语音识别技术学习：原理部分

FOROSS

wushexu@163.com

2013/12

1 概述

本文档介绍语音识别技术，具体来说是基于 HMM 的语音识别技术。

语音识别本身是很复杂的技术。前面先用几章来介绍相关的概念和原理，这些是比较基础的知识，大部分都是理解实际例子的必要基础知识，语音应用的主要开发人员都应该了解。为易于理解，尽量不涉及数学公式和算法细节。如果要深入理解相关原理，还要进一步阅读相关资料。

文档用两章来分别介绍两个语音平台：HTK 和 CMU Sphinx，包括基于这两个平台的应用开发的例子，也包括建立声学模型、语言学模型和字典的步骤。在前面的章节也会提到这两个平台。

1.1 语音识别

语音识别（Speech Recognition, SR）是机器通过识别和理解过程把人类的语音信号转变为相应的文本或命令的技术。除了 SR，它也被称为 ASR (automatic speech recognition)、computer speech recognition、STT (speech to text)。

语音识别技术的根本目的是研究出一种具有听觉功能的机器，这种机器能直接接收人的语音，理解人的意图，并作出相应的反应。技术上看，它属于多维模式识别和智能接口的范畴。语音识别技术是一项集声学、语音学、计算机、信息处理、人工智能等于一身的综合技术，可广泛应用在信息处理、通信与电子系统、自动控制等领域。

要达到让机器听懂人类的语言的目标，面临着诸多的困难。这些困难具体表现在：

语音信号的声学特征随与之前后相连的语音的不同而又很大的变化，且连续语音流中各语音单位之间不存在明显的边界

语音特征随发音人的不同、发音人生理和心理状态的变化而发生很大的差异

环境噪声和传输设备的差异也将直接 影响语音特征的提取

一个语句所表达的意思与上下文内容、说话时的环境条件及文化背景等因素有关，而语句的语法结构又是多变的，并且语境信息几乎是计算机语音识别无法利用的，所有这些都给语意的理解带来很大的困难

由于出发点不同，识别又分为**说话人识别**和**语音识别**。

就说话人识别来看，可分为与文本有关和与文本无关两类。从用途上看，可分为说话人辨认和说话人确认。前者判定某一待识别的声音是多个说话者中的一个，是多选一的问题，属于闭集识别范畴。后者判定一个待识别的声音是或不是某一特定话者的语音，其输出只有两种结果，为肯定或者否定的问题。

1.2 语音识别的分类方法

就语音识别而言，也存在着以下几种不同的分类方法。

1.2.1 按词汇量大小分

每个语音识别系统都有一个词汇表，系统只能识别词汇表中所包含的单词。通常按词汇量可分为小词汇量、中词汇量和大词汇量。一般小词汇量包括 10~100 个词，中词汇量包括 100~500 个词条，相应的大词汇量至少包含 500 个以上的词条。一般情况下语音识别的识别率会随着词汇量大小的增加而下降，因此，语音识别的研究困难是随着词汇量的增加而逐渐增加的。

1.2.2 按发音方式分

语音识别可以分为孤立词识别、连接词识别、连续语音识别以及关键词检出等。

在孤立词识别中，机器只是识别一个个孤立的音节、词或者短语等，并给出具体识别结果

在连续语音识别中，机器识别连续自然的书面朗读形式的语音

在连接词识别中，发音方式介于孤立词和连续语音之间，它表面上看像连续语音发音，但能明显感觉到音与音之间有停顿。这是通常可以采用孤立词识别技术进行串接来实现

对关键词检出，通常用于说话人以类似自由交谈的方式的发音，这种发音称为自发发音方式；在这种发音方式下，存在着各种各样影响发音不流畅的因素，如犹豫、停顿、更正等，并且说话人发音中存在大量不是识别词表中的词，判断理解说话人的意思，只从其中一些关键的部分就可做出决定，因此只需要进行其中的关键词识别

1.2.3 按说话人分

可分为特定说话人和非特定说话人两种。前者只能识别固定某个人的声音。其他人要想使用这样的系统，必须实现输入大量的语音数据，对系统进行训练；而对后者，机器能识别任意人的发音。由于语音信号的可变性很大，这种系统要能从大量的不同人的发音样本中学习到非特定人的发音速度、语音强度、发音方式等基本特征，并归纳出其相似性作为识别的标准。使用者无论是否参加过训练都可以使用一套参考模板进行语音识别。

从难度上看，特定说话人的语音识别比较简单，能得到较高的识别率，并且目前已经有商品化的产品；而非特定人识别系统，通用性好、应用面广、但难度也大，不容易获得较高的识别率。

1.2.4 从语音识别的方法分

有模版匹配法，随机模型法和概率语法分析法。见下一节。

1.3 语音识别的主要方法

主要识别方法有模版匹配法、随机模型法和概率语法分析法。这些方法都属于统计模式识别方法。

其识别过程大致如下：首先提取语音信号的特征构建参考模版，然后用一个可以衡量未知模式和参考模板之间的似然度的测度函数，选用一种最佳准则和专家知识做出识别决策，给出识别结果。

模版匹配法是将测试语音与参考模板的参数一一进行比较与匹配，判决的依据是失真测度最小准则。

随机模型法是一种使用隐马尔可夫模型（HMM）来对似然函数进行估计和判决，从而得到相应的识别结果的方法。由于隐马尔可夫模型具有状态函数，所以这个方法可以利用语音频谱的内在变化（如说话速度、不同说话人特性等）和它们的相关性。

概率语法分析法适用于大范围的连续语音识别，它可以利用连续语音中的语法约束知识来对似然函数进行估计和判决。其中，语法可以用参数形式来表示，也可以用非参数形式来表示。

语音识别中最简单的是特定人、小词汇量、孤立词的语音识别，最复杂最难解决的是非特定人、大词汇量、连续语音识别。无论是哪一种，当今采用的主流算法仍然是隐马尔可夫模型方法。

在隐马尔可夫模型方法之前，比较成功的方法是 DTW（Dynamic Time Warping），这是模板匹配的一种方法，能把不同时间长度的两个序列进行对齐，再计算距离。它是把时间归正和距离测度结合起来的一个种非线性归正技术。目前 DTW 在孤立词识别、连接词识别中仍是很有效的方法。

近年来，基于神经网络、支持矢量机、遗传算法等语音识别技术方兴未艾。

1.4 学习资源

HMM	维基条目（http://en.wikipedia.org/wiki/）： Hidden_Markov_model、Viterbi_algorithm、Baum–Welch_algorithm、 Forward_algorithm、Forward-backward_algorithm、EM_algorithm
	http://jedlik.phy.bme.hu/~gerjanos/HMM/hoved.html
	http://cs.brown.edu/research/ai/dynamics/tutorial/Documents/HiddenMarkovModels.html
	http://www.comp.leeds.ac.uk/roger/HiddenMarkovModels/html_dev/main.html
语音 信号 处理	维基条目： Fourier_transform、Fast_Fourier_transform、Audio_signal_processing、 Window_function、Linear_predictive_coding
	http://practicalcryptography.com/miscellaneous/machine-learning/guide-mel-frequency-cepstral-coefficients-mfccs/
	http://www.data-compression.com/vq.shtml
语音 识别	维基条目： Speech_recognition 、List_of_speech_recognition_software、 Dynamic_time_warping、Beam_search、A*_search_algorithm、 Perplexity
	http://www.fifthgen.com/speaker-independent-connected-s-r.htm
	http://src.chromium.org/viewvc/chrome/trunk/src/content/browser/speech/
HTK	http://htk.eng.cam.ac.uk/
	官方手册 The HTK Book（PDF）

	http://nclab.kaist.ac.kr/~twpark/htkbook/node2_tf.html
Sphinx	http://cmusphinx.sourceforge.net/
	Sphinx4 文档： Sphinx4Whitepaper.pdf、Architecture.pdf、sphinx4Overview.pdf、 sphinx4ArchitectureOverview.pdf
	sphinx4 API： http://cmusphinx.sourceforge.net/sphinx4/javadoc/index.html
	pocketsphinx API： http://cmusphinx.sourceforge.net/api/pocketsphinx/
图书	语音信号处理（第 2 版）（韩纪庆等编著）
	Spoken Language Processing A Guide to Theory Algorithm and System Development

2 HMM 与语音识别

隐马尔可夫模型（Hidden Markov Models, HMM）作为语音信号的以一种统计模型，在语音处理的各个领域中获得了广泛的应用。现在已成为公认的有效的语音识别方法。

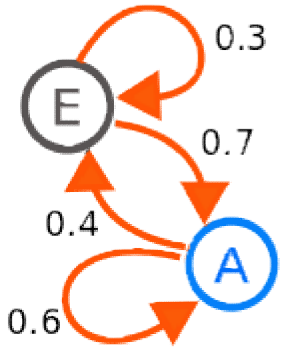
2.1 马尔可夫链

马尔可夫性质是概率论中的一个概念。当一个随机过程在给定现在状态及所有过去状态情况下，其未来状态的条件概率分布仅依赖于当前状态；换句话说，在给定现在状态时，它与过去状态（即该过程的历史路径）是条件独立的，那么此随机过程即具有马尔可夫性质。具有马尔可夫性质的过程通常称之为马尔可夫过程。

马尔可夫链是数学中具有马尔可夫性质的离散时间随机过程。该过程中，在给定当前知识或信息的情况下，只有当前的状态用来预测将来，过去（即当前以前的历史状态）对于预测将来（即当前以后的未来状态）是无关的。

马尔可夫链由发射状态集和状态之间的转移集来定义。状态用圆表示，而转移用箭头表示。每个转移有一个相关联的转移概率，表示从给定的当前状态转移到下一步状态的条件概率。给定状态的所有转移概率之和必须等于 1。

下面是一个有两个状态的马尔可夫链的例子。

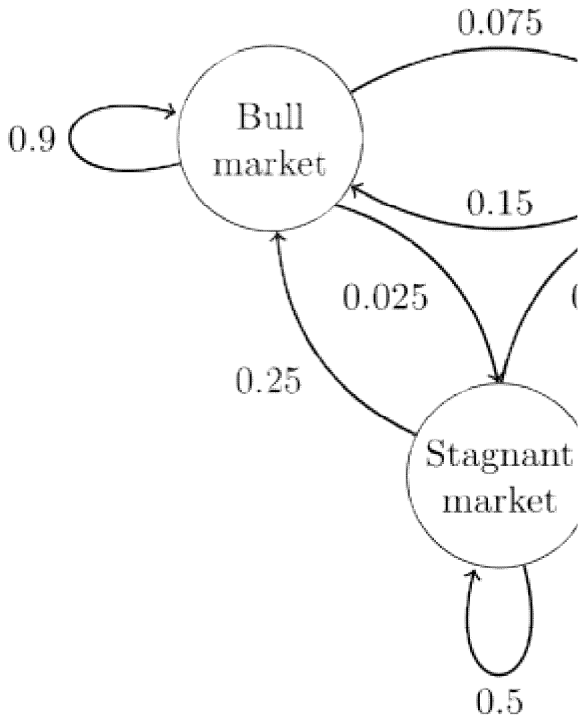


上面例子表明：

如果当前的状态是 E，那么下一个状态有 0.3 的概率仍为 E，有 0.7 的概率变为 A；

如果当前的状态是 A，那么下一个状态有 0.6 的概率仍为 A，有 0.4 的概率变为 E。

以下是一个假想的股市的状态转换图，构成一个马尔可夫链。它给出了给定一周是牛市、熊市，还是萧条的走势。



它的状态空间为{1 = bull, 2 = bear, 3 = stagnant}，它的转移矩阵为：

$$P = \begin{bmatrix} 0.9 & 0.075 & 0.025 \\ 0.15 & 0.8 & 0.05 \\ 0.25 & 0.25 & 0.5 \end{bmatrix}.$$

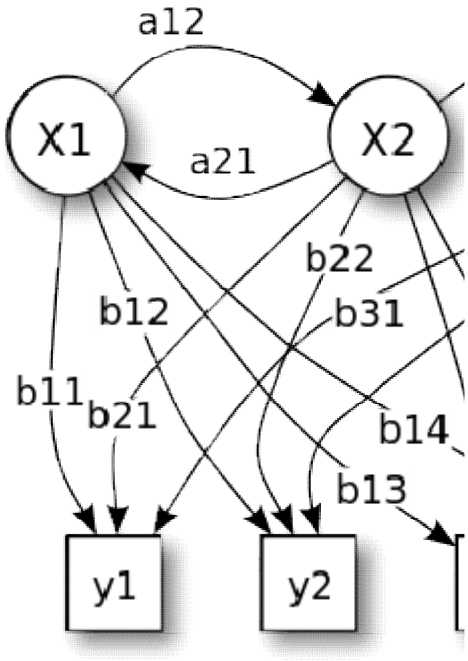
马尔可夫链是生成序列的模型，其中一个符号的概率仅依赖与前一个符号。描述马尔可夫链的最重要的参数就是转移概率矩阵。但转移矩阵决定不了初始分布，因此除了转移矩阵之外，还必须引进初始概率（一般记为 π）。

在实际当中，马尔可夫链的每一状态可以对应于一个可观测到的物理事件。比如天气预报中的雨、晴、雪等，这时可称之为天气预报的马尔可夫链模型。根据这个模型，可以计算出各种天气（状态）在某一时刻出现的概率。

2.2 隐马尔可夫模型（HMM）

HMM 是在马尔可夫链的基础之上发展起来的。由于实际的问题比马尔可夫链模型描述的更为复杂，观察到的事件并不是与状态一一对应，而是通过一组概率分布相联系，这样的模型就成为 HMM。它是一个**双重随机过程**，其中之一就是马尔可夫链，这是基本随机过程，它描述状态的转移。另一个随机过程描述状态和观察值之间的统计对应关系。这样站在观察者的角度，只能看到观察值，不像马尔可夫链模型中的观察值和状态一一对应，因此，不能直接看到状态，而是通过一个随机过程去感知状态的存在及其特性。因而称之为“隐”马尔可夫模型。

以下是一个 HMM 例子，状态为 X1、X2、X3，观察值为 y1、y2、y3、y4。



一个 HMM 可以由下列参数描述：

N：模型中马尔可夫链状态数目

M：每个状态对应的可能的观察值数目

π：初始状态概率

A：状态转移概率矩阵

B：观察值概率矩阵

这样可以记一个 HMM 为

λ=(N,M,π,A,B)

或简写为

λ=(π,A,B)

2.2.1 一个具体例子

假设你有一个住得很远的朋友,，他每天跟你打电话告诉你他那天做了什么。你的朋友仅仅对三种活动感兴趣：公园散步，购物以及清理房间。他选择做什么事情只凭天气。你对于他所住的地方的天气情况并不了解，但是你知道总的趋势。在他告诉你每天所做的事情基础上，你想要猜测他所在地的天气情况。

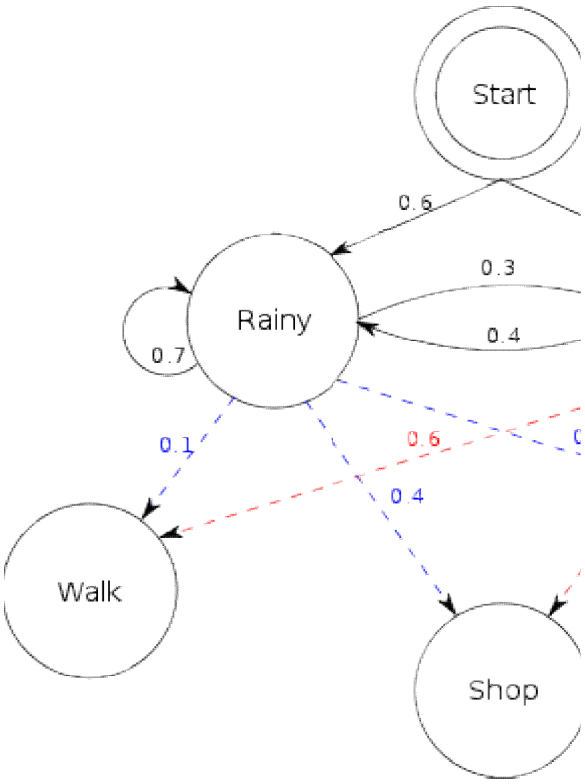
你认为天气的运行就像一个马尔可夫链，其有两个状态 "雨"和"晴"，但是你无法直接观察它们，也就是说，它们对于你是隐藏的。每天，你的朋友有一定的概率进行下列活动：

“散步”，“购物”，或“清理”。因为你朋友告诉你他的活动，所以这些活动就是你的观察数据。这整个系统就是一个隐马尔可夫模型 HMM。

你知道这个地区的总的天气趋势，并且平时知道你朋友会做的事情。也就是说这个隐马尔可夫模型的参数是已知的。你可以用程序语言(Python)写下来：

```
states = ('Rainy', 'Sunny')

observations = ('walk', 'shop', 'clean')

start_probability = {'Rainy': 0.6, 'Sunny': 0.4}

transition_probability = {
  'Rainy' : {'Rainy': 0.7, 'Sunny': 0.3},
  'Sunny' : {'Rainy': 0.4, 'Sunny': 0.6},
  }

emission_probability = {
  'Rainy' : {'walk': 0.1, 'shop': 0.4, 'clean': 0.5},
  'Sunny' : {'walk': 0.6, 'shop': 0.3, 'clean': 0.1},
  }
```

在这些代码中，start_probability 代表了你对于你朋友第一次给你打电话时的天气情况的不确定性（你知道的只是那个地方平均起来下雨多些）。在这里,这个特定的概率分布并非平衡的，平衡概率应该接近（在给定变迁概率的情况下）{'Rainy': 0.571, 'Sunny': 0.429}< transition_probability 表示基于马尔可夫链模型的天气变迁,在这个例子中，如果今天下雨，那么明天天晴的概率只有 30%。代码 emission_probability 表示了你朋友每天做某件事的概率。如果下雨，有 50% 的概率他在清理房间；如果天晴，则有 60%的概率他在外头散步。



2.2.2 三个基本问题

对于一个特定的 HMM，有三个感兴趣的问题，分别是估计问题、解码问题和学习问题。这些都属于推理问题。

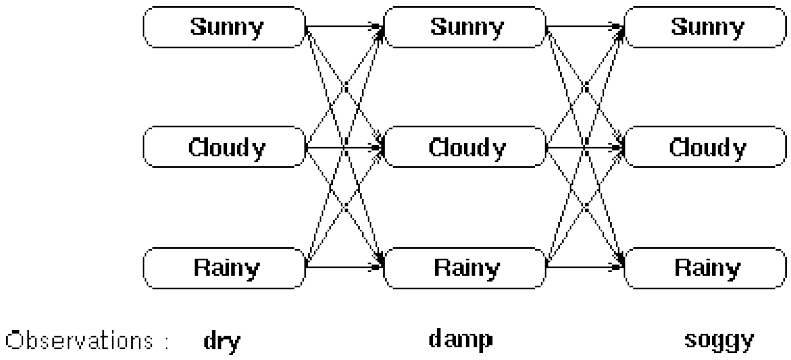
HMM 是语音识别的核心技术，而理解这三个问题对于理解 HMM、乃至语音识别系统至关重要。

2.2.2.1 估计

给定一个 HMM 模型（λ）和一个特定的观察序列（$O=o_1,o_2,...o_T$），计算这个序列由这个模型产生的概率（求 $p(O|\lambda)$），或者说模型产生这个观察序列的概率是多少。

由于多个状态都可能产生同一个观察值，因此有很多状态序列都可能产生同样的观察序列。每个状态序列都有一定的发生概率，而在每个状态序列下产生某个观察序列也有一定的概率。要计算模型产生这个观察序列的概率，就要把所有可能的状态序列下的产生这个观察序列的条件概率累加起来。

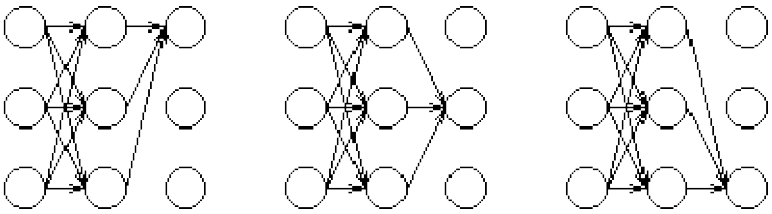
例如有一个 HMM 模型，有三个状态：Sunny、Cloudy、Rainy，三个状态之间都可以互相转换。产生观察序列为 dry,damp,soggy，则可以由下面的格型结构来表示：



要计算模型产生观察序列 dry,damp,soggy 的概率，则要考虑所有的路径（状态转移序列），每个路径的发生概率，以及这个路径下产生这个观察序列的概率（条件概率）。

如果直接累加所有可能的条件概率，计算量会非常惊人，大约 $2TN^T$ 的数量级（N 是状态数量，T 是序列长度）。

前向算法（forward algorithm）是解决这一问题的有效算法。前向算法是对观察序列长度从 1 到 T 的递推过程。每一次递推，都考虑了所有的状态转移。例如下面从 t=2 到 t=3 的递推，共有 9 种状态转移。而从 t=1 到 t=2，也有 9 种状态转移。



前向算法是一种动态规划算法，计算序列的概率时会使用已计算的子序列的概率，因此能够大大减少计算量，只需要进行 N^2T 次运算。

这个问题也可以用**后向算法**来解决。它与前向算法类似，只不过是从 t=n 到 t=n-1 来进行递推。

2.2.2.2 解码

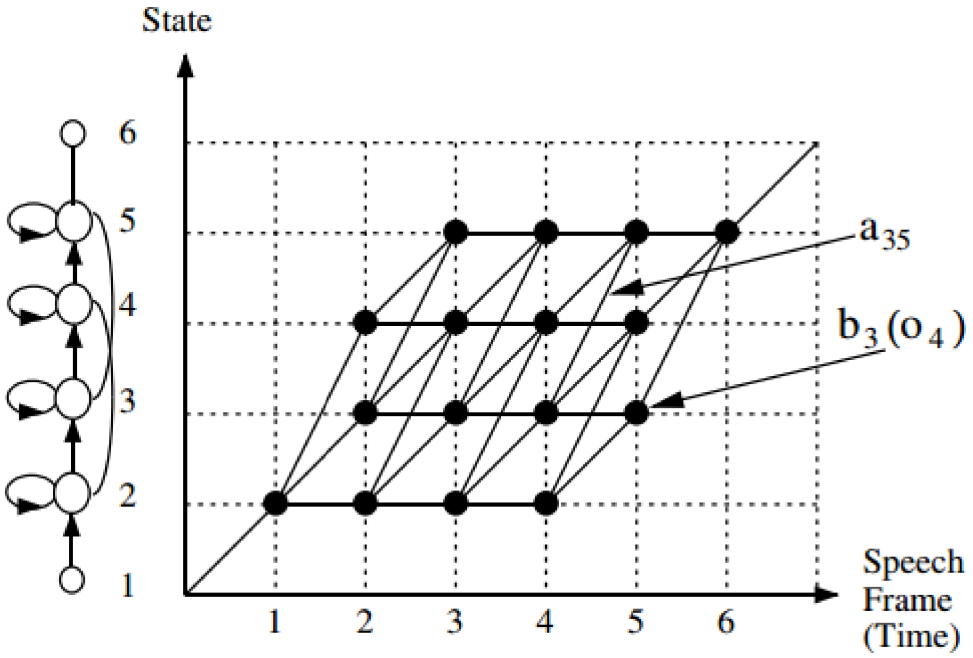
给定一个 HMM 模型（ λ ）和一个特定的观察序列（$O=o_1,o_2,...o_T$），寻找最可能的能产生这个序列的隐含状态的序列。即什么样的状态转移最可能产生这个观察序列。

这个问题可以用**维特比算法**（Viterbi 算法）来解决。

对于给定一个状态序列，可以直接计算出产生某个观察序列的概率（每个状态的转移概率乘以发射概率）。如果计算出所有可能的状态序列下的概率，从而找出最大的概率对应的状态序列，是可以的。但是可能的状态序列数量会非常大（指数级），计算量会非常大。

维特比算法同样是使用了动态规划算法，因而可以大大减少计算量。维特比算法也是一种格型结构，而且和前向算法类似。同样，有后向算法的思想出发，也可以推导出维特比算法的另一种实现方式。

下面是维特比算法的一个格型结构（纵轴是状态，横轴是时间）。



2.2.2.2.1 一个例子

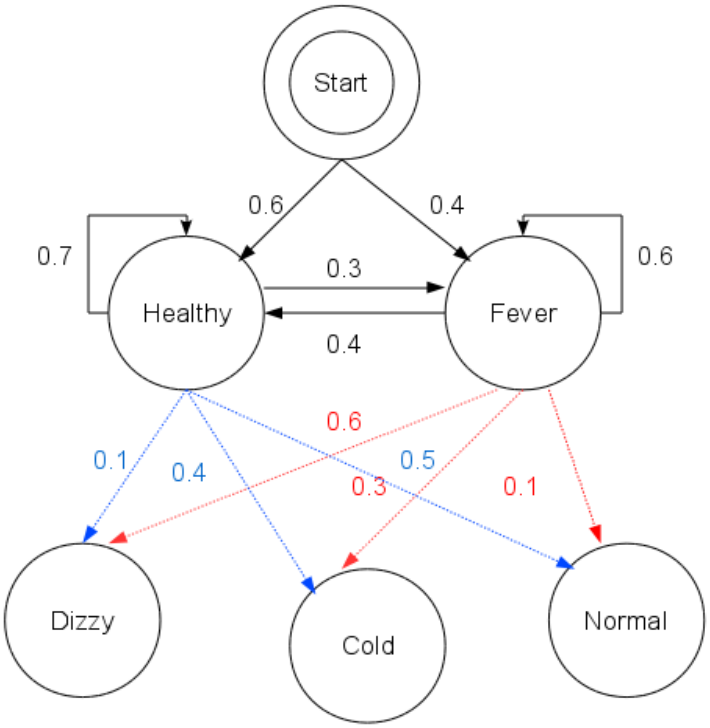
想象一个乡村诊所。村民有着非常理想化的特性，要么健康要么发烧。他们只有问诊所的医生的才能知道是否发烧。 聪明的医生通过询问病人的感觉诊断他们是否发烧。村民只回答他们感觉正常、头晕或冷。

假设一个病人每天来到诊所并告诉医生他的感觉。医生相信病人的健康状况如同一个离散马尔可夫链。病人的状态有两种“健康”和“发烧”，但医生不能直接观察到，这意味着状态对他是“隐含”的。每天病人会告诉医生自己有以下几种由他的健康状态决定的感觉的一种：正常、冷或头晕。这些是观察结果。 整个系统为一个隐马尔可夫模型(HMM)。

医生知道村民的总体健康状况，还知道发烧和没发烧的病人通常会抱怨什么症状。 换句话说，医生知道隐马尔可夫模型的参数。 这可以用 Python 语言表示如下：

```
states = ('Healthy', 'Fever')

observations = ('normal', 'cold', 'dizzy')

start_probability = {'Healthy': 0.6, 'Fever': 0.4}

transition_probability = {
  'Healthy' : {'Healthy': 0.7, 'Fever': 0.3},
  'Fever' : {'Healthy': 0.4, 'Fever': 0.6},
  }

emission_probability = {
  'Healthy' : {'normal': 0.5, 'cold': 0.4, 'dizzy': 0.1},
  'Fever' : {'normal': 0.1, 'cold': 0.3, 'dizzy': 0.6},
  }
```

在这段代码中, 起始概率 start_probability 表示病人第一次到访时医生认为其所处的 HMM 状态，他唯一知道的是病人倾向于是健康的。这里用到的特定概率分布不是均衡的,如转移概率大约是{'Healthy': 0.57, 'Fever': 0.43}。 转移概率 transition_probability 表示潜在的马尔可夫链中健康状态的变化。在这个例子中，当天健康的病人仅有 30%的机会第二天会发烧。发射概率 emission_probability 表示每天病人感觉的可能性。假如他是健康的，50%会感觉正常。如果他发烧了，有 60%的可能感觉到头晕。



病人连续三天看医生，医生发现第一天他感觉正常，第二天感觉冷，第三天感觉头晕。于是医生产生了一个问题：怎样的健康状态序列最能够解释这些观察结果。维特比算法解答了这个问题。

以观察序列['normal', 'cold', 'dizzy']为例，通过运行维特比算法，可以发现这个观察序列最有可能由状态序列 ['Healthy', 'Healthy', 'Fever']产生。 换句话说，对于观察到的活动, 病人第一天感到正常，第二天感到冷时都是健康的，而第三天发烧了。

2.2.2.3 学习

给定一个 HMM 模型（λ）和一个特定的观察序列（$O=o_1,o_2,...o_T$），怎样调整模型参数 (π,A,B)，才能使产生这个观察序列的可能性最大（即 p(O|λ)最大）？

这是 HMM 的训练问题，通过学习或调整模型的参数，以便更好地预测未来状态序列的路径。由于给定的训练序列有限，因而不存在一个估计模型参数的最佳方法。与前面的两个问题相比，是最困难的问题。

解决这个问题，依据不同的最优准则，有不同的方法。主要有两个最优准则，最大化似然度（Maximum Likelihood）和最大化互信息（Maximum Mutual Information）。最大化似

然度准则是在给定模型参数的条件下，让产生给定序列的概率（条件概率）最大化；最大化互信息准则是让模型参数和产生给定序列的联合概率最大化。

2.2.2.3.1 最大化似然度准则

Baum-Welch 算法是最大似然度准则下的一个参数估计方法，它是一种期望最大化（Expectation Maximization, EM）算法。EM 算法是一族算法，用于学习涉及隐藏状态问题的概率模型。

Baum-Welch 算法利用递归的思想，寻找局部最优解。首先，把这些概率（模型参数）估计为任意值。然后，不断地重新估计这些概率，直到收敛。重新估计这些概率是通过著名的**重估（re-estimation）公式**（请参考相关资料）来计算的。

最大似然度准则下，除了 Baum-Welch 算法，还有基于梯度的方法。

Baum-Welch 算法只是得到广泛应用的解决这一问题的经典方法，但并不是唯一的，也远不是最完善的方法。

2.2.2.3.2 最大化互信息准则

最大似然度并不是唯一的准则，也不是所有情况都适用的准则。研究表明：当事先假定的模型不正确时，最大互信息估计器优于最大似然估计器。

但目前对最大互信息估计还没有找到类似于最大似然估计中的前向-后向算法那样有效的方法，因此，一般采用经典的最大梯度法。

2.3 离散、连续和半连续的 HMM

典型的 HMM 是离散的，但是可以扩展为连续和半连续的 HMM。离散 HMM 的精确性比连续 HMM 低，但计算量更少。

HTK 对离散、连续和半连续的 HMM 均支持。CMU Sphinx 目前不支持离散 HMM。

2.3.1 离散 HMM

前面所说的 HMM 基本都是离散 HMM，离散 HMM 的观察值空间是一个有限集合。有限集合意味着要把连续的值空间映射到一个离散空间，这就涉及矢量量化技术。

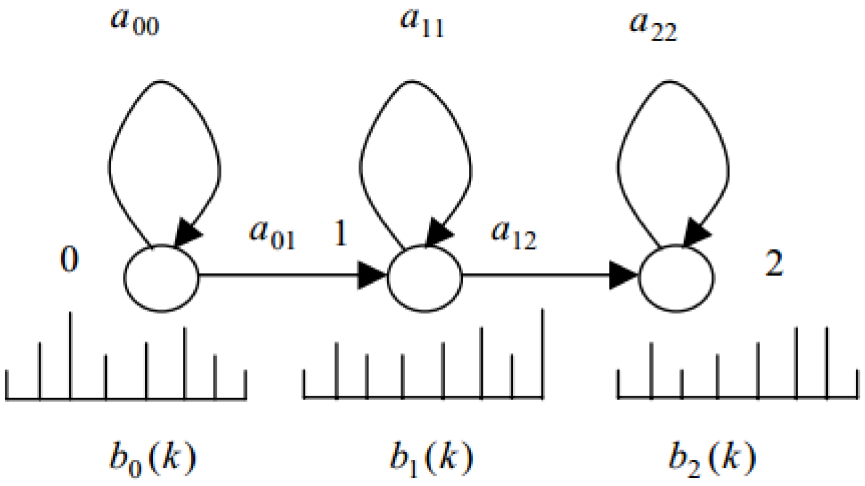
矢量量化（VQ）技术把特征向量空间分为若干子空间，每个子空间用一个中心向量来表示，表征这个中心向量的是一个码字，所有码字的集合构成码本。这样计算输出概率时，是针对矢量量化后的码字进行的，通过简单的计数就可以实现，缺点是描述误差比较大。

2.3.2 连续 HMM

如果观察值不是来自于有限的集合，而是来自于一个连续空间，前面讨论的离散输出分布则需要修改。离散 HMM 和连续 HMM 之间的差异就在于输出概率函数的形式。对于语音识别，连续 HMM 的使用意味着，对于离散 HMM 中需要的从连续空间映射观察矢量到离散空间的量化过程是不必要的。

在选择一个连续概率密度函数时，首选就是高斯混合密度函数（Gaussian mixture density，GMD）。当混合数足够大时，GMD 可以比较准确的描述特征向量的概率密度，可以用 EM 算法估计出概率密度。

以下是一个连续 HMM，其输出概率是一个概率密度函数。

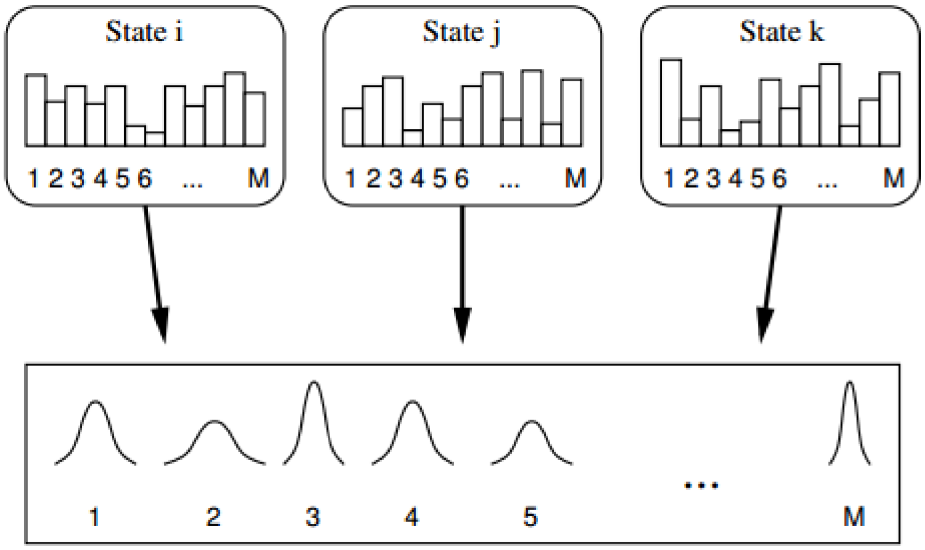


2.3.3 半连续 HMM

传统上，离散和连续混合密度 HMM 是分别地对待的。事实上，这两者之间的缺口是可以缩小的。虽然 GMD 描述方法中需要存储的参数不多（均值、方差、转移矩阵等），但是当混合数很大时，也比较浪费空间，半连续 HMM 结合 VQ 技术和连续密度描述的特点 较好的解决了这个问题。这里，所有模型公用 L 个类似于码字的密度函数，记录一个模型中不同状态的概率密度函数只需要一组系数即可。

半连续 HMM 其实就是所说的 Tied-Mixture System（在 HTK 中就只说 Tied-Mixture）。

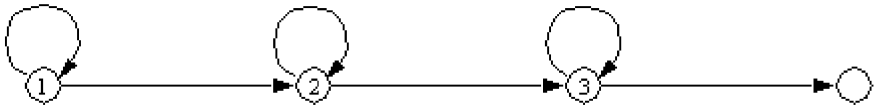
如下图所示，所有高斯密度函数组件都存储在一个池里（Tied-Mixture Codebook），所有状态输出分布共享这个池。



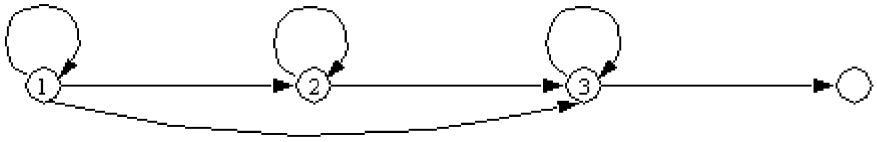
2.4 HMM 实现、训练中的问题

2.4.1 拓扑结构

在语音识别应用中，一个 HMM 表示一个基本声学单元（基元），拓扑一般为有限的几个状态，每个状态都包括自跳转弧。如下图是 HMM 的马尔可夫链部分，有 4 个状态，最后一个是不发射的终结状态。



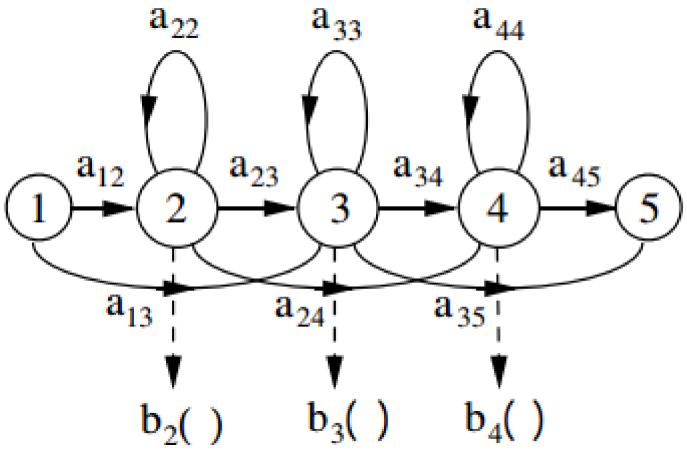
下图与上图类似，但是有状态跨越。



上面两种马尔可夫模型有一个特点，从状态 1 出发，沿状态序号增加的方向转移，最终停在最后一个状态。由这种马尔可夫链构成的 HMM，一般称之为左－右模型。这种模型在实际语音处理应用中被广泛采用，尤其是在孤立词识别中。

为了方便，一般在实现中都会给模型增加不发射的初始和终结状态。在语音识别工具 HTK 中，HMM 模型一般是 5 个状态，包括不发射的初始和终结状态。

下面是一个简单的左－右模型，中间三个状态是发射状态。



2.4.2 初始模型选取

Baum-Welch 算法由训练数据得到 HMM 参数时，一个重要问题就是初始模型的选取。不同的初始模型有不同的将产生不同的训练结果。选取好的初始模型，是最后求出的局部极大与全局最大接近是非常重要的。

实际处理是有一些经验方法，π（初始概率）和 A（转移概率）对初始值的选取结果影响不大，只要满足约束条件。B（发射概率）的初值对训练出来的 HMM 影响较大，一般倾向采取较为复杂的初值选取方法。初始模型可以任意选取，在用重估公式递推改进。

HMM 有很多种类型，针对不同的 HMM，也可以选取不同的初值选取方法。

2.4.3 数据下溢问题

前向-后向算法和 Baum-Welch 算法中，都有概率的递推计算，所有量都小于 1，因此随着 t 的增加，它们将迅速趋向零。为了解决这种数据下溢问题，通常可以采取增加比例因子的方法，对有关算法加以修正。

做了上述处理后，为了保证所有公式计算结果不变，必须在常用计算公式中做相应处理，以消去比例因子的影响。如概率 P(O| λ)的计算公式，重估公式，Veterbi 算法的处理。

对 Viterbi 算法，为防止数据下溢可采用对数化处理。事实上，语音识别中通常是比较多个概率值之间的相对大小，并由此做出决策。因此取对数运算后，既可以防止概率值的下溢，又不会影响多个概率值之间的大小关系。

2.4.4 训练数据的不足

根据 HMM 的定义，一方面，一个 HMM 的模型还有很多待估计的参数，因此为了得到满意的模型，必须要有很多训练数据，实际中很难办到。另一方面，选择规模较小的模型，既减少模型中的状态数和每个状态上的混合高斯分量数，也有实际的困难。训练数据少的情况下，一些出现次数很少的观察值没有包含在整个训练数据库中，这样训练出来的 HMM 参数中就会有不少为零的概率值。事实上，在实际语音识别测试中，这样观察值有可能出现，因而需要对训练好的模型进行相应的处理。

一种常用的方法是将一个训练较充分，但细节较差的模型与一个训练虽不充分，但细节较好的模型进行混合。前一个模型可以在 HMM 模型结构中将有些状态转移概率及观察输出概率相近的进行“捆绑”，即一些转移概率或观察输出概率共享相同的值，从而可以减少模型参数。这样使用相同的训练数据就可以对这种“捆绑”后的模型进行较充分的训练。问题的关键是合并的权值的估计。

2.4.5 处理说话人的影响

由于语音的动态范围很大，不同说话人的语音，甚至同一说话人在不同时间和场合的语音都有很大的不同，因此训练 HMM 时，充分考虑说话人的影响，对于较好的估计 HMM 参数是十分重要的。

这个问题可以表述为：设训练数据集 A，所训练出来的模型 λ 。模型 λ 较好的反映了 A 的特性。如果又增加一个训练数据集 B，希望经过一个处理过程，B 的特性也能反映在

结果模型之中。B 相对于 A 来说，可以使不同说话人的语音，也可以是同一说话人在不同时间所发出的语音。因此，这个问题对语音识别，尤其是非特定人语音识别是很有意义的。

根据 Baum-Welch 算法，一个直接处理方法是一起使用 A 和 B 重新训练一个模型。但这样做，一方面不经济，没有利用已经训练好的模型 1 的信息，另一方面，实现起来也有困难，因为在很多实际场合中并没有保留训练数据集 A，而只保存了反映其特性的占用很少存储空间的模型 λ 。

另一个既简单又容易的想到的方法为，以 λ 为初始模型，用数据 B 通过重估公式进行若干次的迭代，得到新模型 λ^*。但是很显然，这个 λ^*只能较好的反映数据集 B 的特性，而不可能同时很好的反映出 A 的特性。

针对这个问题，经过分析 Baum-Welch 算法，可以给出一种处理说话人影响的方法。它在小词汇量语音识别和大词汇量语音识别中都有成功的应用。

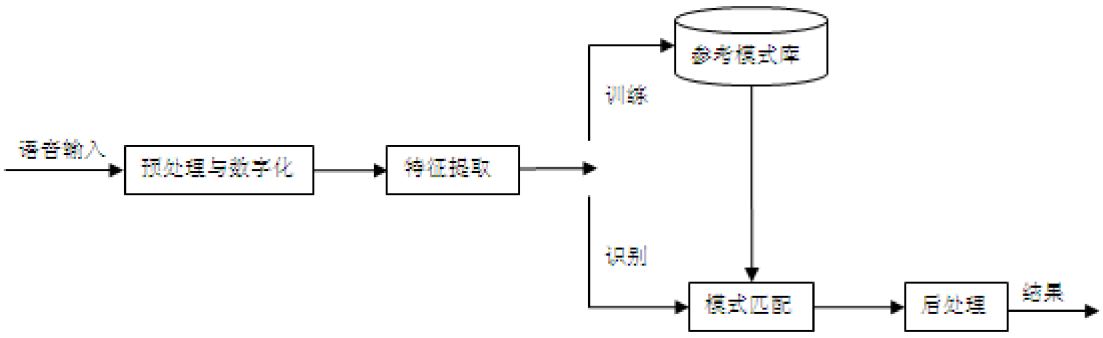
这个方法要改写重估公式（请参考相关资料），改写之后就能使 HMM 参数估计的过程具有很好的自适应性和很强的自学能力。只要增加新的训练数据，通过这种方式最后产生的模型就能反映这些新增数据的信息。

3 语音识别系统

介绍语音识别系统的一般处理过程，和基本架构。展开讲述处理过程中的信号处理、特征分析和特征提取的内容。其他部分在后两章讲述。

3.1 一般过程

语音识别系统本质上是一种模式识别系统。它的基本框架是：

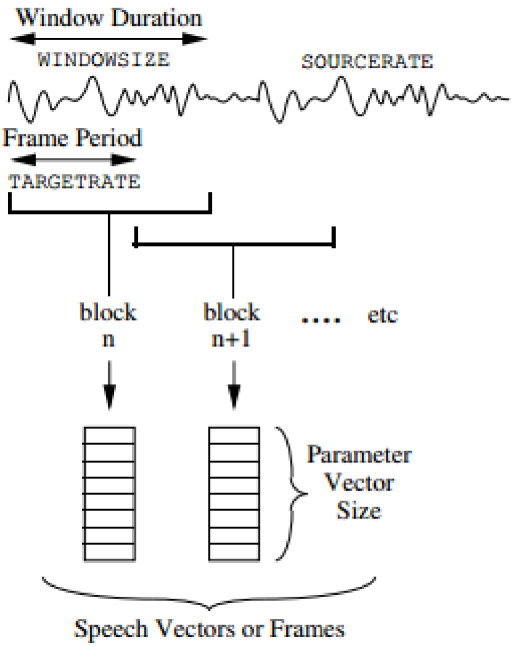


与常规的模式识别系统一样，包含有特征提取、模式匹配和参考模式库三个基本单元。由于语音识别系统所处理的信息是结构非常复杂、内容及其丰富的人类语言信息，因此它的系统结构比通常的模式识别系统要复杂得多。

后处理单元，可能涉及句法分析、语音理解、语意网络以及语言模型等。它往往不是一个孤立的单元，而是与模式匹配计算单元、参考模式库融合在一起，构成一个逻辑关系复杂的系统整体。

对基于 HMM 的语音识别系统来说，“参考模式库”部分就是声学模型、语言学模型，“模式匹配”过程就是基于 HMM、格型结构的搜索、解码过程。

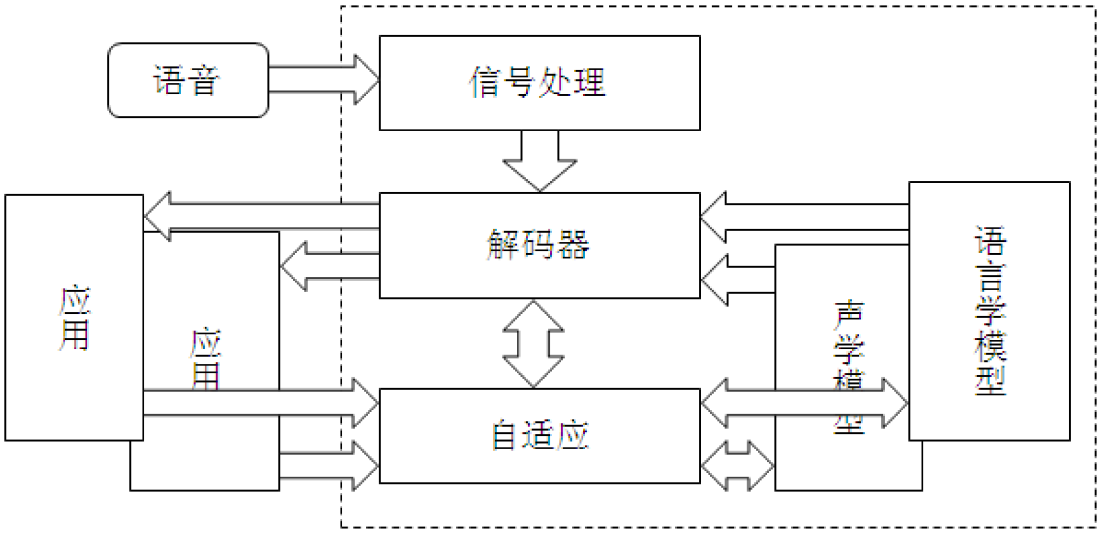
以下是“预处理与数字化”和“特征提取”两个步骤的示意图。



上图显示了对语音信号的数字化、加窗、分帧、提取特征的过程，最后得到标示语音特征的矢量。

3.2 基于 HMM 的基本架构

基于 HMM 的语音识别系统的基本架构，一般如下图（虚线框是系统的边界）。



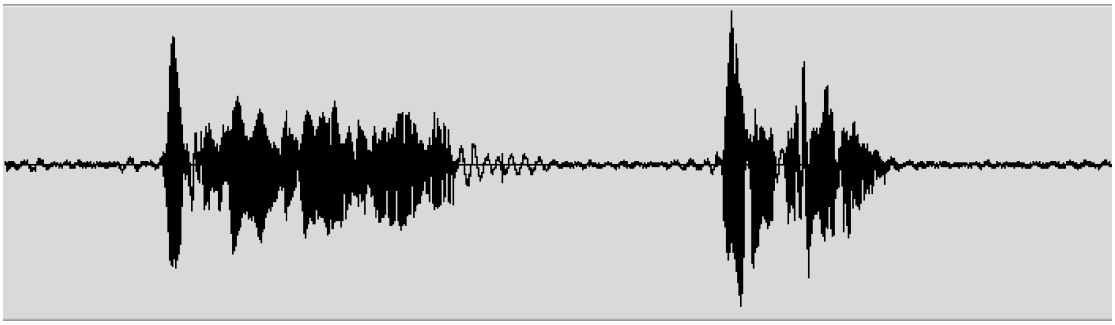
信号处理包含了语音特征参数的提取；解码器需要声学模型和语言学模型的支持；自适应是对模型进行训练。

3.3 信号处理、特征分析

只有将语音信号分析表示成其本质特性的参数，才可能建立用于识别的模版或者知识库。语音识别率的高低，取决于对语音信号的分析的准确性和精度。 因此，应先对语音信号进行特征分析，得到提高语音识别率的有效数据，并据此来设计语音识别系统。

语音信号分析可分为时域、频域、倒谱域等方法。

贯穿语音信号分析全过程的是“短时分析技术”。根据对语音信号的研究，其特性是随时间而变化的，所以它是一个非稳态过程。但从另一方面看，在一个短时间范围内，其特性基本保持不变，即相对稳定，所以可以将其看做一个准稳态过程。基于这样的考虑，对语音信号的分析和处理必须建立在“短时”的基础上，即进行“短时分析”。将语音信号分为一段一段来分析，其中每一段称为一帧。由于语音信号通常在 10~30ms 之内是保持相对平稳的，因而帧长一般取 10~30ms。



3.3.1 数字化

语音信号是时间和幅度都连续变化的一维模拟信号，要想在计算机中对它进行处理，就要先进行采样和量化，将它变成时间和幅度都离散的数字信号。

语音信号数字化之前必须先进行放混叠滤波和防工频干扰滤波。

3.3.1.1 采样

根据采样定理，当采样频率大于信号最高频率的两倍时，在采样过程中就不会丢失信息，并且可以用采样后的信号重构原始信号。

在实际的语音信号处理中，采样频率一般为 8~10kHz。

对于语音识别系统，不同采样率对识别性能的影响，大致如下：

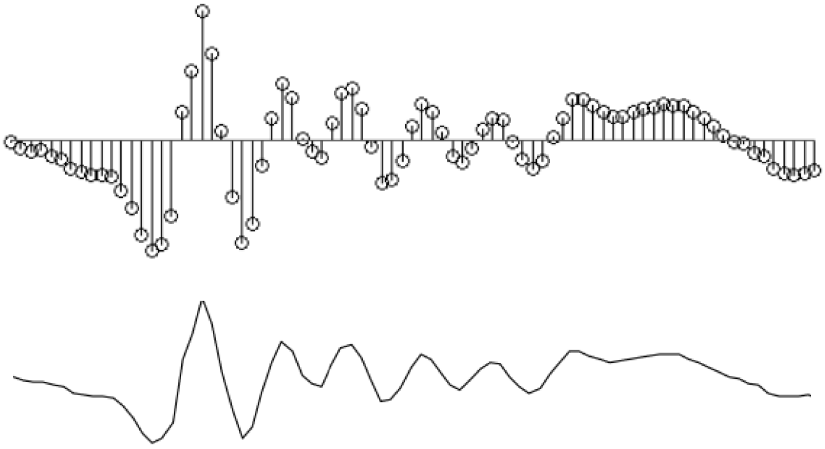
采样率	相对误识率的降低程度
8kHz	基线系统
11kHz	+10%
16kHz	+10%
22kHz	+0%

在上表中，将 8kHz 采样率时的系统作为基线系统，当采样率为 11kHz 时，系统的误识率有 10%的降低；继续升高采用率到 16kHz 时，系统的误识率与 11kHz 相比有 10%的降低；当采样率继续增加时，误识率几乎没有降低。因此在一般的语音识别系统中，采样率最高选择在 16kHz。

语音识别系统一般需要知道语音信号的采样率。如果语音信号来自输入设备（麦克风），语音识别系统要能够读取输入设备的采样率，如果不能读取，则需要手工配置。

3.3.1.2 量化

下图是模拟信号和采样后的离散信号。



可以看出，采样后的信号在时间域上是离散的形式，但在幅度上还保持着连续的特点，所以要进行量化。

量化器就是将整个信号的幅度值分成若干个有限的区间，并且把落入同一个区间的样本点都用同一个幅度值表示，这个幅度值称为量化值。量化方式有 3 种：零记忆量化，分组量化和序列量化。

零记忆量化是每次量化一个模拟采样值，并对所有采样点都使用相同的量化器特性。分组量化是从可能输出的离散集合中，选出一组输出值，代表一组输入的模拟采样值。序列量化是在分组或非分组的基础上，用一些邻近采样点的信息对采样序列进行量化。

3.3.1.3 短时加窗处理

经过数字化的语音信号实际上是一个时变信号。为了得到短时的语音信号，要对语音信号进行加窗操作。窗函数平滑地在语音信号上滑动，将语音信号分成帧。分帧可以连续，也可以采用交叠分段的方法，交叠部分称为帧移，一般为窗长的一半。

窗函数的形状可以有矩形窗，汉明窗，或者汉宁窗。

对语音信号的时域分析来说，窗函数的形状是非常重要的，矩形窗的谱平滑性较好，但波形细节丢失，并且矩形窗会产生遗漏现象；汉明窗可以有效的克服泄漏现象，应用范围也最广泛。

不论什么样的窗函数，窗函数的长度对能否反应语音信号的幅度变化起到决定性的作用。

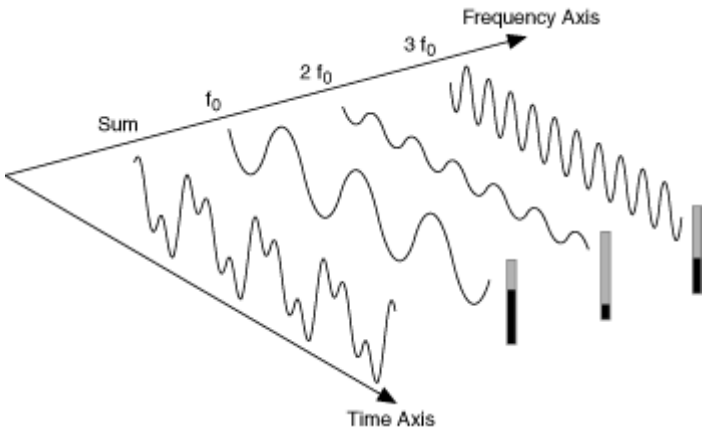
3.3.2 时域分析

进行语音信号分析时，最先接触到的、也是最直观的是它的时域波形。语音信号本身就是时域信号，因而时域分析分析是最早使用且应用范围最广的一种方法。时域分析具有简单直观、清晰易懂、运算量小、物理意义明确等优点。

语音信号典型的时域特征包括短时能量，短时平均过零率，短时自相关系数和短时平均幅度差等。

时域图以时间为横轴，频域图以频率为横轴，两者都是以振幅为纵轴。时域图是不同频率信号的叠加，而频域图则是展现信号中不同频率的振幅。

下图揭示了时域和频域的区别和关系。从时间轴（Time Axis）看是时域图，从频率轴（Frequency Axis）上看在是频域图。



3.3.3 频域分析

尽管时域分析具有多个优点，但更为有效的分析多是围绕频域进行的，因为语音中最重要的感知特性反映在其功率谱中，而相位变化只起到很小的作用。

时域波形较易随外界环境变化，但语音信号的频谱对外界环境变化具有一定的顽健性。另外，语音信号的频谱有非常明显的声学特性，利用频域分析获取的语音特征具有实际的物理意义，如共振峰参数，基音周期参数等。

常用的频域分析方法有带通滤波器组方法、傅里叶变换法和线性预测分析法。

信号处理中的“不确定原理”指出，对给定的信号，其时宽与带宽的乘积为一常数。当信号的时宽减小时，其带宽将相应增大，当时宽减到无穷小时，带宽将变成无穷大；反之亦然。即信号的时宽与带宽不可能同时趋于无穷小。也就是说，信号分析时的时间分辨率和频率分辨率是一个矛盾，不能同时取得很小。

3.4 特征提取

声学特征的提取与选择是语音识别的一个重要环节。声学特征的提取既是一个信息大幅度压缩的过程，也是一个信号解卷过程，目的是使模式划分器能更好地划分。

正如前面所提到的，由于语音信号的时变特性，特征提取必须在一小段语音信号上进行，也即进行短时分析。这一段被认为是平稳的分析区间称之为帧，帧与帧之间的偏移通常取帧长的 1/2 或 1/3。通常要对信号进行预加重以提升高频，对信号加窗以避免短时语音段边缘的影响。

常用的一些声学特征有线性预测系数（LPC）、梅尔频率倒谱系数（MFCC）、感知线性预测（PLP）等。HTK 和 CMU Sphinx 均支持这几种声学特征。

3.4.1 线性预测系数（LPC）

线性预测分析从人的发声机理入手，通过对声道的短管级联模型的研究，认为系统的传递函数符合全极点数字滤波器的形式，从而 n 时刻的信号可以用前若干时刻的信号的线性组合来估计。通过使实际语音的采样值和线性预测采样值之间达到均方差最小 LMS，即可得到线性预测系数（LPC）。

对 LPC 的计算方法有自相关法（德宾 Durbin 法）、协方差法、格型法等等。计算上的快速有效保证了这一声学特征的广泛使用。

与 LPC 这种预测参数模型类似的声学特征还有线谱对 LSP、反射系数等等。

3.4.2 倒谱系数

倒谱域是将对数功率谱进行反傅里叶变换后得到的，它可以将声道特性和激励特性有效的分开，因此可以更好的揭示语音信号的本质特征。

利用同态处理方法，对语音信号求离散傅立叶变换 DFT 后取对数，再求反变换就可得到倒谱系数。对 LPC 倒谱，在获得滤波器的线性预测系数后，可以用一个递推公式计算得出。

实验表明，使用倒谱可以提高特征参数的稳定性。

3.4.3 梅尔频率倒谱系数（MFCC）

不同于 LPC 等通过对人的发声机理的研究而得到的声学特征，Mel 倒谱系数（MFCC）是受人的听觉系统研究成果推动而导出的声学特征。

对人的听觉机理的研究发现，当两个频率相近的音调同时发出时，人只能听到一个音调。临界带宽指的就是这样一种令人的主观感觉发生突变的带宽边界，当两个音调的频率差小于临界带宽时，人就会把两个音调听成一个，这称之为屏蔽效应。Mel 刻度是对这一临界带宽的度量方法之一。

MFCC 的计算首先用 FFT 将时域信号转化成频域，之后对其对数能量谱用依照 Mel 刻度分布的三角滤波器组进行卷积，最后对各个滤波器的输出构成的向量进行离散余弦变换 DCT，取前 N 个系数。

3.4.4 感知线性预测（PLP）

感知线性预测（PLP）同样是受人的听觉系统研究成果推动而导出的声学特征。它将人耳听觉的各种特性通过工程化处理，采用简化的模型来加以模拟。因而经过这样处理后获得的频谱更符合人耳的听觉特点，有利于进行语音信号处理。

PLP 仍用德宾法去计算 LPC 参数，但在计算自相关参数时用的也是对听觉激励的对数能量谱进行 DCT 的方法。

一些研究表明，对噪声环境下的语音识别，采用 PLP 特征比 MFCC 特征的性能更好一些。

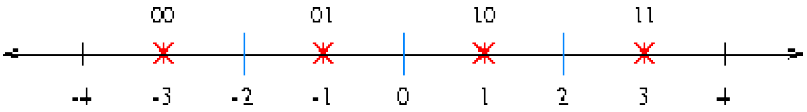
3.5 矢量量化

语音信号在提取出特征后，得到一个个标示某一短时间内（一帧）的语音特征的矢量。矢量的维数跟特征的类型有关，比如 MFCC，有三十几维。这些矢量作为 HMM 的观察序列，用于训练模型、语音解码。

这些特征矢量的取值并不是离散的，连续 HMM、半连续 HMM 可以处理它。为了能适用于离散 HMM，就要对这些矢量进行量化，即得到离散的特征值，特征值空间是有限的。这就是矢量量化（Vector Quantisation）技术。

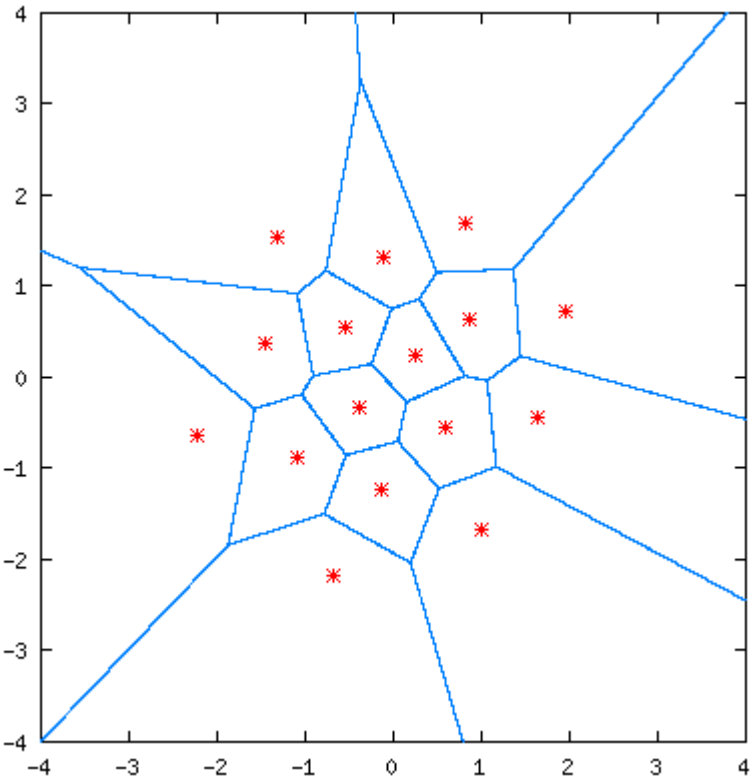
量化可以分为两类：一类是标量量化，一类是矢量量化。

标量量化是将采样后的信号值逐个进行量化，这时将一维的零到无穷大值之间设置若干个量化阶梯，当某个输入信号的幅度值落在某相邻的两个量化阶梯之间时，就被量化为与其最近的一个阶梯的值。如下图，把无限的值空间映射到 00、01、10、11 四个离散值。



矢量量化是将若干个采样信号分成一组，即构成一个矢量，然后对此矢量一次进行量化。它是将 d 维无限空间划分为 K 个区域边界，每个区域称为一个胞腔，然后将输入信号的矢量与这些胞腔的边界进行比较，并被量化为“距离”最小的胞腔的中心矢量值。

以下是 2 维矢量的矢量量化示意图，把平面上无限的空间划分为有限个胞腔。



这里胞腔的中心称为码字，而码字的组合称之为码书（codebook）。

在矢量量化中主要有两个问题：

如何划分 K 个区域的边界。这需要大量的 输入信号矢量，经过统计实验才能确定。这个过程称为“训练”或建立码本，一般采用 K-means 算法或 LBG 算法

如何确定两个矢量在进行比较时的距离测度，可以采用欧拉距离（均方差距离）或 Itakura-Satio 距离，以及似然比失真等。

K-means 矢量量化算法的设计原则是使整体的均方差达到最小。它是直接将 d 维空间分成 K 个胞腔， 然后不断迭代，更新码本，直到相邻迭代的误差不高于阀值。

K-means 算法是在码书大小已知的情况下对样本聚类的方法，但在很多应用中，事先聚类中心的个数未知，这是可以采用 LBG 算法。LBG 算法的核心思想是先生成一个聚类中心的码本，然后逐层分裂，直到聚类误差达到要求（可参考：http://www.data-compression.com/vqanim.shtml）。

3.6 后续步骤

在提取语音信号的特征以及矢量量化之后，就到了解码过程。对于基于 HMM 的语音识别系统，HMM 就是解码的核心。解码过程要有声学模型的支持；对于大词表的语音识别来说，语言学模型也很重要。对于大词表连续语音识别，还涉及到网络的构建（格型结构，lattice）和图搜索算法。

这些内容将在后续章节阐述。

4 声学、语言学模型

在语音识别系统中，声学模型包含了基元的发音模板和词汇的发音字典，发音模板是一套训练好的 HMM 模型及其参数，发音字典包含了语言中每个词的发音，一个词用一行文本或一个网络结构表示。声学模型是语音识别系统的基础和核心。

在连续语音识别系统中，除了声学模型，一般还要建立语言学模型，以利用语言中的句法规则、词频等知识，提高识别率。词汇量越大，语言学模型就越重要。

4.1 声学模型

4.1.1 基本声学单元

基本声学单元（简称基元）的选择是声学模型建模中一个基本而重要的问题。基元的选择有音素、半音节、音节、词等几种。

声学模型的建模单元的选择需要考虑三方面的因素。

其一是该单元的可训练性，亦即是否能够得到足够的语料对每个单元进行训练，以及训练所需要的时间长短是否可接受

其二是该单元的可推广性，当语音识别系统所针对的词汇集发生变化时，原有建模单元是否能够不加修改的满足新的词汇集

最后还需要考虑建模的精确性

以词作为基本单元建立模型，对于简化识别系统的结构和训练过程是很有效的。但对大词汇量连续语音识别系统来说，采用此作为基本单元建模就不合理了。在连续语音识别中，以词作为基本单元，各种音联关系可能得不到充分的训练；并且以词为单位构成的系统，需要的存储量很大，计算复杂度很高；由于词内的各因素重复出现，造成大量不必要的冗余存储和计算。因此在大词汇连续语音识别中，一般采用比词小的子词识别基元，如音节、半音节等。一般来说，声学单元越小，其数量也就越小；但是另一方面，单元越小，对上下文的敏感性就越大，越容易受到前后相邻的影响而产生变异，因此，其类型设计和训练样本的采集更困难。

一般在声学建模中，考虑上下文相关信息，这样识别基元就会变成上下文相关的基元。当考虑上下文信息时，基元的数目会变得非常庞大，这将导致声学模型的规模变得无法接受。因此在进行上下文相关建模时，不适宜采用音节模型。

基于音素的基元在英语连续语音识别系统中得到了广泛的 应用。声韵母基元是适合汉语特点的一种识别基元。声韵母跟半音节在形式上非常接近。

4.1.2 基元的扩展

单纯的一个音素，称为上下文无关音素，简称单音素。所谓上下文相关音素，就是考虑一个音素与其左或右相邻音素的相关情况后选取的基元。这样对 N 个基元，就可能存在 N^2 个坐或右上下文相关基元，称为双音素，可能存在 N^3 个左或右上下文相关的音素，称为三音素。

三音素又分为两种，逻辑三音素和物理三音素。前者指语言上可能的音素组合，即在语言中可能出现的音素组合；后者指训练语音数据中出现的 音素组合。

在训练语音模型时，一般应该保证每个三音素在训练数据中出现的次数不少于 10 次。如果出现次数太少，则不能保证模型的准确性，这称为训练数据稀疏。最直接解决这种问题的方法是，根据一些准则对上下文相关的音素进行聚类，并根据聚类进行状态共享，以此来解决数据稀疏的问题。常见的状态共享策略有基于数据驱动的和基于决策树两种。

对于基于数据驱动的状态共享策略，HTK 提供了一种基于最小类合并的聚类方法，它在初始时将所有状态都作为一个类，每次合并两个最小的类，直到最大类的大小达到一个阀值或者类的数目达到聚类的要求。

基于决策树的聚类方法，可以获得与数据驱动聚类方法类似的聚类效果，同时还能处理训练数据中没有出现的三音素。

4.2 字典

声学－语音学层之上有一个词层，词层中应有一部字典来规定词表中每一个词是用哪些子词单元以何种方式构筑而成的。

最简单实用的方案是每个词用若干子词单元串接而成。但是，每个词的发音可能有多种变化方式，因而串接也有相应的困难。

发音的变化有两个方面：

替换，即词中的某个音子可能被用其他相似而略有差异的子词单元所替换，这种替换具有一定的随机性

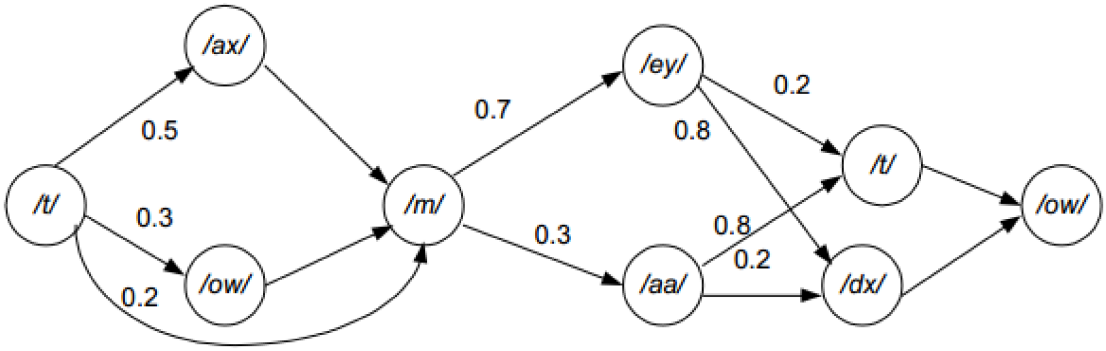
插入和删除错误，即词中有时候增加一个不是本词成分的子词单元，有时又将本词成分中的某个子词删除了，何时插入以及何时删除也是随机的

针对这些问题有以下的几种方案：

第一种方案是在词典中为每一个词建立多套子词单元串接规则来代替单一的规则，这样可以表现同一个词的不同发音变异。这种方案使词典容量扩充很大，但对识别效果收效甚微，因而不是一种优选方案

第二种方案将子词单元构成词的规则用一个网络图来描述，其中包含替代和插入、删除等各种变化

以下就是单词 tomato 的一个可能的发音网络，还加上了转移概率。



4.3 语言学模型

语言学模型（或叫语言模型）分为基于文法的语言模型和基于统计的语言模型。基于文法的语言模型是总结出语法规则乃至语义规则，然后用这些规则排除声学识别中不合语法或语义规则的结果。基于文法的语言模型在特定任务系统中获取很好的应用，可以较大幅度地提高系统的识别率。在大词汇量的语音识别系统中，统计语言模型由于可以克服文法规则难以处理真实文本的局限性，因而获得了越来越广泛的应用。

统计模型的基本原理是，采用大量的文本资料，统计各个词的出现概率及其相互关联的条件概率，并将这些知识与声学模型匹配相结合进行结果判决，以减小声学模型不够合理而产生的误识。

然而，要可靠地估计一种语言所有词在所有序列长度下的条件概率几乎是不可能的事，因此也就出现了几种常用的简化模型。

以下 N 元文法、基于类的 N 元文法是很常用的基于统计的模型，HTK 和 CMU Sphinx 都支持。但采用的词的聚类算法不太一样。

小词汇量的应用一般不用 N 元文法模型。如果词汇量很小（例如 20 个词以内，在一些命令和控制系统中），可以直接定义语言文法（通过产生式），或者直接从句子建立语言文法。

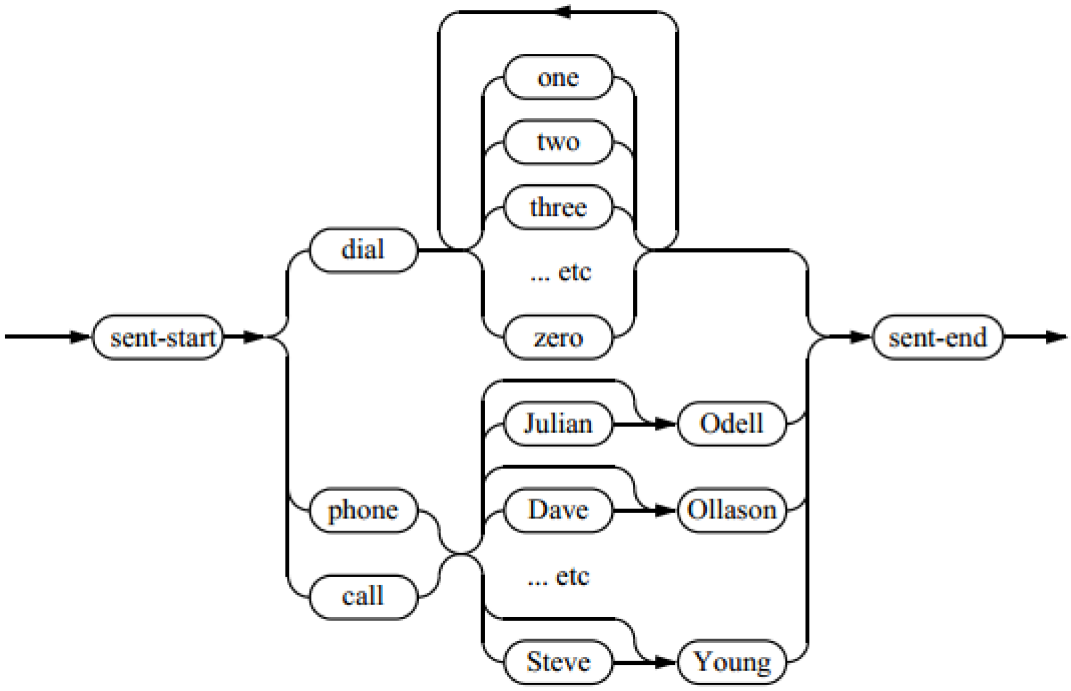
4.3.1 基于文法的模型

通过语法规则来描述一个语言，语法规则是一系列的产生式。根据产生式可以判定一个句子在这个语言中是否合法。

以下是语音拨号应用中的语法文件（来自 HTK 的例子）：

```
$digit = ONE | TWO | THREE | FOUR | FIVE |
         SIX | SEVEN | EIGHT | NINE | OH | ZERO;
$name  = [ JOOP ] JANSEN |
         [ JULIAN ] ODELL |
         [ DAVE ] OLLASON |
         [ PHIL ] WOODLAND |
         [ STEVE ] YOUNG;
( SENT-START ( DIAL <$digit> | (PHONE|CALL) $name) SENT-END )
```

根据语法，可以构造一个词网络：



4.3.2 基于统计的模型

4.3.2.1 N 元文法

条件概率假定只考虑与前 N-1 个词相关，即为 N 元文法模型。

实际上，一般的 N 元文法是难以估计的，通常系统中采用的也只有二元文法和三元文法。

N 元文法统计语言模型的词串的条件概率，一般是通过相对频率计数得到。

然而，即使在 N 较小的情况下，要统计的条件概率也是一个非常庞大的数字，因而常常会出现条件概率为 0 或接近于 0 的情况，这样得到的结果将不可靠。解决这种训练数据稀疏的方法是采用一些平滑技术。

4.3.2.2 基于类的 N 元文法

对一些具有同样语义的字词，可以归并到一类，这是在语言学模型中处理数据稀疏的一个有效方法。基于类的语言学模型，在同等的性能上需要更少的训练以及内存空间。

对于词如何聚类，可以有多种方法。总体而言，可以划分为基于规则聚类和数据驱动聚类两种方法。

基于规则聚类多从句法－语义的角度考虑聚类。如果有限定领域的知识，则在聚类中可以有效利用这部分信息将具有相同语义信息的词聚为一类。

对于一般的识别系统，很难像上述基于规则的方法将一些具有同样功能的词划分到同一个类中。这是可以采用基于数据驱动聚类方法。在这种方法中，一个重要的概念是词的相似度，基于该相似度来定义目标函数。然后通过优化该目标函数将不同的词聚类到不

同的类别中。这里可以采用最大似然估计准则保证最后得到的聚类结果的困惑度（Perplexity）最小。

4.3.2.3 平滑技术

平滑和自适应是应用于基于统计的语言模型的技术。

平滑技术的基本思想是将模型中可见事件的概率值进行折扣，并将改折扣值重新分布给不可见事件的元素序列，所以它可以保证模型中任何概率均不为零，且可以使模型参数概率分布趋向更加均匀。因此，平滑方法由概率值折扣的策略和折扣值的分布方法所决定。

通常使用的平滑技术有加法平滑、图灵估计、Katz 平滑、线性插值平滑和 Kneser-Ney（K-N）平滑。具体请参考相关资料。

从文献的评价结果看，目前性能最好的平滑算法当属 Kata 和 K-N 方法。

4.3.2.4 自适应

在一些自由对话应用中，交谈的主题会随时发生变化，这时需要对语言学模型的一些参数，如 N 元文法的概率、词表的大小、词表内的词进行适当的调整。这就是语言学模型的自适应。自适应方法有基于缓存的语言学模型和主题自适应模型等。

基于缓存的语言学模型假设：在文本中刚刚出现过的一些词在后边的句子中再次出现的可能性往往较大，一般会大于 N 元文法中预测的概率。于是可以采用动态缓存语言学模型根据当前的话题来调整词频。

主题自适应为了减少主题差异对语言学模型的影响，将语言学模型划分成 n 个子模型，整个语言学模型的概率通过插值公式来计算。

4.3.3 性能

语言模型的性能通常用交叉熵和复杂度（Perplexity，或困惑度）来衡量。

交叉熵的意义是用该模型对文本识别的难度，或者从压缩的角度来看，每个词平均要用几个位来编码。

复杂度的意义是用该模型表示这一文本平均的分支数，其倒数可视为每个词的平均概率。

语言模型的性能衡量是在对词采用聚类、平滑技术等时候的依据。

5 识别过程

对于连续语音识别，识别过程可称为解码（decode）。解码器是语音识别系统的核心之一，其任务是对输入的信号，根据声学、语言学模型，寻找能够以最大概率输出该信号的词串。

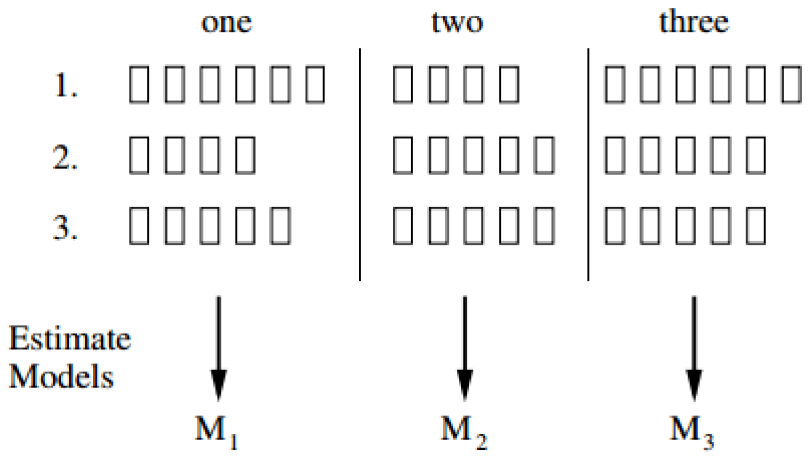
在 HTK 中，一般解码模块是 HVite，用于大词汇量的解码模块是 HDecode。在 CMU Sphinx 中，解码模块就叫 Decoder。

5.1 孤立词语音识别

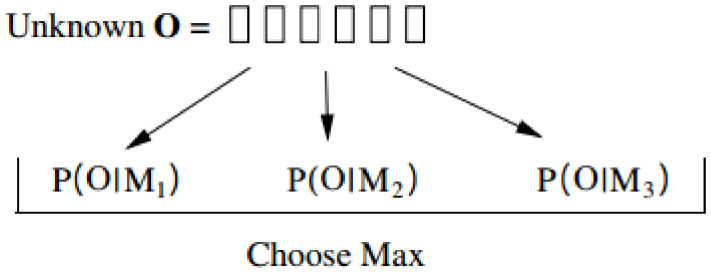
孤立词识别是语音识别中最基本的问题，对该问题的研究开展得最早，也是目前最成熟的技术。孤立词识别可以采用矢量量化方法、DTW 方法及 HMM 方法等。

基于 HMM 的孤立词识别系统的基本思想为：在训练阶段用 HMM 训练算法为系统词汇表中每个词 W_i 建立对应的 HMM，记为 λ_i；在识别阶段，用前向-后向算法或 Viterbi 算法求出各个概率值 $P(O|\lambda_i)$，其中，O 为待识别的观察值序列；在后处理阶段，选取最大的 $P(O|\lambda_i)$值所对应的词 W_i 的为 O 的识别结果。

下图显示了训练过程，分别训练三个词（one、two、three），得到三个模型。



而下图则显示 了识别过程，对观察序列 O，用各个模型计算概率，选取最大概率所对应的模型，从而得到模型所对应的词。



需要注意的是，对于不同类型的 HMM，送入 HMM 处理的观察值序列 O 有所不同。例如，对于离散 HMM，一般求出语音特征参数后，还必须做矢量量化，这样观察序列就

是由 VQ 码字序号组成的序列。对于连续型 HMM，语音信号经过预处理、特征提取之后的特征参数序列就是相应的观察值序列。

对孤立词识别，它要求将词表中的每个词或短语单独发音，之后将该发音作为一个整体使用识别算法来判断结果。建模和识别过程中，词表中的每个词都作为一个整体处理。这种系统结构简单，主要用于命令和控制系统。

5.2 连接词语音识别

对于词表比较大，又希望能灵活的组成各种各样的短语和句子的场合，孤立词识别的系统结构就显得力不从心。一方面它不便于结合句法规则提高识别率，另一方面，对于一个数字序列或词序列，以孤立词方式发音是非常不自然的，且发音不流利，表达的效率低。因此，将孤立词做技术扩展，进行流利语音的识别具有重要意义。

从语音识别算法的角度看，有两类流利语音，第一类为有中小词表组成的字串，包括数字串、拼写的字母串等。这类问题中基本的语音识别单元，可以像孤立词识别一样使用词或短语。第二类为由中到大词表组成的连续语音识别，对于这样的问题，由于复杂性的限制，基本的语音识别单元不能使用词，需要使用比词小的子词作为基本的识别单元。

所谓连接词识别，就是指系统存储的 HMM 是针对孤立词的，但是识别的语音却是由这些词构成的词串。它是根据给定的发音序列，找到与其最优匹配的参考模板词的一个连接序列。为此，必须解决如下的问题：首先，尽管有些时候知道序列中词长度的大致范围，但序列中词的具体数量 L 未知；其次，出了整个序列首末端点外，并不知道序列中每个词的边界位置。由于连音的影响，很难指定具体的词边界，因此，词的边界常常是模糊的或不是唯一的；V 个词在词串长度为 L 的情况下，将有 V^L 种可能的匹配串组合，除非在 V 和 L 均很小的情况下，否则对这种指数量级的匹配用穷举的方法很难进行。

对连接词的识别，有两种有效的方法，分别是二阶动态规划算法和分层构筑算法。

二阶动态规划算法的基本思想是将计算分成两个阶段完成，也称为两个层来完成。第一层进行词内匹配，利用 DTW 算法，找出测试发音中可能构成词的一段，并与词表中的所有词具有最佳匹配的一个发音，将其距离值作为最好打分，并记住对应的词标号。第二层用动态规划算法进行词间的匹配。

分层构筑算法最早用于解码中，后来分别有人将其与 DTW、HMM 结合，用于连接词语音识别，取得了非常好的效果。

5.3 大词表连续语音识别

语音识别研究中意义最重大、应用成果最丰富，同时最具有挑战性的研究课题是大词汇量、非特定人连续语音识别。一般连续语音识别系统的词误识率大致等于特定人识别系统的 3~5 倍，而非特定人识别系统的词误识率大致是特定人识别系统的 3~5 倍。此外，当词汇量大于 1000 词时，易混淆的相似词数量将大大增加。

此外，在连续语音识别系统中，下面两个重要问题是孤立词识别没有的。

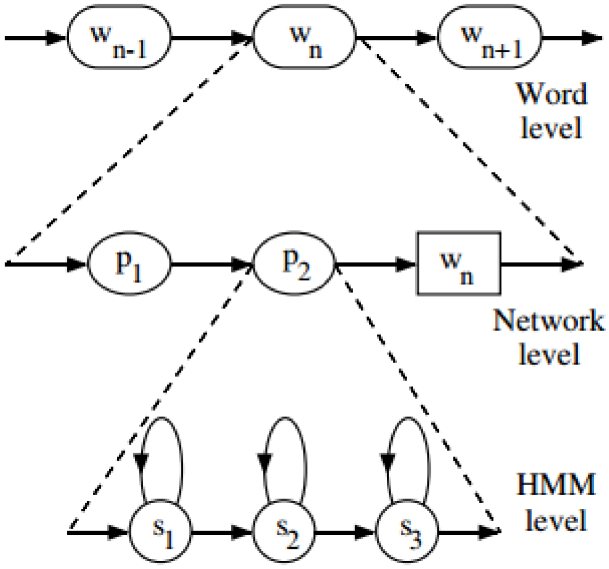
切分：对整个短语进行识别显然是不可能的，因为语言中短语的数量太大，必须把输入的语流切分为更小的组成部分，人类感知语音也是这样做的。因为连续语音中间没有间歇，所以在识别前必须先把各字分开，这要求系统必须能够识别单词之间的边界。但这

是非常困难的，因为确定单词间的边界位置还没有现成的方法。尽管有时可以采用能量最低点作为边界，但通常还要根据发音信息再加以验证。

发音变化：连续语音的发音比孤立词发音更随便，受协同发音的影响也更为严重。灵位，连续语音识别系统中的很多问题都与语言学知识有关，特别是大词汇量识别系统要更多地强调语言学只是的运用。

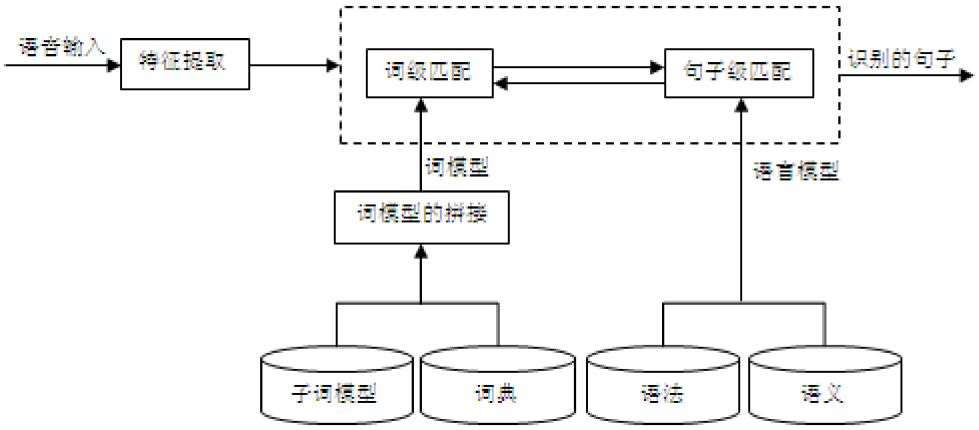
现在统一的做法是将整个识别系统分为 3 层：声学－语音学层、词层和句法层。声学－语音学层是识别系统的底层，它接收输入语音，并以一种“子词”单位作为其识别输出。词层规定词汇表中每个词是有什么音素－音子串接而成的。最后的句法层中规定词按照什么规则组合成句子。

下图体现了这个层次。



声学－语音学层，每一个音子用一个 HMM 模型及一套参数来表示。每一个 HMM 模型中最基本的构成单位是状态与状态之间的转移弧。这样，从状态出发逐层扩大到音子、词、句子。每一个句子是包含许多状态的复杂的状态图，该句子就是用所有状态形成的结构、状态之间的转移概率，以及每个转移弧产生某个特征输出的概率来描述的。对于特定的词表和句法，所有可能出现的句子构成了一个更大的状态图。在完成识别任务的时候，要根据一个输入语音特征矢量序列来确定一个最可能的句子。这就需要在这个大的状态图中搜索一条路径，该路径上产生上述特征矢量的概率最大，由路径可以进一步确定句子中的每一个词。

大词汇连续语音识别总体框架可由下图来描述。



语音信号先经过分析后形成特征矢量，并按字典要求和词模型（由子词模型集合串接成）进行识别，然后根据语言模型的句法限制在句子级进行输入语音与参考模板间的匹配，最后识别出相应的句子。

5.4 解码技术

对大词汇量连续语音识别，最终目的是从各种可能的子词序列形成的一个网络中，找出一个或多个最优的子词序列。这在本质上属于搜索算法或解码算法的范畴。

5.4.1 搜索策略

根据语音识别系统对不同先验知识源的利用方式，可以把搜索策略分为一遍搜索和多遍搜索两种。

一遍搜索策略倾向于将所有可能的知识源全部集成在一起，只对输入的语音序列进行一次处理，完成所有的搜索步骤，并直接给出最优的搜索结果。由于使用的知识源越多，搜索的计算代价就越大，因此，一遍搜索比较适合构建实时系统，但很难应用到比较复杂的声学和语言学模型中。

多遍搜索的基本思想是，将各种知识源，包括声学模型和语言学模型，由简单到复杂逐渐加入到搜索过程中，每一边搜索只使用一部分知识，并为随后的搜索构建缩小的搜索子空间，使后面的搜索过程在前一级产生的子空间上进行。多遍搜索策略的优势在于，可以方便的导入各种复杂的声学模型和语言学模型，以及采用多种识别算法和后期处理算法。但多遍搜索需要精确的控制前一级为下一级提供的搜索子空间，防止正确的结果在前一阶段被错误删除。此外，多遍搜索必须要等待语音输入完成后才能开始，不适应实时系统的要求。

根据搜索过程中路径的扩展方式，可以分为基于词和基于时间的两种方法。在基于词的搜索算法中，搜索路径的扩展取决于前面已经识别的词，而且词边界已经在前面的搜索过程中确定了。在基于时间 的搜索算法中，搜索路径的扩展取决于前一个词结束的时间，即词边界在搜索回溯过程中才能确定。

根据搜索时考虑的语音信号范围，可以将搜索算法分为时间同步和时间异步两种策略。时间同步策略按照从左到右的时间拓扑结构进行搜索路径概率得分的更新；时间异步的

方法则优先处理得分较高的搜索路径，完全打乱了时间的次序。一般 Viterbi 算法需要采用时间同步的方式实现，A*算法则需要采用时间异步的方式实现。

5.4.2 搜索算法

在解码时，会先构建一个网络，解码的过程主要就是在网络中搜索一条最优路径的过程。因此，解码算法基本就是网络的搜索算法，或叫图搜索算法。一些一般的图搜索算法也可以用在解码中，如 A*算法、Beam 算法。

解码中的图搜索算法一般是启发式的算法，这样能够利用已有的知识，缩减搜索路径的数量，减少计算量。

对一般的图搜索过程，可以使用一个著名的图搜索框架（由 N.J.Nilesson 提出），这是一个表达能力很强的搜索框架，可以囊括常见的深度优先搜索和广度优先搜索。它引入了 OPEN 表和 CLOSE 表，OPEN 表有进有出，存放刚生成的节点，CLOSE 表只进不出，存放将要扩展和已经扩展的节点。

关于 A*算法、启发式搜索、图搜索框架，可参考维基条目 A*_search_algorithm 及相关链接。

5.4.2.1 启发式搜索

启发式（Heuristic）搜索就是在状态空间中搜索时，对每一个搜索的位置进行评估，通过设计估价函数来控制搜索方向。这样可以省略大量无谓的搜索路径，提高了效率。

一般而言，估价函数综合考虑了两个方面的因素—已付出的代价及将要付出的代价。

常见的启发式搜索算法有两种，一个是 A*搜索，也成为 Best-First 搜索；另一个是 Beam Search。这两种方法广泛应用于语音识别系统中。

依据启发函数，A*优先搜索最小代价的结点。A*采用保守的未来代价的估价函数（admissible heuristic），不会丢失最优解。

Beam 是一个广度优先并结合深度考虑的算法，与传统的广度优先算法相比，它在每个层次上只是对那些有可能有后继的结点进行扩展。Beam 可以在寻找最优结点和计算量之间权衡，通过确定一个启发函数在同一层次结点间的相差阀值。

5.4.2.2 Viterbi Beam

标准的 Viterbi 算法是在状态空间中的全局搜索，不存在搜索的误差，因此搜索结果是全局最优的。识别的结果也只与模型的精确度有关，在孤立词识别系统中，得到了普遍的应用。但是在连续语音识别系统中，由于搜索的空间比较大，全局搜索将导致性能的下降。而且在搜索过程中，对一些可能性很小的路径进行搜索的效率是很低的。为了解决这些问题，需要采取剪枝策略，放弃可能性小的路径，只在可能性大的路径上进行搜索，这就是 Viterbi Beam 的基本思想。

Viterbi Beam 搜索算法是一个广度优先的帧同步算法。它在不丢失全局最优解的条件下，在搜索中同时解决连续语音识别中 HMM 模型状态序列和声学观测序列的非线性时间对

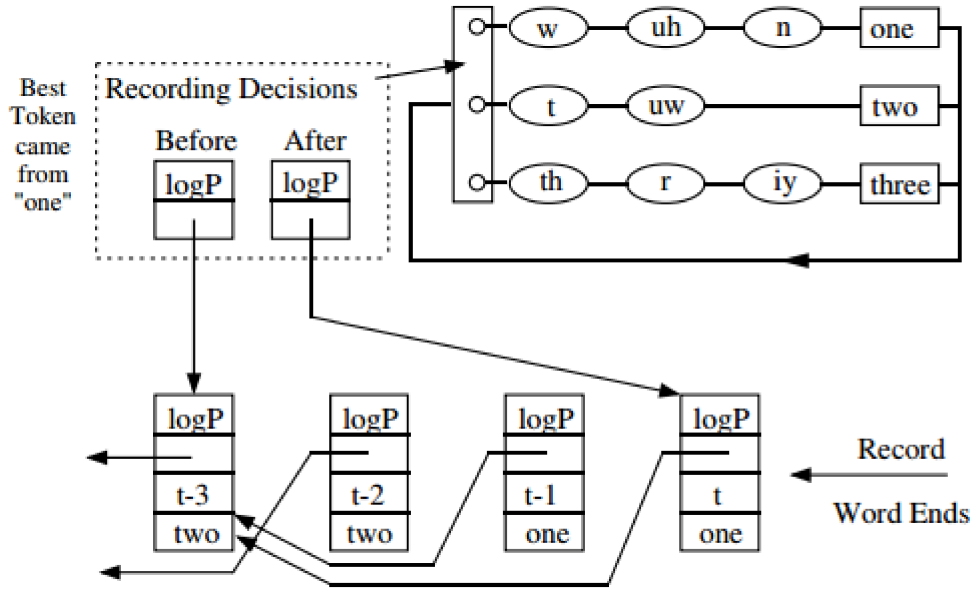
准、词边界监测和词识别等问题。由于 Viterbi 算法中的路径扩展具有时间同步性，同一时刻的各条路径扩展对应于完全相同的观测序列，因此其似然度得分具有可比性。Viterbi Beam 搜索算法在每一时刻有效的剪裁低得分路径，大大地提高了搜索效率。当设置一个比较保守的门限值时，基本上不会影响识别的准确率，因此，Viterbi Beam 搜索算法在连续语音识别中得到了广泛的应用。

5.4.2.3 令牌传递模型

Viterbi Beam 算法在搜索过程中不断寻找可能的最优状态序列，记录回溯信息，直到最后时刻找出得分最高的哪条路径作为识别的结果。在这个过程中，要保留大量的回溯信息，因此如何设计一个好的结构使其存储空间尽可能小是面临的一大难题。S.J. Yong 等人提出一个简单的概念模型来表示搜索过程的信息存储与提取，即令牌传递模型（token passing）。

令牌传递模型是一种时间同步的搜索算法，其核心思想是用令牌的形式标记搜索路径，每输入一个语音帧，对处于激活状态的所有令牌进行处理，通过令牌的产生、复制及传递等操作实现搜索路径的扩展，直到搜索完成。由于声学模型采用的是 HMM 模型，因此令牌的操作主要在 HMM 模型的状态中进行。考虑到输出的识别结果只是此条，所以不需要在 HMM 状态级别上进行回溯，只需要在词级别上进行回溯。令牌传递模型可以看做是 Viterbi Beam 搜索算法的一种实现。

下图阐述了令牌传递模型。一个令牌包含了指向前一个状态的指针，因此，一个令牌就可以表示一条路径。LogP 是概率的对数，因为只需要比较概率的相对大小。



5.4.2.4 N-BEST 算法

Beam 搜索算法尽管减少了计算量，但是他只是一个次优算法，而且只能得到一条最优的路径。N-best 算法是一种两步搜索算法，它能保证全局最优，而且能依次得到全局得分最高的 N 条候选路径。

该算法的第一部分是从初始帧到最末帧的帧同步前向各点搜索，另一部分是从最末帧到初始帧的帧异步后向树搜索。