\prod_{t=1}^{T}b_{q_t}(x_t)\cdot a_{q_{t-1}q_t} \tag{1.9}$$

上式中 $a_{q_0q_1}$ 理解为模型初始处于 q_1 状态的概率，而 $a_{q_Tq_{T+1}}$ 等于 1，所以为了计算和表达上的方便，通常加上首尾两个非发射状态。

实际应用中，将公式 1.8 计算出来十分困难，因此常常采用观察向量的序列与最大可能路径的联合分布来估计，表示为：

$$p(X\mid a;b)\approx\max_{q}p(X,q\mid a;b) \tag{1.10}$$

目前经常采用的前向后向算法和维特比算法都用于在所有 HMM 模型的状态和时间帧构成的栅格上计算满足公式 1.10 的路径，该步骤实际上也就是一个对齐，或者说网络搜索的过程，本文 1.3 节将会详细介绍这个过程。

如图 1.2 所示，对于一个大词表的连续语音识别系统（Large Vocabulary Continuous Speech Recognition, LVCSR）来说，通常是分层构建的，一个 HMM 对应的最小单位通常是音子（Phone），然后由音子所对应的 HMM 首尾相连组成词，以此类推最终一句话则由词典内的词汇组成句子。

语音识别按照任务的复杂程度和规模，依次可以分为如下四类：

(1) 孤立词识别：整个待识别语音段代表识别系统词表内的一个单独的词汇。

(2) 连接词识别：整个待识别语音段的内容是由识别系统词表内的单词首尾相连而成的。

(3) 关键词检出：预定义的关键词词表内的单词有可能出现在待识别语音段的任何位置。

连续语音识别：整个待识别语音段代表一个连续的句子，其中每一个单词都属于预定义的词表。

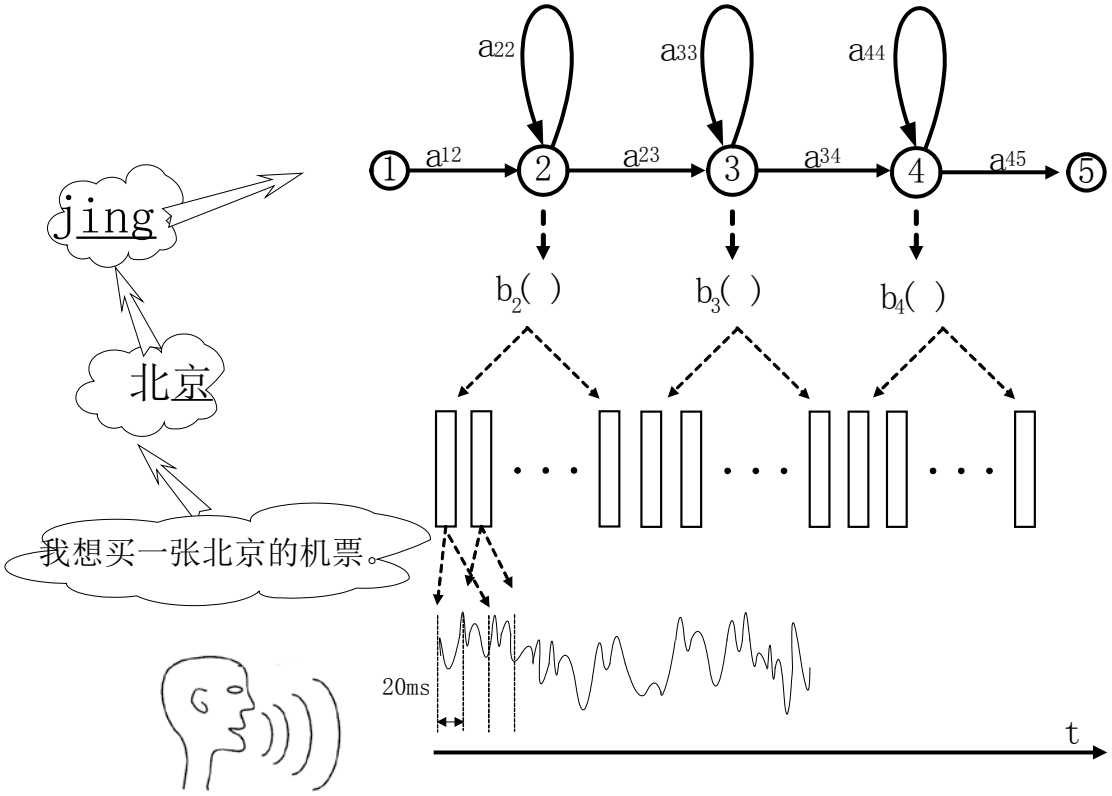

图 1.2 一个连续语音识别系统的示意图

由于硬件的限制，问题的规模大小直接影响到所使用的算法，所以，以上几种问题所采用的算法通常都有较大的差别。对于孤立词识别来说，由于可以确定语音段所代表的词汇的边界，所以只需要找到一种比较好的模板匹配的方法就可以了。若词表大小约束在一个比较小的范围，比如 20 个词汇以内，则只需要对整个单词建模，在识别的时候将语音与模型计算匹配程度便可以完成识别，即匹配程度最高的模型所代表的单词便是识别结果。连接词识别多用于数字串的识别，当串长固定的情况下，可以采用分层构筑[1]的方法。关键词识别与连续语音识别面临的困难比较相似，同样是面对比较大的词表，通常采用子词（Sub word）的方法以有限的模型

来组成词汇，而子词作为构词的最小单元，通常为音子（Phone），在中文语音识别任务中通常采用声韵母（Initial/Final）作为构词的方法。对子词的训练和识别通常都采用基于隐马尔科夫模型（Hidden Markov Model, HMM）的统计方法，按照贝叶斯原理，取在最大后验概率得分的模型状态序列作为声学特征矢量的识别结果。

一个完整的连续语音识别系统应该包括端点检测、特征提取、网络搜索和置信度计算三个模块，后面依次介绍这几个模块的工作原理。

1.2 语音识别的工作流程

1.2.1 端点检测

端点检测也称为 VAD（Voice Activity Detection），是确认语音有效信息的起止的方法，较为普通的做法是对语音信号的过零率（Zero-Crossing Rate, ZCR），短时平均幅值和短时能量等时域参数采用一定的策略来摒弃首尾两端的非语音信息。

若设 N 为窗宽度，$S_t(n)$表示第 t 帧的第 n 个采样点信息的话，则对以上时域参数，我们有以下定义：

(1) 过零率 ZCR(t):

$$ZCR(t)=\sum_{n=0}^{N-1}\frac{1}{2}\left[Sgn\left(S_t(n)\cdot S_t(n-1)\right)+1\right] \tag{1.11}$$

其中 $Sgn(x)$为普通的符号函数，定义为：

$$Sgn(x)=\begin{cases}1 & x>0\\ -1 & x<0\end{cases} \tag{1.12}$$

(2) 短时平均能量 Eng(t):

$$Eng(t)=\frac{1}{N}\sum_{n=0}^{N-1}\left|S_t(n)\right| \tag{1.13}$$

端点检测的方法就是利用诸如以上时域参数或者自相关方法，以及一些包括削波等方法，确定语音段在时间上的起迄。由于不是本文讨论的重点，概不详述。

1.2.2 声学特征的选择和计算

语音识别的最前端就是将语音信号数字化并转化为声学特征的过程。第一步首先需要进行 A/D 转换。声学信号（表现为随时间变化的声压信号）由麦克风等设备采集得到（转为电压等模拟信号），经过采样和量化，完

成 A/D 转换。采样就是记录下某个时间点上的振幅，采样率反映的是每秒钟采样的次数。显然，要想表示一个信号，至少每个周期有两次采样，分别在波峰和波谷，而采样的次数越高，显然原信号丢失的细节越少，得到的采样信号还原度越高。显然，波形信号的最大频率最多只能是采样信号的 1/2，这个也称为奈奎斯特频率。人耳只能辨别大约 10,000Hz 以下频率的信号，所以 20,000Hz 的采样率可以保证足够的精度；而电话交换网络只支持 4,000Hz 以下的信号，所以例如 Switchboard 这样的语音库的采样率只有 8,000Hz，而麦克风采集的信号通常为 16,000Hz 采样率。量化就是将采样得到的连续幅值用一定范围的整数来表示，比如 8 位的位宽（Resolution，也称为位率，采样精度等）就是-128~127，而 16 位的位宽则是-32768~32767。所以在计算机中可以用 1 个字节来表示 8 位位宽的采样点，或者用 2 个字节来表示 16 位位宽的采样点。尽管如此，在短时分析中，20ms 左右的 16KHz、16bits 的语音共包含 320 个采样点，占用 640 个字节。因此特征提取的作用就是压缩冗余数据，提取特征信息的过程。目前最为主流的 MFCC（Mel-frequency cepstrum coefficients）特征[2]提取的过程如下：

(1) 预加重

由于人的声道具有低通特性，大部分能量集中在低频范围内。这就造成消息信号高频端的信噪比可能降到不能容许的程度。所以为了加强高频共振峰的信息，需要对高频信号能量进行提升。预加重可以在 A/D 转换之前或者之后进行，用具有 6d/B 倍频程的数字滤波器来实现：

$$Z = 1 - 0.97 \cdot Z^{-1} \tag{1.11}$$

时域上可表示为：

$$y_1(n) = s(n) - 0.97 \cdot s(n-1) \tag{1.12}$$

其中 0.97 也称为预加重因子。

(2) 分帧

由于我们进行声学特征提取的任务就是为了分析语音的频谱特性，以便构造我们的分类器，因此有必要对非平稳的语音信号进行一定的假设，就是认为语音信号在一个比较短的时隙内，比如 10 毫秒到 40 毫秒的范围内可以近似看做平稳信号。于是，我们需要用一个窗函数对语音信号进行逐帧的处理。这个过程也叫做加窗，其中，采用什么形状的窗，窗的长度和前后两次加窗过程之间的偏移量都是需要预先决定的。目前一般采用所谓的 hamming 窗函数与原信号在时域上相乘完成语音信号的分帧：

$$y_2(n) = w(n) \cdot y_1(n) \tag{1.13}$$

其中 $y_1(n)$来自预加重的过程。而 hamming 窗函数 $w(n)$定义为：

$$w(n)=\begin{cases}0.54-0.46\cos(\frac{2\pi n}{L}) & 1\le n\le L-1\\ 0 & \text{其他}\end{cases} \tag{1.14}$$

L 表示窗长。hamming 窗比普通的矩形窗具有更平滑的低通特性，可以有效的克服泄漏现象。

(3) 离散傅里叶变换

进行离散傅里叶变换（Discrete Fourier Transform, DFT）是为了提取语音信号中的频谱信息:

$$X[n]=\sum_{n=0}^{N-1}y_2[n]e^{-j2\frac{\pi}{N}kn} \tag{1.15}$$

实际应用中一般采用补零的方法运用快速傅里叶变换进行计算。

(4) 求取 MFCC 倒谱系数

由于人耳对高于 1000Hz 的声音分辨较不敏感，所以为了模拟在低于 1000Hz 时候的线性变化和高于 1000Hz 以后对数特性的听觉边界，将实际频率弯折到 Mel 刻度上：

$$mel(f)=1127\ln\left(1+\frac{f}{700}\right) \tag{1.16}$$

按照 Mel 刻度把频率分为 K 等分，构造 K 个三角滤波器，其中相邻滤波器的下限频率 F_o、中心频率 F_c 和上限频率 F_h 满足：

$$F_c(k)=F_h(k-1)=F_o(k+1) \tag{1.17}$$

接下来利用 K 个三角带通滤波器分别与离散谱做卷积求出每个频段的对数能量后利用离散余弦变换（Discrete Cosine Transform, DCT）：

$$C_{mfcc}(d)=\sum_{k=1}^{K}X(k)\cdot\cos\left(\left(k-\frac{1}{2}\right)\cdot\frac{d\pi}{K}\right),d=1,...D \tag{1.18}$$

其中 d 表示 mfcc 参数的第 d 维。

必须说明的是，一般的特征矢量能量信息。能量维的计算有：

$$C_{mfcc}^{E}(d)=\sum_{n=1}^{N}y_2^{\,2}[n] \tag{1.19}$$

若在公式 1.18 中 D=12，则目前为止我们得到一个 13 维的特征矢量，然而，为了表征语音的动态特性，我们也常常使用 mfcc 的一阶差分甚至二阶差分，差分运算的一种比较简单的做法就是计算 t+1 帧和 t-1 帧的倒谱系数的算术平均。于是，比如一个 39 维的特征矢量对应于 13 维的向量

以及它的一阶和二阶差分。

值得说明的是，一般还有一些做法是对 mfcc 向量做一些变换以期望特征矢量各维有比较好的统计独立性，使得稍后在计算发射概率的时候可以用对角阵来近似模型的协方差矩阵，加速计算。

1.2.3 网络搜索

网络的搜索也称为解码，是识别器的主要研究方向，目的就是给定一个特征矢量的序列和一个 HMM 模型的集合，如何找到一个 HMM 模型的状态序列，使得在该序列上，此特征矢量序列有最大的概率。比较著名的时间异步搜索算法有栈解码（Stack Decoding）[3]和 A*算法[4]；比较著名的时间同步算法有维特比算法（Viterbi Algorithm）[5]等。为了高效的得出搜索结果，常常还采用两遍搜索，对搜索路径进行剪枝等。由于孤立词识别和连续语音识别在解码算法上有较大不同，在此分开描述。

(1) 孤立词识别

孤立词识别就是一个基于最大后验概率（Maximum a Posteriori, MAP）的问题。简言之，对于一个给定的观察向量 $X = x_1 x_2 \ldots x_T$，找到一个模型 w_i，使得概率 $P(w_i \mid X)$具有最大值：

$$w_i = \arg\max_i \{P(w_i \mid X)\} \tag{1.20}$$

由于直接计算 $P(w_i \mid X)$是不可能的，但是根据贝叶斯公式，我们有：

$$P(w_i \mid X) = \frac{P(X \mid w_i) P(w_i)}{P(X)} \tag{1.21}$$

由上式可以知道，对于给定的模型的先验概率 $P(w_i)$来说，问题转化为估计联合条件概率 $P(x_1 x_2 \ldots x_T \mid w_i)$。若假定了语音信号的一阶马尔可夫性，则 $P(x_1 x_2 \ldots x_T \mid w_i)$可以近似计算为：

$$\begin{aligned} P(x_1 x_2 \ldots x_T \mid w_i) &= P(x_1) P(x_2 \mid x_1) P(x_3 \mid x_2) \cdots P(x_T \mid x_{T-1}) \\ &= \sum_q \prod_{t=1}^{T} b_{q_t}(x_t) \cdot a_{q_{t-1} q_t} \end{aligned} \tag{1.22}$$

由于将所有的状态序列都求出来在实际中是不现实的，于是公式 1.22 就简化成：

$$P(x_1 x_2 \ldots x_T \mid w_i) = \max_X \left\{ \prod_{t=1}^{T} b_{q_t}(x_t) \cdot a_{q_{t-1} q_t} \right\} \tag{1.23}$$

求取公式 1.22 的方法主要有前向后向算法（Forward-Backword

Algorithm）[6]，求取公式 1.23 的方法主要有维特比算法及其各种改进算法，其中前向后向算法是由 Baum-Welch 模型训练算法[7]推算出来的，而较之前者，维特比算法不仅可以用于孤立词识别，也比较容易推广到连续语音识别，因此在这里就不再详述 Baum-Welch 算法及前向后向算法，而维特比算法在接下来的连续语音识别中再讨论。

(2) 连续语音识别

连续语音识别广义上也包含了连接词识别、关键词检出等应用。通常是采用对声学基元建模，并对一个给定的观察向量的序列，求出最大似然的模型序列。维特比算法与前向算法计算方法类似，不过将历史路径上的概率由求和改为只保留最大值。

具体来说，当前 w_i 模型中，让 $\phi_j(t)$ 代表从 x_1 至 x_t 历史序列为止的最大似然率，且目前是 t 时刻，处于模型的 j 状态。则根据以下递归公式：

$$\phi_j(t)=\max_i\left\{\phi_i(t-1)a_{ij}\right\}b_j(x_t),\quad 1\le j\le N \tag{1.24}$$

其中 N 是 HMM 模型的状态数，包括 1 和 N 两个非发射状态；$\phi_1(1)=1$，$\phi_j(1)=a_{1j}b_j(x_1)$。于是模型在观察向量序列下的最大似然率可以表示为：

$$\phi_N(T)=\max_i\left\{\phi_i(T)\right\}a_{iN} \tag{1.25}$$

在遍历的过程中采用一定的数据结构保留历史路径，回溯便可以得到模型的状态序列。对于公式 1.24 来说，由于直接计算似然率的乘积会导致实际中数值下溢，所以通常采用对数似然率：

$$\psi_j(t)=\max_i\left\{\psi_i(t-1)+\ln\left(a_{ij}\right)\right\}+\ln\left(b_j(x_t)\right),\quad 1\le j\le N \tag{1.26}$$

搜索算法里面比较流行的是帧同步算法（Frame-synchronous Algorithm）[8]，在每一步向前搜索一帧，都有可能是在某个模型的状态内跳转，或者跳出当前模型，进入任何一个模型。在一个上下文相关的大词表连续语音识别系统里面，哪怕是在模型训练的时候采用决策树对 HMM 模型的状态的高斯混合或模型转移矩阵进行合并和共享，仍然会有好几千甚至上万的模型数。维特比算法复杂度为 $O(N^2T)$，其中 N 表示全部模型的所有状态总数，T 表示观察向量的总帧数。显然，对于维特比算法来说，完全的计算每一条搜索的路径是极其耗时且不合理的。面对如此庞大数目的模型状态，普遍的做法是维特比集束搜索（Viterbi Beam Search）[9]，就是采用一定的方法对路径进行剪枝，对累计对数似然分小于一定阈值的路径，不再考虑。这种做法也有自己的不足，例如，在路径的开始，正确的

路径候选往往由于刚开始的似然分数比较低而遭到淘汰，而且这种剪枝在后期无法得到恢复，导致识别器无法找到正确的结果。然而，Jurafsky 提到，Huang (2001)等人的实验表明，集束搜索在 5%~10%的搜索空间里面达到了比较好的效果，这就意味着高达 90%~95%的状态不必在每一帧都参与计算，这会极大的提高搜索的效率。

根据经典的维特比算法，只能得出最大概率的路径，然而有时候，比如在做关键词检出等应用中，我们还需要对识别结果计算置信度，这时候则希望识别器能给出一个包含多候选的中间结果。于是，维特比算法被扩展为输出 N-Best 的算法。维特比算法虽然比复杂度是 N 的指数级的前向算法快了不少，但是仍然是开销极大。而且许多情况下，具有最高似然分的路径并非就是正确的结果序列，因为人耳接收信息还受到许多更高阶的信息的影响，比如语言模型。因此，试图直接在维特比解码的时候，在 HMM 模型边界（跳出某个 HMM 模型的最后一个状态，进入下一个模型的第一个状态）加入高阶语言信息被认为是不现实的。于是还有一种比较流行的两遍搜索的算法，也即是在第一遍搜索中采用比较低阶的语言信息，得到一部分剪枝后的网络，第二遍在缩小了的网络中再采取高阶的语言信息，对搜索路径进行重打分，得出最终的结果。Frank Soong 等人提出的基于 tree-trellis 的快速搜索算法[10]是其中的代表。这种做法就是第一遍采用维特比算法与低阶的语言信息进行快速剪枝，然后在第二遍从句子的最末帧往前采用基于 A*的算法得出 N-Best 路径候选，事实证明，该方法是比较有效的，日本京都大学开发的著名识别器 Julius[11][12]正是采用了这种思路。

1.2.4 性能评价

在语音识别系统中，单词错误率（Word Error Rate, WER）是最常见的性能评价参数。单词错误率是根据识别器的召回数中与正确的标注相比错误的个数在全部召回数中所占的比例来决定的。给定正确的标注，将识别候选与其计算最小编辑距离（Minimum Edit Distance）,在这样的最小编辑距离下，记录单词的插入、删除和替换错误，然后按照下式计算：

$$WER = \frac{Insertions + Deletions + Subsitutions}{Total \quad words \quad in \quad transcriptions} \times 100\% \tag{1.27}$$

有时候也有采用句子错误率（Sentence Error Rate, SER）来衡量系统的性能。此时 SER 的计算就是含有至少一个错误的句子占全部句子的比

例。此外还有许多各种评价参数。应该看到，在一些应用中，比如对需要理解语义的场合，一个句子中每一个词所承载的语义是有多和寡的区别的，此时单独凭 SER 是无法正确估量系统的性能的，这时候采用 WER 就显得更为合理，另外在关键词识别等应用中，识别的正确率还应该在一定的误警率（False Alarm Rate）下讨论，借助 ROC 曲线或者 DET 曲线，可以很容易的分析出系统在不同的操作点下的性能。

1.3 关键词检出的介绍及近年来的研究进展

关键词检出就是在连续的语音流中检测出目标定义的关键词，一个关键词检出系统不仅仅需要检出关键词，一般还包括后期对关键词的处理。所以，关键词检出系统也常常是对话处理系统的一部分。比如，自动语音台利用关键词检出系统识别出对话中所关注的关键词，采集所需的信息，完成语音内容的理解；安全部门利用关键词检出系统可以进行电话监听，完成敏感信息的来源追踪；民航系统可以利用关键词检出系统完成语音自动售票等等。关键词识别与连续语音识别既同又不同，具体说来有如下几点：

(1) 可以允许集外词。关键词检出需要处理连续语音，但是它对于关键词以外的词并无要求；

(2) 连续语音识别要识别出整句语音，还涉及到句法等自然语言处理的内容等，而基本的关键词识别系统几乎可以完全抛弃句子层次的上下文信息；

(3) 不仅可以用到连续语音识别中的算法，也可以借鉴孤立词、连接词识别中的方法；

(4) 应用场景上的差异对关键词识别有特别的要求。比如使用关键词识别技术的常常是非说话人，也就是说说话人是被动使用者，比如在监听，人工台转接等应用场景，此时说话人作为被动使用者，会采取更为随意和自然的方式，且电话信道上的识别也需要对算法做相关的修改。这些无疑是关键词识别的适用领域，不过也给语音信号的识别带来了相应的困难；

(5) 在性能上，普通连续语音识别系统的错词率（Word Error Rate, WER）一般在百分之几的水平，而一个关键词识别系统在某个阈值上（ROC 曲线上某点，也称为 KWS 系统的操作点）的错词率则常常为百分之几十。

以上这些特点都决定了关键词检出既要依托于连续语音识别，又不能直接照搬连续语音识别的方法，而应该根据自身应用场景所面对的挑战，研究一些新的方法。

近年来，国外关键词检出的研究逐渐增多。其中，Ketabdar[13]等提出了采用长上下文并结合先验知识累加每一帧候选关键词（或填充模型）与特征矢量序列的联合概率进行对齐的方法，避免了传统的维特比方法仅能得到近似路径最优的缺点，重点突出了关键词的局部似然度；Keshet[14]等则采用了非 HMM 框架的基于大间距线性分类器和核机器的方法进行置信度计算；Kim[15]将类似 N-Gram 的语言模型得分加入到基于填充模型的关键词检出系统中，获得了比较好的效果；Jin[16]提出可以将词格网络和音子网络的两层对齐结果利用混淆网络进行综合决策，获得候选关键词的输出结果。

1.4 论文结构

本文共分五章，第二章介绍一个关键词检出系统的整体结构和面临的难点。 第三章介绍如何训练关键词检出系统所需的声学模型。第四章介绍关键词检出系统的搜索策略；第五章介绍关键词检出系统的置信度方法。第六章则是总结了本文所做工作，并探讨了进一步的研究方向。

1.5 本章小结

本章作为全文的绪论，主要详细介绍了语音识别的基本原理和语音识别系统的工作流程，并由此介绍语音识别里比较重要的一个分支，关键词检出的技术，意在比较关键词检出技术与其他语言识别技术之间的异同，并对关键词检出技术这几年的一些进展进行了一些概述。

第二章 关键词检出系统的设计

关键词检出是语音识别的一个分支，其目的是在说话人的连续话语中辨认和确定给定单词表的特定词，这些话语可以包括许多其他的词，还可以包括说话人的非话语声音（如停顿声、呼吸声等）和说话时的背景声音（如背景噪声、音乐声、关门声等）。它可以借用连续语音识别的方法，不过由于和连续语音识别在任务要求和使用背景上的较大差异，在具体的实现上还是存在较大差别。关键词识别技术有着广泛的应用领域，比如在银行和票务代售中心，自动语音台等对话管理系统的应用场景，或者用于公共安全管理上的电话自动监听等领域。本章主要介绍一个完整的关键词检出系统的结构以及功能模块。

2.1 关键词检出系统的整体框架

一个基于统计型连续语音识别框架的关键词识别系统通常包括分为两个阶段，模型的训练，和关键词的识别。不管是训练还是识别阶段，首先需要采用声学特征矢量生成模块将声音信号转化为声学特征矢量序列作为系统前端输入。系统的 HMM 模型的选取确定以后，通过 Baum-Welch 算法或前向—后向算法对模型参数进行迭代和优化。关键词的识别则是采取基于维特比的帧同步算法由关键词检出模块对输入的声学特征矢量序列进行解码，得到 HMM 的状态序列，再采取最大后验准则选出获得候选的关键词。由关键词检出模块所得到的候选关键词，不一定就包含在待识别的连续语音段中，必须通过关键词确认过程(Utterance Verification)来确定它是否真的存在于语音段中。所以还需要置信度模块对识别结果进行判断，拒绝可能错误的命中结果，通常的确认方法是把维特比解码时最佳路径的平均分数作为确认的条件，只有分数在一定的阈值范围内的才是关键词，目前国内外对于置信度确认的研究比较多。一个普通的关键词检出系统的框图如图 2.1 所示。

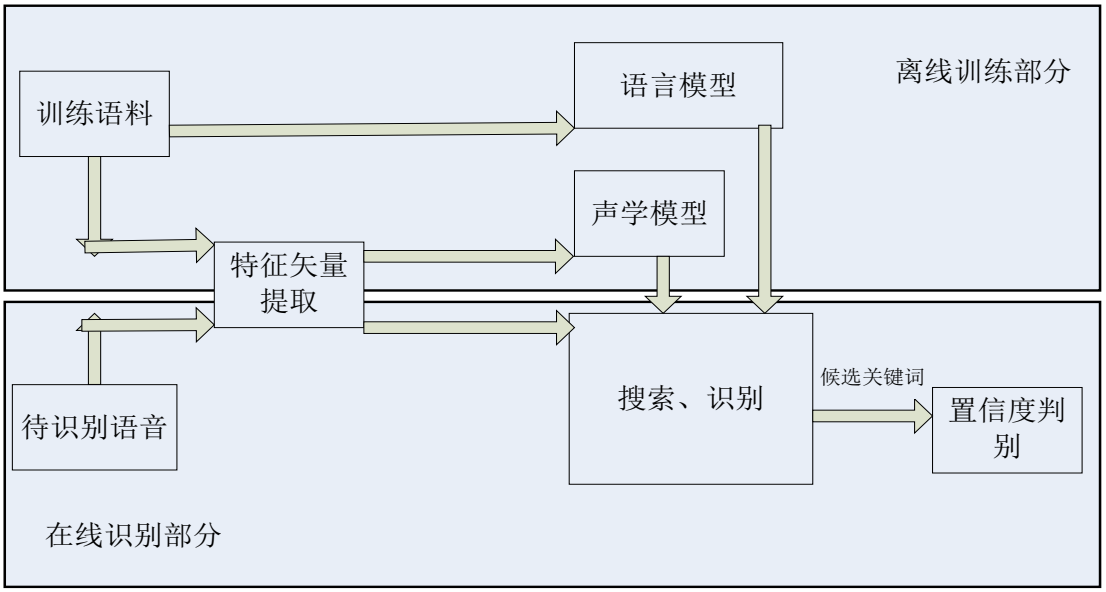

图 1.2 一个连续语音识别系统的示意图

2.2 系统各功能模块的确定

构建一个关键词检出系统，第一步是确定采用的特征。作为关键词识别的最前端，特征矢量提取模块是利用语音信号的短时平稳特性，在几十毫秒的时域上，对语音信号进行倒谱转换，将每帧语音片段对应于一个声学特征矢量进行转换，以便更准确的刻划语音信号的特征和去冗。本文的声学特征矢量采用符合人耳接收特性的 Mel 倒谱特征(Mel Frequency Cepstral Coefficients, MFCC)。取一阶的 MFCC 矢量及其差分系数组成一个矢量。为了提高特征的区分度，还需要加上差分信息，以反映语音的动态特性；对各维分量做均值和方差的归一化，部分去除信道和说话人相关的成份，提高特征的顽健性；最后做特征矢量的解相关线性变换，使得各维特征之间的统计独立性得到增强，提高对角方差建模的精度。这部分设计与孤立词和连续语音识别等系统的无异，并已经在绪论中有所讨论，这里不再详述。

其次，采用 HMM 作为语音识别的建模框架来构建模型还需要决定最小单元的选取，HMM 模型状态参数的确定和模型的拓扑结构的决定。英语等表音文字常常采用三音子模型(Triphone)或称上下文相关音素作为 HMM 最小建模单元，而近年来在中文语音识别方面的研究表明汉字等单音节采用声韵母或者上下文相关扩展声韵母(Tri-Extended Initial/Final, Tri-XIF)作为最小单元是比较合适的[17]。

本文的系统采用上下文相关扩展声韵母作为系统建模单元。并采用

Baum-Welch 算法进行训练。训练采用 HTK 的相关工具集，由于不是本系统设计的主要特性，且已经在绪论中有所讨论，这里不再详述。

一个普通的采取连续 HMM 的关键词识别方法通常是预先采用标注好的语音库构建子词单元(比如对于中文语音的识别，通常采用 Tri-XIF)的 HMM 模型，然后将关键词对应的音节的 HMM 模型连接成某个关键词 HMM 模型。采用上下文相关的单元由于增加了单元数目，通常是采用决策树将其划分为若干等价类。有些则是部分采取上下文相关的单元建模，比如一般情况下，填充模型设计的越复杂，数目越多，则对非关键词的描述能力越强，但是盲目的增大的填充模型也会带来计算上的开销，且识别率增长不多，Wilpon、Higgins 等人[18][19]对这方面的研究做了探讨。本文采用决策树方法，对模型参数进行共享，然后再进行迭代训练以达到减少模型数量的效果，也有利于减少后面阶段的网络搜索空间。

为了描述非关键词，常常采用以下两种策略：有填充模型(Filler Model)和无填充模型。填充模型又称为垃圾模型(Garbage Model)，用来描述集外词和背景噪声，有时候若集外词无法用单独一个填充模型来描述，也常常将所有的子词单元聚类成 N 个类，为每一类建立一个 HMM 模型。一个普通的有填充模型的 HMM 模型拓扑结构如图 2.2 所示。

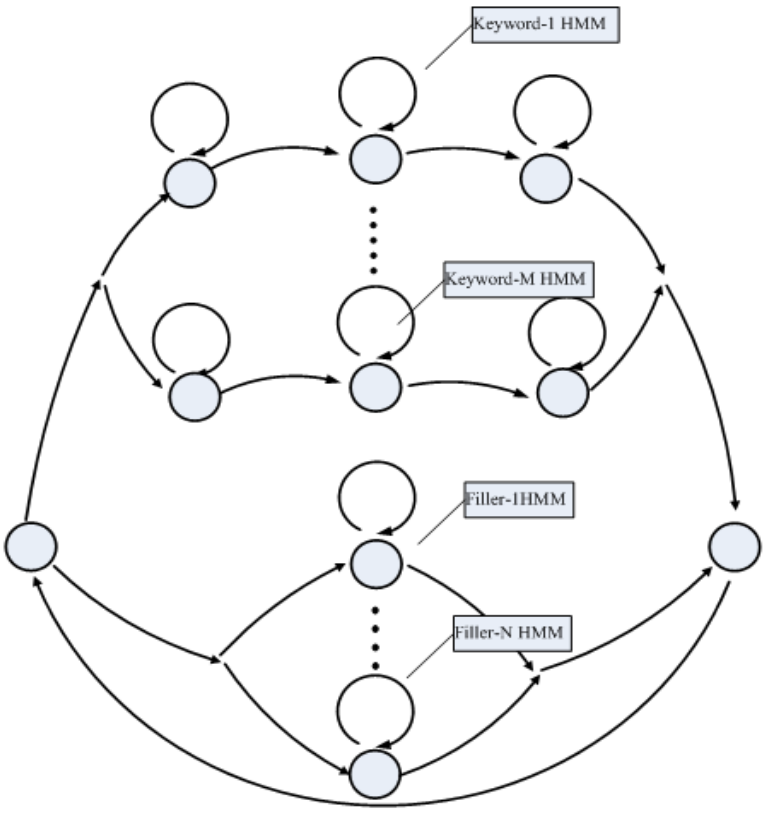

图 2.2 一个关键词识别系统的示意图

根据填充模型的生成阶段位于识别前还是识别时，还可以分为两种：离线垃圾模型和在线垃圾模型。离线垃圾模型在系统建立过程中预先对词库外的词(垃圾)建立模型，此时的模型称之为离线垃圾模型。通常用最大似然法训练得到离线垃圾模型，可以是单一的模型，也可以是多个分类模型。在数据量较小时，离线垃圾模型能够比较精细地刻画垃圾的特性，但是由于系统词库外的词相当广泛，要想得到比较好的结果，离线垃圾模型的设计与训练变得相当困难； Caminero 等人[20]提出了在线垃圾建模方法(On-line Garbage Modeling)，此方法没有明确地建立垃圾模型，而是当识别单位是音素时，在线地计算音素模型局部每一帧的得分，其中 N 个最好的得分的平均分作为垃圾似然得分。此算法计算简单，且有一定的抗噪性。用此垃圾得似然分与关键词似然得分相比较，可以对识别结果的正确性做出判断。但问题是这种方法仅仅考虑到外界环境的影响及输入语音的变化，而并没有考虑到关键词之间的混淆，所以对关键词之间的识别错误没有确认作用，系统的整体效果受到影响。

无填充模型的方法则是基于一个连续语音识别系统，这种模型是先得到音节的识别结果，然后再对这个结果利用关键词表进行处理。所以前述 Caminero 的方法也可以看成是无填充模型。这样做非常灵活，往往可以把 N-Best 音节序列或者是音节网格的结果用文本的方式存储起来，方便后续的灵活处理。但是这样做也有缺点，因为如果把音节的识别和关键词的识别分开处理，那么在对音节进行连续音节解码时，根本没有利用到音节之间会出现的特定连接关系的信息，也很难在这个阶段运用上层知识指导搜索。这样得到的音节序列的识别结果往往会增加大量的插入、删除错误，并且替代错误也会有所增多。这将严重的影响后期的关键词确认过程。

本文的关键词检出系统采用基于连续语音识别的方法。采用这种方案的做法有以下几点考虑：

第一，基于连续语音识别的方法避免了训练填充模型，而填充模型的设计和精度是采用填充模型的关键词检出方案的一个瓶颈。一种常用的设计填充模型的方法如下：

例如将语音库 100 个人分成两部分，拿 90 个人的数据来训练，得到 401 个音节模型及一个静音模型。再用这个模型来为所有的 100 个人的数据进行解码，将识别中的替代错误进行统计再填入音节混淆矩阵中。最后再对这个矩阵进行分析，找出较易混淆的将之归为一类，这样得到十三个类，再根据每个类内的数据来对这个类训练出一个填充模型来。这样的填充模型依赖于训练样本的数量，并且，由于语音识别技术在声学环境和说

话人说话方式，性别等因素的失配情况下有较差的鲁棒性，这类方法训练出来的填充模型对于拉开关键词和非关键词的声学得分并无很好的效果。

第二，基于填充模型的关键词检出系统无法方便的变更关键词词表。由于填充模型的拓扑结构往往是采用除了关键词以外的音子模型的自由连接网络，任何对关键词词表的改动都会影响填充模型的拓扑，从而需要重新训练新的填充模型，这种设计在一些需要经常修改业务环节的应用场景上是不合适的。

综上，本文的关键词检出系统采用基于连续语音识别的方法做为系统的网络框架，相应的，也确定了采用帧同步的维特比算法进行模型状态与特征矢量的空间进行搜索。

在关键词确认阶段，就是对置信度做出估计，关键词确认给出的置信度得分不仅能有效地去除错误候选，而且还可以为后续的语言理解提供信息。置信度问题从本质上说，就是对识别结果的正确与否进行判决的一个两类分类问题，因此，置信度研究的关键和重点在于如何寻找有效的特征，并找到从这些特征计算置信度的方法，使其区分能力达到比较好的效果。如果采用人工判决的方法，在没有参考答案的情况下，我们判断识别结果的正确与否，往往需要借助语义层次的信息，结合自己的经验和知识，才能得到比较好的判决效果，而这些信息在计算机中是很难利用的。目前用于置信度计算的信息主要包括以下三类：

(1) 识别结果本身的信息：如声学得分、语言得分、状态驻留时间、语言模型回退情况等信息，可以直接从识别结果中得到。

(2) 搜索过程中的信息：如竞争路径条数、词图密度（Word Graph Density）等信息，在识别结果中不能直接得到，是在搜索的动态过程中体现出来的。

(3) 辅助模型提供的信息：如声学似然比等，不仅需要识别结果本身，还需要借助额外的模型(通常称为背景模型或反模型)计算得到。

置信度计算的难点就在于，以上所有这些信息各自的区分能力都不是很强，而彼此也很难整合到一起。另外，在不同的语法约束条件下，能够利用的语言信息往往是不一样的，很难得到统一的置信度计算方法。例如，在关键词检测或者孤立词系统识别中，就很难利用语言模型的信息进行置信度计算。

因此，置信度研究的重点和难点在于，如何有效地选择和综合利用上述信息，计算在不同语法约束条件的置信度，使其区分能力达到最好，这也是本文的研究重点所在。并非所有的信息都能有效地整合到置信度计算

中，也并非所有的特征都能具有很好的区分能力，因此，在置信度计算过程中，往往就需要根据具体情况，选择比较有效的信息进行整合，才能达到比较好的效果。

本文采用一种基于 N-Best 的不匹配帧加权的置信度惩罚因子，来改进经典的基于驻留归一化的置信度方法。基于 N-Best 的候选路径可以综合更多一遍搜索过程中的声学似然度信息，同时相对于经典的基于 N-Best 的置信度方法，本文提出一个不匹配帧的概念，通过比较关键词候选分段与全部语音的不匹配帧的数量分布，强调了 N-Best 路径中候选关键词分段内具有较好声学匹配的路径在全部路径中的加权作用。具体的方法，在接下来的第五章中重点论述。

2.3 本章小结

本章主要是介绍关键词检出系统的整体框架，并对本文所提出和实现的中文关键词检出系统的各模块设计和采取的一些方法做了初步的介绍，主要是为了介绍本文所做的主要工作，按照系统的功能流向提供一个思路。

第三章 可定制的声学模型的训练

关键词检出面对的应用环境比较复杂，使用者口音、性别、发音方式的变化以及新的需求的变更都要求系统应该是可定制的。一个可定制的关键词检出系统包括可以调整检出点阈值，方便增减关键词表而无需做到重新训练模型等，为了做到这一步，首先必须对语音基元建模。

3.1 声学基元与隐马尔科夫模型表示

隐马尔科夫模型作为一种统计学方法，可以有力的刻划离散时域观察序列样本，并且可以很好的将动态规划的过程应用到时变序列的建模过程中去，并且时间序列的数据样本可以有离散或者连续的分布，既可以是标量也可以是向量。主要的一些关于隐马尔科夫模型的文献可以见 Baum 等人的论文[21]，而自从提出了建模的算法之后，隐马尔科夫模型在自动语音识别、口语理解和机器翻译中均获得了大量成功的应用。

马尔科夫链（Markov Chain）是一类随机过程。下面主要讨论离散马尔科夫链。假设随机变量序列 $X = x_1, x_2, \ldots, x_n$，其中每一个随机变量的取值来自于集合 $O = \{o_1, o_2, ..., o_M\}$，根据贝叶斯原理，我们有：

$$P(x_1, x_2, ..., x_n) = P(x_1)\prod_{i=2}^{n} P(x_i \mid x_1^{i-1}) \tag{3.1}$$

其中 x_1^{i-1} 表示 $x_1, x_2, ..., x_{i-1}$，则当：

$$P(x_i \mid x_1^{i-1}) = P(x_i \mid x_{i-1}) \tag{3.2}$$

的时候，我们称 X 为一个一阶马尔科夫链，则此时公式(3.1)变成：

$$P(x_1, x_2, ..., x_n) = P(x_1)\prod_{i=2}^{n} P(x_i \mid x_{i-1}) \tag{3.3}$$

公式(3.2)也称为马尔科夫假设，也即对于一个时间序列来说，给定时刻的随机变量的概率仅依赖于前一个随机变量的取值。如果将 x_i 与一个状态对应起来，则一个马尔科夫链则可以用一个有限自动机来表示，其中状态之间的跳转概率由：

$$P(x_i = s \mid x_{i-1} = s') = P(s \mid s') \tag{3.4}$$

来决定。考虑一个状态变量 s 由集合{1,…,N}组成的马尔科夫链，s_t 表示 t 时刻的状态变量的值，则一个马尔科夫链的参数可以表示如下：

$$a_{ij} = P(s_t = j \mid s_{t-1} = i) \quad 1 \le i, j \le N \tag{3.5}$$

$$\pi_i = P(s_1 = i) \quad 1 \le i \le N \tag{3.6}$$

其中 a_{ij} 表示从状态 i 跳转到 j 的概率，π_i 表示以 i 状态作为马尔科夫过程的起始状态的概率。且以上两个参数均有如下限制：

$$\sum_{j=1}^{N} a_{ij} = 1; \quad 1 \le i, j \le N \tag{3.7}$$

$$\sum_{j=1}^{N} \pi_j = 1; \quad 1 \le j \le N \tag{3.8}$$

以上模型其实就是将观察变量序列 X 与状态序列 $S = s_1, s_2, ..., s_n$ 对应起来了，换句话说，就是对每一个观察变量 x_i 均会对应模型的一个状态 s。然而在我们目前为止讨论的模型里，每一个状态对应的观察向量均是一个确定性的事件，是一个非随机信号。进一步的，我们讨论一种新的观察随机变量，该变量在所有的状态上都符合一定的概率分布，则此时观察变量与模型的某个状态不在一一对应，此时我们称该模型为一个隐马尔科夫模型。构造和训练隐马尔科夫模型的算法详细见文献[22]。

对于连续语音识别来说，我们已经知道可以利用音素（phoneme）单元构造连续语音的模型。音素对于不同的语言具有不同的意义，对于汉语普通话来说，通常采用音节(syllable)来构造一个汉字，而构成一个音节的声韵母（Initial/Final），就是音素。对于汉语普通话来说，409 个无调音节中还包含部分单元音，即只有韵母的音节，为了更好的规范音素的连接规则，方便在上下文相关情况下建模的时候减少模型数量，提出了扩展声韵母（eXtended Initial/Final，XIF）的概念，见表 3.1：

表 3.1 扩展声韵母表

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

与标准的声韵母表相比，增加了_a, _o, _e, _I, _u, _v个零声母，当使用标准的声韵母基元集合时，有一些音节只有韵母部分，而没有声母部分。所以，当考虑上下文相关信息时，这些韵母既可以搭配声母，又可以搭配韵母，因此，上下文相关声韵母基元数目会很大。而使用扩展的声韵母基元集合时，韵母的上下文只能是声母或静音，声母的上下文只能是韵母或静音，所以，上下文相关基元数目会明显减少。Jing LI等人通过实验也证

明，如果没有引入零声母，那些带有零声母的音节将会和其它音节的韵母部分共享模型参数，从而在识别中增加了许多插入错误[23]。

需要注意的是，尽管我们已经知道所有的音节均可由以上扩展声韵母组成，汉语音节的同一个音素在不同的汉字，甚至同一个汉字的不同词汇或者上下文中的发音都是有所不同的，这种现象称为“音素变体”(Allophone)。例如：声母[sh]在发“诗”这个音与发“书”这个音时，发音方式不完全一致，前者是非圆唇音，而后者是圆唇音。这种现象在构建声学模型时都需要正确的标注。

在建模的时候，根据需要，可以按照以下方式建模：

- 上下文无关；
- 上下文左相关；
- 上下文右相关；
- 上下文无关；

上下文的相关性是和“音素变体”现象相对应的，相同的音素在不同的上下文环境中发音的方式也会不尽相同。声学基元的层次大概可以分为以下几类：

- 声韵级；
- 音节级；
- 词级；

不同层次的声学模型是针对不同范围的发音进行建模的，一般的连续语音识别系统或者关键词识别系统采用声韵级模型或音节级模型。由于汉语普通话的特殊性，模型还可以按下面的方式划分：

- 有调模型；
- 无调模型；

本文实现的系统中主要采用的是无调声韵级模型。在汉语普通话中只有四种声调，它们是阴平、阳平、上声、去声，或者称之为一声、二声、三声、四声。声调在普通话中也承担着重要的构字辨意的作用，但是在实际的发音中声调的变化是多种多样的，会受语调、语气等多方面的影响，给声学建模和识别带来了很多不稳定的因素，所以在本系统中主要采用的是无调模型。

3.2 状态共享与决策树方法的引入

声学模型是关键词识别系统的基础。声学模型的分辨率直接影响识别

过程中不同词的声学似然分，从而对最终的关键词识别结果造成显著的影响。每个词都是由若干个声学基元组成的，每个声学基元都有逻辑名称。之所以称为逻辑名称是因为相同的声学模型可能存在多个与之对应的逻辑名称，也就是说逻辑名称和真正的物理模型参数不是一一对应的关系，而是多对一的关系。这种现象在上下文相关模型中更加普遍，这是声学模型训练时有效的实现信息共享造成的，逻辑名称不同的声学基元对应的物理模型的参数很可能相似或者相同，共享这些相似或相同的部分，不但可以减小模型的规模，还可以在数据有限的情况下训练出更精细的模型。另外，不仅存在逻辑名称不同物理参数相同的模型，而且不同物理模型之间的状态参数也有可能是相同的，模型参数在合理的范围内，共享程度越高，模型的规模越小，而且这样的共享对最终的识别结果也不会有影响。

音变现象在连续语流中是普遍存在的，当同一音素出现在不同的上下文语境时，它的声源的激励方式和声道的调制方法是不一样的。因此，在选择模型基元时就不能忽视它的影响。当采用基于上下文语境的基元建模时，所获得的声学模型会更加精细，模型的数目也比相应的上下文无关的模型数目要多。精细的声学模型将会更加精确地描述语音信号的特征，在识别阶段可以根据模型基元所出现的上下文语境的不同，选择不同的上下文相关的声学模型。而在采用上下文无关的声学模型时，不管上下文语境如何，都采用同一种模型。两者相比，显然前者比后者要优很多。

对于连续语流中的某个音素，可能同时受到来自它前面 n 个音素或后面 m 个音素的影响。我们只考虑前后 1 个音素对中心音素的影响，并且，汉语音节也确定了不必考虑完所有的音素的组合。进一步的，我们还需要一些语音方面的专家知识，对一些上下文相关的音素进行聚类。

决策树（Decision tree）方法是基于数据驱动和专家知识引导的分类法。其基本思想是首先根据专家知识精心设计问题集，选择相似度度量方法及决策树停止分裂准则。在决策树每次分裂时，都要从问题集中选择一个最优的问题即根据该问题分裂时所获得的相似度的增加值最大，并按照该问题进行分裂。

在本文的决策树的设计中，问题集也就反映为相似音子集的划分，我们参考了语音学的一些准则，比如声母按照发音方式分为塞音，塞擦音，擦音，鼻音，边音和通音。韵母根据韵头的不同分为开口呼，齐齿呼，合口呼和撮口呼四大类。然后再根据韵母的最后一个音尾的不同进行分类，比如鼻音 n 和 ng 划分为一类；或者根据韵头和韵腹来分类，在每一类的内部，又可根据音位的细小差别做进一步的分类。总之，考虑了各种可能

的分类，各类之间的音素相互交叉和包容。这些都是根据汉语语音学的已有结论。

3.2.1 决策树的建立

(1) 问题集的设计

在应用于模型状态参数的共享的时候，问题集即反映为音子集类别的划分。

所有的分类可以根据其组成分成两大类，即由声母构成的分类和由韵母构成的分类。如果中间建模单元是声母,它的左边只能是有关静音或韵母的分类，而右边只能是有关韵母的分类。如果中间建模单元是韵母，那它的左边只能是有关声母的分类，而右边是有关静音或韵母的分类。根据汉语音学知识和一些初步的实验比较，我们最终设计了一个有 124 个分类的音子类别集，其中分为左上下文和右上下文两类，举例如下：

表 3.2 决策树上下文相关音子类别集举例

编号	音子集名称	音子集集合	意义
1	"L_I_Sonorant"	{m-*, n-*, l-*}	左上下文是否是作为声母的响音
2	"R_I_Affricate"	{*+z, *+zh, *+j, *+c, *+ch, *+q}	右上下文是否为做为声母的塞擦音
3	"L_F_Open"	{a-*, o-*, e-*, ai-*, ei-*, er-*, ao-*, ou-*, an-*, en-*, ang-*, eng-*, ii-*, iii-*}	左上下文是否为做为韵母的开口音

(2) 停止分裂准则

决策树的停止分裂准则则是采用对数似然分，设定一个阈值，当分裂集合带来的似然分增加低于该阈值的时候停止分裂，将该树节点设为叶节点。注意在生成决策树之前，必须先根据每状态对应的样本数量的多寡，设定一个阈值，将那些统计上比较稀疏的状态去除。

(3) 评估函数的选择

评估函数是为了度量样本之间的相似度，它可以是任何一种距离测度，如均方根距离函数和似然比函数等。当决策树的某一个节点称之为父节点分裂成两个节点(称之为子节点)时，似然度会增加。假设 L_{parent} 表示父节点的相似度，$X=\{x_1,x_2,...,x_N\}$ 表示父节点有 N 个样本。假设 L_{lchild} 和 L_{rchild} 分别

表示两个子节点的相似度，$X^1=\{x_1^1,x_2^1,...,x_{N_1}^1\}$和$X^2=\{x_1^2,x_2^2,...,x_{N_2}^2\}$分别表示两个子节点的样本，并且$X^1$和$X^2$构成父节点样本集$X$的划分。在本文中，每个样本就是一个 M 维的特征矢量，假设相似度的增加以ΔL来表示的话，则：

$$\Delta L = L_{lchild} + L_{rchild} - L_{parent} \tag{3.9}$$

值得注意的是，决策树分裂过程中无需额外的计算高斯混合的似然分，可以利用上下文无关的第一遍训练过程中保留的信息。为了高效的进行决策树的分裂，训练上下文无关中文声韵母模型的时候可以保留状态的驻留样本个数，似然分数等信息，因此，决策树的分裂的样本集可以直接从该统计信息中得到，极大的加快了决策树的建立。具体做法就是利用 HTK 工具中的维特比强制分段对语音库文件进行切分，训练出上下文无关的 27 个声母（包括 6 个零声母）和 38 个韵母，外加一个静音模型。根据保存出来的统计文件，依照如下的决策树生成算法进行状态参数的共享。决策树生成算法流程如下：

1. 选择某一个模型的某个输出分布，记样本集为X。定义开始节点为根节点，包含样本集中的所有样本，标记根节点为“没有处理过”。
2. 从所有节点中选择一个“没有处理过”的节点，如果当前节点所包含的样本数小于停止阈值，则记录该节点为叶节点。否则，计算该节点的相似度，然后对音子类别集中的每一个分类都计算该节点如果按照这个分类分裂为两个子节点时的相似度，然后根据式计算相似度的增加值。选择使得相似度增加最大的分类对该节点进行分裂，并记录该分类，标记该节点“ 已经处理过” ，同时标记新产生的节点为“ 没有处理过”，计算新产生节点的相关参数。.
3. 如果所有的节点都已经处理过，则决策树已经形成否则，转入 2。

一个典型的决策树分裂以划分共享模型的状态参数的示意图如图 3.1 所示。

3.3 本章小结

本章主要是介绍本文所设计和实现关键词检出系统的模型训练部分所用到的基于决策树的模型共享方法。首先分析了中文音子的上下文相关建模的基本原理和方法，以及基于汉语语音学的一些音子集划分原则，最后给出了基于决策树的参数共享的策略和算法流程。

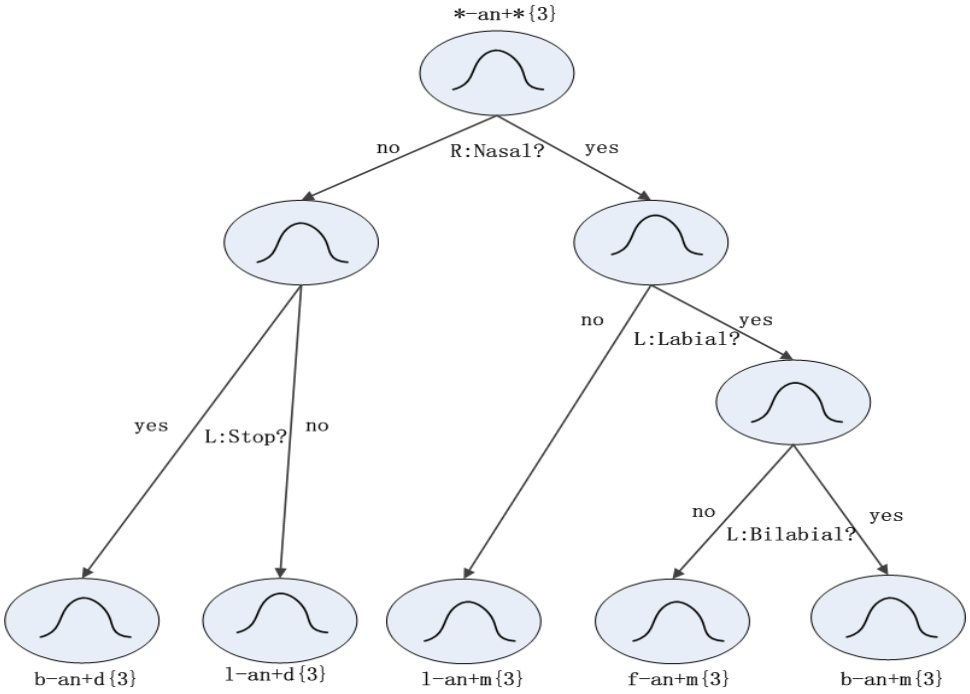

图 3.1 利用决策树共享模型状态参数的示意图

第四章 网络搜索策略

本章将首先介绍连续语音识别中常见的网络搜索策略并对几种搜索策略进行比较，然后对本文采用的修改的帧同步算法做一个详细的介绍。

根据信息理论的定义，搜索算法也被称为解码算法。由于基于 HMM 的连续语音识别的问题本质上来说，就是一个在多层次的网络上寻找最大（小）路径或者说最优路径的问题。基于 Bellman 原理，就是把一个全局最优的问题转化为求解局部最优的子问题并递归求解的过程。这个动态规划（Dynamic Programming）的过程是所有搜索算法的基础。评价一个搜索算法的复杂度和性能是在一定的网络规模下前提下进行的，而在连续语音识别问题中，一个网络的规模与模型基元的数量，上层的语言模型对网络的语法约束相关。经典的搜索算法包括维特比算法和 A*堆栈算法。维特比算法能够得到最优解，是基于动态规划恒定的假设：如果最佳路径经过状态 q_i，那么这个最佳路径一定包含 q_i 之前的最佳路径。因此维特比算法并不能用于所有可能的语言模型，例如对于三元语法模型就不适用。维特比算法属于宽度优先（Breadth-First）搜索，一般各种改进的维特比算法都具有帧同步的特点，结合一定的剪枝策略可以减少计算量，但是却有可能丢失次优解。宽度优先和深度优先（Depth-First）算法均为 A*算法的特例，而 A*算法属于最佳优先（Best-First）搜索。Nilsson[20]证明，满足一定的条件，A*算法可以保证完备性和最优性，因此，在语音识别领域也得到了实际的应用。而堆栈解码算法（Stack Decoding Algorithm）属于 A*算法的一种。随着更优秀的剪枝策略的提出，80 年代以来基于维特比算法的帧同步算法以及其衍生算法得到了更广泛的应用。根据遍历次数，还可以将搜索分为一遍搜索（One-Pass）和两遍搜索（Two-Pass），Frank Soong 提出的基于维特比算法的搜索和 A*算法的两遍算法则成功的应用于日本京都大学的 Julius 语音识别器中。接下来将会逐步介绍以上提到的算法。

4.1 帧同步算法

帧同步算法（Frames-Synchronous Viterbi Search）的基础就是维特比算法。对于维特比算法，若将横轴定义为由相同时间间隔的语音帧提取的特征矢量组成的序列，则对应于一个顺序增长的帧号。纵轴则是对应于所有语音模型的状态，于是，搜索网络组织成一个有着规整结构的网络，这种网络往往在文献中称为格栅（trellis），在一些文献中，常常将搜索网络

分为两层甚至三层，第一层或者说最高层往往是与语言模型相关，该层网络由语法树和发音词典构造，而最低的层次则是与声学对齐相关，最低层的网络就是由 HMM 的状态与时间轴的最小单位共同组成，也被称为状态格栅（state trellis），如图 4.1 所示。帧同步算法就是描述在该层网络上的维特比搜索过程。

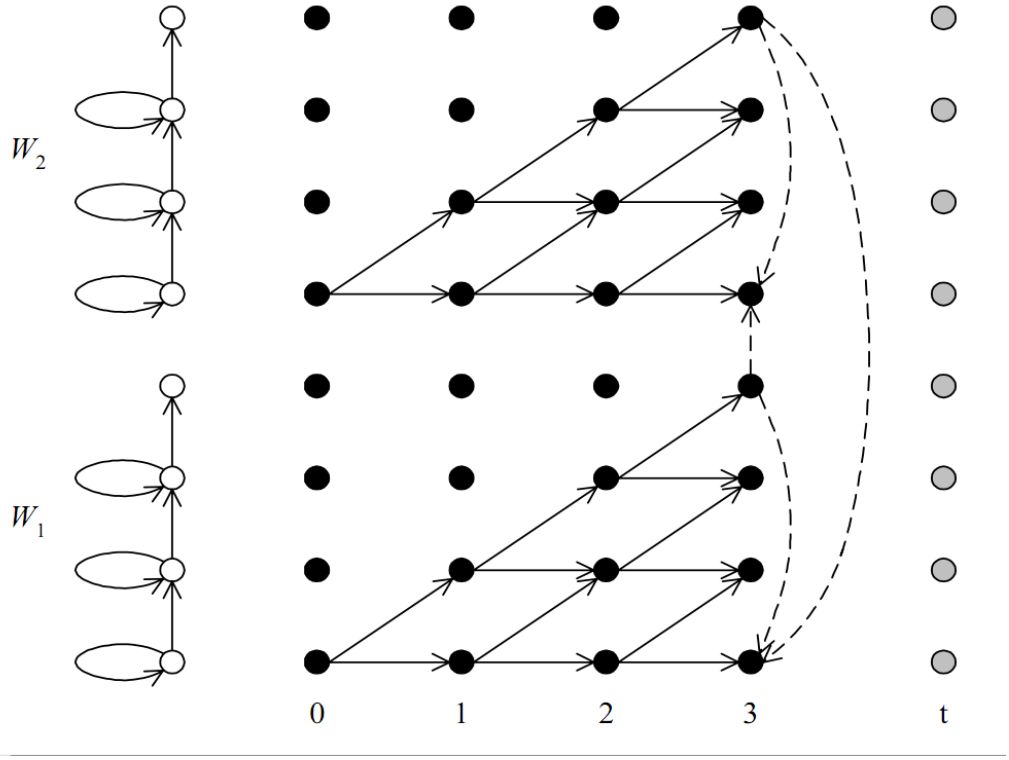

图 4.1 一个典型的状态格栅的局部

语音识别的任务是要找到与观察序列对齐的最优路径，而经典的维特比算法仅能得到最优的得分，因此，还需要对算法进行一些修改，比如通过回溯（back-tracking）得到通过的最优路径的状态序列。S. J. Young 提出的 Token Parsing[25]就是一个在著名语音识别工具集 HTK 上完整实现的帧同步算法。由于众所周知的所谓维特比近似（Viterbi Approximation）假设，帧同步算法是次优的，因为一个最优的状态路径，并不一定对应于一个最优的词路径。因此，帧同步算法需要加入语言模型。帧同步算法在每一帧达到计算之前，都必须更新完上一步的所有路径得分，因此也被称为时间同步的维特比算法（Time-synchronous Viterbi Algorithm）。基于维特比的算法具有多项式的时间复杂度，在一个模型空间巨大的网络中进行搜索，计算量是相当大的，因此还必须要有有效的剪枝算法，以及一些实现上的设计。此外，由于高层网络搜索的需要，要求低层次网络的搜索不但能够保持最优的路径，还要能够输出次优的，或者 N 最优（N-Best）的路

径，这些都需要对算法进行相应的修改，以上问题也在接下来分别讨论。

4.1.1 语言模型的加入

关键词识别系统如果要达到比较好的识别性能，就需要充分的利用高层的知识。最高层次的句法信息描述词汇之间的连接约束，一般用有限状态网络或者统计语言模型来表示。网络的最上层起到了一个语法约束的作用，其节点表示语法单元的边界，弧表示语法单元及其转移。网络越简单，表明语法约束越松弛；反之，表明更强的语法约束。网络的中层是对上层语法弧的展开，可以认为对每个词或短语的声学基元的约束，词对应的声学基元的连接关系、构成词的不同声学基元所可能产生的混淆情况都可以在这个层次的网络中进行描述。

有限状态网络的灵活性比较好，需求的训练数据要求比较小。类似图 2.2 的有限网络就是用一个 BNF 范式描述的。有限状态网络可以由专家在对语料库进行分析的基础上很快的总结出来，而统计语言模型则需要比较充分的训练数据才能得到。综合以上的考虑，本文的关键词识别系统，是以一个有限状态网络组织语法规则，作为最高层的网络。与图 2.2 的例子不同的是，本文的关键词系统并未建立垃圾模型，而是尝试用连续语音识别的方式，将所有模型单元的连接和定义的关键词均考虑在内，建立了一个全连接的网络。而对关键词，则增加了跳转权重。这样的好处是对非关键词语音段的对齐会比采用垃圾模型的句法网络具有更好的效果，并且由于并未加入统计语言模型，在进行一遍搜索的时候维特比对齐的开销会比较低，若应用于特定的应用场景，可以加入更细致的语法规则，进一步限制网络规模，减少搜索时间。

网络的中层一般采用词典来描述，词典按照组织方式分为线性词典和树形词典。对于一个逻辑上的词典如图 4.2 所示，一个典型的线性词典如图 4.3 所示，一个树形词典如图 4.4 所示。

对于一个线性词典来说，由于每个词独占一个搜索的路径，因此在进入词条的时候，就已经可以确定进入的词条，对于某些应用比如关键词识别，可以方便的对关键词提高权重，然后，在识别路径进入关键词的时候，将权值分配到关键词的音子对应的 HMM 模型的状态对应的似然分上，减少关键词检出中关键词内部插入，删除和替换的错误。值得注意的是，这个权值不能在网络中词汇的入口处一次性加入，因为对于一个帧同步的搜索过程，这样会导致路径似然分不平衡，因此，应该将该权值分摊到词内

的每个模型上，在搜索步进中逐渐加入。如果采用线性词典，这种分摊是十分容易实现的，而对于树形词典，则需要进一步的考虑，文献[22] 也讨论了树形词典的权值分摊问题。

...	...
syllable n	[]: l iu
Word 1	“流行”: l iu x ing
Word 2	“刘德华”: l iu d e h ua
Word 3	“张飞”: zh ang f ei
...	...

图 4.2 一个逻辑词典的例子

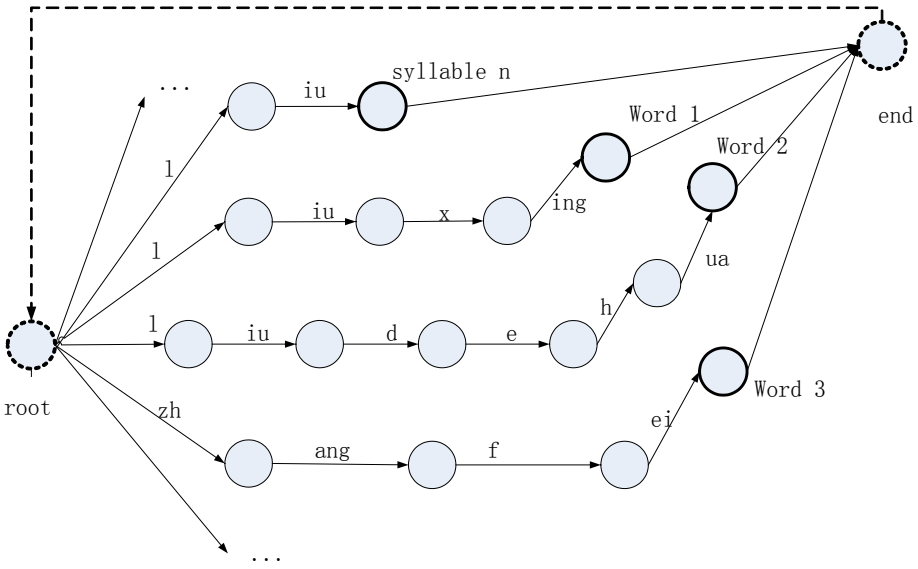

图 4.3 线性词典

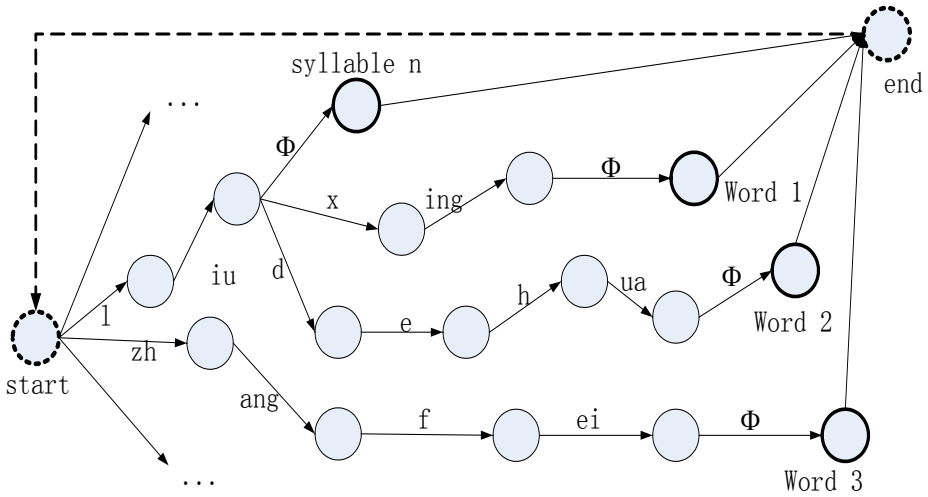

图 4.4 树形词典

树形词典有一个特点，就是线性词尾。对于一个单词来说，若从词中间某条弧直至末尾，不再有其他单词与其共享弧，则从该弧直至末尾的弧称为一个单词的线性词尾。搜索到达线性词尾的开端时候，便可以确定一个单词而无需遍历至单词的结尾。因此，线性词尾的语言模型的概率为 1，在加入上层语义信息的时候，无论是采用语言模型还是关键词加权，均可以将线性词尾合并。这样的做法显然是有优势的，因为越早加入高层知识源信息，便可以越准确的扩展出最优路径。加上相同前缀的词共享相同的树枝节点，极大的减少了模型数量，因此，采用树形词典的组织方式作为算法操作的数据结构的优势则在于可以大幅度的减少搜索空间。Ney[26]等人通过实验证明，一个 12,306 个词表大小的连续语音系统，采用树形词典只需要 43,000 个语法弧，而采用线性词典则需要 100,800 条语法弧，是采用树形词典的 2.5 倍。

一个自然而然的想法就是，将单词首尾相同的部分合并，形成一个音子网络以进一步缩小搜索空间。因此，这样的前缀树应该是一棵包含音子网络的树[27]，然而这样形成的树的形态过于复杂，导致算法需要进行较大的改动，不过，实验证明，在采用跨词的上下文相关音子而非上下文无关音子构建前缀树的时候可以极大的提高搜索的效率。

树形词典的缺点也是由于它的前缀共享造成的，由于所有单词共享相同的前缀，导致进入路径的时刻，当未通过表示词汇末尾的标志的时候，识别器并不知道当前正在进入哪一个单词，这样当前词候选的权重或者语言模型概率只有在到达词候选的最后一个结点（或者线性词尾）时才能计算。

为了能够应用动态规划原理，我们可以采用如下的方式来组织搜索空间：

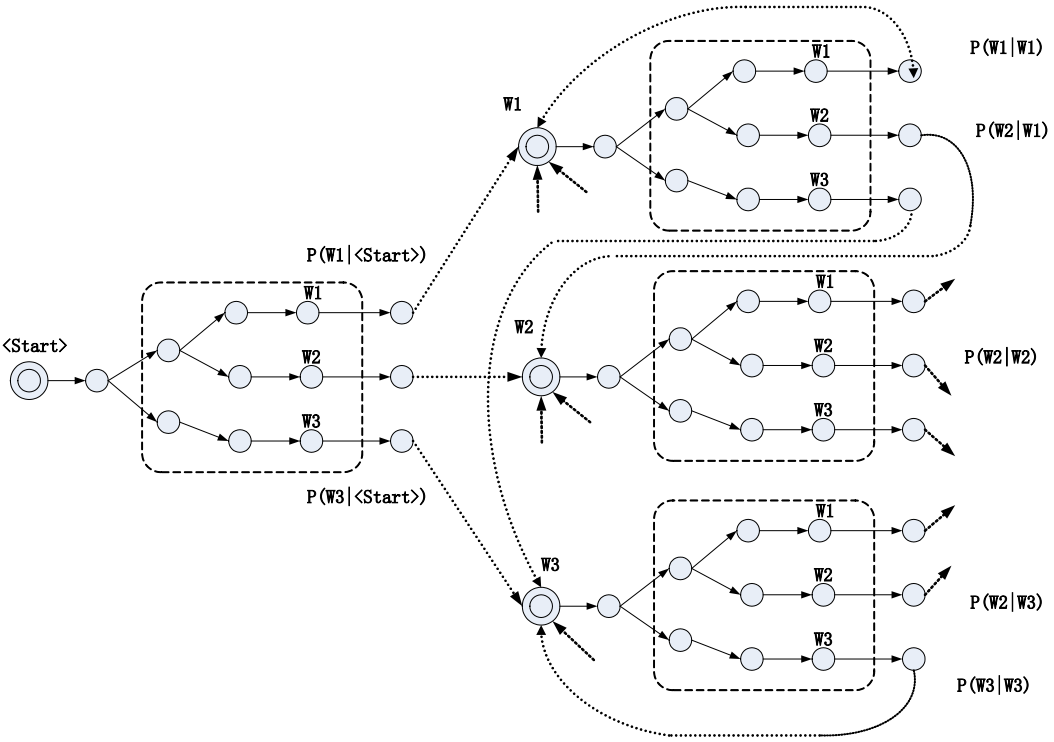

图 4.5 前缀树词典搜索过程

对于每个前驱词 v，我们引入词典的一份拷贝，这样在搜索的过程中，当词结束的假设 w 出现时，我们总能够知道它的前驱词为 v。一个典型的基于 bigram（二元文法）的非上下文相关的前缀树搜索过程如图 4.5 所示。当然如果采用特殊的词典结构，是可以避免搜索过程中树的拷贝的。Schwartz 就曾经提出了一种语法指导下的单树搜索算法[28]。

确定了语言层次的网络结构以后，可以进一步描述在该层次上的搜索策略了，首先定义两个变量：

$Q_v(t,s)$ 表示时刻 t 到达前驱词为 v 的词法树的状态 s 的最佳部分路径的得分；

$B_v(t,s)$ 表示时刻 t 到达前驱词为 v 的词法树的状态 s 的最佳部分路径的起始时间。

这两个变量的计算根据以下递推公式：

$$Q_v(t,s)=\max\left\{p\left(x_t,s\mid s'\right)\cdot Q_v(t-1,s')\right\} \tag{4.1}$$

$$B_v(t,s)=B_v(t-1,s_v^{\max}(t,s)) \tag{4.2}$$

以上公式 4.2 中 $s_v^{\max}(t,s)$ 表示前驱词为 v 时假设(t,s)的最佳前驱状态。

后向指针 $B_v(t,s)$只是简单的根据动态规划的决策进行传播。和前驱词 v 不同的是，正在处理的词 w 的索引只有当路径假设到达词法树的结束结点后才有可能知道，因为词法树的每个结束结点标记的是词典中的对应词。

在词的边界，我们需要为每个词 w 找到它的最佳前驱词 v。我们定义：

$$H(w;t) := \max\left\{p(w|v)\cdot Q_v(t,s_w)\right\} \tag{4.3}$$

这里 S_w 表示词法树中词 w 的结束状态。为了能够传播路径假设，我们需要在处理时刻 t 的数据帧前传递分数和时间索引：

$$Q_v(t-1,s=0) = H(v;t-1) \tag{4.4}$$

$$B_v(t-1,s=0) = t-1 \tag{4.5}$$

于是，前缀树搜索算法流程参见图 4.6 所示：

按照时间顺序从左至右：

声学层：处理（tree, state）假设

初始化：$Q_v(t-1,s=0) = H(v;t-1)$

$B_v(t-1,s=0) = t-1$

时间对准：使用 DP 计算 $Q_v(t,s)$

传播后向指针 $B_v(t,s)$

对不可能的假设进行剪枝

词对层（word-pair）：处理词边界假设

对每一个词 w：

计算 $H(w;t) := \max\{p(w|v)\cdot Q_v(t,s_w)\}$

$V_0(w;t) := \max\{p(w|v)\cdot Q_v(t,s_w)\}$

存储最佳前驱词 $V_0 := V_0(w;t)$

存储最佳边界 $\tau_0 := B_{v_0}(t;S_w)$

对每个词对（v,w）存储：

词边界 $\tau(t;v,w) := B_v(t;S_w)$

词的得分 $h(w;\tau,t) := Q_v(t,s_w)/H(v;\tau)$

图 4.6 前缀树搜索算法流程

4.1.2 剪枝策略

若设全部模型的状态总数是 N，待识别语音段的总帧数是 T 的话，维特比算法的搜索空间复杂度是 $O(NT)$，时间复杂度则是 $O(N^2T)$，因此，对

于一个大词表的连续语音识别器来说，就算是借助于动态规划的方法，基本的维特比算法仍然需要遍历数量巨大的网络，在这样的情况下，在每一帧的路径分更新的过程中，及时的淘汰掉似乎不可能在接下来的搜索中有机会保留下来的路径，会极大的节省搜索的事件，这种思想就是称为剪枝。剪枝算法是一种启发式的算法，因为在全部语音段对齐完成之前，并没有办法获取确定性的信息以帮助淘汰掉必然没有机会成为最优的路径。

比较简单的剪枝算法就是所谓的集束搜索（Beam Search）。一种策略就是给定一个阈值，在搜索的 t 时刻，若之前的路径的分数到当前节点的开销大于最优路径的得分与阈值的和，则应该淘汰。也有一种做法是限定每一步搜索中保留的最大的 w 个部分路径，此时我们说这个束的宽度就是 w。这种看似简单的策略与帧同步算法相结合，还是取得了不错的效果，这是由于帧同步算法本身的时间同步性决定的。由于相比较的路径都对应于同样的局部观察向量序列，因此局部路径的累积似然分作为评价最优路径的度量，可以取得不错的效果。

剪枝可以分为几个层次，这几个层次在剪枝的时机上也不一样。

- 声学剪枝：在到达每一帧之前，按照剪枝策略进行剪枝。
- 语言模型剪枝：在每一个语法节点的边界，在加入语言层的得分之后，按照剪枝策略进行剪枝。

4.1.3 保存N-Best路径

经典的维特比算法在每一步搜索中仅仅保留最优的路径，而在关键词检出中，常常还需要后处理阶段对一遍搜索的结果进行置信度分析。相对简单的语言层设计，让关键词识别相比于连续语音识别的一遍搜索的结果更多的反映了声学的似然度，因此，最优路径往往并非一定是正确结果。若能保留尽可能多的中间结果，并借助于某种后处理方法，将声学似然度不是最大的，但是却更有可能成为正确结果的路径进行重新打分，便可以提高系统的识别正确率，N-Best 结果的生成，就成为识别器设计的一个重要需求。因为，有必要对经典的维特比算法进行修改，使识别结果具有更多的信息。

常用的的中间结果的数据结构有词格网络（word lattice）、词图（word-graph）和 N-Best 路径，而 N-Best 列表的生成需要借助于词格网络。词格网络的生成与帧同步搜索是同时进行的，

一个典型的词格网络如图 4.7 所示。在一个完整的词格网络中，边表

示词（这里指识别基元）对应的模型，端点表示词边界的时间信息，还应该记录词的累积声学概率对于不同上下文的同一个音子，应该处于不同的路径中，而对于历史路径上音子序列相同但是时间点有所变化的，按照 Schwartz 和 Chow[29]的算法，则应该合并。

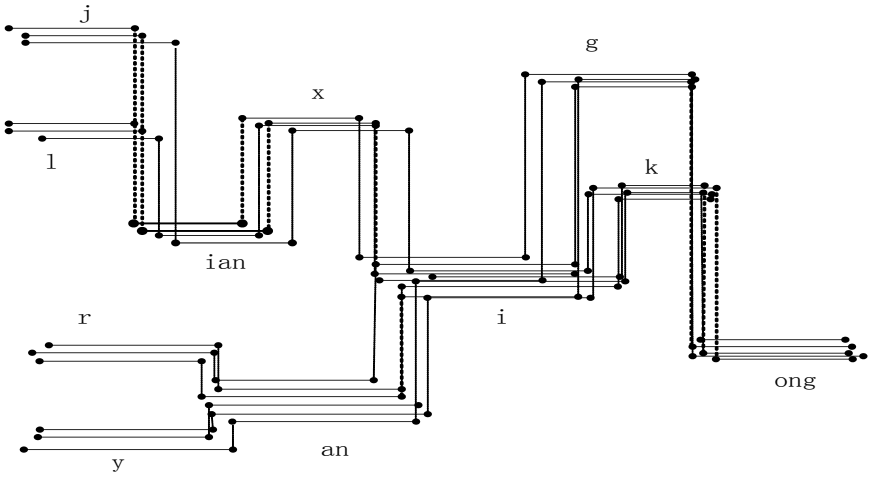

图 4.7 词格网络示意图

本文采用修改的帧同步算法产生词格，可以避免在遍历词格以产生 N-Best 列表的时候重新遍历 HMM 的状态-特征空间格栅（state-feature trellis），并能够避免由于维特比近似造成的某些路径的丢失，以下详细描述。

若采用不修改的采用基于维特比的帧同步算法，对于 t 时刻到达 s_t 状态，处于 w 模型的格栅中的某个点 $Cell(t;s_t;W)$ 来说，只能保留最小似然分的路径，这个在孤立词识别中是没有任何问题的，但是，对于连续语音识别来说，由于单词的边界有可能出现于任何一帧，于是，有必要保留任何一条似然分数处于集束宽度内的路径，然而，此时 $Cell(t;s_t;W)$ 还有可能有来自于不同的前驱词的入弧，这时，若按照标准的维特比算法求最小值，则会造成该路径的丢失，如图 4.8 所示。

因此，需要维持一个后向的列表，该列表中的每一个元素代表词格中的每一条边，其中需要保存的信息应该包括前驱词指针，当前部分路径得分，以及前驱词入弧的时刻。这样，在向前扩展的时候，保存相同前驱词的路径得分的总和而不是最小值，并将此语法节点放入队列；对于不同前驱词的路径应该单独建立节点放入队列。对于状态-特征空间中的路径，按照最小路径（最大似然分）的分数和集束宽度（阈值）进行剪枝；对于不同历史语法节点路径，则维护一个 N-Best 的列表，其中每一条路径按照分数进行排序。

由于区分了前驱词，可以保证不同上下文的路径不会被丢失，如图 4.9 所示。

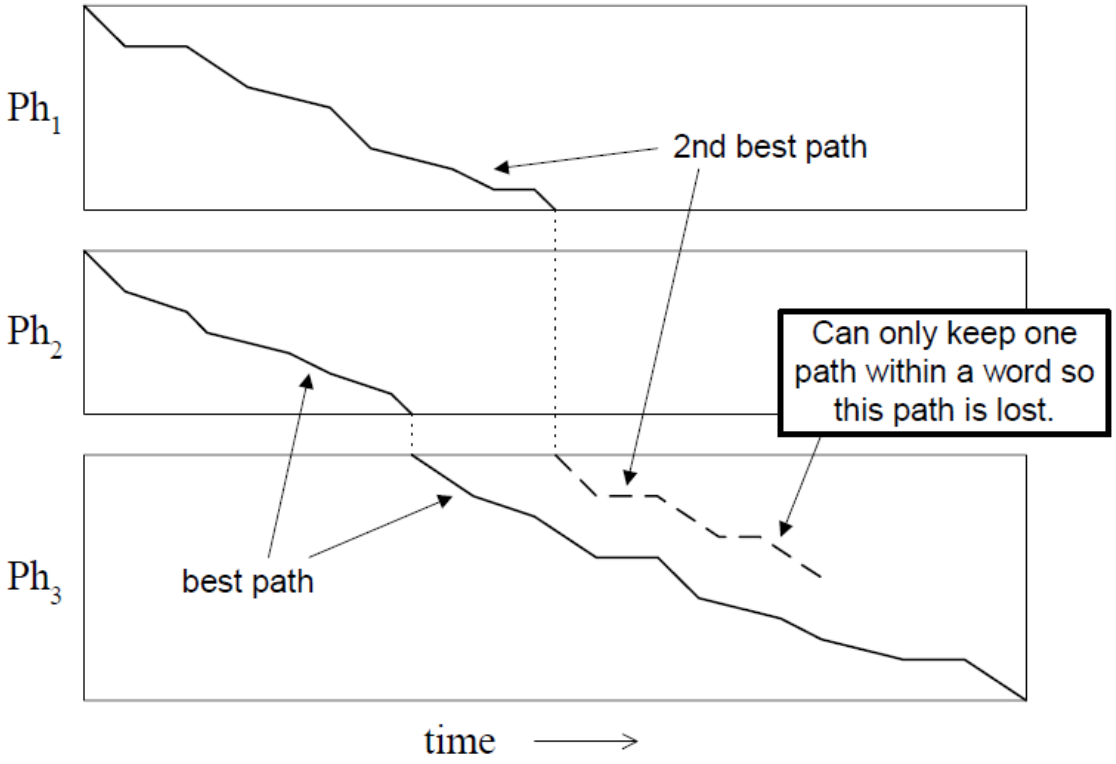

图 4.8 帧同步算法输出 N-Best 路径的示意图

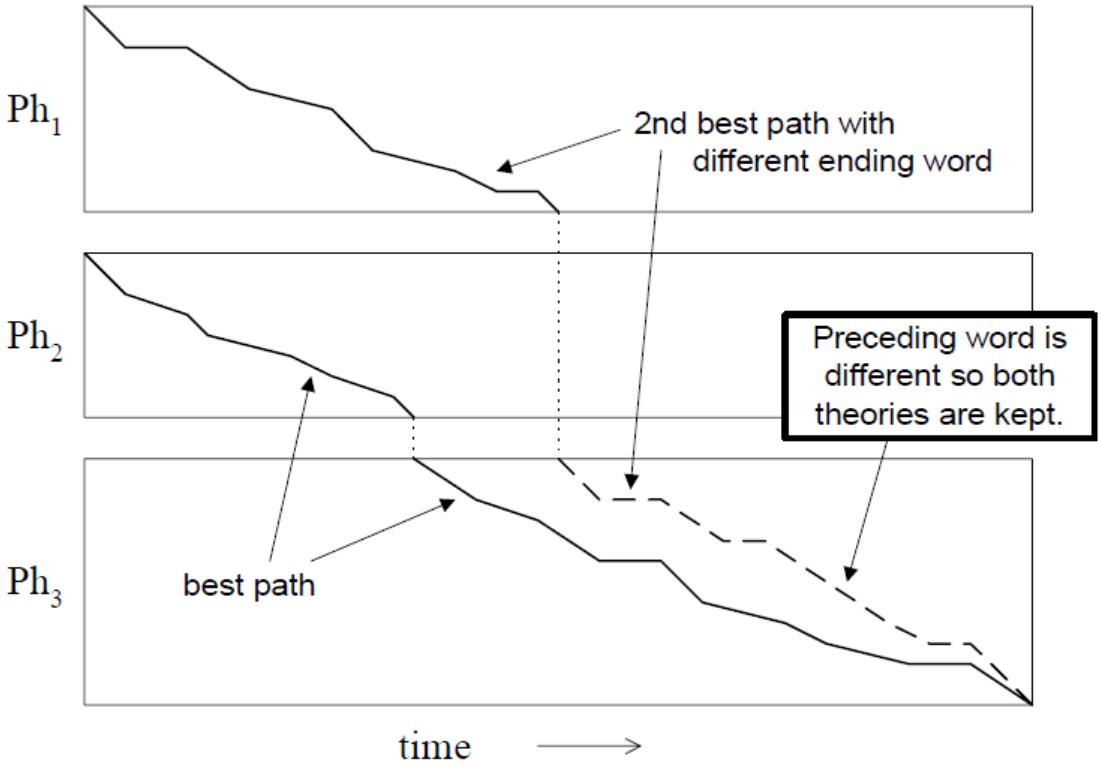

图 4.9 修改的帧同步算法输出 N-Best 路径的示意图

设 $\tau(t;w_i,w_j)$ 表示最佳前驱词的入弧时刻， $h(w_j;\tau(t;w_i,w_j),t)$ 也即是 $-\log P(x_\tau^t \mid w_j)$ 表示从时刻 τ 到当前时刻单词 w_j 的累积似然分。则修改的帧同步算法流程如图 4.10 所示。

按照时间顺序从左至右：

执行前缀树搜索；

对与 t 时刻的任一词对 (w_i,w_j)

存储最佳前驱词的入弧时刻 $\tau(t;w_i,w_j)$

存储当前路径的累积分数 $h(w_j;\tau(t;w_i,w_j),t)$

存储指向前驱词的指针

输出 1-Best 结果；

回溯生成词格或者 N-Best 列表；

图 4.10 修改的帧同步算法流程

4.2 堆栈解码

对于时间同步的搜索算法，前面已经给出了比较详细的论述。下面将对时间异步的搜索算法做一简单介绍。

时间异步的搜索算法可以通过堆栈解码器（Stack decoder）来实现。堆栈解码器在解码的过程中将使用一些堆栈，这些堆栈包含着一定数目的词的假设。

通常，这些假设将通过使用词典得到扩展，扩展出来的新的假设则插入到对应的堆栈中。当所有的堆栈（除了结果堆栈）都变为空时，结果堆栈里将包含最佳假设、N-Best 假设或者网格（lattice），具体的结果形式依赖于搜索的模式。

通常堆栈 stack 指的是后进先出(Last-In-First-Out, LIFO)缓冲区，但是实际上它可以是一个非常简单的按照某种分数排序的的假设的列表（优先队列，Priority queue）。排序所基于的分数可以是：

- 部分假设的对数似然度，
- 整个完整的句子的对数似然度的预测（A*准则）[10]，
- 其它一些能够反映部分假设的正确性的分数[30]。

所有堆栈解码器都至少包括如下两个层次：

- 外层，在堆栈之间循环（词级的搜索）；
- 内层，在时间和状态之间循环来搜索词（状态级的搜索）[31]。每当找到一个可能的词边界时，该词的语言模型分数就可以加上去。由

于这个动态的语言模型分数可以考虑任何历史词，因此在堆栈解码器中可以很容易地使用任何类型的 N 阶马尔科夫语言模型或者非马尔科夫语言模型。

把语言模型的使用从状态空间的维特比搜索过程分离开来还有其它一些好处。由于词假设的生成完全和词内的搜索独立开来，因此词内的搜索可以采取非常高效的方式来实现，并不需要回溯指针的存储。词的网格可以很轻松地生成。采用类似的过程，N-best 列表也能够很容易地产生。

由于本文所实现的关键词检出系统并未采用时间异步算法，在这里就不多介绍了。

4.3 本章小结

本章主要是介绍基于连续语音识别方法的关键词检出系统的网络结构以及搜索算法。首先对各种常用的搜索算法进行了介绍并比较相互的优点和缺点，然后对本文所提出和实现的基于帧同步的维特比网络搜索算法的修改算法进行了阐述，包括多个层次上搜索网络的结构和数据结构的介绍，相应算法的分析以及 N-Best 候选路径生成的算法。

第五章 置信度分析

语音识别系统的鲁棒性问题，是语音识别技术走向实际应用的主要问题，包括两个方面：一是移植问题，在语音识别技术走向实际应用过程中，人们发现，一个在实验室中非常成功的语音识别系统（识别率可以达到90％以上），在实际应用环境中，效果往往不是很理想，识别率甚至可能不到50％，根本无法使用；二是拒识问题，就是在系统遇到非预期的输入时，拒绝识别，防止误操作，这对检测系统或命令识别系统来说尤其重要。

语音识别系统移植性差的根源在于训练和应用的不匹配性：训练数据不可能包含所有的说话人、说话方式和背景噪声等，这样，训练出来的模型就与实际应用存在不同程度的差异，导致模型精度下降，识别率降低。为了提高系统对应用环境的鲁棒性，可以通过声学特征的补偿来实现，如各种抗噪算法，也可以通过模型层的自适应调整来实现，通常这两种方法是结合在一起使用的。为了调整系统的模型参数，我们往往需要花费较大的精力去准备训练数据，对模型进行有监督的自适应调整。如果我们能够从系统在实际应用环境下的输出结果中挑选出正确的结果，对系统模型参数进行无监督的自适应调整，那么系统的性能将会在实际应用过程中不断提高[32][33]。

语音识别中的拒识问题，则是实际应用中提出来的：在实际应用环境下，系统接收到非预期输入的情况是在所难免的，如果系统没有拒识的能力，在收到非预期输入时也根据最大似然的识别结果作出动作，则会出现很多误操作，带来许多的不便。

语音识别系统鲁棒性的这两个典型问题，实际上可以归结为同一个问题，即置信度（Confidence Measure，CM）问题。所谓置信度问题，就是在没有参考答案的情况下，如何让计算机对语音识别结果的可靠性给出一个“客观”的度量，从而使得系统可以根据这个可靠性的度量，对识别结果的对错进行判决。利用置信度信息，我们就选择识别结果可靠性比较高的部分，对系统模型进行无监督自适应，提高系统性能；或者根据识别结果的置信度，对识别结果可靠性比较低的结果进行拒识，减少系统的误操作。

可见，置信度问题的解决，对提高语音识别系统的鲁棒性具有重要作用。因此，在本文的关键词检出系统中，置信度的设计成为了主要的工作。

5.1 置信度的原理

回忆第一章绪论，曾经谈到过语音识别算法一般利用最大后验概率决策规则进行识别,得到的识别结果满足公式 1.4。

给定 X，对于所有词模型来说,$P(X)$是常量.在比较词模型后验概率的相对大小时，$P(X)$忽略不计,即识别器给出的识别结果是词表中相对最匹配的词,而不是置信度足够大的词.在实际的语音识别系统中,后处理的语音确认过程给出识别候选结果的置信度水平,并根据置信度大小接受或拒绝候选结果。

我们将影响置信度参数 cm 的因素分成两个部分：一是单独的声学置信度，即语音特征和该词的声学模型的匹配程度；二是考虑语言环境的置信度，即包含该词的上下文环境出现的概率及其声学的相似程度。这样，词的置信度就可以表示为：

$$CM\left(W[t_s,t_e]\right)=\alpha\cdot CM_{AM}\left(W[t_s,t_e]\right)+\beta\cdot CM_{LM}\left(W[t_s,t_e]\right) \tag{5.1}$$

其中，$CM_{AM}(W[t_s,t_e])$和$CM_{LM}(W[t_s,t_e])$分别为$W[t_s,t_e]$的“纯声学”和“带语言”的置信度，α 和 β 则为加权系数，用于平衡动态范围或者两者的影响比例，在不同的语法约束条件下，两个置信度的影响是不一样的，这样，词的置信度就分为“纯声学”和“带语言”两个部分。其中，“纯声学”的置信度是最基本的，与具体的语法约束无关；而“带语言”的置信度则与具体的语法约束条件有关，通常是结合声学信息一起计算的。在本文的关键词检出系统中仅仅考虑声学置信度的计算。

目前声学置信度的一般计算方法大致有两种：基于声学似然比的方法和基于声学后验概率的方法，这两种方法都是对声学似然度进行某种方式的归一化，只是归一化的分母不同而已。对词$W[t_s,t_e]$，其声学置信度的计算如下[33]：

$$CM_{AM}(W[t_s,t_e])=\frac{1}{N_W}\sum_{i=1}^{N_W}CM(ph_i) \tag{5.2}$$

式中，N_W为词 W构成音素的个数，ph_i为其第 i 个三音子（tri-phone），$CM(ph_i)$为该三音子的声学置信度，因此，词 W的声学置信度是用其三音子声学置信度的算术平均来计算的。三音子的声学置信度 $CM(ph_i)$计算如下[30]：

$$CM(ph_i)=\frac{\sum_{t=t_s^i}^{t_e^i}\log p\left(q^{(t)}\mid o^{(t)}\right)}{t_e^i-t_s^i+1}=\frac{1}{t_e^i-t_s^i+1}\sum_{t=t_s^i}^{t_e^i}\log\frac{p\left(o^{(t)}\mid q^{(t)}\right)p\left(q^{(t)}\right)}{p\left(o^{(t)}\right)}$$

$$= \frac{1}{t_e^i - t_s^i + 1} \sum_{t=t_s^i}^{t_e^i} \left\{ \log p\left(o^{(t)} \mid q^{(t)}\right) + \log p\left(q^{(t)}\right) - \log p\left(o^{(t)}\right) \right\} \quad (5.3)$$

上式中，$q^{(t)}$代表 t 时刻该音子所对应的 HMM 模型的状态，$o^{(t)}$表表示观察向量。t_s^i和t_e^i分别表示音子 i 的起始和结束帧号。$\log p\left(q^{(t)}\right)$表示语言模型的似然分，为常数可以忽略。$\log p\left(o^{(t)}\right)$为第 t 帧的归一化参数，可以是用“反模型”的似然度，也可以用“所有”模型的似然度之和，这样分别得到似然比和后验概率。可见，三音子的声学置信度是其每帧对数似然比或对数后验概率的算术平均。

本文所实现的关键词检出系统采用 N-Best 的方式来估计 $\log p\left(o^{(t)}\right)$，也即：

$$\log p\left(o^{(t)}\right) = \frac{1}{N} \sum_{n=1}^{N} \log p\left(q_n^{(t)} \mid o^{(t)}\right) \quad (5.4)$$

公式（5.2）和（5.3）是目前声学置信度最典型的计算方法，也是试验效果比较好的方法。

在关键词检出系统中，识别的过程是在连续的语音中识别出内嵌在中间的关键词作为候选，并根据置信度对候选结果进行确认的过程。根据检测任务分成几种情况。对于是否含有关键词这样的任务，关键词检出也被称为关键词确认；面对事先可以确认语音中含有并且仅有一个关键词的任务或者无约束的情况，即既可能有也可能没有，有也不一定只有一个的情况，在复杂程度以及所采取的算法上面也有一定的区别。对于可能出现多于一个关键词的情况，通常采取将出现一次以上同样的关键词的情况视为出现了多次不同的关键词并按照关键词的起始时间进行区别，对先后出现的，语音帧上重叠的关键词候选则视为同一个关键词。本文的关键词检出的置信度处理关键词出现无约束的情况。

5.2 置信度问题的难点

在关键词识别问题中，由于使用者说话方式难以限定（自然发音，不规范语法）以及背景噪声，往往会给识别结果引入大量的错误，关键词检出中的置信度问题，就是对识别结果给出一个置信度得分，使得这个置信度得分的高低直接反映识别结果的可靠程度。影响置信度的因素很多，包括识别结果自身的信息、路径搜索过程中的竞争路径信息、辅助模型给出的对比信息等，这些都对置信度的计算有不同程度的影响。

从置信度计算的基本方法来看，大致分为三类：基于特征分类器的方

法、基于似然比检验（Likelihood Ratio Test, LRT）的方法和基于后验概率的方法[35]。基于特征的方法首先是选定一个特征，并训练一个两类的分类器，然后比较待识别样本到两类的距离差，若距离差小于预先给定的阈值，则认为识别错误，反之则认为识别正确。用于设计这类分类器的特征有很多，例如：状态驻留时间、竞争路径密度、相似路径条数等；还有人将多个特征合并进行分类以获得更好的分辨率，但是这样做的前提是在假设以上多种参数之间有较低的统计相关性，显然这个前提并不正确。基于似然比检验的置信度计算方法需要用到所谓的反模型（Anti-Model）[36]，而反模型的精细程度也极大的影响似然比检验方法的有效性。基于后验概率的方法就是利用贝叶斯公式求观察序列的后验概率，最重要的就是估计分母的方法。All-phone[37]的做法是首先离线计算所有上下文相关音子序列的语言模型，然后对所有可能的上下文相关音子序列上计算似然分求和作为先验概率的估计，计算量大；基于词格网络（lattice-based）[38]的方法则是利用前向后向算法计算包含关键词命中的所有可能路径的概率作为先验概率；基于 N-Best[39]的方法则是将前 N 条得分最高的路径代替全部路径来估算先验概率，增大 N 可以提高精度然而这是以性能作为代价。

置信度问题从本质上说，就是对识别结果的正确与否进行判决的一个两类分类问题，因此，置信度研究的关键和重点在于如何寻找有效的特征，并找到从这些特征计算置信度的方法，使其区分能力达到比较好的效果。如果采用人工判决的方法，在没有参考答案的情况下，我们判断识别结果的正确与否，往往需要借助语义层次的信息，结合自己的经验和知识，才能得到比较好的判决效果，而这些信息在计算机中是很难利用的。目前用于置信度计算的信息主要包括以下三类：

(1) 识别结果本身的信息：如声学得分、语言得分、状态驻留时间、语言模型回退情况等信息，可以直接从识别结果中得到。
(2) 搜索过程中的信息：如竞争路径条数、词图密度等信息，在识别结果中不能直接得到，是在搜索的动态过程中体现出来的。
(3) 辅助模型提供的信息：如声学似然比等，不仅需要识别结果本身，还需要借助额外的模型（通常称为背景模型或反模型）计算得到。

置信度计算的难点就在于，以上所有这些信息各自的区分能力都不是很强，而彼此也很难整合到一起。另外，在不同的语法约束条件下，能够利用的语言信息往往是不一样的，很难得到统一的置信度计算方法。例如，在关键词检测或者孤立词系统识别中，就很难利用语言模型的信息进行置信度计算。

因此，置信度研究的重点和难点在于，如何有效地选择和综合利用上述信息，计算在不同语法约束条件的置信度，使其区分能力达到最好，这也是本文的研究重点所在。并非所有的信息都能有效地整合到置信度计算中，也并非所有的特征都能具有很好的区分能力，因此，在置信度计算过程中，往往就需要根据具体情况，选择比较有效的信息进行整合，才能达到比较好的效果。

5.3 算法的提出

置信度定义在不同的层次上（状态，音子，词或者句子），反映了不同的物理意义。比如对于一个局部来说，低的音子级别的置信度和音子内某个 HMM 状态的高的置信度说明了该音子是一个被错误匹配的模型，而对于一个由多个音子组成的词来说，这样错误匹配的音子个数越多，反映了该词被误识的可能性越大[40]。Cao 提出可以用不匹配音子惩罚对命令词的识别结果进行修正[41]，首先介绍几个概念：

某一帧的帧最佳音子是指识别到该帧时似然分最高的路径(token)所在的 HMM 状态代表的音子。

对某一识别结果音子，出现频率定义为这个音子内的帧最佳音子所占的比率。

因此给定一个阈值，路径上识别到该帧时的非最佳状态占当前音子所占据的帧数的比例低于该阈值的音子，定义为不匹配音子。

图 5.1 就是一个不匹配音子在识别路径上的示意图，其中大的圆形代表模型，下方的小的圆形代表状态序列，其中深色的小圆圈代表当前最佳似然分的状态，浅色的小圆圈代表非最佳似然分的状态。

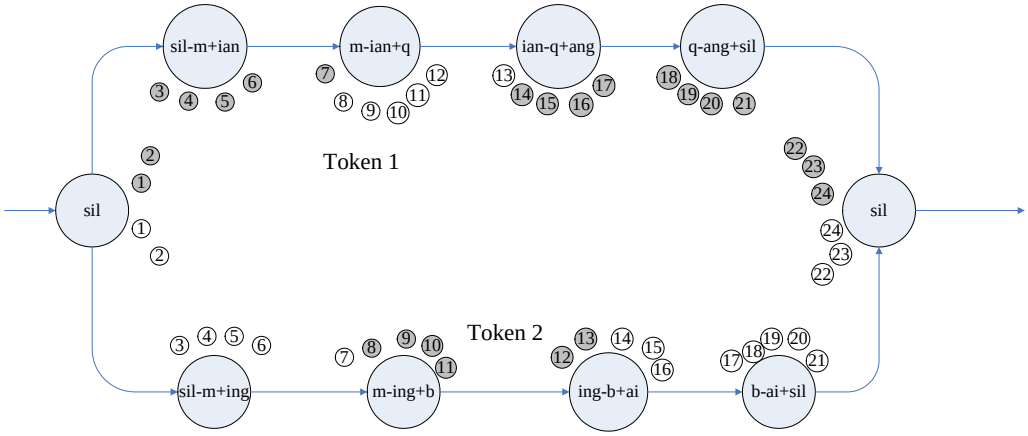

图 5.1 局部不匹配音子示意图

Cao 的做法是对传统的置信度分数乘以一个指数函数的方法作为惩罚

函数，采用统计最终结果路径上不匹配音子出现的个数对置信度分数进行修正，然而，该方法无法利用 N-Best 路径上的信息，且也仅仅适合词边界确定的情况，比如孤立词识别中。

Weintraub 提出的 LVCSR 对数似然率打分方法[42]则是充分利用了 N-Best 路径的信息，在 SRI 的 DECIPHER™ 系统中，他们将经过关键词的路径的得分与 N-Best 路径的总似然分的比率作为驻留归一化似然得分的修正。然而，面对 N-Best 路径上每一次出现候选关键词的时间无法对齐的问题，采取了一个简单的做法，就是将最大驻留归一化的候选关键词作为该置信度得分。

本文所提出的方法则是综合了以上两种方法的思路。为了在更精细的层次上描述路径上每一个分段的似然分数与当前最佳帧的最高分相互间的大小，直接采用帧而非音子来描述模型对观察序列的分数情况。若识别结果为某个音子序列，设函数 p_i 表示当前音子内不属于最佳分数的帧，也即不匹配帧的个数，对整个句子上，求不匹配帧的对全部帧数的比例：

$$M = \frac{\sum_{i=1}^{S} p_i}{T} \qquad 1 \le i \le S \tag{5.5}$$

其中 S 表示路径上音子的个数，T 为句子的全部帧数。接下来，对 N-Best 结果中某条路径上的某个关键词候选，统计不匹配帧个数占关键词长度的比例：

$$m_{key} = \frac{\sum_{i=1}^{m} p_i}{t_e - t_s + 1} \tag{5.6}$$

其中 m 表示该关键词内的音子个数，而 t_s 和 t_e 意义同上。于是，只有当 $m_{key} < M$ 的时候，认为该路径上的关键词对估计置信分数有贡献。于是，设定新的打分因子：

$$c = \frac{\sum_{q \in N-Best} \sum_{key \in q} H(m_{key}, M)\, p(o|q)\, p(q)}{\sum_{q \in N-Best} p(o|q)\, p(q)} \tag{5.7}$$

其中 q 代表 N-Best 路径当中的某一条，公式 3.3 中 $p(o|q)p(q)$ 在计算中可以用对数似然分代替。$H(m_{key}, M)$ 定义为：

$$H(m_{key}, M) = \begin{cases} 1 & M - m_{key} \ge 0 \\ 0 & otherwise \end{cases} \tag{5.8}$$

于是，c 为一个[0,1]区间上的值。由于 N-Best 输出的 N 条路径中，出现候选关键词的起始时间实际上是无法对齐的，因此，为了利用基于驻留归一化的置信度方法，我们还需要做一些约定，最自然的做法就是 Weintraub 提出的，选用平均帧似然分最大的候选关键词分段。于是，对于一个关键词,根据输出的 N-Best 结果，我们有置信度：

$$CM_{key} = c \cdot \max_{\substack{q \in N-Best \\ key \in q}} CM_{key}^{q} \tag{5.9}$$

5.4 算法评估以及实验

语音识别中的置信度问题，本质上就是一个两类分类问题：我们根据识别结果的置信度，选定一个置信度门限 T_{cm}，当某个词 $W[t_s,t_e,\mathrm{cm}]$ 的置信度 $CM \geqslant T_{cm}$ 时，就认为该词识别正确；反之，就认为该词识别错误。显然，无论置信度门限 T_{cm} 如何选定，从理论上说都会存在两种潜在的分类错误[38]：

(1) 错误拒绝（false reject, fr）：正确的识别结果，由于置信度低于给定门限而拒绝掉的部分，也称为漏报。

(2) 错误接收（false alarm, fa）：错误的识别结果，由于置信度高于给定门限而接收的部分，也称为误警。

显然，这两类错误是此消彼长的关系：为了降低漏报，就必须降低置信度门限，这样就带来更多的误报；为了降低误报，就必须提高置信度门限，这样就带来更多的漏报。因此，度量一个置信度的性能，通常必须同时考虑误报率和漏报率两个方面因素，并结合实际应用选择恰当的置信度门限。对不同的系统，误报率和漏报率的定义是不一样的。

在关键词检出系统中，置信度的误报率和漏报率定义如下：

错误拒绝率（False Rejection Rate, FRR）

$$= \frac{\text{标注有而未识别的词数}}{\text{标注的总关键词数}} \tag{5.10}$$

错误接受率（False Alarm Rate, FAR）定义为：

$$= \frac{\text{标注没有而识别出来的词数}}{\text{识别出来的个数（正确识别词数+错误识别词数）}} \tag{5.11}$$

其中正确识别词数指的是标注有并且识别正确，而错误识别词数则包括标注有而未识别的词数和标注有但是识别错误的词数。

显然，fa 和 fr 都是随着置信度门限变化的，如果从低到高调整置信

度门限，就可以得到一条 *fa-fr* 的曲线，称之为检测错误折中曲线（DET），反映漏报率随误报率变化的趋势，在门限取某特定值时，可以使得 *fa=fr*，此错误率称为置信度的等错误率（EER）。如果用正确接收率 *pd*＝100%－*fr* 代替漏报率 *fr*，同样做一条曲线，则称为工作特性曲线（ROC），反映检出率随误报率变化的趋势。

为了评估本文提出的算法，我们利用一个基于 HTK[43]的中文关键词检出系统进行了实验。实验采用 863 数据库的重庆口音普通话的 100 个男性的语音数据，全部语音在室内办公环境下录制，采样率为 16kHz,16bit。其中平均每个人用自然的语调朗读 10 个单词，20 个问句，100 句中等长度的句子作为训练集。为了避免方言标注的问题，筛掉了原有数据库的纯方言语音。每个人抽取 20 句话作为测试数据。测试数据中关键词表大小为五个，均为两字词。模型采用上下文相关扩展声韵母（eXtended Initial/Final, XIF）。

基线系统采用驻留归一化的方法,而实验中基于 N-Best 的不匹配帧加权置信度方法中 N 取 4，得到系统 EER 如表 5.1 所示：

表 5.1 两种置信度方法下系统 EER

置信度方法	EER(%)
Baseline	29.5
N-Best with MFPW	25.3

而系统的 DET（Detection Error Trade-off）曲线如图 5.2 所示：

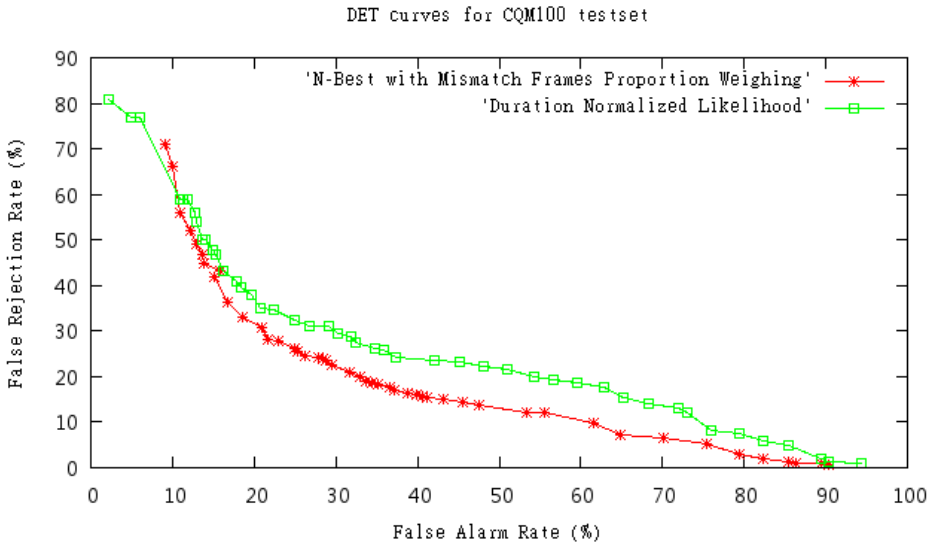

图 5.2 两种置信度方法下 DET 曲线对比

可以看到，对比单纯采用驻留归一化的置信度方法，基于 N-Best 的

不匹配帧加权的方法相比之下对于系统的 EER 有大约 4.2%的降低，相对降低了 14.2%，而适当提高系统的检出阈值（FRR = 20% 附近），系统的虚警率相对于基线系统下降较快（26%）。

综上，在 863 重庆口音普通话数据上的测试表明，采用基于 N-Best 的不匹配帧加权的置信度方法强调了候选关键词分段内声学模型的匹配程度与整个待识别语音的声学匹配程度平均水平的比较，并综合了包含候选关键词的 N-Best 候选路径的模型似然分数信息，相比与仅依靠基于驻留归一化的置信度方法，具有更好的系统性能。

5.5 本章小结

本章首先介绍了置信度的原理，并对常用的各种置信度方法进行了简单的比对，并根据关键词检出的角度深入讨论了置信度方法的难点。随后对本文提出的一种基于 N-Best 的不匹配帧加权的置信度方法进行了理论分析和推导，对算法的评估参数进行了介绍。最后，本章设计了一个对比实验。在 863 重庆口语普通话数据库上进行的实验，其结果验证了本文所提出的新的置信度方法相对于传统方法的有效性和由此带来的系统性能的提高。

第六章 结论及未来的工作

6.1 结论

本文的主要工作是实现的一个可定制的中文关键词识别系统。通过对模型、关键词集、基本语法网络的定制性问题的研究，本文提出和实现了可定制关键词识别系统。不但可以方便的使用高层语法知识，而且还可以对搜索网络通过共享机制进行有效的优化。在模型的训练方面，结合中文语音学的先验知识与决策树方法，对上下文无关的扩展声韵母模型进行了状态的共享，有效的约束了模型的数量；采用一种相对简单的语言模型和前缀树词典来扩展搜索网络，搜索算法采用基于帧同步的改进维特比算法生成 N-Best 候选作为识别器的中间结果；提出了基于不匹配帧加权的置信度方法，该方法在传统的基于驻留归一化的方法上利用不匹配帧在语音段和关键词切分段的不同比例，对 N-Best 中包含关键词的候选进行对数似然分的加权，进而得到对驻留归一化测度的惩罚。实验证明相比于单纯的基于驻留归一化的方法取得了较好的性能。

6.2 未来的工作

未来需要进一步解决的问题有几个方面。

更高的知识源的引入方法和时机，以及对搜索网络架构的影响，是需要考虑的问题。目前的关键词检出系统有效的利用增加关键词权重的方法提高了检出率，然而对于新加入的关键词而言，并无一种有效的确定权重的方法，实际应用中需要通过实验来确定有效的权值。语言置信度的使用和加入的时机，有待于进一步的实验。

噪声的鲁棒性问题。噪声问题无论对于关键词识别还是连续语音识别，均具有非常重要的作用。在一个投入真实场景的应用中，系统的性能受到说话人失配、方言、集外词以及声学和信道的失配的严重影响，而目前对于噪声问题的解决常常是从两条途径来考虑，一个就是将现实环境的噪声消除，进而还原到模型训练数据的声学场景；另外一条路径就是将模型进行变换和修正，与现实的声学场景进行匹配。对于前者而言，需要解决的问题包括寻找一种噪声鲁棒的特征，或者对现有的特征信号通过滤波消除噪声影响；对于后者而言，则包括对模型参数进行变换或者与现场快速生成的模型进行合并，生成新的模型，并利用新的模型进行传统的语音

识别。因此，对于一个关键词检出系统来说，在综合考虑实时性的同时，首先需要确定采取哪种方向的思路来解决噪声问题。

参考文献

[1] Rabiner, J. G. Wilpon and F. K. Soong. High performance connected digit recognition using hidden Markov models[C]. IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-37, PP. 1214-1225, Aug. 1989

[2] J. S. Bridle, M. Brown. An Experimental Automatic Word-Recognition System[M]. JSRU Report No. 1003, Joint Speech Research Unit, Ruislip, England. 1974

[3] F. Jelinek. A Fast Sequential Decoding Algorithm Using a Stack[J]. IBM J. Res. Develop., vol. 13, November 1969

[4] Hart, P. E.; Nilsson, N. J.; Raphael, B. Correction to "A Formal Basis for the Heuristic Determination of Minimum Cost Paths"[J]. SIGART Newsletter 37: 28–29.1972

[5] Forney GD. The Viterbi algorithm[C]. Proceedings of the IEEE 61 (3): 268–278. March 1973

[6] Lawrence R. Rabiner. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition[C]. Proceedings of the IEEE, 77 (2), p. 257–286, February 1989

[7] L. E. Baum, T. Petrie, G. Soules, and N. Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains[J]. Ann. Math. Statist, vol. 41, no. 1, pp. 164–171, 1970

[8] C. H. Lee, L. R. Rabiner. A Frame Synchronous network search algorithm for connected word recognition[C]. IEEE Trans. Acoustic Speech and Signal Processing. v37(11): 1649-1658, 1989

[9] Lowerre B. The HARPY speech recognition system[D]. PhD thesis, Dept. of Computer Science, Carnegie-Mellon University, Pittsburgh PA, USA,

1976

[10] Frank. K. Soong and Eng-Fong Huang. A Tree-Trellis Based Fast Search for Finding the N-Best Sentence Hypothesis in Continuous Speech Recognition[C]. Proceedings of the Workshop on Speech and Natural Language, Albuquerque, NM, U.S.A, pp. 705-708, 1991

[11] T.Kawahara, A.Lee, T.Kobayashi, K.Takeda, N.Minematsu, S.Sagayama, K.Itou, A.Ito, M.Yamamoto, A.Yamada,T.Utsuro, and K.Shikano. Free software toolkit for Japanese large vocabulary continuous speech recognition[C]. In Proc. IC-SLP, volume 4, pages 476–479, 2000

[12] A.Lee, T.Kawahara, and K.Shikano. Julius – an open source real-time large vocabulary recognition engine[C]. In Proc. EU-ROSPEECH, pages 1691–1694, 2001

[13] Posterior Based Keyword Spotting with A Priori Thresholds, Hamed Ketabdar, Jithendra Vepa, Samy Bengio and Hervé Bourlard, in: International Conference on Spoken Language Processing (ICSLP), 2006

[14] Discriminative Keyword Spotting , Joseph Keshet, David Grangier and Samy Bengio, Speech Communication,Volume 51, Issue 4, pp. 317-329, April 2009

[15] Joo-G. Kim, Ho-Y. Jung and Hyun-Y. Chung, “A Keyword Spotting Approach based on Pseudo N-gram Language Model”, Proc. SPECOM'04, pp.156-159, September 2004

[16] Shan Jin, Thomas Sikora, Combining Confusion Networks with Probabilistic Phone Matching for Open-vocabulary Keyword Spotting in Spontaneous Speech Signal, 17th European Signal Processing Conference (EUSIPCO 2009), Glasgow, Scotland, August 24-28, 2009

[17] Jiyong ZHANG, Fang ZHENG, Jing LI. Improved Context-Dependent Acoustic Modeling for Continuous Chinese Speech Recognition [C].

Eurospeech 2001, 3: 1617–1620

[18] J.G. Wilpon, L.G. Miller, P. Modi. Improvements and applications for key word recognition using hidden Markov modeling techniques[C]. Acoustics, Speech, and Signal Processing, ICASSP-91, vol.1, 309-312

[19] A. L. Higgins, and R. E. Wohlford. Keyword Recognition Using Template Concatenation[C]. ICASSP-85, vol.3 pp.1233-1236

[20] J. Caminero, C. De La Torre, L. Villarrubia, C. Martín, L. Hernández. On-line garbage modeling with discriminant analysis for utterance verification[C]. ICSLP-96

[21] Baum, L.E. and J.A. Eagon. An Inequality with Applications to Statistical Estimation for Probabilistic Functions of Markov Processes and to a Model for Ecology[J]. Bulletin of American Mathematical Society, 73, pp. 360-363, 1967

[22] Xuedong Huang, Alex Acero, Hsiao-Wuen Hon. Spoken Language Processing: a guide to theory, algorithm, and system development. Prentice Hall, 2001

[23] LI Jing, ZHENG Fang, and WU Wenhu. Context-independent Chinese initial-final acoustic modeling[C]. International Symposium on Chinese Spoken Language Processing (ISCSLP'00), pp. 23-26, Oct. 13-15, 2000

[24] N. J. Nilsson, Principles of artificial intelligence, Morgan Kaufmann Publishers Inc., San Francisco, CA, 1980

[25] S. Young, N. Russell, and J. Thornton. Token Passing: a Simple Conceptual Model for Connected Speech Recognition Systems[M]. Technical Report: Cambridge University Engineering Department, 1989

[26] Ney, H., et al. Improvements in Beam Search for 10000-Word Continuous Speech Recognition[C]. Proc. of the IEEE Int. Conf. on Acoustics, Speech and Signal Processing, San Francisco, California, pp.

9-12, 1992

[27] Demuynck, K., Duchateau, J. & Van Compernolle, D. A static lexicon network representation for crossword context-dependent phones[C]. Proceedings of the European Conference on Speech Communication and Technology, volume 1, Rhodos, Greece, pp. 143–146, 1997

[28] Long Nguyen and Richard Schwartz. Single-tree Method for Grammar Directed Search[C]. Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing, vol.2, Phoenix, AZ, pp.613-616, Mar. 1999

[29] Schwartz, R. and Chow, Y.-L. The N-best algorithm: An efficient and exact procedure for finding the N most likely sentence hypotheses [C]. In IEEE ICASSP-90, (1): 81–84 , 1990

[30] Renals, S. & Hochberg, M. Efficient Evaluation of The Search Space Using the NOWAY Decoder[C] .Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, Atlanta, GA, volume I, pp.149-153, 1996

[31] Robinson, T. & Christie, J. Time-first Search for Large Vocabulary Speech Recognition[C]. Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, Seattle, WA, volume II, pp.829-833, 1996

[32] Pitz, M., F. Wessel, and H. Ney. Improved MLLR speaker adaptation using confidence measures for conversational speech recognition[C]. ICSLP 2000, vol. 4, pp. 548-551

[33] Wessel, F. and H. Ney. Unsupervised training of acoustic models for large vocabulary continuous speech recognition[C]. in Proceedings Automatic Speech Recognition Workshop 2001

[34] Abdou, S. and M. Scordilis. Beam search pruning in speech recognition using a posterior probability-based confidence measure[J]. Speech

Communication 42 (2004) 409-428

[35] Hui Jiang. Confidence Measures for Speech Recognition: A Survey [J]. Speech Communication, pp. 455-470, 2005

[36] Sukkar R., Setlur A.R., Rahim M.G. and Lee C.H. Utterance Verification of Keyword Strings using Word-Based Minimum Verification Error (WB-MVE) Training [C]. IEEE International Conference on Acoustics, Speech, and Signal Processing, Atlanta, GA, May, 1996

[37] Young YS. Detecting misrecognitions and out-of-vocabulary words [C]. Proc. of the IEEE Int'l Conf. on Acoustic, Speech and Signal Processing, Vol 2. Adelaide: IEEE, 1994, 21-24

[38] Wessel F, Schluter R, Macherey K, Ney H. Confidence measures for large vocabulary continuous speech recognition [C]. IEEE Trans. on Speech and Audio Processing, 2001, 9(3):288-298

[39] Wessel F, Schluter R, Ney H. Using posterior word probabilities for improved speech recognition [C]. Proc. of the IEEE Int'l Conf. on Acoustic, Speech and Signal Processing, 2000, Vol 3.,1587-1590

[40] G. Evermann. Minimum Word Error Rate Decoding [D]. MPhil thesis, Cambridge University, 1999

[41] CAO Wenxiao, LIU Yi, Thomas Fang Zheng. Local Mismatch Phone for Confidence Measure in Standard and Accented Chinese Speech Recognition [C]. Proceedings of ISCSLP2008, Kunming, China, 2008

[42] Weintraub. LVCSR Log-Likelihood Ratio Scoring For Keyword Spotting [C]. Proceedings of the 1995 ICASSP Conference, 1995

[43] Young S., et al. The HTK book [M]. Cambridge University Engineering Department Speech Group, 1995

致 谢

首先，我衷心的感谢我的导师李银国老师和郑方老师，以及实验室的程安宇老师。三年前，是程安宇老师给予了我无与伦比的机会，让我有机会就读于美丽的重庆邮电大学，他对当时处于困境中的我的帮助和鼓励让我有机会重新回到校园；李银国老师作为我的指导老师，宛若慈父般的给予我信任与支持，并让我有机会在研究生阶段获得在清华大学学习的宝贵机会，是我学术的启蒙和步入学术殿堂的第一级阶梯；清华大学的郑方老师作为我的导师，在生活和学习上给予了我充分的指导，深厚的学术功底和非凡的个人魅力给予我深刻的影响。在清华大学的那段时光艰苦但是快乐，并对我的人生产生了奇妙的影响，激发了我对学术的兴趣和信心。在语音和语言技术中心每一位老师和同学的帮助下，我得到了飞速的成长，对自己的研究方向获得了广泛而有益的建议和认识。

还要特别感谢的老师包括清华大学电子系的肖熙老师，相处的时间不多但是您的人格魅力让我为之倾倒，祝您好人一生平安；以及实验室的老大哥刘建师兄，您好像一盏路灯默默的照料我以及和我一样的其他师弟师妹们，让我获益匪浅，祝福您有美好的明天。一并感谢的实验室同学包括清华大学的全刚同学、王宏显同学、侯珏同学、王琳琳同学、罗灿华同学、曹文晓同学、宇航同学、曹曜师兄、高嵩师兄、新加坡的留学生赖瑞平同学以及其他未能一一提到的同学，与你们相识给予我最美好的记忆，也希望你们都有灿烂的未来。

最后我要感谢我的家人，感谢我的爸爸妈妈，是你们在生活和精神上的支持得以让我完成三年的研究生学习；感谢我亲爱的姐姐，希望你能够为我而自豪；感谢我的未婚妻，是你的爱让我平静。