

Lecture 05: 高阶语音识别

预训练模型与大语言模型

王帅

南京大学智能科学与技术学院

2026 年 10 月 9 日



南京大學
NANJING UNIVERSITY



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

- ① **语音预训练**: 自监督目标的构造与语音表示学习。
- ② **比较与动机**: LLM-ASR 的优势来源、实验证据与适用条件。
- ③ **跨模态连接**: 语音表示的压缩、投影与输入组织。
- ④ **代表性架构**: 连续表示、离散单元与指令条件化。
- ⑤ **系统评测**: 识别精度、上下文效应与推理延迟。

先修概念: 语音编码器、注意力机制、CTC 与自回归序列建模。

同一段语音：“嗯，我想订周五，不，周六的票。”

逐字转写 嗯，我想订周五，不，周六的票。

整理转写 我想订周六的票。

意图理解 订票，日期为周六。

- 研究任务为**语音到文字转写**，采用转写参考答案评价识别性能。
- 逐字转写与整理转写对应不同的输出规范，需分别定义参考答案。

系统	语音部分	文字输出部分
常规端到端 ASR	面向 ASR 训练的编码器	CTC、AED 或 RNN-T
预训练语音模型 + ASR	复用语音预训练编码器	ASR 输出层与任务微调
LLM-ASR	语音编码器 + 连接模块	复用文本或多模态 LLM，生成转写

预训练策略与端到端建模属于不同的设计维度。

Decoder-only 通常描述文本生成模块，系统仍可包含独立语音编码器。

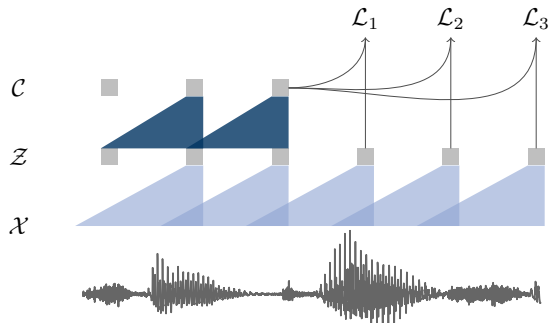
数据	已知信息	监督信号
人工转写语音	音频和可靠文字	直接学习语音与文字的对应
无转写音频	波形及其上下文	从音频构造自监督目标
带字幕或伪标签的音频	可能有误的文字	筛选、对齐后进行弱监督训练
大规模文本	词汇及语言上下文	预训练文本 LLM

两种知识来源：语音预训练学习声学结构，文本预训练学习语言规律。

- ① **预训练**: 从大量音频中构造预测任务, 训练语音编码器。
- ② **迁移**: 保留编码器, 添加 **ASR** 输出层或 **LLM** 连接模块。
- ③ **适配**: 用语音与文字配对数据学习转写任务。

预训练与任务适配的分工

语音预训练学习声学表示, 任务适配建立语音表示与文字序列的对应关系。



- 1 CNN 编码波形，得到短时向量 z_t 。
- 2 上下文 CNN 汇聚当前及此前表示，得到 c_t 。
- 3 用 c_t 区分真实未来表示 z_{t+k} 与采样的负例。

预测目标来自音频本身，不需要人工转写。

wav2vec (2019): 预训练后将上下文表示用于监督 ASR。

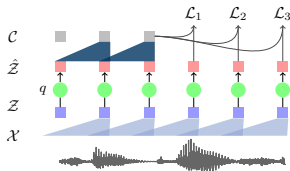
以 $t = 3$ 、预测步长 $k = 1$ 为例，真实目标为 z_4 。

时间位置	1	2	3	4	5	6
音频表示	z_1	z_2	z_3	z_4	z_5	z_6
构造 c_3 的局部上下文	可用	可用	可用			
正例目标				z_4		
示意负例	z_1					z_6

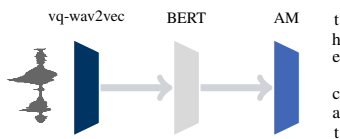
① 根据 c_3 计算对未来一步的预测表示 $h_1(c_3)$ 。

② 与真实 z_4 的匹配得分提高，与采样负例的得分降低。

负例不一定对应不同音素。原方法使用正负配对的二分类对比目标。



(a) vq-wav2vec

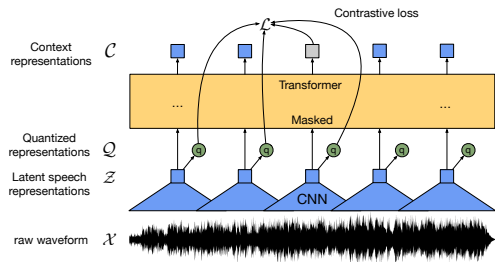


(b) Discretized speech training pipeline

- 1 **训练量化器**: 将连续声学表示映射到码本, 用未来预测学习语音单元。
- 2 **训练 BERT**: 把音频转换为离散单元序列, 遮盖部分单元并预测它们。
- 3 **训练识别器**: 利用 BERT 表示和人工转写, 学习 ASR 任务。

vq-wav2vec (2019 / ICLR 2020): **量化与上下文预训练分阶段进行。**

离散单元是码本编号及对应向量, 尚不等于文字或人工音素。



- 1 CNN 输出连续表示 z_t 。
- 2 掩码后的连续输入经过 Transformer, 得到 c_t 。
- 3 未遮盖的目标支路量化 z_t , 得到 q_t 。
- 4 用 c_t 从候选集合 Q_t 中识别 q_t 。

单个掩码位置 t 的核心对比损失:

$$\ell_t = -\log \frac{\exp(\text{sim}(c_t, q_t)/\kappa)}{\sum_{\tilde{q} \in Q_t} \exp(\text{sim}(c_t, \tilde{q})/\kappa)}$$

Q_t 包含正例与负例, sim 为余弦相似度, κ 为温度。

分子为正例的指数得分, 分母汇总所有候选; 提高正例概率可降低损失。

沿用六个时间位置，遮盖位置 2–4；本页重点预测 $t = 3$ 。

时间位置	1	2	3	4	5	6
CNN 输出	z_1	z_2	z_3	z_4	z_5	z_6
Transformer 输入	z_1	MASK	MASK	MASK	z_5	z_6
量化单元编号 (示意)	9	7	12	4	7	3
量化目标支路	q_1	q_2	q_3	q_4	q_5	q_6

- ① Transformer 由掩码后的输入计算位置 3 的上下文表示 c_3 。
- ② 从未遮盖的 z_3 得到正例 q_3 ，从其他掩码位置采样负例。
- ③ 本例候选为 $Q_3 = \{q_3, q_2, q_4\}$ ，对 c_3 与各候选计算得分。

wav2vec 预测未来；2.0 利用上下文识别被遮盖位置的目标。

沿用 c_3 与候选 $Q_3 = \{q_3, q_2, q_4\}$ 。得分 $s = \text{sim}(c_3, q)/\kappa$ 为教学设定。

候选	角色	得分 s	e^s	归一化概率
q_3	正例, 来自 z_3	2	7.389	0.665
q_2	负例, 来自 z_2	1	2.718	0.245
q_4	负例, 来自 z_4	0	1.000	0.090

$$p(q_3 | c_3, Q_3) = \frac{e^2}{e^2 + e^1 + e^0} \approx 0.665, \quad \ell_3 = -\ln(0.665) \approx 0.408$$

- 优化使正例相对于负例的得分提高, 正例概率增大。
- 若正例概率升至 0.9, 损失降为 $-\ln(0.9) \approx 0.105$ 。

概率在本次候选之间归一化, 尚不是某个文字 token 的识别概率。

$$\mathcal{L} = \mathcal{L}_m + \alpha \mathcal{L}_d, \quad \mathcal{L}_d = \frac{1}{GV} \sum_{g=1}^G \sum_{v=1}^V \bar{p}_{g,v} \log \bar{p}_{g,v}$$

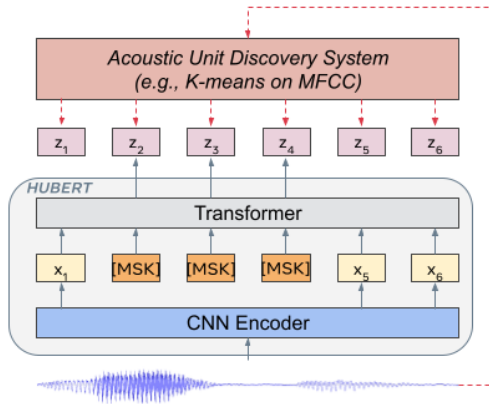
$\mathcal{L}_m$ 聚合掩码位置的对比损失， α 控制多样性项的权重。

G 为码本组数， V 为每组条目数， $\bar{p}_{g,v}$ 为批次平均选择概率。

单组、三条目的简化示例

	条目 1	条目 2	条目 3
使用集中在单一条目	1	0	0
使用更均衡	1/3	1/3	1/3

- 对比项要求区分正负例，多样性项鼓励充分使用码本，抑制使用塌缩。
- 预训练学习语音表示；ASR 适配再加入文字输出层，用转写训练。



- ① 对未遮挡的声学特征进行离线聚类。
- ② 为各时间位置分配聚类标签。
- ③ 对部分输入施加掩码，根据上下文预测聚类标签。

Hidden Unit 表示聚类获得的隐单元类别，与人工音素标签不同。

时间位置	1	2	3	4	5	6
聚类目标	7	7	12	12	4	9
预测网络输入	h_1	MASK	MASK	MASK	h_5	h_6
需要预测的目标		7	12	12		

- 目标标签由未遮挡特征生成，预测网络仅接收掩码后的输入。
- 聚类编号用于标识类别，其数值大小不具有语言含义。
- 预测同时利用局部声学信息与较长范围的上下文。

阶段	聚类特征	聚类结果的用途
第一轮	MFCC 等较简单特征	产生初始目标，训练编码器
后续轮次	已训练模型的中间层表示	产生新目标，再训练模型

迭代聚类的依据

模型隐藏表示可能比初始特征更充分地反映语音结构，从而改善后续聚类目标。

- ① 保留预训练的 CNN 与 Transformer 语音编码器。
- ② 移除预训练任务的输出模块，添加**文字 token + blank**的分类层。
- ③ 利用语音与转写配对数据，通过 CTC 目标进行微调。

预训练与 ASR 微调的输出空间

wav2vec 2.0: 量化目标的对比学习。 HuBERT: 聚类类别预测。

ASR 微调: 两者均可接文字分类层, 预测文字 token 与 blank。

facebook/wav2vec2-base-960h: 经 CTC 微调的英语识别模型。

```
import torch
from transformers import Wav2Vec2Processor, Wav2Vec2ForCTC
import soundfile as sf

model_id = "facebook/wav2vec2-base-960h"
processor = Wav2Vec2Processor.from_pretrained(model_id)
model = Wav2Vec2ForCTC.from_pretrained(model_id)
model.eval()
speech, sr = sf.read("audio.wav", dtype="float32")
assert sr == 16000 and speech.ndim == 1, "Need 16 kHz mono audio"
inputs = processor(speech, sampling_rate=sr,
                  return_tensors="pt")

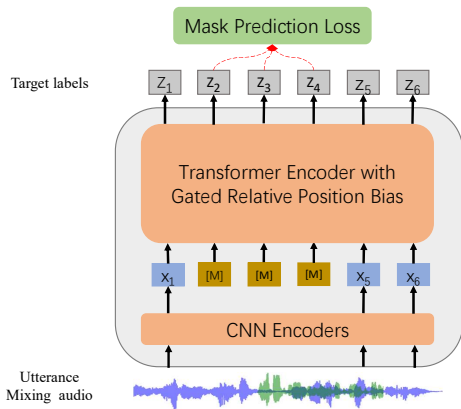
with torch.no_grad():
    logits = model(**inputs).logits
pred_ids = torch.argmax(logits, dim=-1)
print(processor.batch_decode(pred_ids)[0])
```

输入需为实际 16 kHz 单声道音频；填写采样率不会自动重采样。

- 编码器不同层可能保留不同程度的局部声学 and 上下文信息。
- **冻结特征提取**：固定编码器参数，使用隐藏表示训练下游模型。
- **微调**：下游任务的梯度进一步更新语音编码器。

表示分析：探针任务 (Probing)

冻结编码器，用某一层表示训练简单分类器，观察能否预测音素或说话人等标签。



- 对部分输入加入噪声或其他语音。
- 根据扰动后的输入，预测原始语音的目标单元。
- 预训练表示可迁移至说话人识别等语音任务。

预训练目标影响表示中保留的信息。

模型	训练目标	目标生成方式
wav2vec 2.0	掩码位置的对比学习	学习量化器，区分正确目标与干扰项
HuBERT	掩码位置的类别预测	离线聚类，可迭代更新目标
WavLM	扰动输入后的掩码预测	聚类目标配合噪声、混合语音

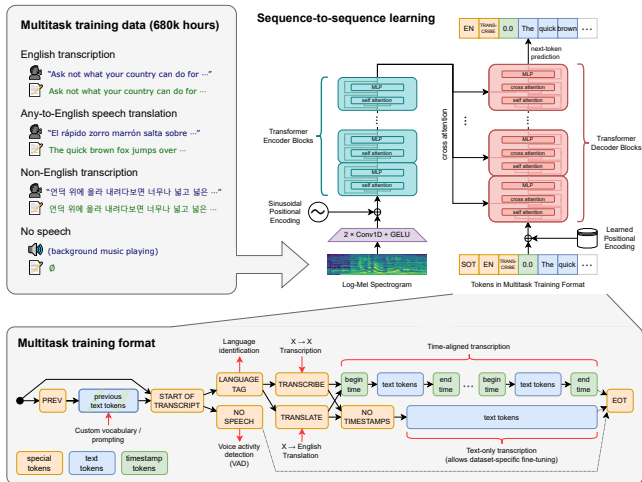
共同点：从音频中学习可迁移表示。

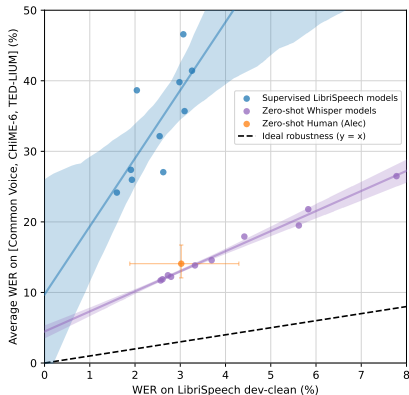
ASR 仍需要文字监督或其他语音与文字对齐信号。

- 自监督预训练能够学习语音表示，文字解码通常仍需监督适配。
- 单一数据集上的低 WER，未必对应不同口音、设备和场景下的识别精度。
- Whisper 在多来源语音与转写数据上联合训练编码器和解码器。

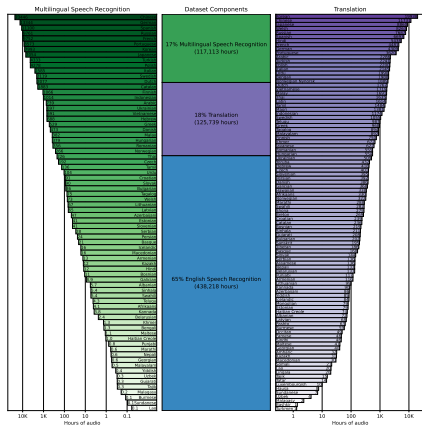
研究目标

在目标测试集上无需专门微调，评估跨领域、多语言与多任务泛化能力。





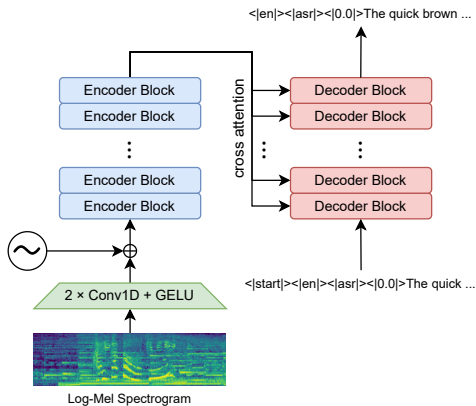
- 横轴：LibriSpeech dev-clean WER。
- 纵轴：其他测试集上的平均 WER。
- 相近的域内性能，可对应不同的跨域性能。



- 原论文训练数据共约 680,000 小时。
- 数据包括英语转写、其他语言转写及非英语到英语的翻译。
- 对比数据规模：SpeechStew 5,140 小时、GigaSpeech 10,000 小时、People's Speech 30,000 小时。

- 1 **配对数据收集**: 获取互联网音频及其对应转写。
- 2 **质量过滤**: 识别机器生成转写, 检查音频与文本的语言匹配, 并进行模糊去重。
- 3 **来源检查**: 根据初始模型错误率与数据源规模, 人工检查低质量来源。
- 4 **音频分段**: 构造 30 秒音频片段及对应文字序列。
- 5 **任务格式化**: 加入语言、任务、时间戳和序列边界标记。

音频来源的多样性与转写质量分别影响泛化能力和监督可靠性。



- 原论文前端：80 维对数 Mel 频谱与两层卷积。
- Transformer 编码器生成声学表示。
- 解码器以交叉注意力读取语音，逐 token 生成文字。
- 字节级 BPE 构造文字词表。

模型	参数量	英语专用版本	多语言版本
tiny	39M	tiny.en	tiny
base	74M	base.en	base
small	244M	small.en	small
medium	769M	medium.en	medium
large	1550M	无	large 系列
turbo	809M	无	large-v3 的优化变体

原论文覆盖 tiny 至 large。后续发布包含 large-v2、large-v3 与 turbo；turbo 面向转写加速，未针对翻译任务训练。

序列组成

含义

◁startoftranscript▷

标识预测序列起点

语言 token

指定或预测音频语言

◁transcribe▷ / ◁translate▷

指定转写或翻译任务

时间戳 token / ◁notimestamps▷

指定时间戳输出格式

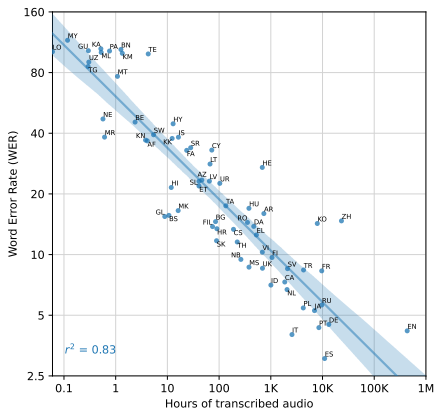
文字 token 与结束标记

生成目标文本并结束序列

无语音片段可预测 ◁nospeech▷。同一解码器通过特殊标记实现任务与语言条件化。

数据集	wav2vec 2.0 Large	Whisper Large V2	相对降幅 (%)
LibriSpeech Clean	2.7	2.7	0.0
Artie	24.5	6.2	74.7
Common Voice	29.9	9.0	69.9
Fleurs En	14.6	4.4	69.9
Tedlium	10.5	4.0	61.9
CHiME6	65.8	25.5	61.2

数值为 WER (%), 应用论文的文本规范化。wav2vec 2.0 不使用外部 LM; Whisper 不进行目标测试集微调。原表完整跨域集合的平均 WER 为 29.3 与 12.8, 平均相对降幅为 55.2%。

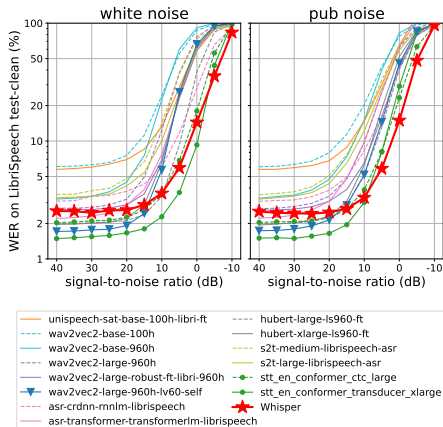


- FLEURS 评测用于分析语言间性能差异。
- 训练数据量与识别错误率呈明显相关性。
- 文字系统、语言差异及标注质量也影响结果。

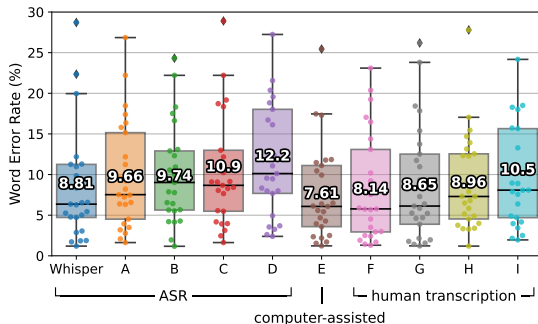
非英语到英语	高资源	中资源	低资源	全部
XMEF-X	34.2	20.2	5.9	14.7
XLS-R (2B)	36.1	27.7	15.1	22.1
mSLAM-CTC (2B)	37.8	29.6	18.5	24.8
Maestro	38.2	31.3	18.4	25.2
Zero-shot Whisper	36.2	32.6	25.2	29.1

- 指标为 BLEU，数值越高越好，与 ASR 的 WER 不同。
- 原论文在 CoVoST2 的总体、中资源和低资源分组中获得较好结果。
- 高资源分组仍低于部分直接监督训练的模型。

Whisper: 加性噪声鲁棒性



- 在 LibriSpeech test-clean 上加入白噪声或酒吧背景噪声。
- 随信噪比降低，比较各模型 WER 的变化。
- 该实验反映特定噪声条件下的鲁棒性，不代表所有噪声与录音环境。



- 从 Kincaid46 选择 25 段不同录音条件下的音频。
- 对照专业人工转录与计算机辅助转录服务。
- 论文结果接近该评测中的人工转录精度，结论受样本与评测条件限制。

- **弱监督数据扩展**: 通过数据收集与过滤扩大语音-文本监督规模。
- **联合模型训练**: 编码器与解码器共同学习语音到文字的映射。
- **多任务条件化**: 采用特殊 token 指定语言、任务与输出格式。
- **跨域评测**: 在多个目标数据集上检验无需专门微调的识别能力。

Whisper 的方法重点是训练数据与任务组织, 模型采用标准 encoder-decoder Transformer。

- **应用**: 多语言转写、语言识别及非英语到英语的语音翻译。
- **局限**: 低资源语言、长音频重复、时间戳稳定性及错误文本生成。
- **适配方向**: 领域微调、解码改进、轻量化与部署优化。
- **LLM-ASR 接口**: 复用语音编码器, 将连续表示连接至文本 LLM。

官方仓库提供推理代码与模型权重, 采用 MIT 许可。

比较维度	GMM-HMM / DNN-HMM	CTC / AED / RNN-T	LLM-ASR
语言信息	独立语言模型参与解码	输出历史建模或外部 LM 融合	复用文本预训练 LLM 生 成转写
场景信息	词表、LM 适配与解码约 束	热词偏置、上下文模块 等	文字上下文与语音共同 条件化
任务接口	各组件与任务分别配置	专用转写或多任务训练	可通过指令组织转写及 其他任务

LLM-ASR 的研究动机：复用文本知识，并建立灵活的上下文接口。

端到端系统也可使用预训练、语言模型与热词。LLM-ASR 仍可端到端优化。

- **可复用的资源**：文本 LLM 从大量文字中学习词汇、表达和长距离依赖。
- **识别中的作用**：声学候选接近时，语言信息可以辅助选择符合语境的文字。
- **迁移方式**：将语音表示接入已有生成模型，再学习语音到转写的关联。

教学示例：专有名词的书写

领域说明为“语音自监督学习”，音频中出现模型名称的读音。

领域信息有助于确定“HuBERT”的专名书写，仍需声音与相关训练支持。

相比从头训练文字生成部分，可复用已有文本参数与知识。

独立语言模型也能利用文本；迁移收益需通过实验检验。

文字条件 可帮助识别的内容

参会名单 同音姓名、机构名及缩写的书写

历史对话 当前片段中的简称与此前提及的对象

领域说明 术语及符合当前场景的候选表达

- LLM 可将这些文字与语音表示一起用于转写生成。
- 在支持该接口的模型中，可更新场景信息而不重新训练模型权重。
- 上下文训练与解码约束影响实际效果，错误提示可能引入偏置。

传统热词与上下文方法同样有效；此处关注**自然语言上下文的统一处理**。

组件复用：SLAM-ASR

复用语音编码器与文本 LLM，在连接阶段固定两者，只训练 projector。

收益：减少该阶段更新的参数，便于组合与比较已有组件。

任务扩展：Qwen2-Audio

通过音频表示与任务指令，组织转写、翻译及音频问答等输出。

收益：在一个接口中复用音频与文字处理能力。

- 冻结参数仍可能需要梯度传播，LLM 的计算与存储开销仍然存在。
- 多任务能力需要相应训练；Whisper 等端到端模型也支持多任务。

关键词召回率 (%)，越高越好。对话历史为上下文，分强依赖与弱依赖子集。

模型	解码设置	强依赖	弱依赖
Transducer 端到端基线	Context FST 热词偏置	72.77	84.58
Seed-ASR (CN)	Beam Search, 无上下文	65.45	89.33
Seed-ASR (CN)	Joint Beam Search, 有上下文	80.63	93.89

- 强依赖子集：Seed-ASR 加入上下文及配套解码，提升 **15.18 个百分点**。
- 无上下文时，该子集的 Seed-ASR 低于已有热词基线。

收益与上下文、训练和解码共同相关；这是作者构建的测试集，指标为召回率。

LibriSpeech WER (%), 越低越好。选取同一报告中的比较结果。

系统	test-clean	test-other
Conformer-large	2.1	4.3
Conformer-large + 域内 LM	1.9	3.9
Zipformer-large + 域内 LM	1.9	3.9
SLAM-ASR	1.9	3.8

- 该配置达到与强端到端基线相近的精度，test-other 有小幅改善。
- 域内 LM 本身也带来收益；LLM-ASR 的优势不能概括为普遍更低 WER。
- 训练、预训练与计算预算有差异，表中结果未隔离架构的单独贡献。

组件	预训练数据	可以复用的能力
语音编码器	音频，或音频+文字	声学特征与语音上下文表示
文本 LLM	大规模文本	词汇、语言规律与文本生成
连接模块	通常需要语音+文字	将语音表示映射至 LLM 输入空间

跨模态连接需要匹配输入维度，并学习语音与文字的关联。

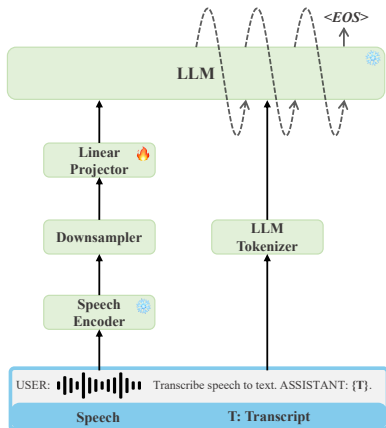
文本生成示例：token 划分仅作参考

“今天” “学习” “语音” “识别” EOS

$$P(y \mid c) = \prod_{u=1}^U P(y_u \mid y_{<u}, c)$$

- y_u 为当前预测的 token, $y_{<u}$ 为此前的 token 序列。
- c 是生成条件，在文本模型中可以是提示或上下文。
- LLM-ASR 以**语音表示作为生成条件**，预测转写序列。

表示	例子	LLM 输入方式
原始波形	高采样率的时域信号	需要声学前端处理
连续语音表示	$h_1, h_2, \dots, h_T$	连接模块映射至 LLM 输入空间
离散语音单元	7, 7, 12, 4, ...	语音 token 的嵌入查找
文字 token	“语音”、“识别”	文本 token 的嵌入查找
连续语音表示经投影后可作为输入嵌入， 无需经过中间文字转写 。		



- 1 语音编码器提取连续表示。
- 2 降采样缩短语音序列。
- 3 Projector 映射至 LLM 输入嵌入维度。
- 4 LLM 结合语音与提示生成文字。

连接训练阶段冻结编码器与 LLM，只训练 projector。

教学设定：10 秒音频，编码器每秒输出约 50 个向量。

阶段	张量尺寸（忽略 batch）	操作含义
编码器输出	约 500×1024	帧级声学表示序列
每两帧拼接	约 250×2048	时间长度减半，保留相邻帧信息
Projector 输出	约 250×4096	匹配假设的 LLM 输入维度

语音输入长度与文字输出长度独立，**无需逐位置一一对应**。

输入部分	作用
提示：“请转写这段语音”	指定任务，经过文本 tokenizer 与 embedding
语音向量： $a_1, a_2, \dots, a_M$	提供声学条件，由 projector 产生
输出起始标记	标识文字生成的起始位置
已生成文字： $y_1, \dots, y_{u-1}$	帮助预测下一个文字 token

LLM 预测 $P(y_u \mid a_{1:M}, p, y_{<u})$ 。

具体提示顺序与特殊标记由模型模板决定。

预测位置	可访问的条件	受因果掩码约束的信息
第一个输出 token	已提供的语音与提示	整段参考答案
第二个输出 token	语音、提示、第一个 token	后面的输出 token
第 u 个输出 token	语音、提示、此前文字	y_u 及后续参考 token

输出文字保持因果生成。

语音前缀内部可以采用不同注意力设计，**离线模型可使用完整音频上下文。**

阶段	预测下一个 token 时使用	目的
训练	语音、提示、 参考答案的此前 token	最大化目标 token 的条件概率
推理	语音、提示、 模型已生成的 token	逐步生成整段转写

$$\mathcal{L} = - \sum_{u=1}^U \log P(y_u \mid a_{1:M}, p, y_{<u})$$

常见实现仅在目标转写位置计算损失，输入音频向量不作为预测目标。

对齐	解决的问题	例子
时间对齐	语音时间区间与文字单元的对应关系	CTC 路径、注意力、时间戳
模态对齐	语音表示与 LLM 输入空间的适配	利用配对数据训练 projector

- 模态对齐无需约束语音向量与单个文字嵌入完全相等。
- 文字转写与逐词时间戳采用不同的评价标准。

策略	语音编码器	连接模块	LLM
只训练连接模块	冻结	更新	冻结
连接模块 + LoRA	通常冻结或按阶段更新	更新	基础权重冻结，LoRA更新
联合适配	按方案更新	更新	部分或全部更新

- 参数冻结可降低训练开销，并保留预训练能力。
- 连接阶段冻结的编码器，此前可能已经接受过监督微调。

$$Wx \mapsto Wx + BAx$$

- 原始权重 W 保持不变，仅学习低秩矩阵 A 与 B 。
- 低秩分解的中间维度 r 远小于原始矩阵维度。
- 用较少的可训练参数，适配语音输入和目标任务。

连接模块解决**输入接口**，LoRA 调整 **LLM 内部处理方式**。

方法	压缩机制	主要限制
帧拼接或卷积	按固定比例降采样	压缩过强可能丢失细节
平均池化	合并邻近帧	容易平滑短促声音
Q-Former	以查询向量汇聚局部语音表示	查询数量与窗口大小影响信息容量
CTC 引导压缩	按预测标签删帧或合并帧	依赖 CTC 预测质量

压缩序列长度和**匹配 embedding 维度**是不同操作。

CTC 帧标签示意	blank, a, a, a, blank, b, blank, b, b, blank
帧平均策略示例	连续预测相同标签的帧，合并对应隐藏表示
删帧策略示例	删除预测为 blank 的帧，保留其他表示

- 压缩结果为较短的**连续声学表示序列**，作为 LLM 输入。
- 由 blank 分隔的两个 b 段保留各自边界。
- 文本纠错路线则仅以 CTC 解码后的文字作为 LLM 输入。

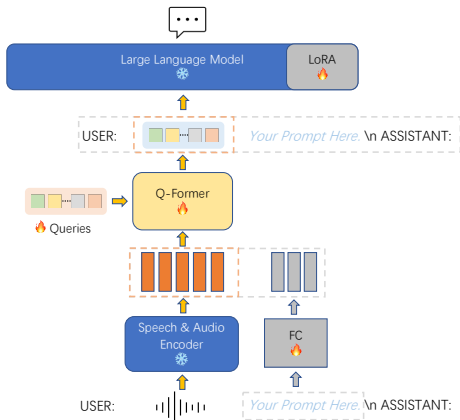
固定 Vicuna-7B-v1.5 和连接方案，仅改变语音编码器。

语音编码器	test-clean WER%	test-other WER%
Whisper-large	2.58	6.47
HuBERT Large	4.53	8.74
HuBERT Large + LS-960 ASR 微调	2.30	4.53
HuBERT X-Large + LS-960 ASR 微调	1.94	3.81

语音编码器及其 ASR 微调状态显著影响识别性能。



- 图中纵轴是**训练时下一个 token 的预测准确率**。
- 不同编码器呈现不同训练动态，该指标与测试集 WER 应分别报告。
- 训练拟合程度、测试集识别精度与跨领域泛化需分别评估。



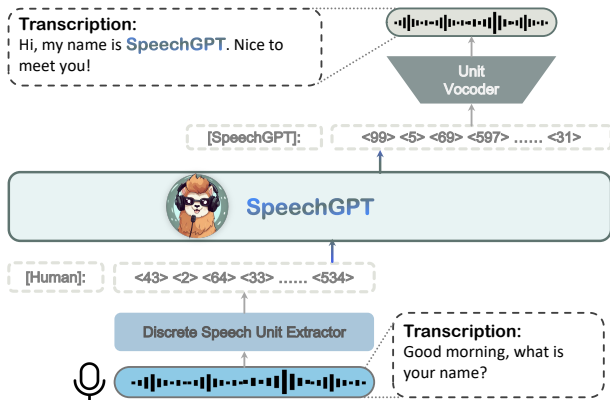
- Whisper 编码器提供语音相关表示。
- BEATs 编码器补充一般音频信息。
- 窗口 Q-Former 连接 Vicuna。
- 训练 Q-Former 与 LLM 的 LoRA。

- ① 按时间顺序把语音表示划分成多个短窗口。
- ② 各窗口通过少量**可学习查询向量**汇聚语音特征。
- ③ 查询向量通过注意力汇聚窗口特征，输出按时间顺序组成语音前缀。

教学设定：每个窗口 10 帧，使用 2 个查询

30 帧被分成 3 个窗口，输出共 6 个向量。

保留窗口的时间顺序，输出向量与文字单元无需一一对应。



- 采用 HuBERT 相关离散单元表示语音，并扩展 token 词表。
- 在 LLM 中联合建模语音单元与文字序列。
- 模型还可以生成语音单元，ASR 是其中一项能力。

帧级语音单元示意	7, 7, 7, 12, 12, 4, 4, 7
去除相邻重复后的示意	7, 12, 4, 7
扩展词表	原有文字 token + 新增语音单元 ID
进入 LLM	根据 ID 查找对应输入嵌入

- 同一个编号在不同时间可以再次出现。
- 语音单元 ID 与文字标签属于不同词表，编号不具有直接文字含义。
- 去重能缩短序列，也会改变时长信息的表达。

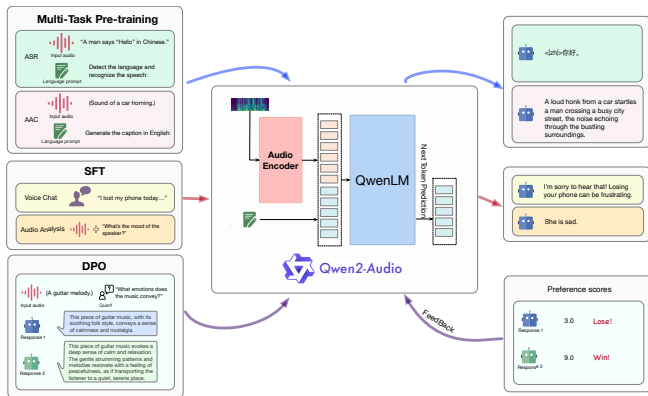
连续向量路线

接口 投影到 LLM 输入空间
信息表示 可保留细粒度变化
训练重点 编码器与连接模块适配
典型代表 SLAM-ASR、SALMONN

离散 token 路线

量化后查 embedding
受离散词表与量化限制
token 化与跨模态序列适配
SpeechGPT

两类接口均需语音与文字的关联训练，识别精度取决于数据与训练配置。



- Whisper-large-v3 初始化音频编码器。
- 语音表示连接到 Qwen 系列 LLM。
- 支持转写、音频分析和语音指令响应等任务。

输入语音：“会议改到周六上午十点。”

指令	目标输出
----	------

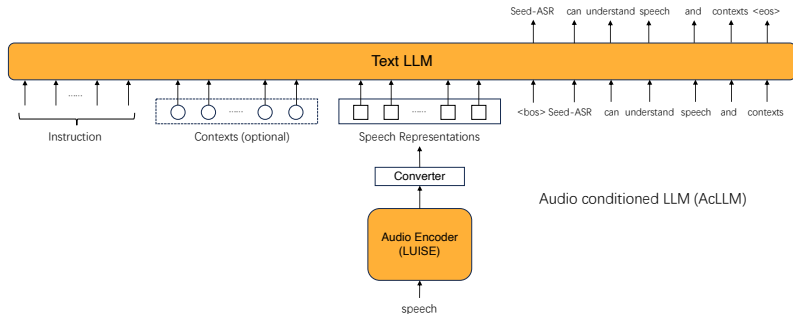
请逐字转写	会议改到周六上午十点。
-------	-------------

提取会议时间	周六上午十点。
--------	---------

翻译成英文	The meeting has been moved to 10 a.m. on Saturday.
-------	--

ASR 评测应使用转写指令与转写参考答案。

多任务能力与逐字转写能力需分别评价。

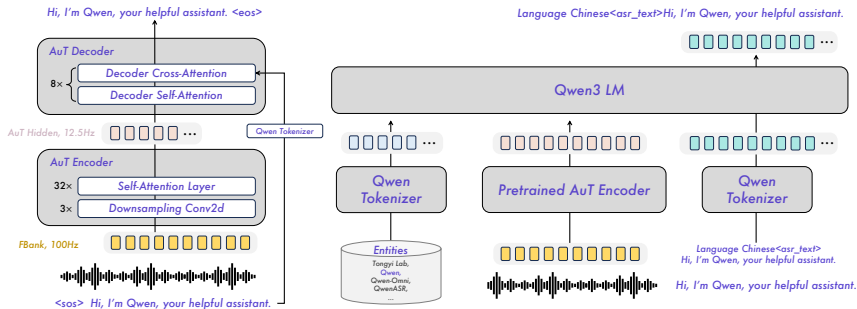


- 输入同时包含语音表示、转写任务提示和可选上下文。
- 上下文可以提供对话历史或领域相关信息。
- 模型在配对与上下文训练中学习如何使用这些条件。

教学场景：同音人名“李明”与“黎铭”。

上下文	可能的影响
本次会议参会人是黎铭	提高与场景一致的人名写法的概率
错误名单写成李明	可能提高错误人名的输出概率
包含大量无关姓名	增加歧义，收益未必稳定

- 专有名词识别需要声学证据与上下文共同支持。
- 对纯同音不同字，声音可能不足以唯一确定正确写法。



- 左侧：先通过 AED 训练 AuT 语音编码器。
- 右侧：语音表示与文本上下文接入 Qwen3 LM。
- 编码器输出约 12.5 个向量/秒，降低长音频输入开销。

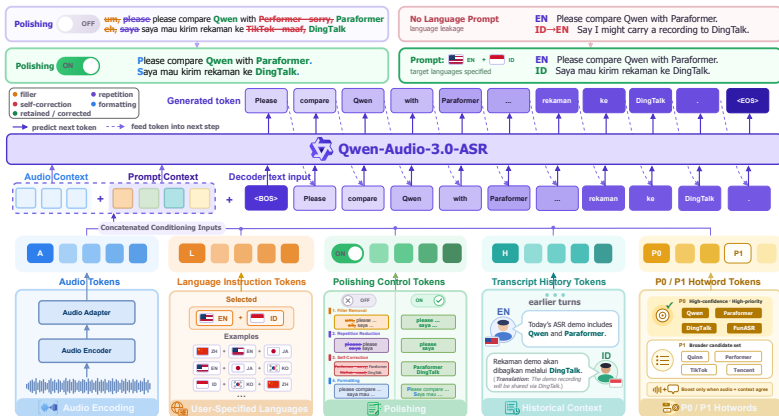
阶段	主要作用
AuT 预训练	用大规模带标签或伪标签数据学习语音表示
Omni 预训练	获得多模态理解的基础模型
ASR 监督微调	适配固定转写格式、上下文与复杂音频
ASR 强化学习	进一步优化转写稳定性等目标

最终模型面向 ASR 专用任务，使用上下文辅助转写，不任意执行自然语言任务指令。

数据集与指标 (%)	Whisper-large-v3	Qwen3-ASR-0.6B	Qwen3-ASR-1.7B
LibriSpeech clean, WER	1.51	2.11	1.63
LibriSpeech other, WER	3.97	4.55	3.38
AISHELL-2, CER	5.06	3.15	2.71
WenetSpeech meeting, CER	19.11	6.88	5.88

- 模型间的相对性能随测试数据集变化。
- 不同模型的训练数据不同，结果不单独反映 LLM 组件的贡献。

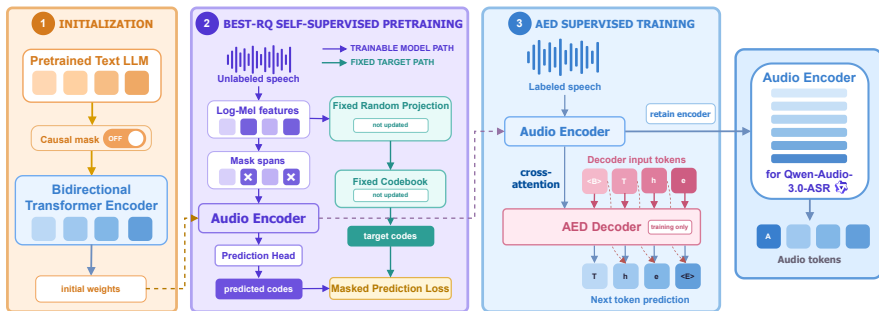
Qwen-Audio-3.0-ASR: 条件解码



语音向量与语言条件、历史转写、热词、润色开关共同控制输出。

组件	分工
SenseVoice 语音编码器	从声音中提取表示
两层 Transformer adapter	连接音频表示与 LLM
辅助 CTC 分支	产生初步转写，辅助热词与长音频上下文处理
Qwen MoE 解码器	根据音频及文本条件生成最终输出

MoE: 每个 token 只激活部分专家，**总参数规模与每次计算量不同。**



- BEST-RQ 自监督阶段：对语音表示施加掩码，预测固定随机量化目标。
- AED 阶段：用文字监督强化转写相关表示。
- 保留语音编码器，接入后续 Audio LLM 训练。

Qwen3-ASR

报告版本 2026 年 1 月, v2
语音前端 AuT, AED 预训练
输出控制 专用转写格式与上下文
解码设计 Qwen3 LM + 连接模块

Qwen-Audio-3.0-ASR

2026 年 9 月, v2
SenseVoice, SSL + AED 预训练
指令控制语言、热词及润色等
Qwen MoE + adapter + 辅助 CTC

两者均采用 LLM-ASR 架构, **任务范围与组件配置存在差异。**

问题	常见处理方式	代价或困难
完整音频依赖	分块或限制编码器未来上下文	减少未来证据，可能影响精度
长语音前缀	降采样、压缩、分段识别	可能丢失细节或上下文
分段间实体不一致	使用历史文字与热词	错误历史也可能传播
自回归解码开销	缓存与高效推理实现	仍需考虑模型规模与资源

流式能力取决于编码器上下文、分块训练与解码实现。

教学例子：录音中说“会议**不**在周五”。

输出	判断
----	----

会议不在周五	保留原话
--------	------

会议在周五	否定词删除，导致语义反转
-------	--------------

会议改到周六	插入录音未提供的信息
--------	------------

常见问题还包括漏字、重复、无语音时生成文字，以及过度润色。

转写忠实性以**音频内容与任务规范**为依据。

$$\text{WER} = \frac{S + D + I}{N} \times 100\%$$

参考	I want a ticket tomorrow
输出	I need a ticket
最小编辑对齐	want 被替换为 need, tomorrow 被删除

$S = 1$, $D = 1$, $I = 0$, $N = 5$, 所以 WER 为 40%。

中文常用 CER, 按字计算, 并明确标点、数字等规范化规则。

- **数据条件**：测试集一致性、训练领域及数据重叠情况。
- **转写规范**：大小写、标点、数字及逐字/整理标准。
- **解码条件**：热词、历史上下文及其他辅助信息。
- **效率条件**：硬件、并发、音频长度与流式等待时间。

模型之间的 WER 差异，**需结合训练数据与其他组件分析。**

- LLM 的预训练语料可能包含公开测试集的文字转写。
- 测试转写文本重叠可能改变语言先验，即使音频未参与训练。
- 应尽量使用独立、近期或明确保留的测试数据，并说明数据来源。

数据污染与结果解释

需核查测试集污染风险，并通过独立测试集判断实际泛化能力。

条件	检查的问题	观察指标
不提供上下文	基础识别是否稳定	WER/CER
提供正确热词	是否改善相关实体	实体召回与整体错误率
提供错误或无关热词	是否被条件误导	错误替换与无关插入
改变转写提示	是否仍执行同一任务	输出格式与错误率稳定性

固定测试音频，通过控制变量评估**上下文与指令变化的影响**。

已知条件：有预训练 HuBERT、文本 LLM、少量校园会议转写数据，以及参会名单。

- ① 需要在 HuBERT 与 LLM 之间增加什么？
- ② 训练时哪个序列作为预测目标？推理时能否提供参考答案？
- ③ 参会名单如何帮助识别？怎样检查它是否造成错误偏置？
- ④ 如果要求实时字幕，还需要考虑什么？

- ① 增加长度压缩与 projector，匹配 LLM 输入维度。
- ② 用参考文字的下一个 token 训练，推理从模型输出起始位置生成。
- ③ 名单作为上下文，比较空、正确、错误和无关名单四种条件。
- ④ 检查编码器未来上下文、分块策略、生成速度及端到端延迟。

有限数据条件下，可先冻结预训练组件并训练连接模块，再依据验证集结果调整适配策略。

- **语音预训练**为 ASR 与 LLM-ASR 提供可复用表示。
- **连接模块**处理语音与文本的输入接口及长度差异。
- **LLM-ASR** 根据语音条件，逐 token 生成目标转写。
- **方案选择**同时考虑配对数据、任务边界、上下文和计算资源。

架构分析的主要维度为**语音编码器、连接模块、解码器与训练数据**。

- Schneider et al. (2019), *wav2vec: Unsupervised Pre-training for Speech Recognition*. 1904.05862v4
- Baevski et al. (ICLR 2020), *vq-wav2vec: Self-Supervised Learning of Discrete Speech Representations*. 1910.05453v3
- Baevski et al. (2020), *wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations*.
arXiv:2006.11477
- Hsu et al. (2021), *HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units*.
arXiv:2106.07447
- Chen et al. (2022), *WavLM: Large-Scale Self-Supervised Pre-Training for Full Stack Speech Processing*.
arXiv:2110.13900
- Mohamed et al. (2022), *Self-Supervised Speech Representation Learning: A Review*.
arXiv:2205.10643

- Wu et al. (2023), *On decoder-only architecture for speech-to-text and large language model integration*.
arXiv:2307.03917
- Zhang et al. (2023), *SpeechGPT: Empowering Large Language Models with Intrinsic Cross-Modal Conversational Abilities*.
arXiv:2305.11000
- Tang et al. (2024), *SALMONN: Towards Generic Hearing Abilities for Large Language Models*.
arXiv:2310.13289
- Ma et al. (2024), *An Embarrassingly Simple Approach for LLM with Strong ASR Capacity (SLAM-ASR)*.
arXiv:2402.08846v1

- Chu et al. (2024), *Qwen2-Audio Technical Report*. 2407.10759
- Bai et al. (2024), *Seed-ASR: Understanding Diverse Speech and Contexts with LLM-based Speech Recognition*. 2407.04675
- Peng et al. (2025), *A Survey on Speech Large Language Models for Understanding*. 2410.18908v6
- Cui et al. (ACL 2025), *Recent Advances in Speech Language Models: A Survey*. 2410.03751
- Shi et al. (2026-01), *Qwen3-ASR Technical Report*. 2601.21337v2
- Bian et al. (2026-09), *Qwen-Audio-3.0-ASR Technical Report*. 2609.07549v2
- Tseng et al. (2025), *Evaluation of LLMs in Speech is Often Flawed: Test Set Contamination in Large Language Models for Speech Recognition*. 2505.22251

章节	本课参考的讲解思路
----	-----------

Lecture 15, Peter Bell	弱监督数据的来源与转写质量问题
------------------------	-----------------

Lecture 16, Hao Tang	自监督目标、微调与特征提取的区别
----------------------	------------------

Lecture 18, Peter Bell	语音前缀、序列压缩与指令接口
------------------------	----------------

模型结构图与实验数据的原始出处保存在源码与资料索引中。

上述课件已保存在课程目录的 `Edinburgh_ASR_2026` 中。