

第四讲：自动语音识别

基本概念

王帅

2025 年 9 月 15 日



南京大学
NANJING UNIVERSITY



智能科学与技术学院
School of Intelligence Science and Technology

1 引言与背景

2 语音识别任务的形式化定义

什么是语音识别

语音识别 (Automatic Speech Recognition, ASR) 是将人类语音信号转换为可编辑文本的技术。简单来说，就是赋予机器“耳朵”来听，“大脑”来理解的能力。



语音 → 文本

应用场景



智能家居



车载导航





语音输入法





在线会议

核心挑战

 口音与方言
不同地区的口音和方言差异巨大。

 背景噪音
嘈杂环境下语音信号易被干扰。

 语速变化
人们说话快慢不一，常有停顿、重复。

 口语化表达
日常交流充满省略、倒装等非规范语言。

“如何让模型在这些复杂多变的情况下依然保持高准确率，是语音识别领域持续研究的重点。”

核心原理：语音识别的“三驾马车”



语音识别系统并非单一的黑箱，而是由三个精密协作的组件构成。我们可以将它们比喻为语音识别的“三驾马车”：声学模型、语言模型和解码器。它们各司其职，又紧密配合，共同将人类的语音转化为可理解的文本。

声学模型 (AM)

功能：“耳朵”

- 将声学信号转化为音素的概率分布。
- 分析语音的声学特征，如音高、音强。
- 从 MFCC 等特征计算音素或状态概率。

技术演进：GMM → RNN/LSTM → Transformer

语言模型 (LM)

功能：“大脑”

- 评估一个词序列出现的概率是否合理。
- 解决同音异义词或发音相似词的歧义。
- 通过上下文判断哪个词序列更通顺。

技术演进：N-gram → RNN → Transformer

解码器 (Decoder)

功能：“决策者”

- 结合声学模型和语言模型的输出。
- 从所有可能路径中搜索最优词序列。
- 常用算法：维特比和 Beam Search。

目标：在计算效率和准确率间取得平衡。

什么是声学模型

声学模型 (Acoustic Model, AM) 是语音识别系统中的“耳朵”，将输入的声学信号转化为更高级的、与语言相关的表示，通常是音素 (phoneme) 的概率分布。



从 GMM 到深度学习的演进



GMM 模型

早期模型，需人工设计特征，表达能力有限。



DNN 模型

自动学习复杂、抽象的声学特征，对人工设计特征依赖有限。



RNN/LSTM

捕捉语音时序依赖关系，适合处理长序列。



Transformer

并行处理特征，捕捉全局依赖关系。

“深度学习模型能够自动学习更复杂、更抽象的声学特征，从而显著提高了识别的准确性。”

①

概率计算

计算声学特征属于某个特定音素或 HMM 状态的概率。

②

示例

以“你好”为例，模型会分析“你”和“好”的声学特征，判断它们最可能对应的音素。



什么是语言模型

语言模型 (Language Model, LM) 是语音识别系统中的“大脑”，主要任务是评估词序列出现的概率，解决同音异义词或发音相似词的歧义问题。



解决同音异义词

当声学模型无法区分多个可能的词时，语言模型通过分析上下文，判断哪个词序列更合理。

| | | |
|----|-------------|------------|
| 发音 | /si:/ | see 或 sea? |
| 上文 | swim in the | 更可能是 sea |



从 N-gram 到神经语言模型



N-gram 模型

统计连续 N 个词的出现频率，学习词语间的统计规律。

✗ 难以捕捉长距离依赖。



循环神经网络 (RNN)

具有记忆能力，能够捕捉序列数据中的长距离依赖关系。

✓ 适合处理长距离上下文。



Transformer 架构

结合自注意力机制，并行处理序列，高效捕捉长距离依赖。

✓ 高效长距离依赖，✓ 更好的并行计算。

“语言模型在提高语音识别准确率、使识别结果更符合语法和语义方面发挥着至关重要的作用。”

典型的语音识别任务主要有如下几类：

- 孤立词 VS. 连续语音识别
- 有限词表（小词表或中等词表）VS. 大词表
- 限制语法网络 VS. 非限制开放网络
- 安静环境（干净信道）VS. 噪声环境（复杂信道）

目前语音识别的主要研究趋势和任务：

- 大词表及复杂语言领域
- 说话人无关
- 复杂多变的噪声信道和环境
- 小型化、本地化、芯片化

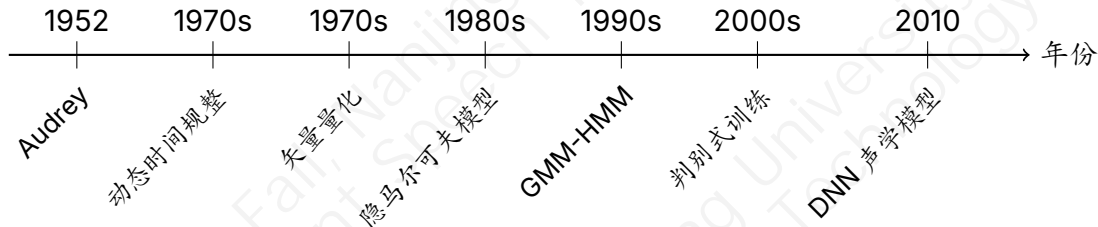
ASR 发展时间线 (深度学习之前)



南京大学
NANJING UNIVERSITY



智能科学与技术学院
School of Intelligence Science and Technology



- 本讲内容聚焦于 2010 年深度学习革命之前的经典方法。

1877 年 — 爱迪生发明留声机：首次实现声音的记录与再现。

1952 年 — Bell Labs 「Audrey」：单说话人数字 (0-9) 识别，准确率 90

1962 年 — IBM 「Shoebox」：可识别 16 个英语词与数字，支持简单算术指令。

1970 年代 — DARPA 语音理解计划：引入 HMM，出现可识别约 1000 词的 Harpy 系统。

1980 年代 — 统计方法主流化：HMM-GMM 框架确立；IBM Tangora 支持约 20k 词汇。

1996 年 — Dragon Dictate 发布：面向消费级的商用听写产品。

2009 年 — 深度学习进入 ASR：DNN-HMM 混合模型显著提升准确率（Hinton 等）。

2014 年 — 端到端兴起：CTC 等方法实现语音到文本的直接映射（Graves）。

2017 年 — Transformer 架构提出：为后续语音与序列建模提供新基石。

2020 年 — OpenAI Whisper：多语言识别，跨域稳健，无需精细调优亦具高准确率。

1877 留声机：声音记录与再现

1996 Dragon Dictate 商用化

1952 Audrey：数字识别

2009 DNN-HMM 大幅降错

1962 Shoebox：16 词/数字

2014 CTC 端到端

1970s DARPA + HMM, Harpy

2017 Transformer 架构

1980s HMM-GMM 主流, Tangora

2020 Whisper 多语言稳健



模板匹配时期

- ✓ 将输入语音与预存的模板进行比较识别。
- ✓ 主要依赖动态时间规整 (DTW) 等方法。
- ✗ 受限于特定说话人和孤立词识别。



统计建模时期

- ✓ 隐马尔可夫模型 (HMM) 和高斯混合模型 (GMM) 成为主流。
- ✓ HMM-GMM 框架确立, 支持大规模词汇量。
- ✓ 开始处理连续语音和非特定人语音。
- ✗ 仍需大量人工设计的声学特征。



深度学习时代

- ✓ 深度神经网络 (DNN) 替代 GMM 作为声学模型。
- ✓ DNN-HMM 混合模型极大提升特征学习能力。
- ✓ 端到端 (E2E) 架构兴起, 实现直接映射。
- ✓ 大规模预训练模型 (如 Whisper) 展现强大能力。

- 目标：将口语语言转换为书面文本。
- 应用：虚拟助手、语音转写服务、无障碍工具、命令与控制。
- 挑战：将变速、含噪、与说话人相关的声学信号 → 转换为离散、合乎语法的文本。

- 目标：将口语语言转换为书面文本。
- 应用：虚拟助手、语音转写服务、无障碍工具、命令与控制。
- 挑战：将变速、含噪、与说话人相关的声学信号 \rightarrow 转换为离散、合乎语法的文本。

数学描述

给定声学观测（特征）序列 $\mathbf{O} = (\mathbf{o}_1, \dots, \mathbf{o}_T)$ ，寻找最可能的词序列 $\mathbf{W} = (w_1, \dots, w_L)$ ：

$$\hat{\mathbf{W}} = \arg \max_{\mathbf{W}} P(\mathbf{W} | \mathbf{O})$$

贝叶斯定理 (Bayes' Rule)

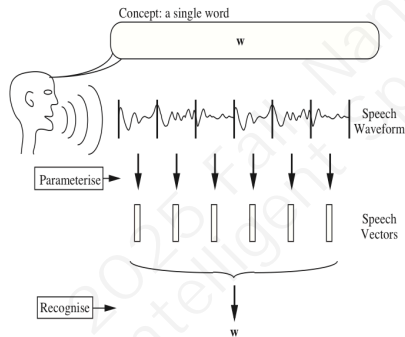
$$P(\mathbf{W} | \mathbf{O}) = \frac{p(\mathbf{O} | \mathbf{W}) P(\mathbf{W})}{p(\mathbf{O})}$$

最大后验概率 (MAP) 决策

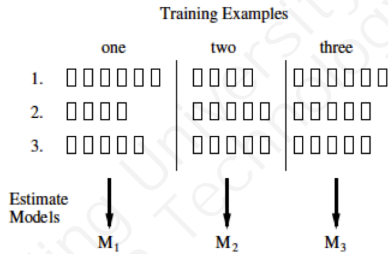
由于 $p(\mathbf{O})$ 与 \mathbf{W} 无关，优化目标可写为：

$$\hat{\mathbf{W}} = \arg \max_{\mathbf{W}} p(\mathbf{O} | \mathbf{W}) P(\mathbf{W})$$

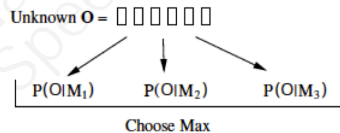
- $p(\mathbf{O} | \mathbf{W})$: 声学模型 (Acoustic Model, AM) —— 观测在给定词序列下的条件似然。
- $P(\mathbf{W})$: 语言模型 (Language Model, LM) —— 词序列的先验概率 (语法与语义合理性)。



(a) Training



(b) Recognition



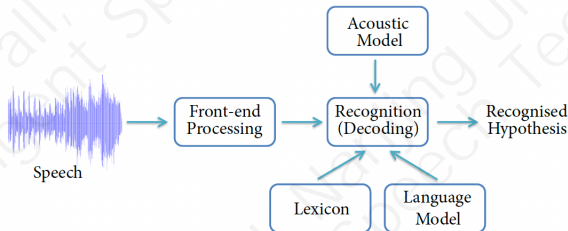
$$\hat{\mathbf{W}} = \arg \max_{\mathbf{W}} p(\mathbf{O} | \mathbf{W}) P(\mathbf{W})$$

$$\hat{\mathbf{W}} = \arg \max_{\mathbf{W}} p(\mathbf{O} | \mathbf{L}) P(\mathbf{L} | \mathbf{W}) P(\mathbf{W})$$

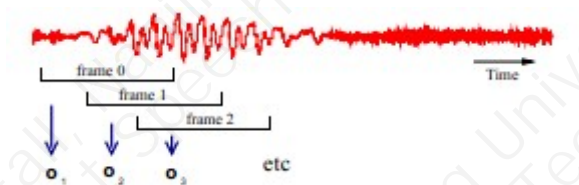
W: 词序列

L: 声学建模单元序列

O: 音频样本序列

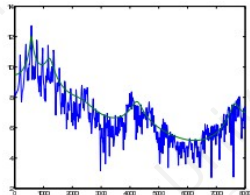
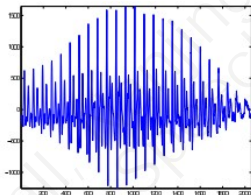


原始语音通过信号处理的方法转换成特征向量的序列。



- 降低信息率，但是保留有用信息
- 去除噪声或者其他的无关信息
 - 识别元音：最低的两个共振峰
 - 识别性别：音调 (pitch) 或者基音周期频率

语音识别中常见的声学特征



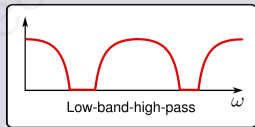
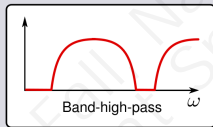
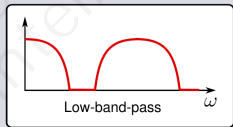
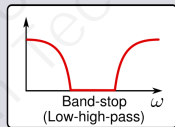
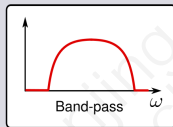
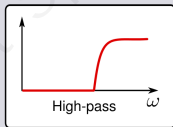
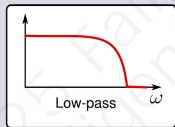
我们往往用短时谱信息来描述语音信号。一些有用的短时谱分析技术用于提取有用的特征，包括：

- 线性预测系数 (LPC)
- 滤波器组系数 (Fbank)
- 梅尔频率倒谱系数 (MFCC)
- etc.

滤波器 一般是一个设备或者处理过程, 它可以从原始信号中将那一些不想要的成分或者特征去除

滤波操作一般情况下作用在频率域上:

$$Y(\omega) = H(\omega)X(\omega)$$



前端处理 — 特征提取

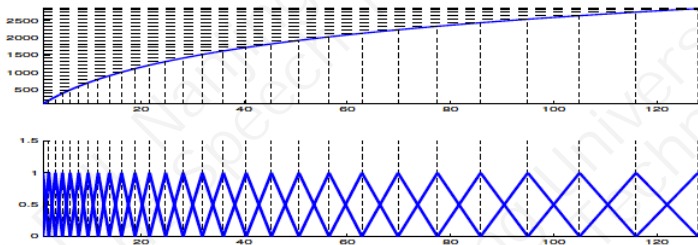
Filter Bank 系数 (FBank)



- 通过短时傅里叶变换得到的幅值谱包含太多的信息
- **Filter bank** 是一系列带通滤波器 (一般用三角窗滤波器)
- 每一个带通滤波器输出一个 FBank 系数, 它等于此带通滤波器内的信号加权和

前端处理 — 特征提取

梅尔域 (Mel Scale)



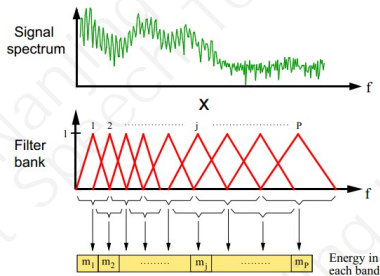
Mel scale 是一个基音感知域，它通过听音者可以区分两种纯音频率的差距来作为标度。Mel 域和线性频域之间通过一个非线性的映射函数可以相互转换：

$$\text{Mel}(f) = 2595 \log_{10} \left(1 + \frac{f}{700} \right)$$

注意：在 Mel 域中，低频部分具有更高的分辨率。

前端处理 — 特征提取

梅尔域的 FBank 系数



$$m_i = \sum_{k=f_i}^{F_i} s(k)T_i(k)$$

其中 m_i 是第 i^{th} 个 FBank 系数, f_i 和 F_i 分别是三角滤波器的开始和结束频率, $s(k)$ 是在频点 k 的谱的能量 (有时为谱的幅值), $T_i(k)$ 是三角滤波器的值。

MFCC 在许多语音处理的领域被广泛使用。它们从 Mel 域的 FBank 系数衍生得到：

- ① 取对数 **logarithm** 得到 N 个对数域的 FBank 系数
- ② 计算倒谱 **Cepstral** 系数，使用 **Discrete Cosine Transform (DCT)** 变换

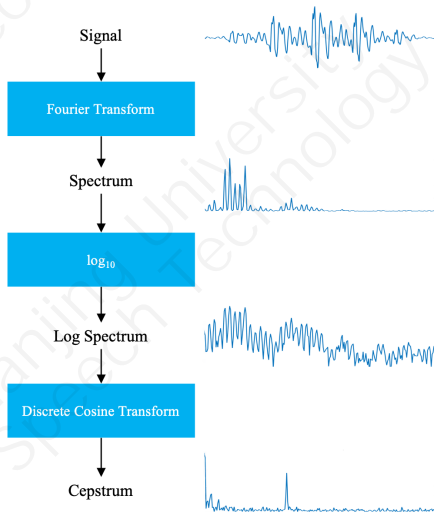
$$c_n = \sqrt{\frac{2}{N_{fb}}} \sum_{j=1}^{N_{fb}} \log(m_j) \cos\left(\frac{\pi n}{N_{fb}}(j - 0.5)\right) \quad n = 1, 2, \dots, N_{mfcc}$$

其中 c_n 是第 n^{th} 个 MFCC 系数, m_j 是第 j^{th} 个 mel 域的 FBank 系数, N_{fb} 和 N_{mfcc} 分别是 FBank 系数和最终的 MFCC 系数的数量 (通常情况下 $N_{mfcc} = 12$, N_{fb} 变化范围是 20-30, 甚至 40)。

语音短时谱能量没有直接用来作为特征参数，因为

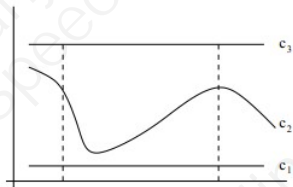
- 语音的能量谱不符合高斯分布
- 所有系数对响度很敏感
- 各个系数之间都高度相关

Discrete Cosine Transform (DCT) 能够有效地去除不同系数之间的相关性。



概念

MFCC 特征很好地描述了即时语音信号谱 (静态), 但是不能描述信号的动态特性。



这个不足可以通过在特征向量中添加静态系数的差分特征来得到改善。

$$\mathbf{o} = \begin{bmatrix} \mathbf{c} \\ \Delta \mathbf{c} \\ \Delta \Delta \mathbf{c} \end{bmatrix}$$

- 简单的差分系数可以计算如下：

$$\Delta_n = \frac{\mathbf{c}_{n+\delta} - \mathbf{c}_{n-\delta}}{2\delta}$$

- 更鲁棒的估计是通过一系列语音帧的最佳线性回归系数来得到：(这里是 $2\sigma + 1$)

$$\Delta_n = \frac{\sum_{i=1}^{\delta} i(\mathbf{c}_{n+i} - \mathbf{c}_{n-i})}{2 \sum_{i=1}^{\delta} i^2}$$

- 更高阶的差分系数可以通过以上过程的迭代形式来获得：

$$\Delta\Delta_n = \frac{\sum_{i=1}^{\delta} i(\Delta_{n+i} - \Delta_{n-i})}{2 \sum_{i=1}^{\delta} i^2}$$

- 通常情况下会使用 2^{nd} or 3^{rd} 系数

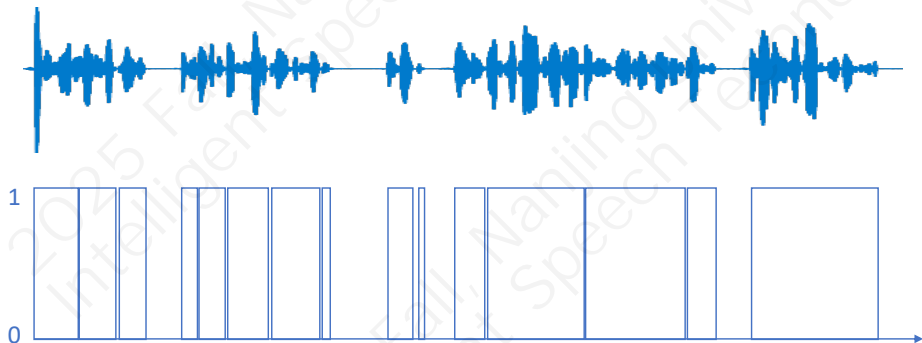
- ① 预加重 (Pre-emphasis): 提升高频分量。
- ② 分帧 (Framing) 与加窗 (Windowing): 将信号切分为约 25ms 的短帧。
- ③ 离散傅里叶变换 (DFT/FFT): 将每一帧从时域转换到频域, 得到频谱。
- ④ 梅尔滤波器组 (Mel Filter-bank): 对频谱进行滤波, 模拟人耳听觉特性。
- ⑤ 对数运算 (Log): 对滤波器组能量取对数。
- ⑥ 离散余弦变换 (DCT): 解除特征之间的相关性, 得到最终的 MFCC 系数。

最终特征向量

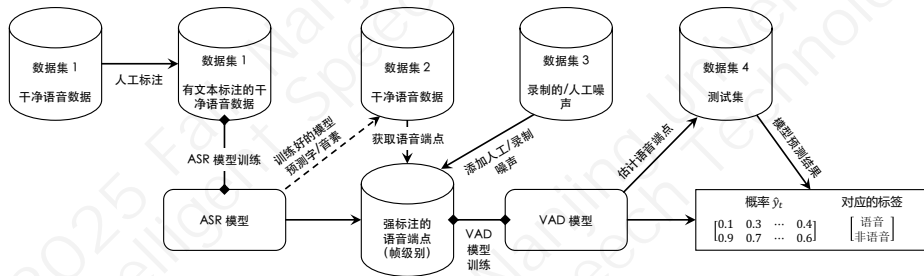
MFCC (12-13 维) + 能量 (1 维) + 一阶差分 (Deltas) + 二阶差分 (Double-Deltas) \rightarrow 39 维特征向量

语音端点检测 — Voice Activity Detection (VAD)

- 主要任务：检测一段音频中是否存在人类的语音
- 作用：
 - 减少不必要的语音处理，节省运算开销和功耗
 - 也能提升后续语音处理的性能（减少了不必要的非语音噪声输入）



传统的有监督 VAD 模型训练流程



前端处理 — 语音端点检测 (VAD)

传统的有监督 VAD 模型训练流程



- 1. 采集干净的单人语音数据，尤其是音频质量相差不大的数据。确保数据中仅包含固定种类的一种/多种语言。
- 2. 手动对这些数据进行文本标注。
- 3. 在这些数据上训练一个 ASR 模型。
- 4. 用训练好的 ASR 模型来预测新的（通常干净）数据集中的音频是否存在语音，以得到帧级别的强标注。
 - ASR 模型可以提供音素级别的对齐，进而转换为二值的语音指示（有语音的帧设为 1，静音段设为 0）
- 5. 在新数据集上用前一步得到的强标注训练一个 VAD 模型（DNN/CNN/RNN 等）。

前端处理 — 语音端点检测 (VAD)

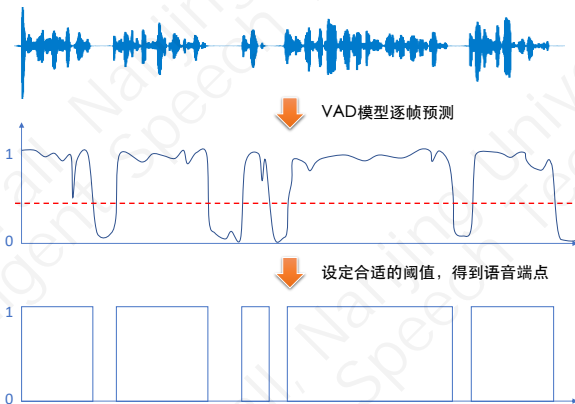
VAD 模型测试阶段



南京大学
NANJING UNIVERSITY



智能科学与技术学院
School of Intelligence Science and Technology



Recap: 统计语音识别

大词汇连续语音识别系统架构



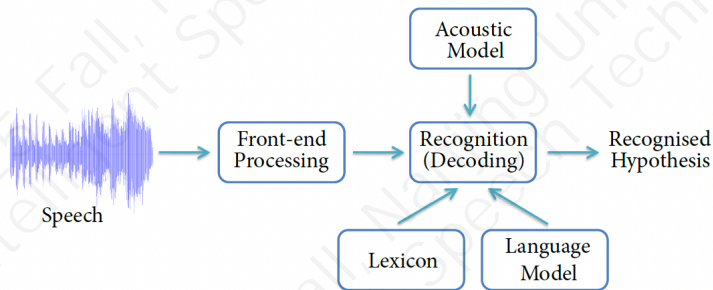
$$\hat{\mathbf{W}} = \arg \max_{\mathbf{W}} p(\mathbf{O} | \mathbf{W}) P(\mathbf{W})$$

$$\hat{\mathbf{W}} = \arg \max_{\mathbf{W}} p(\mathbf{O} | \mathbf{L}) P(\mathbf{L} | \mathbf{W}) P(\mathbf{W})$$

W: 词序列

L: 声学建模单元序列

O: 音频样本序列



声学模型是一个概率模型，它可以刻画用来描述不同声音的声学特性。

- 语音识别最关键的技术之一
- 概率模型 $p(\mathbf{O}|\mathbf{L})$ 用于刻画不同语音单元，如音素，音节，字，词
- **Hidden Markov Model (HMM)** 隐马尔科夫模型由于其概念的简单以及数学的完备，被最广泛地采用。

HMM 可以认为是一个最基本的有限状态转录机 (FST)，它可以将一个用于表示语音的特征向量序列，通过有限状态机，转换成状态机的状态序列（表示音素、音节或词）。

语言模型是一个 概率模型 probabilistic model:

- ① 引导搜索算法 (在给定历史的情况下预测下一个词)。
- ② 消除声学单元之间的混淆性, 特别是那些声学层相似的单元。

Great wine v.s. Grey twine

语言模型将概率分配到一串要识别的 tokens (通常是词) 上:

- 上下文自由语法:
($\langle s \rangle$ \langle one | two | three \rangle $\langle /s \rangle$)
- 统计语言模型: n -gram 语言模型

$$P(w_1, w_2, \dots, w_N)$$

n -gram 统计语言模型和 HMM 声学模型由于容易结合而被广泛地用于语音识别中。

字典模型为声学模型和语言模型之间构建了桥梁。

- 它在词和声学单元之间定义了一个映射。
- 它可以是一个确定化的模型 (**deterministic**)

| Word | Pronunciation |
|----------|-----------------|
| TOMATO | t ah m aa t ow |
| | t ah m ey t ow |
| COVERAGE | k ah v er ah jh |
| | k ah v r ah jh |

- 它也可以是一个概率模型 (**probabilistic**)

| Word | Pronunciation | Probability |
|----------|-----------------|-------------|
| TOMATO | t ah m aa t ow | 0.45 |
| | t ah m ey t ow | 0.55 |
| COVERAGE | k ah v er ah jh | 0.65 |
| | k ah v r ah jh | 0.35 |

ASR 系统性能评估：通过比较第 i^{th} 句话的识别假设标注 H_i 和它的真实参考标注 R_i ，来统计得到最后的识别性能。

- 句子错误率 Sentence error rate (SER)
- 词错误率 Word error rate (WER) / 字错误率 Character error rate (CER)
- 音素错误率 Phone error rate (PER) / 音节错误率 Syllable error rate (YER)

SER 定义：

$$\text{SER} = \frac{1}{N} \sum_{i=1}^N \delta(H_i, R_i) \times 100\% \quad \delta(H_i, R_i) = \begin{cases} 1 & H_i = R_i \\ 0 & H_i \neq R_i \end{cases}$$

其中 N 是测试集合的总语句数。

- 计算两个序列之间的差距 (a.k.a. 编辑距离)
- 定义: 将一个序列变成另一个序列所需要的最少修改次数
- 动态规划算法可以用于计算编辑距离

WER/CER/PER/YER 计算如下:

$$\text{WER} = \frac{\sum_{i=1}^N (I_i + D_i + S_i)}{\sum_{i=1}^N T_i}$$

其中 I_i , D_i 和 S_i 分别是通过编辑距离算法计算每一对 (R_i, H_i) 后得到的各种错误的数量, 包括: 插入 insertions, 删除 deletions 和替代 substitutions。 T_i 是真实参考标注 R_i 中的单元总数 (如: 词, 音素, 字, 音节)。

Q: WER 能否大于 100%?

Levenshtein Distance 示例

SATURDAY v.s. SUNDAY



南京大学
NANJING UNIVERSITY



智能科学与技术学院
School of Intelligence Science and Technology

| | | S | A | T | U | R | D | A | Y |
|---|---|---|---|---|---|---|---|---|---|
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| S | 1 | | | | | | | | |
| U | 2 | | | | | | | | |
| N | 3 | | | | | | | | |
| D | 4 | | | | | | | | |
| A | 5 | | | | | | | | |
| Y | 6 | | | | | | | | |

- 初始化: 填满第一行和第一列
 - 第一行: 插入 (insertion) 错误的数量
 - 第一列: 删除 (deletion) 错误的数量
- 参考标注: SUNDAY
- 假设标注: SATURDAY

Levenshtein Distance 示例

SATURDAY v.s. SUNDAY



南京大学
NANJING UNIVERSITY



智能科学与技术学院
School of Intelligence Science and Technology

| | | S | A | T | U | R | D | A | Y |
|---|---|---|---|---|---|---|---|---|---|
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| S | 1 | 0 | 1 | | | | | | |
| U | 2 | | | | | | | | |
| N | 3 | | | | | | | | |
| D | 4 | | | | | | | | |
| A | 5 | | | | | | | | |
| Y | 6 | | | | | | | | |

- 考虑 entry ($i = 1, j = 1$)

- 插入: $d[1][0] + 1 = 2$
- 删除: $d[0][1] + 1 = 2$
- 替代: $d[0][0] + 0 = 0$

- 最小代价: $\text{cost} = 0$

- 考虑 entry ($i = 1, j = 2$)

- 插入: $d[1][1] + 1 = 1$
- 删除: $d[0][2] + 1 = 3$
- 替代: $d[0][1] + 1 = 2$

- 最小代价: $\text{cost} = 1$

Levenshtein Distance 示例

SATURDAY v.s. SUNDAY



南京大学
NANJING UNIVERSITY



智能科学与技术学院
School of Intelligence Science and Technology

| | | S | A | T | U | R | D | A | Y |
|---|---|---|---|---|---|---|---|---|---|
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| S | 1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| U | 2 | 1 | 1 | 2 | 2 | 3 | 4 | 5 | 6 |
| N | 3 | 2 | 2 | 2 | 3 | 3 | 4 | 5 | 6 |
| D | 4 | 3 | 3 | 3 | 3 | 4 | 3 | 4 | 5 |
| A | 5 | 4 | 3 | 4 | 4 | 4 | 4 | 3 | 4 |
| Y | 6 | 5 | 4 | 4 | 5 | 5 | 5 | 4 | 3 |

因此, 最优的 *Levenshtein distance* 是 3 (2 insertions and 1 substitution)

| REF | S | | | U | N | D | A | Y |
|-------|---|---|---|---|---|---|---|---|
| HYP | S | A | T | U | R | D | A | Y |
| Edits | | I | I | | S | | | |

- 特征提取 **Feature extraction** (前端处理)
- 声学模型 **Acoustic model** $p(\mathbf{O}|\mathbf{L})$: 在给定声学单元 (如音素) 的条件下对特征分布进行建模。
- 字典 **Lexicon** $P(\mathbf{L}|\mathbf{W})$: 声学建模单元和词之间的映射。
- 语言模型 **Language model** $P(\mathbf{W})$: 产生词序列的概率。

$$\hat{\mathbf{W}} = \arg \max_{\mathbf{W}} p(\mathbf{X}|\mathbf{O})p(\mathbf{O}|\mathbf{L})P(\mathbf{L}|\mathbf{W})P(\mathbf{W})$$

- 解码算法 **Decoding algorithm**: 基于以上各种信息源, 找到“最优”词序列。
- 结果评估 **Evaluation**: 对比 *hypotheses* (识别假设标注) 和 *reference* (参考标注)



- **Hidden Markov Toolkit:** <http://htk.eng.cam.ac.uk>
- **Kaldi:** <http://kaldi.sourceforge.net>

- **wenet**: <https://github.com/wenet-e2e/wenet>
- **funasr**: <https://github.com/modelscope/FunASR>
- **k2**: <https://github.com/k2-fsa/k2>
- **firered-asr**: <https://github.com/FireRedTeam/FireRedASR>
- **whisper**: <https://github.com/openai/whisper>
- ...

Assignment 1

设计一个 ASR 打分工具



南京大学
NANJING UNIVERSITY



智能科学与技术学院
School of Intelligence Science and Technology

要求

- 1 实现 Levenshtein Distance 算法
- 2 支持中英双语
- 3 考虑 text normalization
- 4 在现有的开源模型上进行验证
- 5 ...

提交内容

- 1 源代码
- 2 报告

推荐阅读

- 《Fundamentals of Speech Recognition》 Lawrence Rabiner & Biing-Hwang Juang

致谢

本节 Slides 的制作参考了多位学者的优秀资料。部分内容与图示来自上海交通大学俞凯教授的课程 Slides，在此谨致谢意。