语音识别技术基础

罗平峰

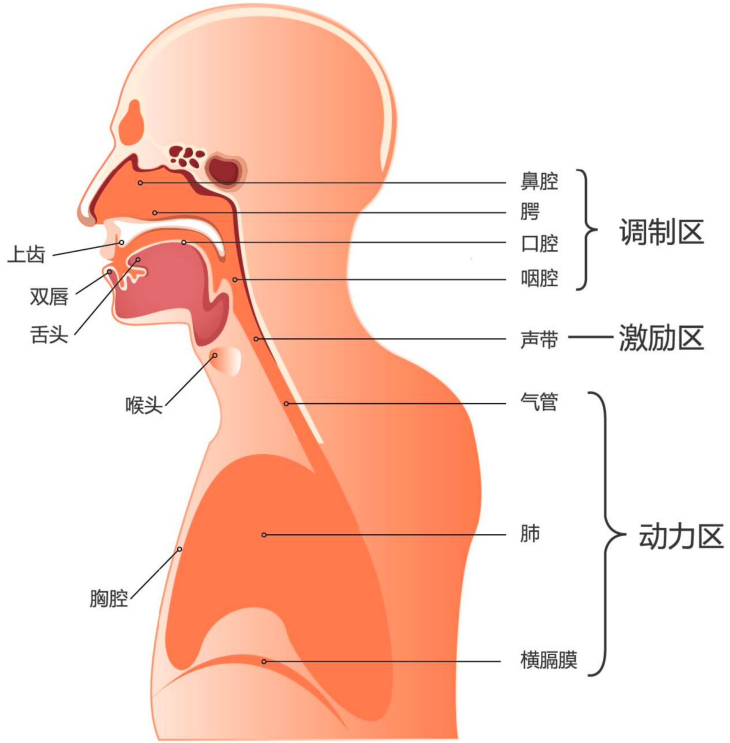
2023/01

目 录

语音信号的产生

激励响应模型：人脑根据要表达的信息，控制肺部产生气流的动力，经过气管引起声带振动形成声源(通常称为激励)；最后经过声道(咽腔、口腔、鼻腔等区域)调制后由口唇辐射出来，最终产生了我们所听到的语音。

信号的采集：麦克风将声波转换成电压（电磁感应），完成模数信号转换，方便计算机处理。



语音识别的原理

基本问题：给定一系列观测信号，在语言空间中找到最可能的文本序列。

数学原理（后验概率最大）：

$$W^* = \text{argmax}_W \ P(W|X)$$

若使用贝叶斯定理，则可得

$$P(W|X) = \frac{P(X|W)P(W)}{P(X)}$$

$$\propto P(X|W)P(W)$$

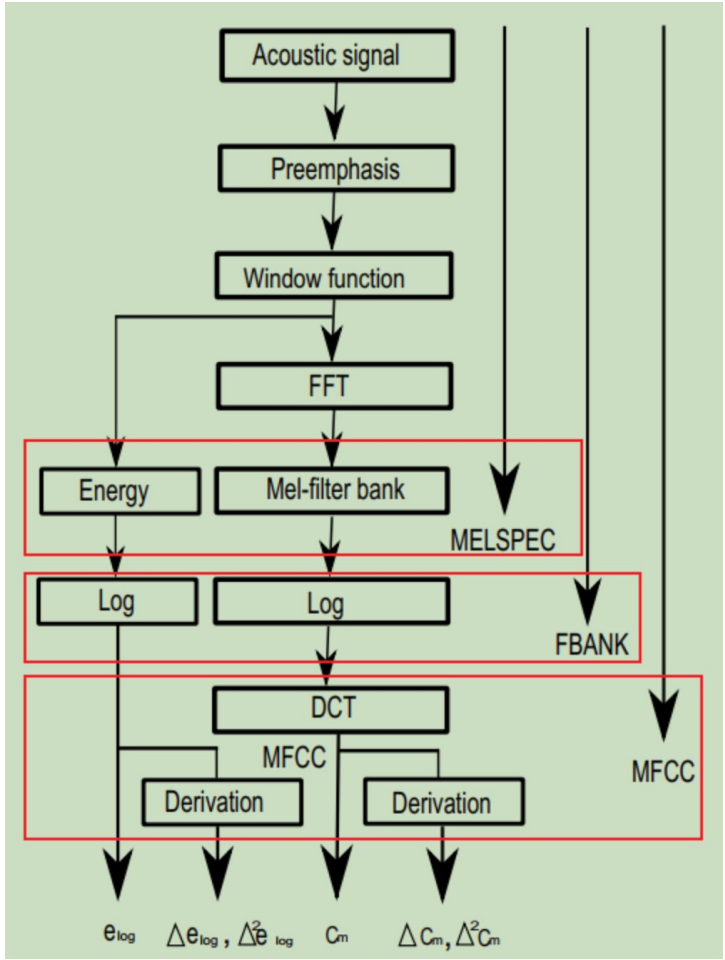
方法一：拆分建模的思路为传统的语音识别方法，即声学模型P（X | W）、语言模型P(W)独立建模。有GMM-HMM、DNN-HMM等，其中Kaldi框架中实现主要就是这两种。

方法二：直接对P(W | X)进行建模，即声学和语言模型放在一个系统进行联合建模，则为目前的端到端的语音识别方法。有CTC、RNNT、Attention等，其中espnet、K2、wenet等框架实现的是这类最新的E2E方法。

语音识别的方法——信号处理

特征提取

- 分帧：一般取**25ms**（太长则不满足短时平稳假设，太短则无法表征特征）
- 预加重：缓解高频能量的衰减
- 加窗：缓解频谱泄漏的现象
- FFT：时频转换（三角函数的正交性）
- 三角滤波：仿人耳声学感知变换和减少参数量（每个bin合并成一个能量点计算）
- 取对数：非线性变换



问题

- 传统傅立叶分析的局限性？（平稳假设、相同频率分辨率、时间分辨率为零）
- 联合时频分析方法：短时傅里叶变换（上述特征提取部分）、小波分析等
- 直接从raw signal建模 （类似图像，RGB信号直接输入模型）

语音识别的方法——信号处理

前端处理（改善信号质量）

- 加性噪声
  - 谱减法，假设噪声和原始语音的能量谱叠加得到带噪信号，估计出噪声能量谱，相减和平滑即可。
- 混响和回声
  - 估计出声源到接收端的传递函数（房间的耐冲响应函数表示RIR），设计一个滤波器和RIR抵消
  - 设计逆滤波器使得生成的LPC参数非高斯化
  - 基于T60(衰减60db)估计RIR，然后利用谱减法
  - 线性预测模型（当前的信号由历史的信号延迟衰减并叠加当前信号形成）
- 信道差异：覆盖和补偿
- 麦克风阵列：
  - 阵列类型：线性阵列、环形阵列
  - 增益是入射角的函数；控制每路麦克风的延时即可控制指向性（相位一致）
  - 不同麦克风接受的噪音不相关，叠加则会抵消
  - 可利用时间和空间信息，实现方向选择(延迟加和)、去噪、去混响
- CMVN：在线滤波器可以理解为一种高通滤波器（滤掉固定不变的成分）
- DAE：去噪自编码器，干净数据和带噪数据

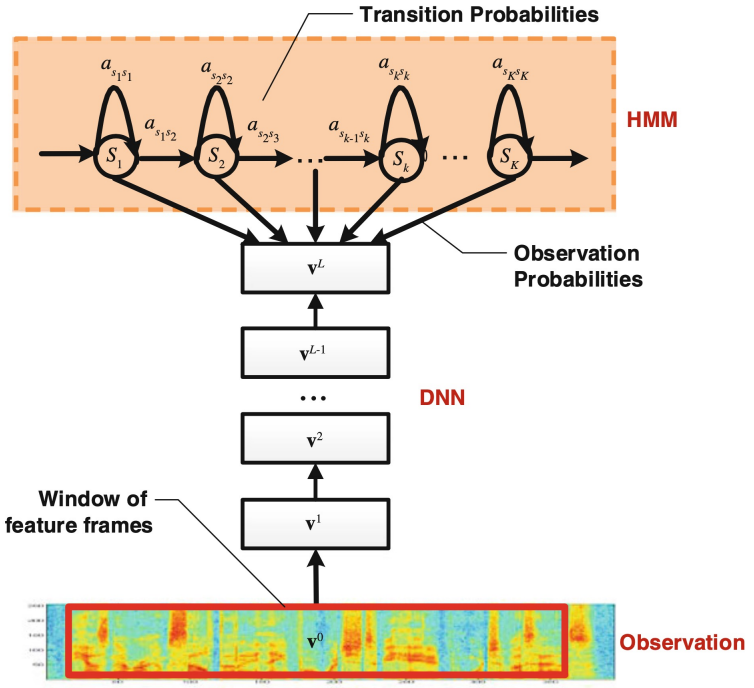
语音识别的方法——传统语音识别

根据前述，是否对后验概率直接建模，语音识别可以分成传统方法和E2E方法。

传统方法的思路：通过声学模型将信号识别成音素序列，音素序列在声学和语言模型的共同约束下识别成字词序列。

P(X | W) 建模（声学模型）

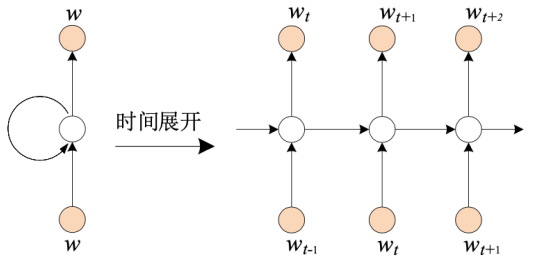
- GMM-HMM
  - 建模单元：一般选择音素，考虑到音素上下文相关和协同发音等信息，会进一步使用三音素或者双音素作为基础的建模单元，并通过聚类（合并相近的类，减少数量）得到最终的建模单元。
  - 模型结构：每个建模单元（三音素或双音素）都用一个HMM表示，包含转移概率和发射概率(GMM模型建模)，两者都可建模，但后者一般更重要而前者可以取固定的值。
  - 参数估计：GMM和HMM的参数使用EM算法
    - E步：
      - 根据现有参数计算$P(ot \mid m)$
    - M步：
      - 根据$P(ot \mid m)$ 更新GMM参数
    - 不断重复E和M步，直到收敛
- DNN-HMM
  - 建模单元：和GMM-HMM相同，为三音素或者双音素
  - 模型结构：
    - 将GMM-HMM中DNN替换成DNN
    - DNN可以是TDNN、LSTM、CNN等网络结构
  - 参数估计：使用梯度下降算法估计DNN的参数



语音识别的方法——传统语音识别

P(W)建模（语言模型）

- Ngram
  - 使用N阶的马科夫模型建模：在给定历史词汇的条件下预测当前词汇出现的概率
  - 通过统计词频和利用平滑算法估计文法的概率值
  - 公式：$P(wi \mid wi-n+1,wi-n+2,...,wi-1) = count(wi-n+1,wi-n+2,...,wi) \ / \ count(wi-n+1, wi-n+2, ..., wi-1)$
- NNLM模型
  - 使用历史的词汇信息预测当前词汇
  - 网络的输出单元为词典的单元

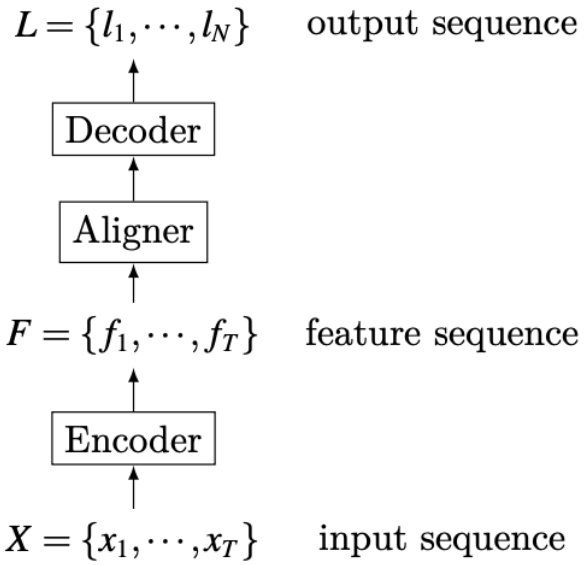


- Ngram和NNLM对比

算法	优势	劣势
Ngram	存储是文法的概率值，计算复杂度O(1) 方便编辑，例如领域适应、文法概率惩罚或激励	建模长历史信息的能力较弱 阶数增加则参数相应增加指数倍数
NNLM	建模较长的历史信息	训练好后参数不易修改 文法概率需要临时计算，耗时较大（会有些优化算法可以将NNLM转成NGRAM格式存储，从而不需要每次进行前向计算）

语音识别的方法——E2E语音识别

E2E的语音识别建模方法（CTC、RNNT、AED）



E2E	model	math
CTC	Encoder	P(Yt | Xt)
RNNT	Transciptor/Preditor/Joiner	P(Yu | X1:t, Y1:u-1)
Attention	Encoder/Attention/Decoder	P(Yu | X1:T, Y1:u-1)

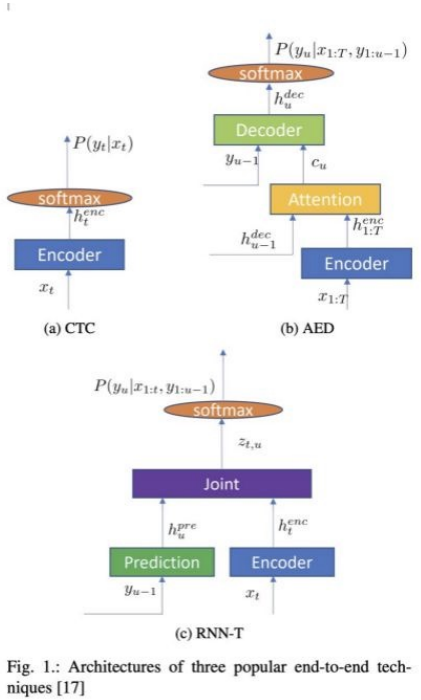


Fig. 1.: Architectures of three popular end-to-end techniques [17]

语音识别的方法——E2E语音识别

CTC

- 基本形式：
  - 给定表示X的条件下，关于当前输出Y的各种对齐的概率
  - 通过引入blank符号和考虑所有可能对齐解决对齐问题
  - 一个网络之后直接接softmax，模型较简单
  - output label序列长度小于等于输入序列长度
  - 各个output的label之间是独立的（没有LM的条件概率建模）
- 数学意义：对P(Yt | Xt)建模
- 公式展开：

$$P_{\text{CTC}}(\mathbf{y}|\mathbf{x}) = \sum_{\mathbf{a}\in\mathcal{B}^{-1}(\mathbf{y})} P(\mathbf{a}|\mathbf{h})$$

$$= \sum_{\mathbf{a}\in\mathcal{B}^{-1}(\mathbf{y})} \prod_{t=1}^{T'} P(a_t|h_t)$$

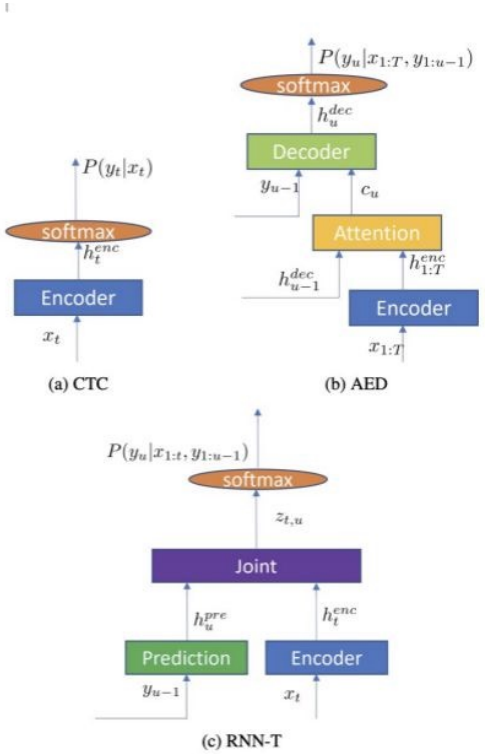


Fig. 1.: Architectures of three popular end-to-end techniques [17]

语音识别的方法——E2E语音识别

RNNT

- 基本形式：
  - 给定X**和历史输出Y的条件下**，关于当前输出Y的各种对齐的概率
  - 通过考虑所有可能对齐解决对齐问题
  - 当前的output依赖于历史的output（显示的LM建模）
  - input和output之间存在一对多的情况
- 数学意义：对P(Yu | X1:t, Y1:u-1)建模
- 公式展开：

$$P_{\mathrm{RT}}(\mathbf{y}|\mathbf{x}) = \sum_{\mathbf{a}\in\mathcal{B}^{-1}(\mathbf{y})} P(\mathbf{a}|\mathbf{h})$$

$$= \sum_{\mathbf{a}\in\mathcal{B}^{-1}(\mathbf{y})} \prod_{t=1}^{T'} P(a_t|h_t, y_{<u_t})$$

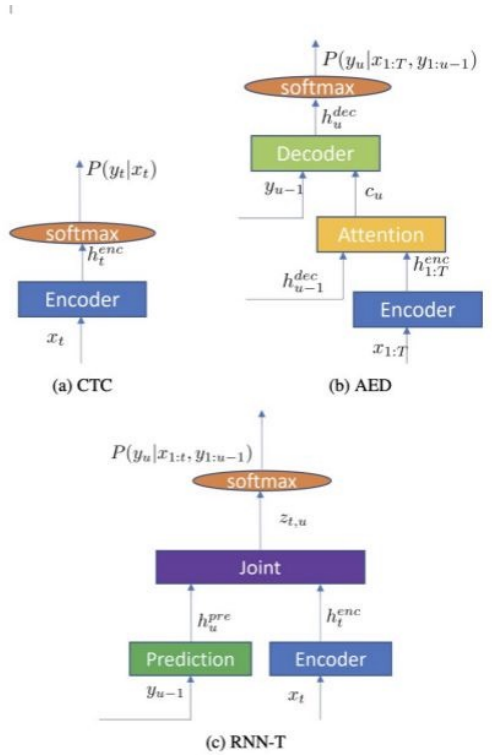


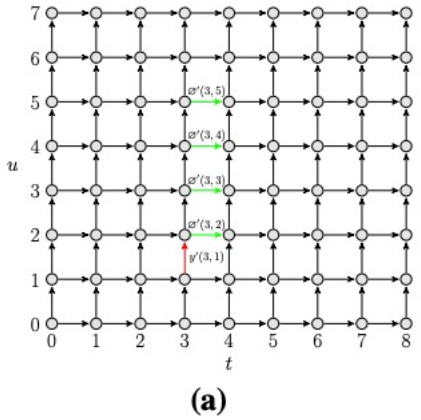
Fig. 1.: Architectures of three popular end-to-end techniques [17]

语音识别的方法——E2E语音识别

Attention

- 基本形式：
  - 给定X的加权注意力和历史输出Y的条件下，关于当前输出Y的概率
  - 通过使用注意力机制解决对齐问题
  - 标准形式是使用全局注意力
  - 流式支持：需要改成局部注意力（WER会有损失）
- 数学意义：对P(Yu | X1:T, Y1:u-1)建模
- 公式展开：

$$P_{\text{Attn}}(\mathbf{y}|\mathbf{x}) = P(\mathbf{y}|\mathbf{h}) = \prod_{u=1}^{U} P(y_u|c_u, y_{<u})$$



$$c_u = \sum_{t=1}^{T} \alpha_{u,t} h_t$$

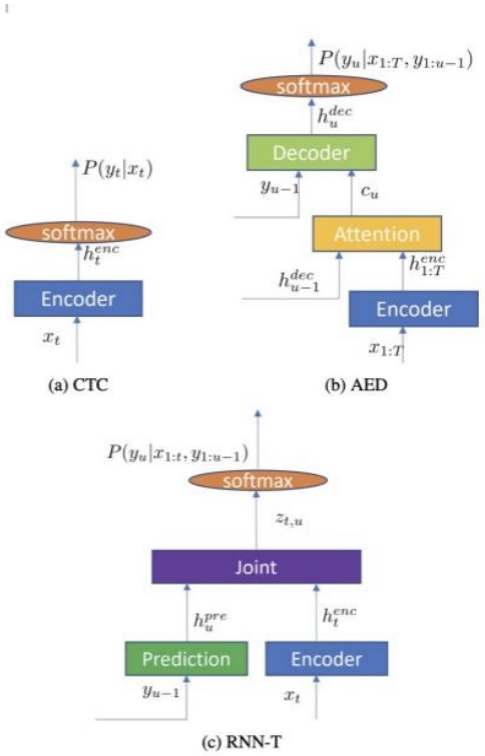


Fig. 1.: Architectures of three popular end-to-end techniques [17]

语音识别的方法——E2E语音识别

语音识别的传统方法和E2E方法对比

算法	优势	劣势
传统	模块拆分，方便独立优化特定子模块	声学和语言模型独立建模（模块建模的累计误差、个别模块需要专家知识、系统复杂模块众多）
E2E	联合建模（无累积误差、统一优化、架构简单）	依赖大量数据、引入额外文本在理论上不太直接

工程实践进展

K2基础功能

算法	核心点
WFST	相对传统WFST（如openwfst）不同主要是可微分（指的是FSA算法、运算是可导，比如求最短路径） 高效的GPU求导和解码（compose、shortest等算法）
LF-MMI	E2E，不依赖HMM（相对Kaldi简洁）
CTC	可以结合各种模型和loss（attention、LF-MMI等）
RNNT	Pruned（训练速度提升2-8倍) Emformer、Chunk-Based-Confomer（支持流式）
Rransfer learning	VQ based（度量模型编码后向量的距离）
Rework	Eva(adam + RMS-control)、ActivationBalancer、BasicNorm、DoubleSwish、Warmup（model level）

Training data preparation

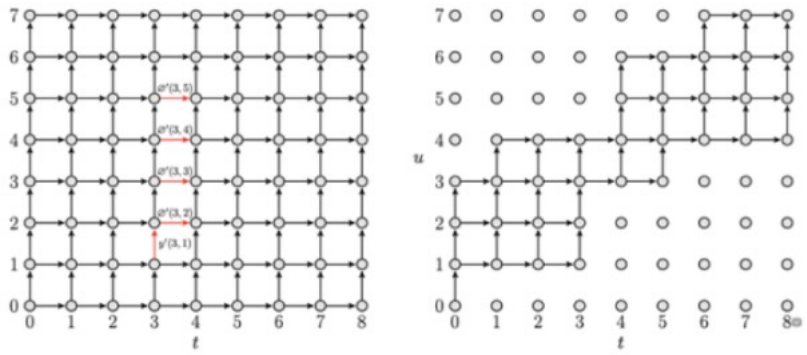
Core Algorithms

Recipes

工程实践进展

RNNT剪枝

- 标准的前后向算法需要考虑T和U两个维度（espnet等框架rnnt速度慢、较难训练）
- 语音和文本满足单调对齐 -> 只有对角线上小部分路径是有效的
- 先用am+lm估算有效的 ‘窄带’ ， 计算am+lm+joiner只考虑窄带内的节点，使得计算复杂度从T * U 降到 T * beam （其中beam << U）



多	更大的batch_size 更多的模型参数
快	计算复杂度降低
省	显存占用更少
好	丢掉噪声干扰

工程实践进展

1.算法维度和业务价值

算法	核心点
WFST	• 训练阶段：方便联合使用传统Ngram模型，支持GPU加速 • 解码阶段：不依赖openwfst，方便使用海量文本（Ngram查找复杂度O(1)，快NNLM几个数量级），on-the-fly-compose（not rescore）

E2E	• 更好的单一模型
	  • RNNT 联合声学语言建模同时支持流式（对比ctc+attention的hybrid模型，ctc的output独立性假设导致ctc本身无lm，而chunk-based流式弱化了hybrid模型基于attention的LM）
	  • 具有更高的准确率(满足全场景识别的需求和一致的体验)、更稳定的流式识别(提升中间结果的效果)、更低的功耗（落地可参考google on-device-speech）
	• 也支持组合优化各种Loss和模型构建复杂系统
	  • LF-MMI、CTC、RNNT、Attention

2.实验效果

Toolkit	Test_Net	Test_Meeting
ESPNET	8.9	15.9
WENET	9.7	15.59
K2_release	8.71	13.41
K2_our_finetune	8.14	13.23

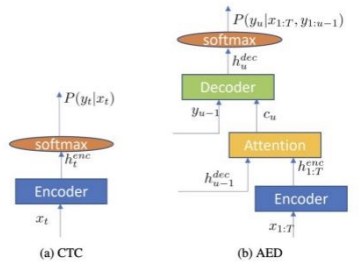


Fig. 1.: Architectures of three popular end-to-end techniques [17]

语音识别未来的发展方向

完全端到端的语音识别	目前端到端的方法通常只是考虑声学和语言模型的同时优化； 如果能进一步，“同时优化信号处理、特征表示、声学模型、语言模型”，那么这种完全端到端方法的建模能力和鲁棒性将会更强。
低资源语音识别	方言、少数语种、特定场景的音频获取和标注都非常困难（音频数据一般成本在100-500元/小时，一些敏感场景或者稀有数据甚至无法获得），如果解决低资源语音识别将极大降低算法落地成本。大致思路有：知识迁移、无监督、半监督（少量标注、海量无标注）。
自适应语音识别	模型根据上下文进行识别结果自适应；例如将历史信息或者用户指令，作为先验塞给prompt，影响模型解码的偏向；例如prompt设定“当前为游戏场景”则可以强化相关游戏领域专有术语的识别率，从而特定领域定制和自适应的效果。
多任务多模态音频理解模型	语音方向从单任务往多任务（多任务音频模型包括whisper、audioPaLM、SpeechPrompt等），从单模态往多模态（多模态模型包括audioPaLM、SeamlessM4T等）发展。