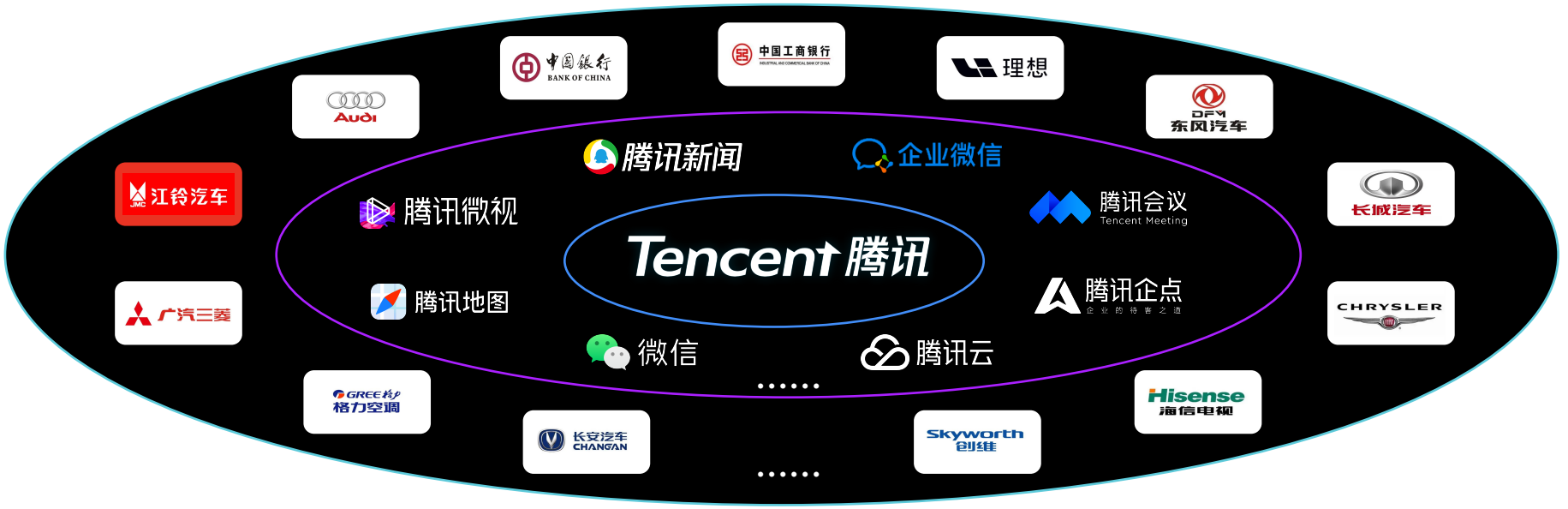
语音识别预训练技术的发展和应用

曹松军
songjuncao@tencent.com
腾讯CSIG云小微

主要内容

- 一、语音识别简介
- 二、语音预训练技术
- 三、我们的进展和应用

腾讯云小微-产品技术生态



能力沉淀

垂直行业应用	智能家居	智能汽车	智慧金融	智慧政务	智慧教育	泛互	...
人机交互	对话机器人	AI语音助手	...	人机协同	智能客服	数智人	AI同传	...
语音语义AI底层能力	对话输入与前端处理	ASR语音识别	NLP语义理解	TTS			

能力迁移

云小微：打造更自然、更智能的交互智能产品

腾讯云小微-语音助手应用

覆盖场景

IOT

应用行业

家居

40+ 高频技能智能家居服务场景

3.5亿+ 激活设备数

已与业内超 60% 品牌和机型合作

汽车

34家 主流汽车品牌

160款 量产车型落地

累计落地车辆总数超过 200万辆

合作客户

TCL · Skyworth 创维 · GREE 格力空调

Hisense 海信电视 · Haier · SHARP

优必选 UBTECH · ling 物灵 · (部分客户)

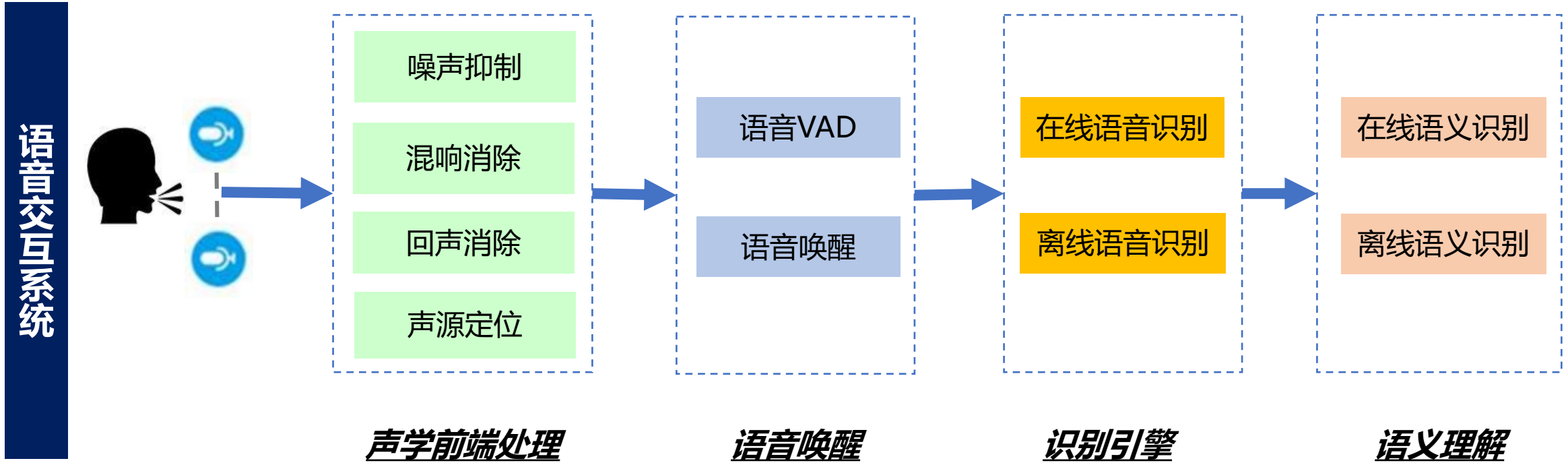
江铃汽车 · 广汽三菱 · 吉利汽车 GEELY AUTO

长安汽车 CHANGAN · MASERATI · Audi

上汽大众 SAIC VOLKSWAGEN · CHRYSLER · 理想

Ford · 长城汽车 · (部分客户)

语音助手中的ASR技术



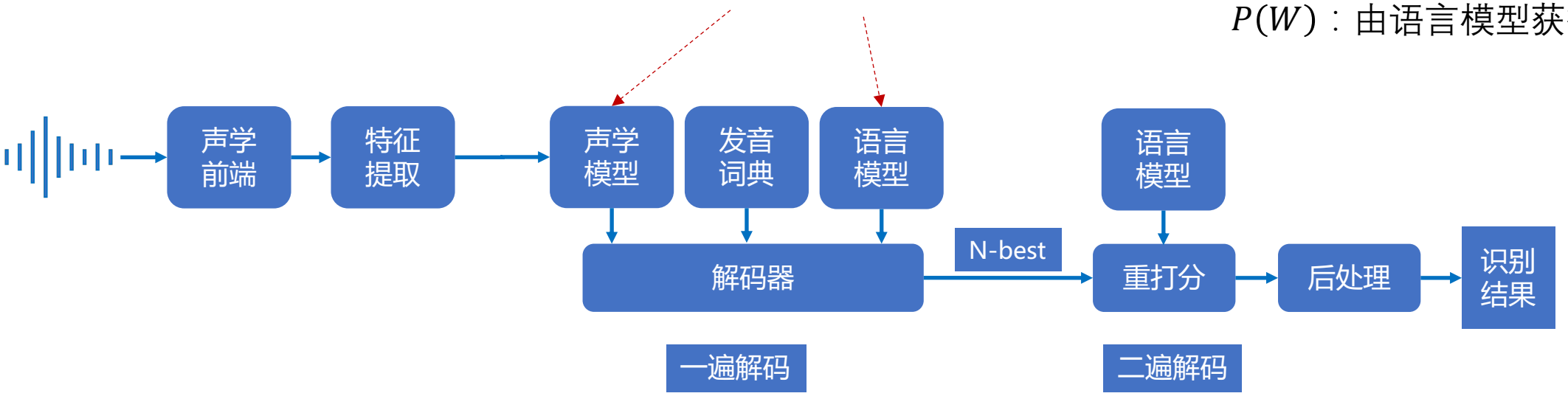
自动语音识别技术(automatic speech recognition)，可以简写为**ASR**，在语音交互过程中，ASR技术模块负责将用户的语音转化为文本，转化得到的文本会传给下游的NLP技术模块做语义理解，所以ASR模块的识别效果对整个语音交互的体验至关重要。

ASR-技术架构

系统架构

$$\widehat{W} = \arg\max P(W|O) = \arg\max(P(O|W)P(W))/(P(O))$$
$$\widehat{W} = \arg\max P(O|W)\,P(W)$$

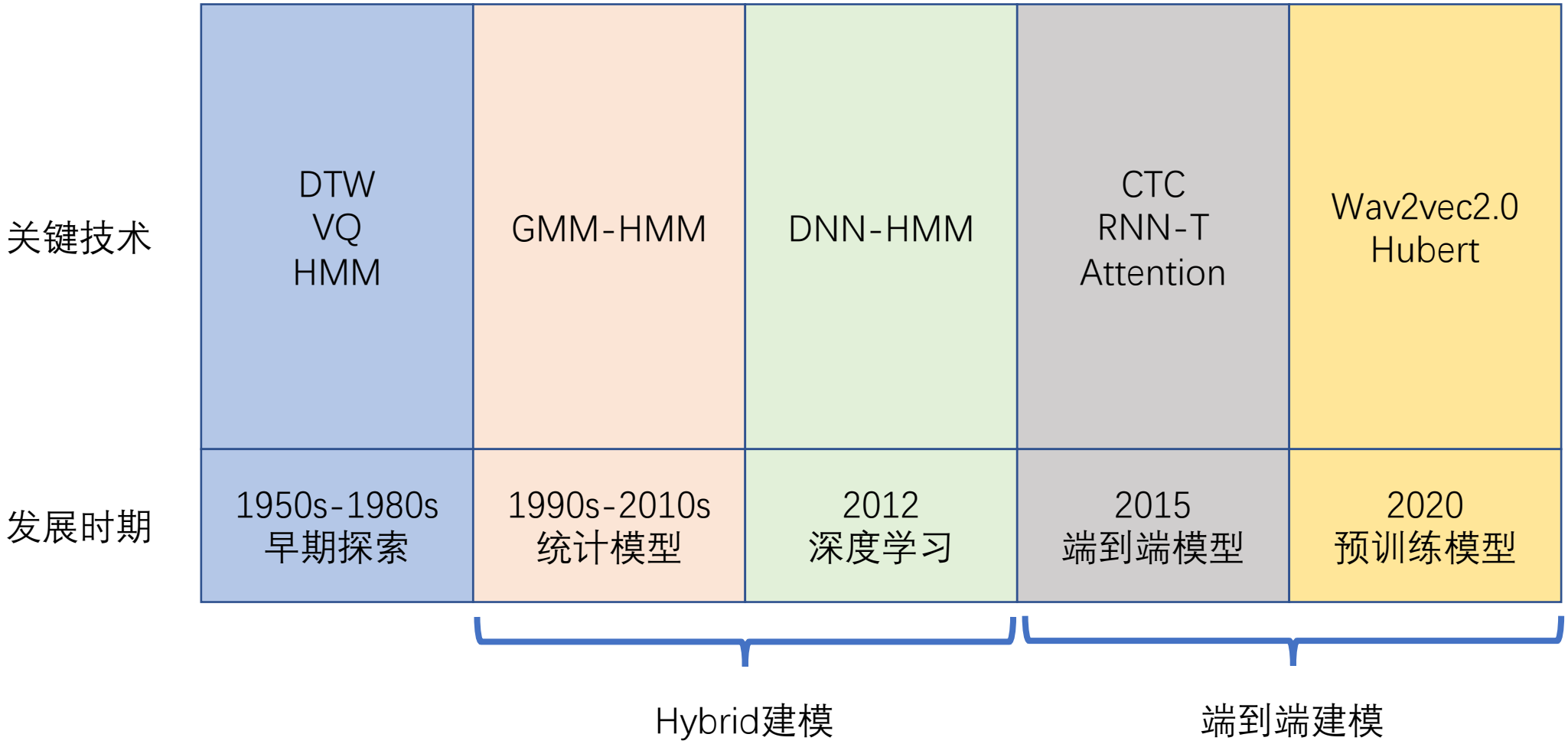
O: 观察值序列（语音特征）
$\widehat{W}$：最优词序列（解码结果）
$P(O|W)$：由声学模型获得
$P(W)$：由语言模型获得



评价指标

- ✓ 字错误率（Character Error Rate，CER）= 识别错误字数 / 测试集中总字数
- ✓ 句错误率（Sentence Error Rate，SER）= 识别错误的句子数 / 测试集中总句子数
- ✓ 句正确率（Sentence Accuracy）= 识别正确的句子数 / 测试集中总句子数 = 1 – SER

ASR-发展阶段


关键技术
DTW
VQ
HMM
GMM-HMM
DNN-HMM
CTC
RNN-T
Attention
Wav2vec2.0
Hubert
发展时期
1950s-1980s
早期探索
1990s-2010s
统计模型
2012
深度学习
2015
端到端模型
2020
预训练模型
Hybrid建模
端到端建模

ASR-hybrid建模

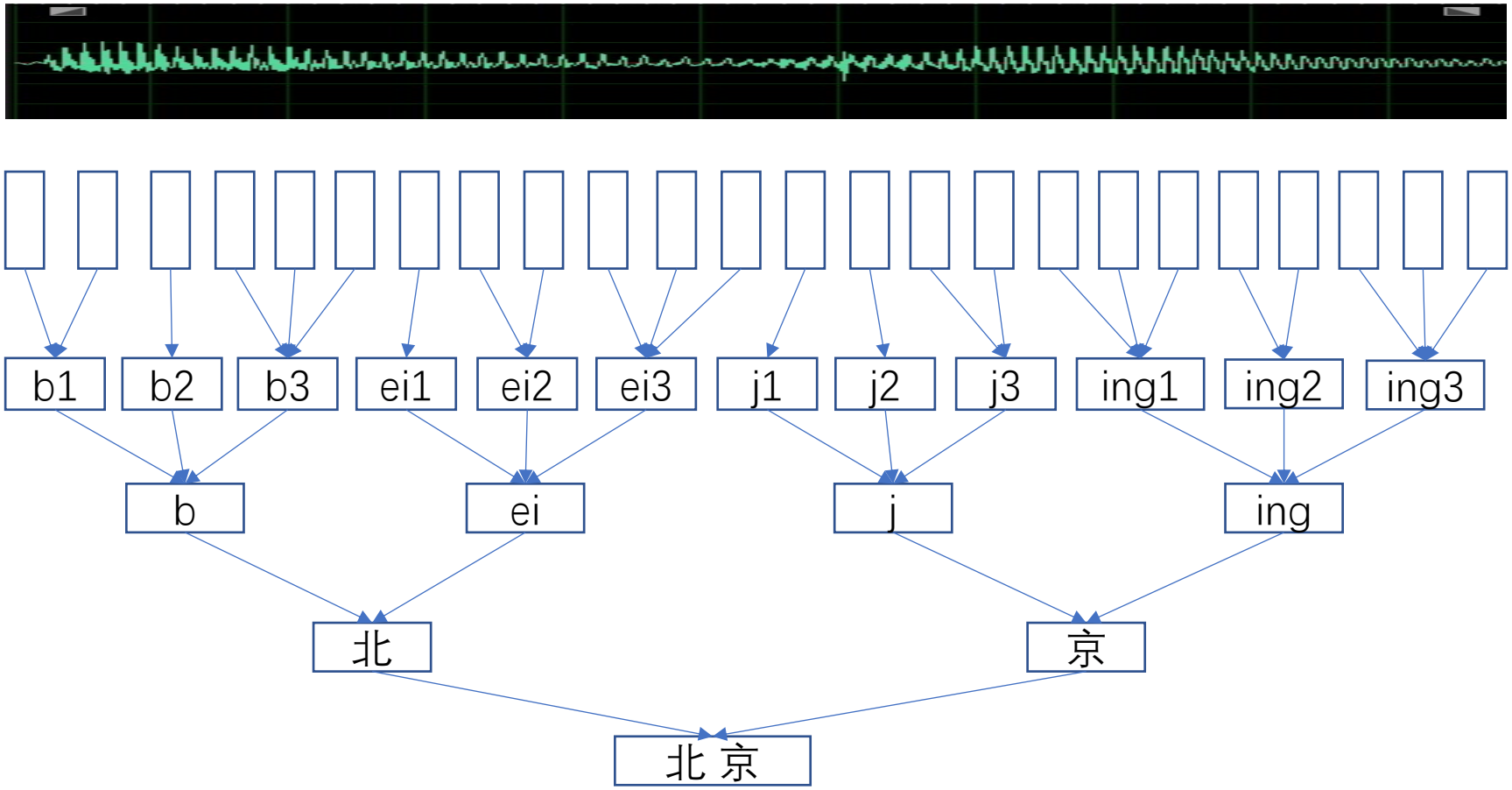
1.特征抽取
时域点->频谱特征

2.1声学模型-DNN
计算p(状态|特征)

2.2声学模型-HMM
状态->音素

3.发音词典
音素->词

4.语言模型
词->句子



建模到HMM的状态

ASR-hybrid建模

问题1：每一帧对应一个HMM状态，拆分科学吗？

- 语音过渡地带

问题2：训练流程复杂？

- 第一步，训练hmm-gmm
- 第二步，使用hmm-gmm来做对齐获取每一帧特征的label
- 第三步，训练帧到label的鉴别模型

问题3：独立性假设？

- 每一帧的输出只跟当前帧有关，跟上一帧的输出无关

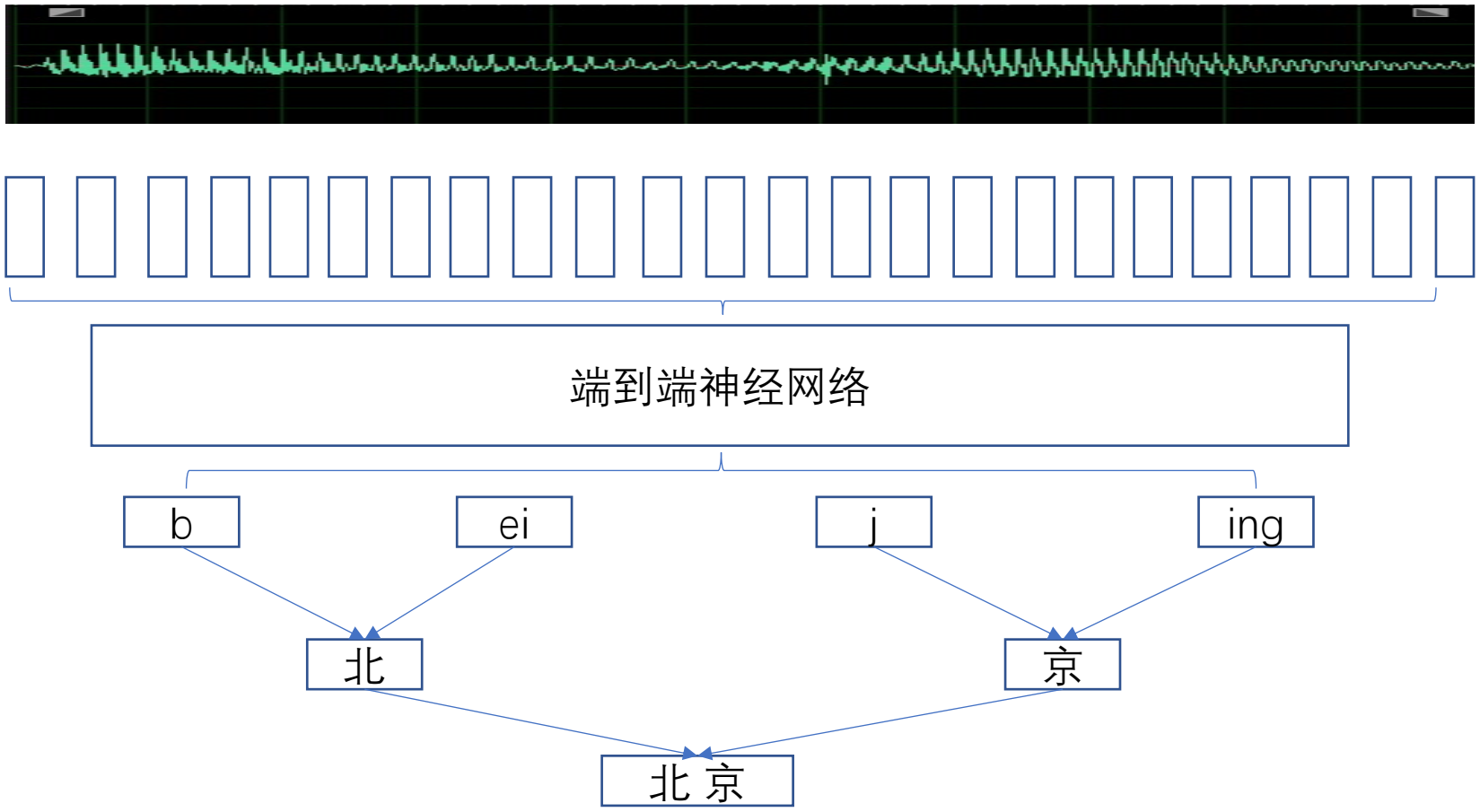
ASR-端到端建模

1.特征抽取
时域点->频谱特征

2.端到端建模
计算p(音素|特征)

3.发音词典
音素->词

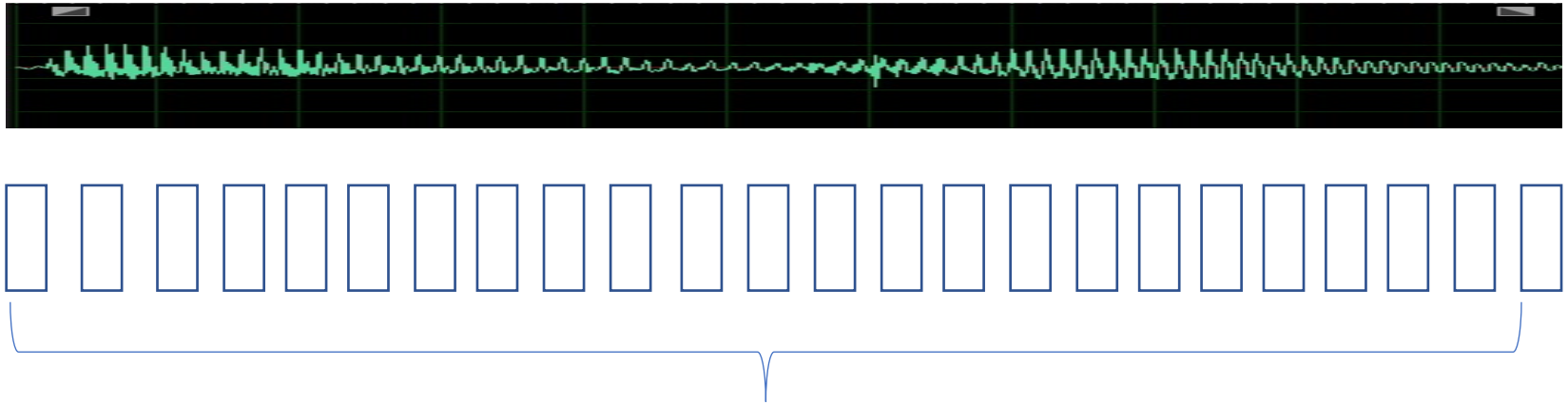
4.语言模型
词->句子



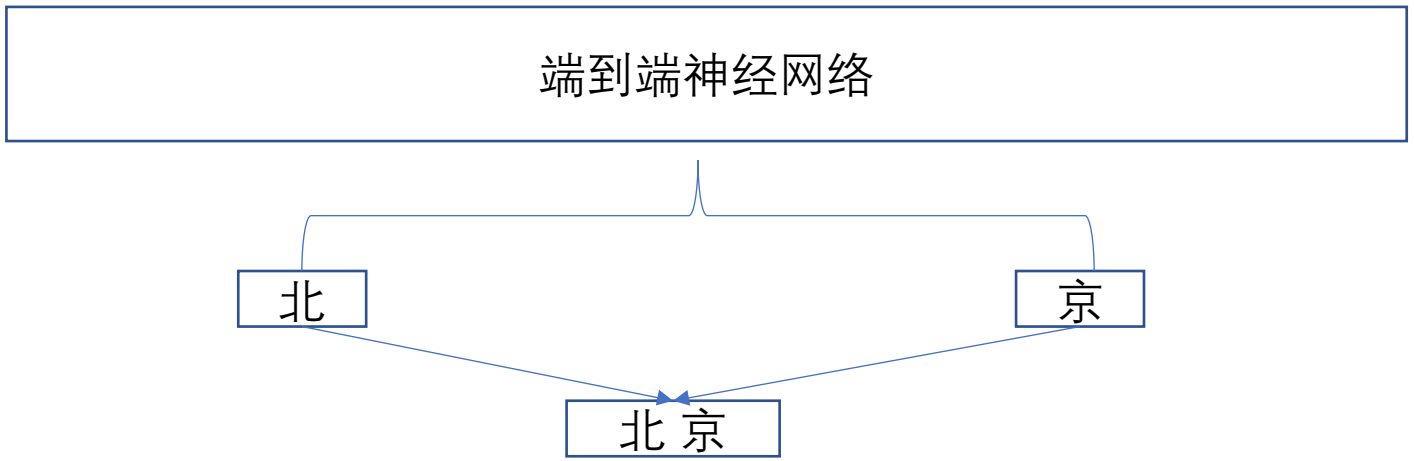
没有HMM，直接建模到音素，需要词典映射为汉字

ASR-端到端建模

1.特征抽取
时域点->频谱特征



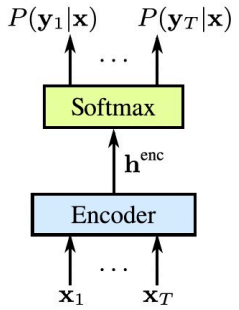
2.端到端建模
计算p(词|特征)



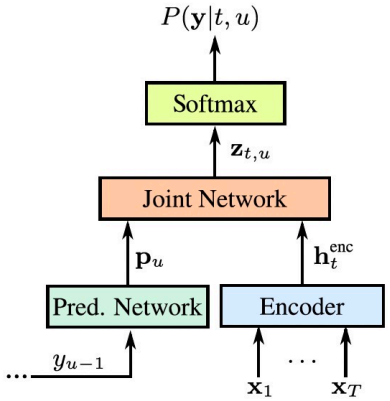
3.语言模型
词->句子

没有HMM，不需要词典，直接建模到汉字

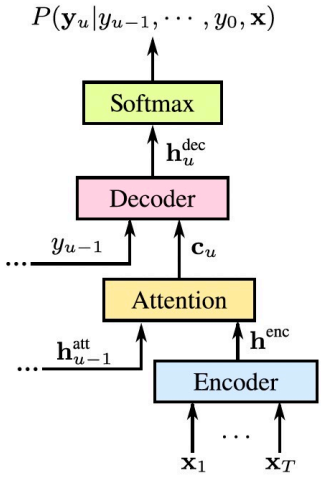
ASR-端到端建模



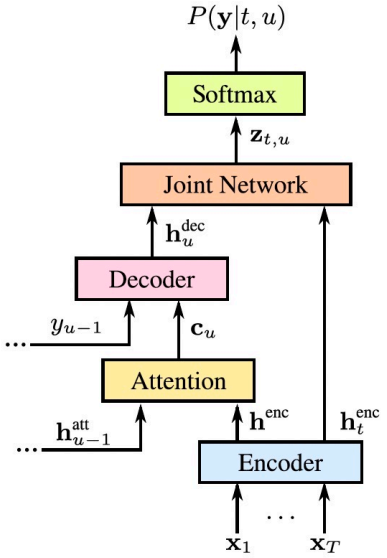
(a.) CTC



(b.) RNN-Transducer



(c.) Attention-based Model



(d.) RNN-Transducer with Attention

Table 1: *WERs (%) on various test sets for the models compared in this work. The attention-based model with two decoder layers is the single best sequence-to-sequence model.*

Model	Clean		Noisy		numeric
	dict	vs	dict	vs	
Baseline Uni. CDP	6.4	9.9	8.7	14.6	11.4
Baseline BiDi. CDP	5.4	8.6	6.9	-	11.4
End-to-end systems					
CTC-grapheme[3]	39.4	53.4	-	-	-
RNN Transducer	6.6	12.8	8.5	22.0	9.9
RNN Trans. with att.	6.5	12.5	8.4	21.5	9.7
Att. 1-layer dec.	6.6	11.7	8.7	20.6	9.0
Att. 2-layer dec.	**6.3**	**11.2**	**8.1**	**19.7**	**8.7**

Hybrid

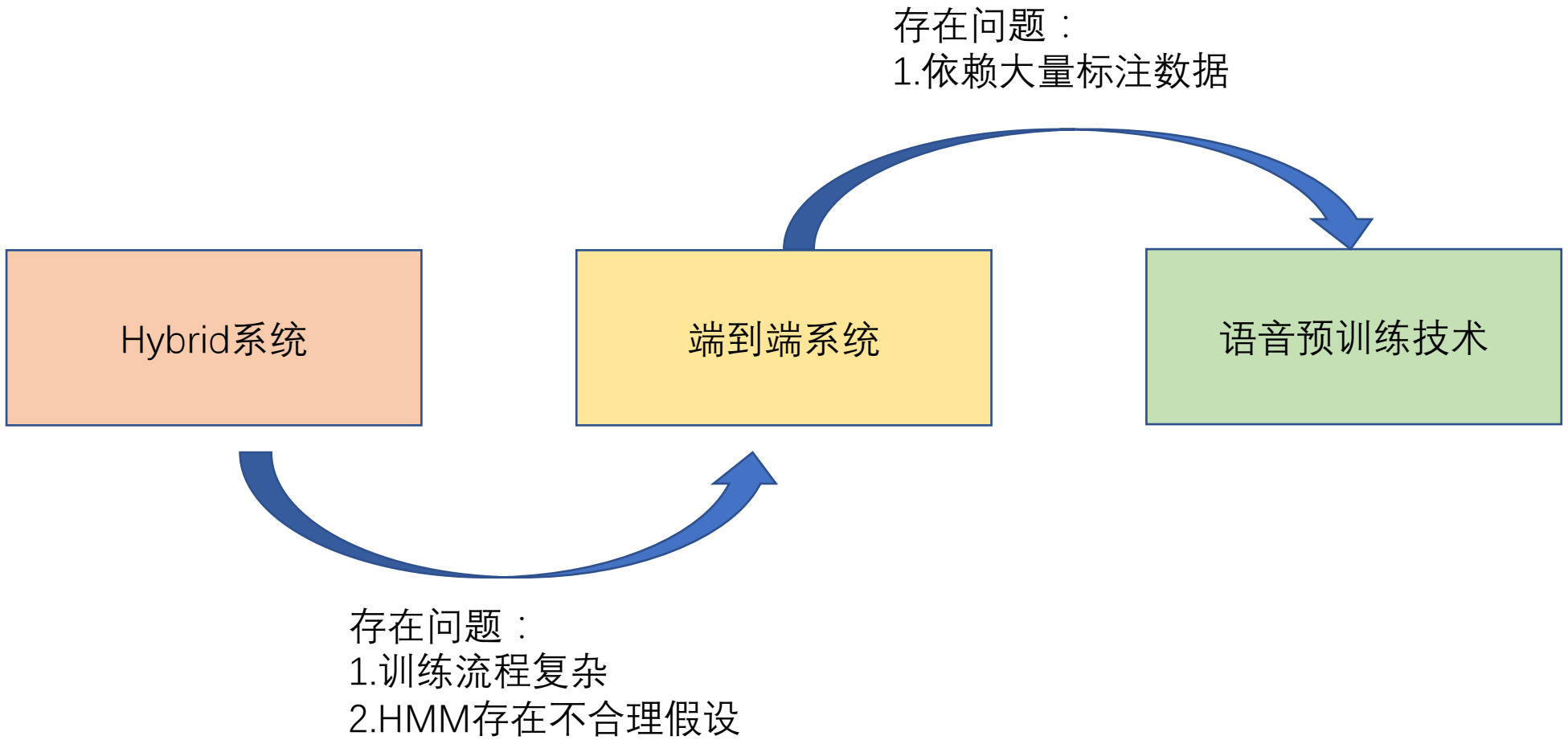
End2end

相比hybrid传统方案，端到端建模有以下特点：

- 训练简单，直接seq2seq进行训练
- Attention/Transducer loss修正了输出独立性假设
- **对标注数据量要求比较多？？**

"A Comparison of Sequence-to-Sequence Models for Speech Recognition." *Interspeech*. 2017.

ASR-建模总结


存在问题：
1.依赖大量标注数据
Hybrid系统
端到端系统
语音预训练技术
存在问题：
1.训练流程复杂
2.HMM存在不合理假设

主要内容

- 一、语音识别简介
- 二、语音预训练技术
- 三、我们的进展和应用

NLP背景

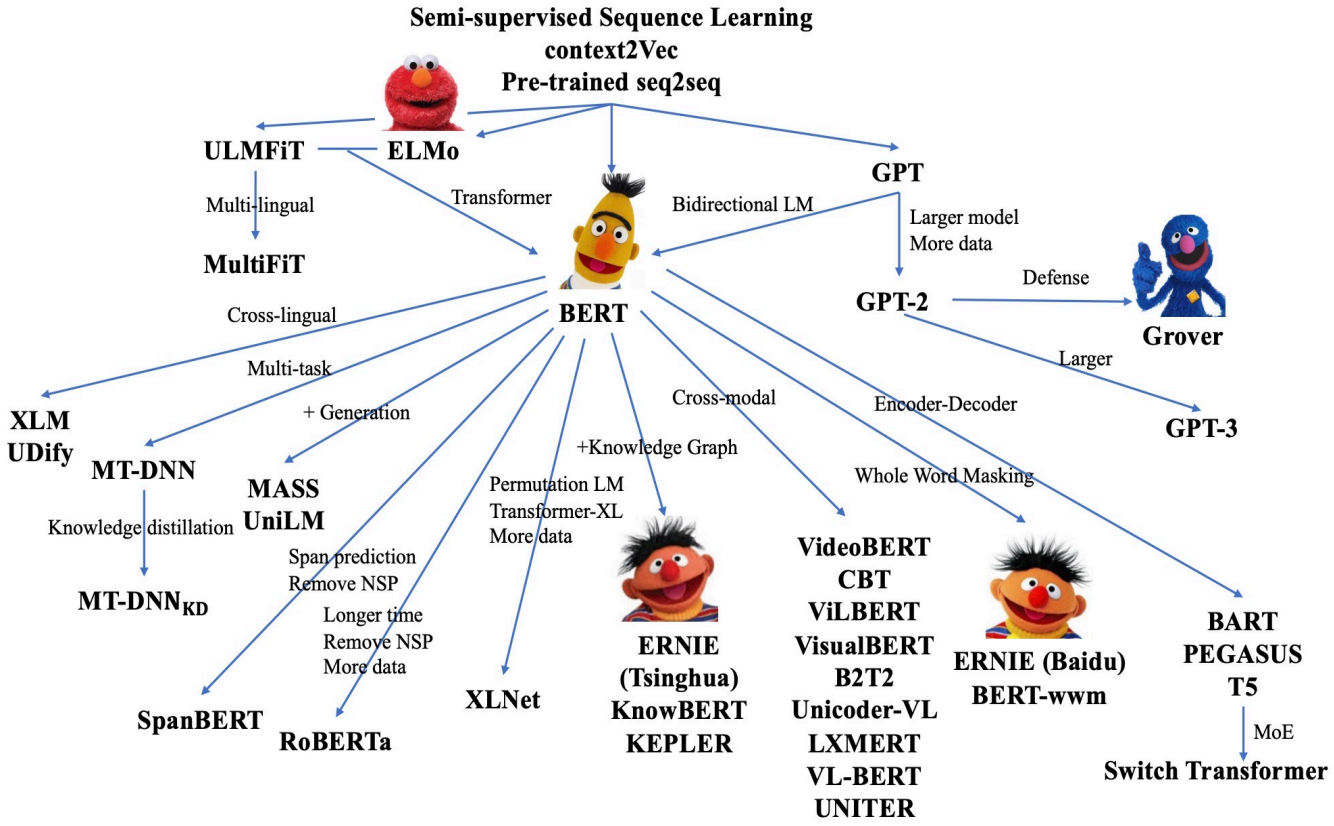


Figure 9: The family of recent typical PTMs, including both pre-trained language models and multimodal models.

预训练模型使用范式：

1. Feature-transfer

使用无监督数据预训练网络，预训练的网络当做特征提取器，为下游的有监督任务提供特征，如ELMo、GloVe

2. Parameter-transfer

使用无监督数据预训练网络，预训练的网络增加新的线性层和loss，进行下游的有监督任务训练，比如BERT、GPT、BART

预训练常用的目标函数：

1. LM（GPT/ELMo）：

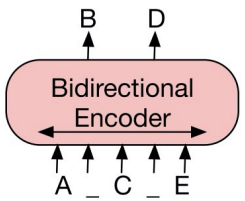
$$\mathcal{L}(\mathcal{X}) = \sum_{i=1}^{n+1} \log P(x_i | x_{i-k}, ..., x_{i-1}; \Theta),$$

2. MLM（BERT/BART）：

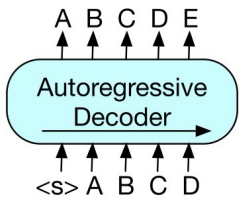
$$\mathcal{L}(\mathcal{X}) = \sum_{i=1}^{m} \log P(\texttt{[Mask]}_i = y_i | \tilde{\mathcal{X}}; \Theta),$$

"Pre-trained models: Past, present and future." *AI Open* 2 (2021): 225-250.

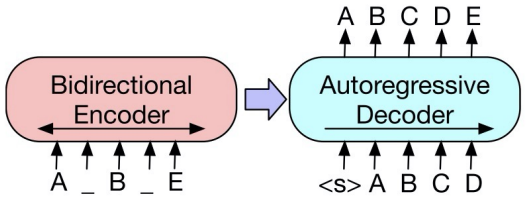
NLP背景



(a) BERT: Random tokens are replaced with masks, and the document is encoded bidirectionally. Missing tokens are predicted independently, so BERT cannot easily be used for generation.



(b) GPT: Tokens are predicted auto-regressively, meaning GPT can be used for generation. However words can only condition on leftward context, so it cannot learn bidirectional interactions.



(c) BART: Inputs to the encoder need not be aligned with decoder outputs, allowing arbitary noise transformations. Here, a document has been corrupted by replacing spans of text with mask symbols. The corrupted document (left) is encoded with a bidirectional model, and then the likelihood of the original document (right) is calculated with an autoregressive decoder. For fine-tuning, an uncorrupted document is input to both the encoder and decoder, and we use representations from the final hidden state of the decoder.

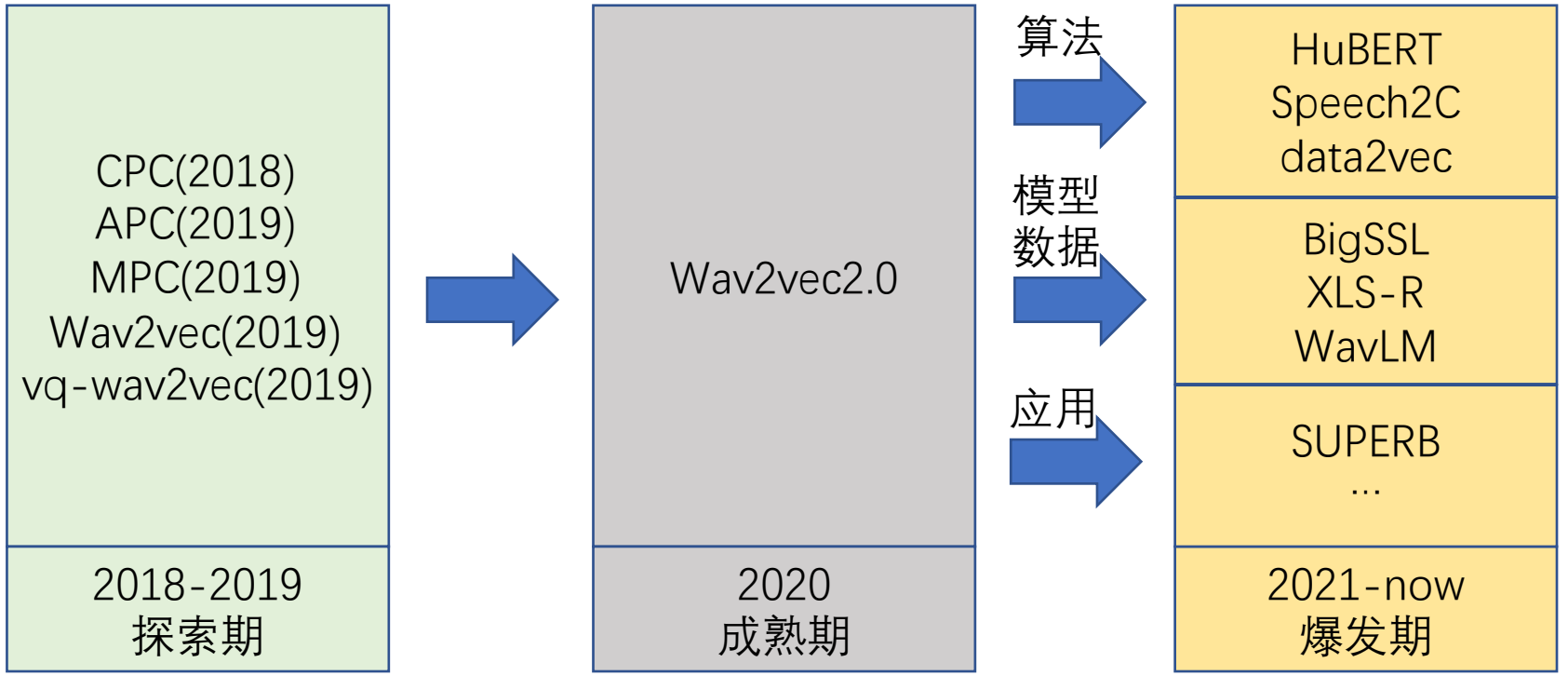
三种NLP预训练模型对比的示意图

模型	使用范式	网络
GPT	Parameter-transfer	Masked transformer
BERT	Parameter-transfer	双向transforemer
BART	Parameter-transfer	Encoder（双向网络） Decoder（单向网络）

NLP预训练模型的使用范式、网络

"Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension." *arXiv preprint arXiv:1910.13461* (2019).

发展阶段

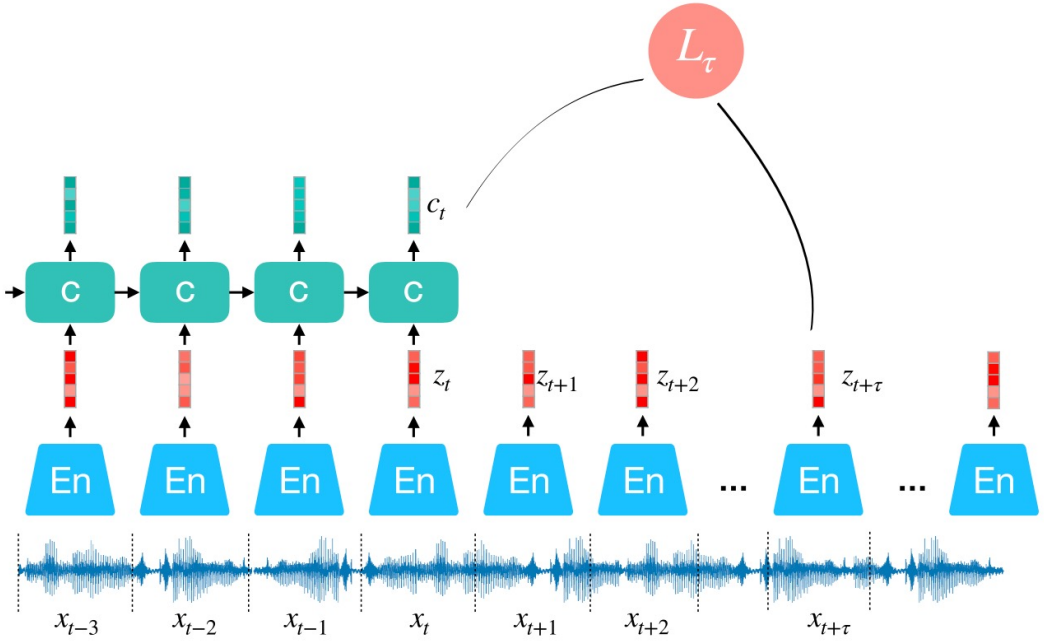


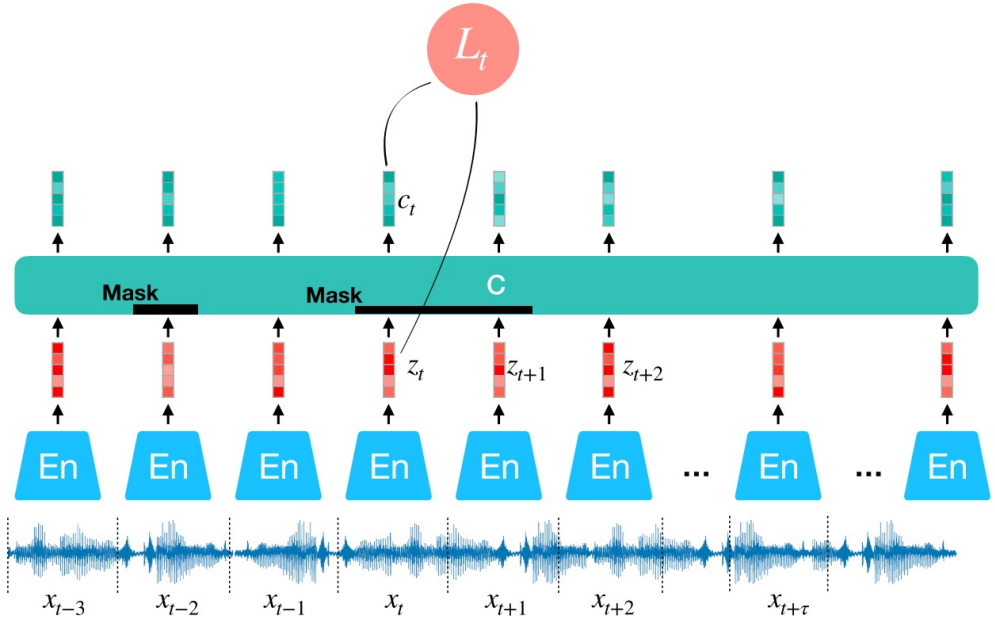
相比baseline系统，还没有取得理想的效果

在已有公开数据集取得SOTA结果，大幅优于已有baseline

成为业界主流架构
在多个方向快速演进

探索期-APC/MPC

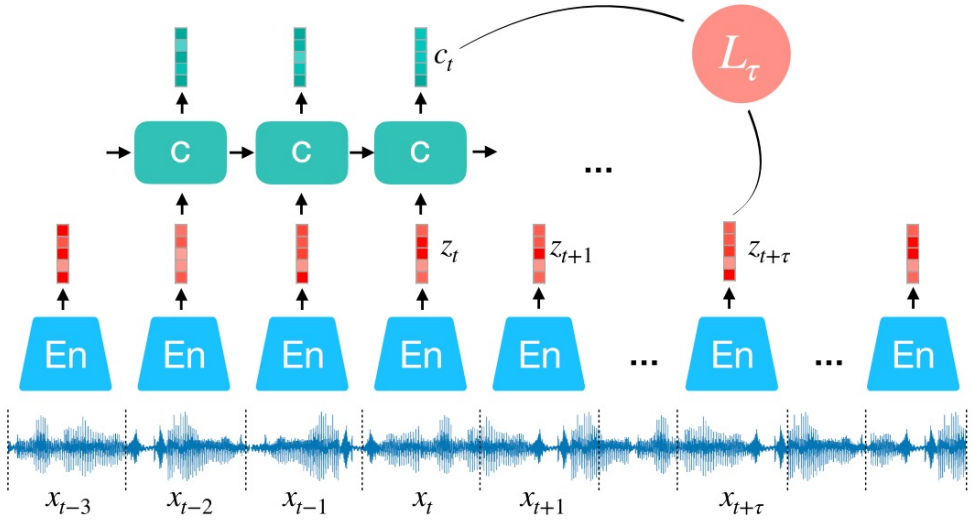


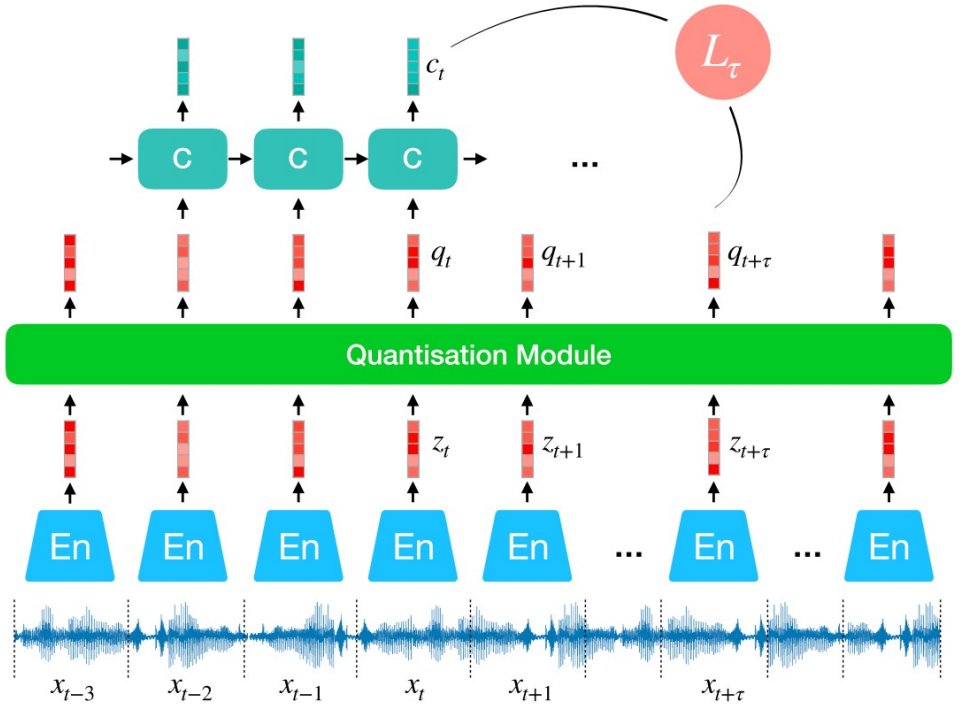


	输入	输出	主要网络	Loss	特点
APC（左图）	Fbank特征	未来$t+\tau$帧	RNN	L1	类似于GPT
MPC（右图）	Fbank特征	Masked帧	transformer	MLM	类似于BERT

"An unsupervised autoregressive model for speech representation learning." *arXiv preprint arXiv:1904.03240* (2019).
"Improving transformer-based speech recognition using unsupervised pre-training." *arXiv preprint arXiv:1910.09932* (2019).
"Audio self-supervised learning: A survey." *arXiv preprint arXiv:2203.01205* (2022).

探索期-wav2vec





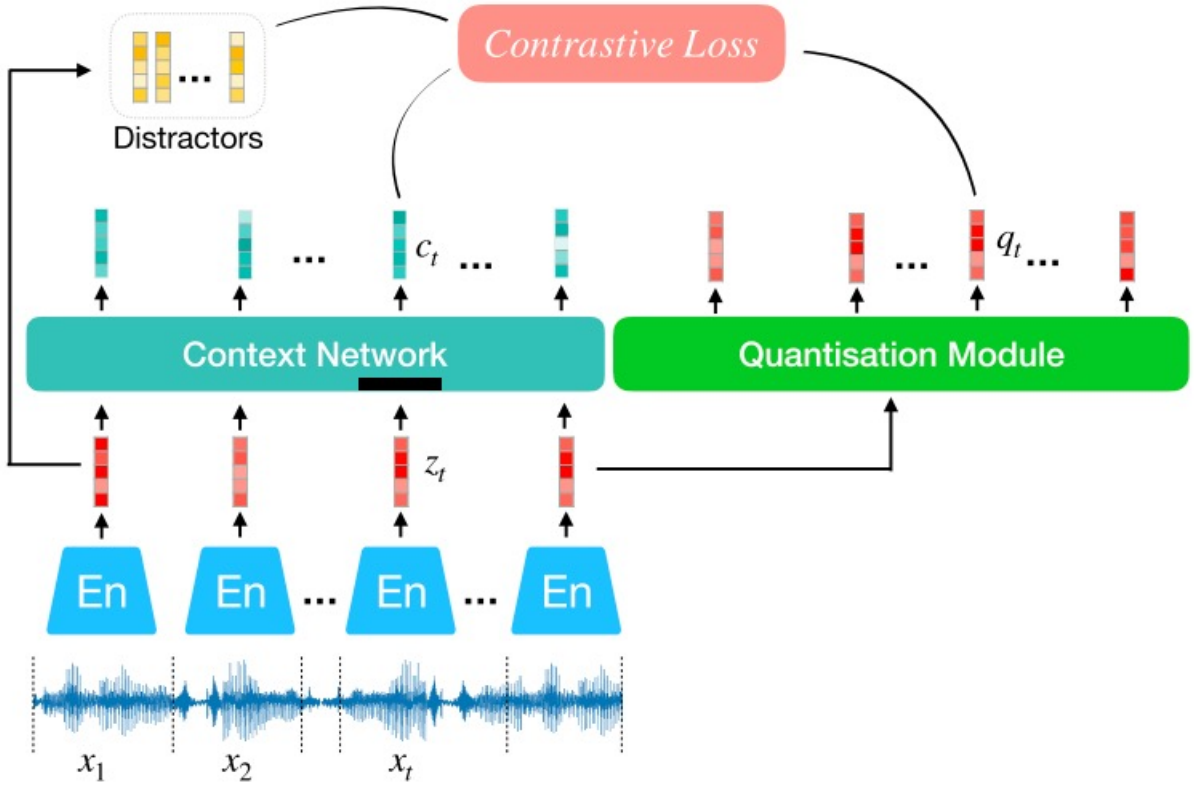
$$\mathcal{L}_k^{\text{wav2vec}} = -\sum_{i=1}^{T-k}\left(\log\sigma(\mathbf{z}_{i+k}^\top h_k(\mathbf{c}_i)) + \lambda\mathop{\mathbb{E}}_{\tilde{\mathbf{z}}\sim p_n}\left[\log\sigma(-\tilde{\mathbf{z}}^\top h_k(\mathbf{c}_i))\right]\right)$$

contrastive loss：$\mathbf{z}_{i+k}$对应正样本，$\mathbf{p}_\mathbf{n}$对应负样本集合

	输入	输出	主要网络	Loss	特点
wav2vec（左图）	wave	未来$t+\tau$帧	CNN	Contrastive loss	预测未来的帧
vq-wav2vec（右图）	wave	未来$t+\tau$帧	CNN+量化	Contrastive loss	分成两阶段
	量化特征	Masked帧	BERT	MLM loss	

"vq-wav2vec: Self-supervised learning of discrete speech representations." ***arXiv preprint arXiv:1910.05453*** **(2019).**
"wav2vec: Unsupervised pre-training for speech recognition." *arXiv preprint arXiv:1904.05862* (2019).

成熟期-wav2vec 2.0



网络分成三部分：
1. Encoder：将语音输入转化为特征序列
2. Context Network：将输入特征进行编码抽象
3. Quantisation Module：将目标进行离散化

Contrastive loss：
1. q_t表示正样本，q̃表示负样本
2. $sim()$表示两个向量的cos距离

$$\mathcal{L}_m = -\log \frac{\exp(sim(\mathbf{c}_t, \mathbf{q}_t))/\kappa}{\sum_{\tilde{\mathbf{q}} \sim \mathbf{Q}_t} \exp(sim(\mathbf{c}_t, \tilde{\mathbf{q}}))/\kappa}$$

Masking：
1. Encoder的输出做mask作为context网络的输入
2. 采取分段连续masking策略，整体占比49%

"wav2vec 2.0: A framework for self-supervised learning of speech representations." *Advances in Neural Information Processing Systems* 33 (2020): 12449-12460.

成熟期-wav2vec 2.0

1h labeled						
Discrete BERT [4]	LS-960	4-gram	8.5	16.4	9.0	17.6
BASE	LS-960	4-gram	5.0	10.8	5.5	11.3
		Transf.	3.8	9.0	4.0	9.3
LARGE	LS-960	Transf.	3.8	7.1	3.9	7.6
	LV-60k	Transf.	3.3	6.4	3.4	6.8
100h labeled						
Hybrid DNN/HMM [33]	-	4-gram	5.0	19.5	5.8	18.6
TTS data augm. [29]	-	LSTM			4.3	13.5
Discrete BERT [4]	LS-960	4-gram	4.0	10.9	4.5	12.1
Iter. pseudo-labeling [56]	LS-860	4-gram+Transf.	5.0	8.72	5.37	9.51
Iter. pseudo-labeling [56]	+LV-60k	4-gram+Transf.	3.19	6.14	3.72	7.11
Noisy student [41]	LS-860	LSTM	3.9	8.8	4.2	8.6
BASE	LS-960	4-gram	2.7	7.9	3.4	8.0
		Transf.	2.2	6.3	2.6	6.3
LARGE	LS-960	Transf.	2.1	4.8	2.3	5.0
	LV-60k	Transf.	2.0	4.1	2.1	4.4

1小时结果

100小时baseline

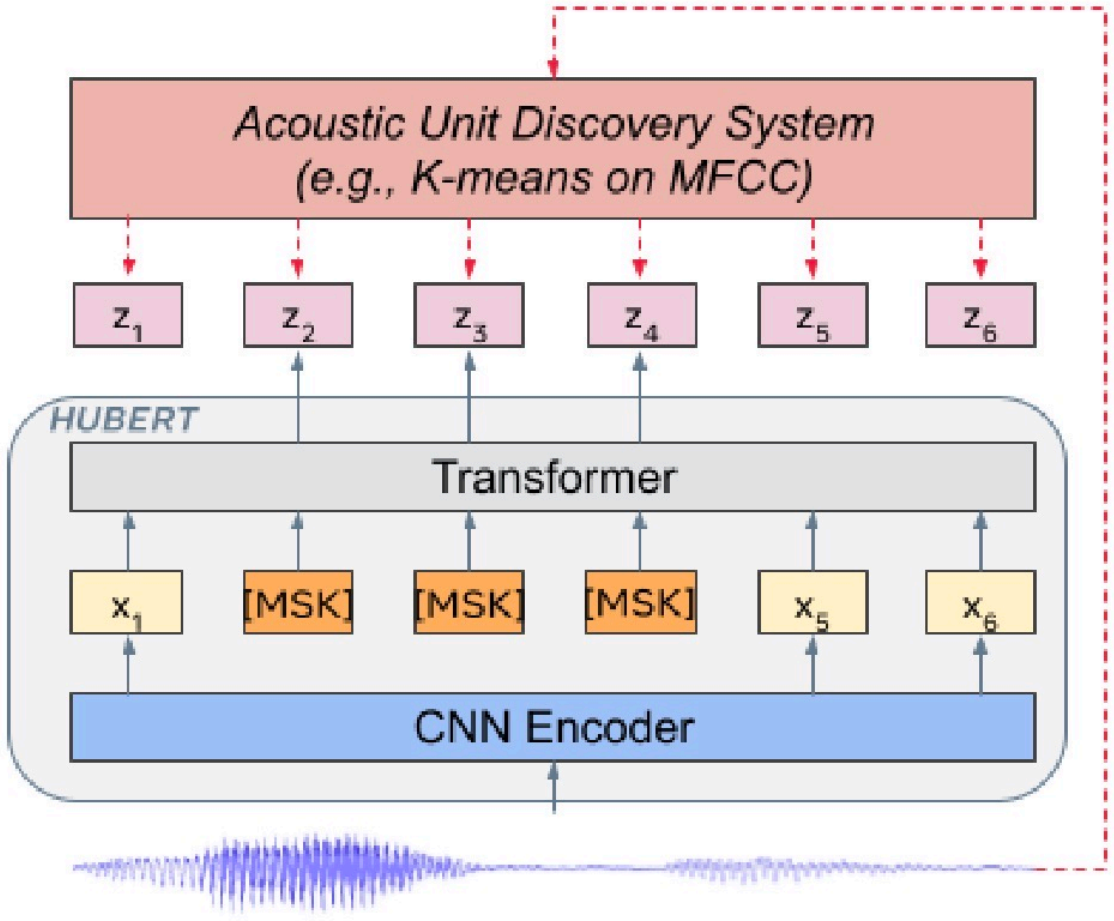
100小时结果

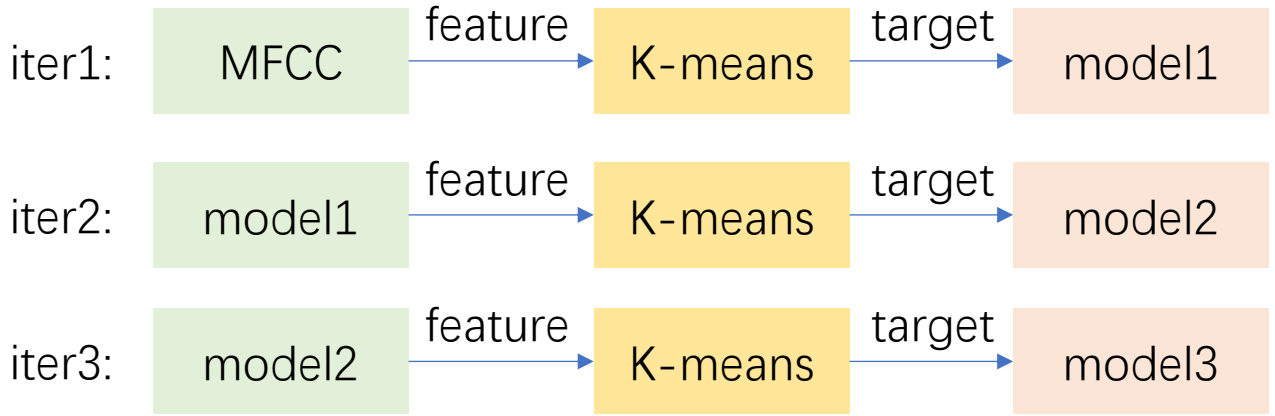
Wav2vec2.0 在语音识别任务上面的实验结果：

- 在Librispeech公开数据集的100小时任务上面取得了SOTA的结果，明显优于之前的工作
- 只使用了1小时的标注数据，wav2vec 2.0的结果就可以优于之前100小时标注数据的最好结果

wav2vec 2.0工作发挥了预训练在语音任务的威力，大幅降低了对标注数据的依赖

爆发期-算法方向-HuBERT



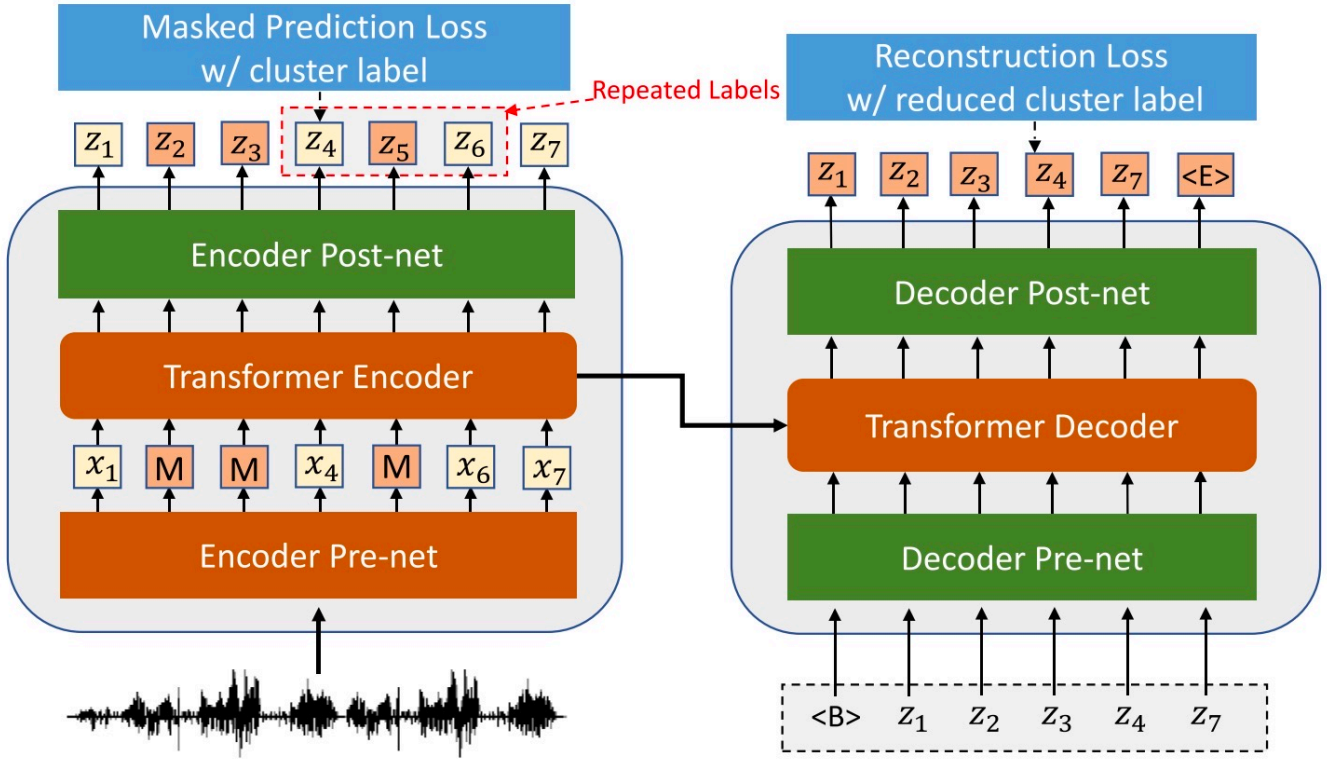


		100-hour labeled				
IPL [12]	LL-60k	4-gram + Transformer	3.19	6.14	3.72	7.11
SlimIPL [54]	LS-860	4-gram + Transformer	2.2	4.6	2.7	5.2
Noisy Student [61]	LS-860	LSTM	3.9	8.8	4.2	8.6
DeCoAR 2.0 [50]	LS-960	4-gram	-	-	5.0	12.1
DiscreteBERT [51]	LS-960	4-gram	4.0	10.9	4.5	12.1
wav2vec 2.0 BASE [6]	LS-960	4-gram	2.7	7.9	3.4	8.0
wav2vec 2.0 LARGE [6]	LL-60k	Transformer	1.9	4.0	2.0	4.0
HUBERT BASE	LS-960	4-gram	2.7	7.8	3.4	8.1
HUBERT LARGE	LL-60k	Transformer	1.8	3.7	2.1	3.9
HUBERT X-LARGE	LL-60k	Transformer	1.7	3.0	1.9	3.5

HuBERT整体上跟NLP领域的BERT模型非常相似，在Librispeech 100h任务上效果上来讲跟wav2vec2.0相当

"Hubert: Self-supervised speech representation learning by masked prediction of hidden units." ***IEEE/ACM Transactions on Audio, Speech, and Language Processing*** **(2021): 3451-3460.**

爆发期-算法方向-Speech2C

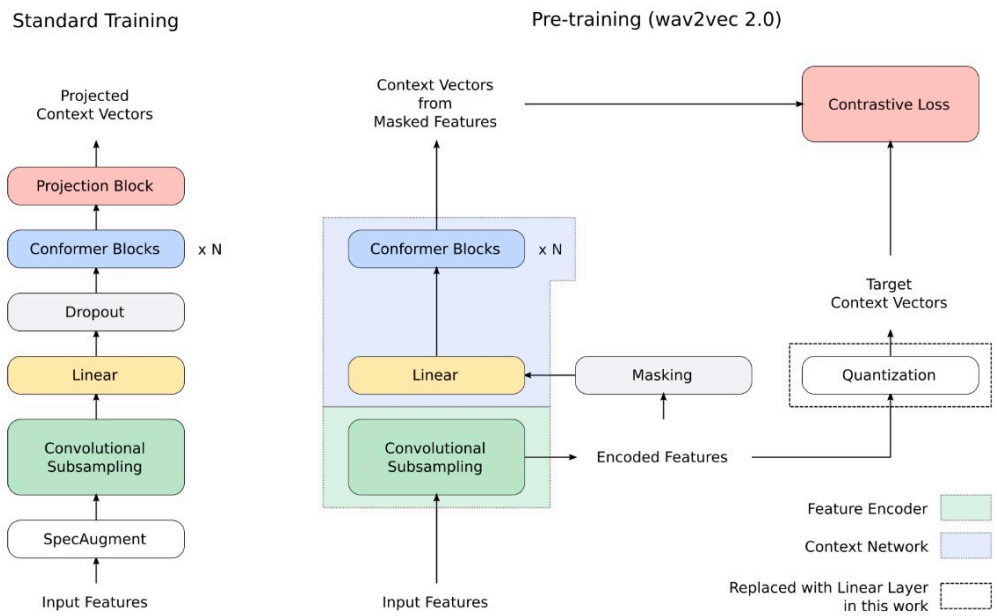


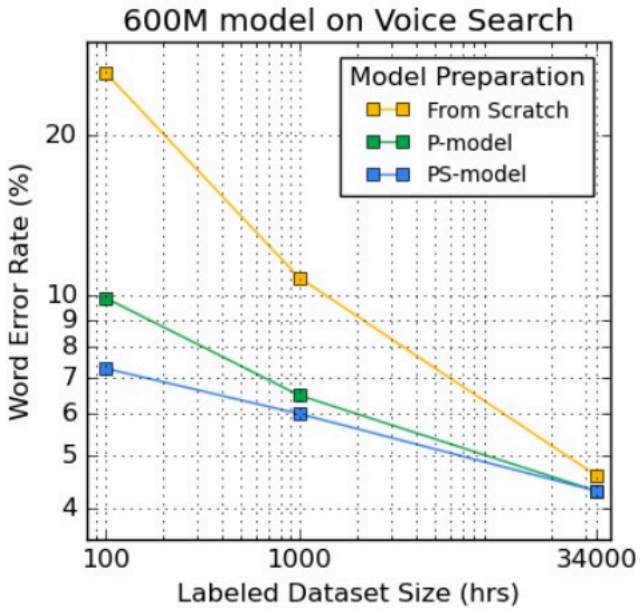
100 hours subset			
wav2vec2.0 BASE [11]	None	6.1	13.3
wav2vec2.0 LARGE [11]	None	4.7	9.0
HuBERT BASE † [12]	None	6.3	13.2
SpeechT5 [15]	None	4.4	10.4
Baseline	None	5.0	11.9
Our Speech2C	None	**4.3**	**9.0**
wav2vec2.0 BASE [11]	4-gram	3.4	8.0
wav2vec2.0 BASE [11]	Transf	2.6	6.3
wav2vec2.0 LARGE [11]	Transf	2.3	5.0
HuBERT BASE [12]	4-gram	3.4	8.1
SpeechT5 [15]	Transf	2.4	5.8
Baseline	Transf	2.5	6.3
Our Speech2C	Transf	**2.4**	**5.2**

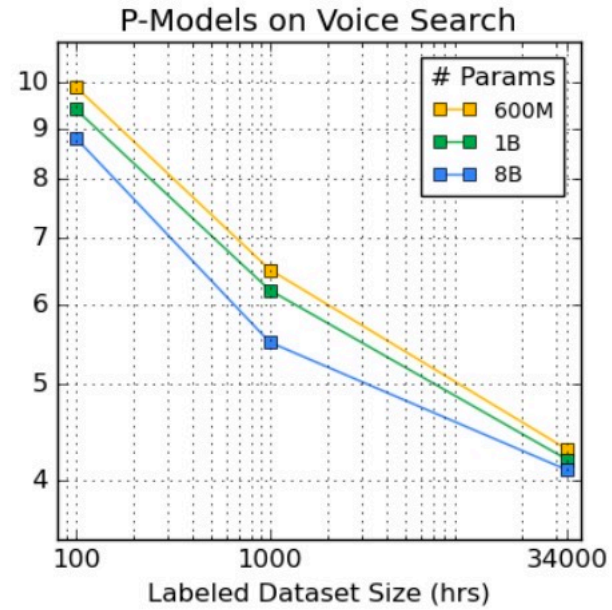
之前的预训练模型主要基于encoder来做，**Speech2C**在HuBERT基础上尝试基于encoder-decoder架构来做预训练，在Librispeech 100h任务上取得了明显的提升，类似于NLP领域的**BART模型**。

"Pre-Training Transformer Decoder for End-to-End ASR Model with Unpaired Speech Data." ***arXiv preprint arXiv:2203.17113*** **(2022).**

爆发期-数据模型方向-BigSSL





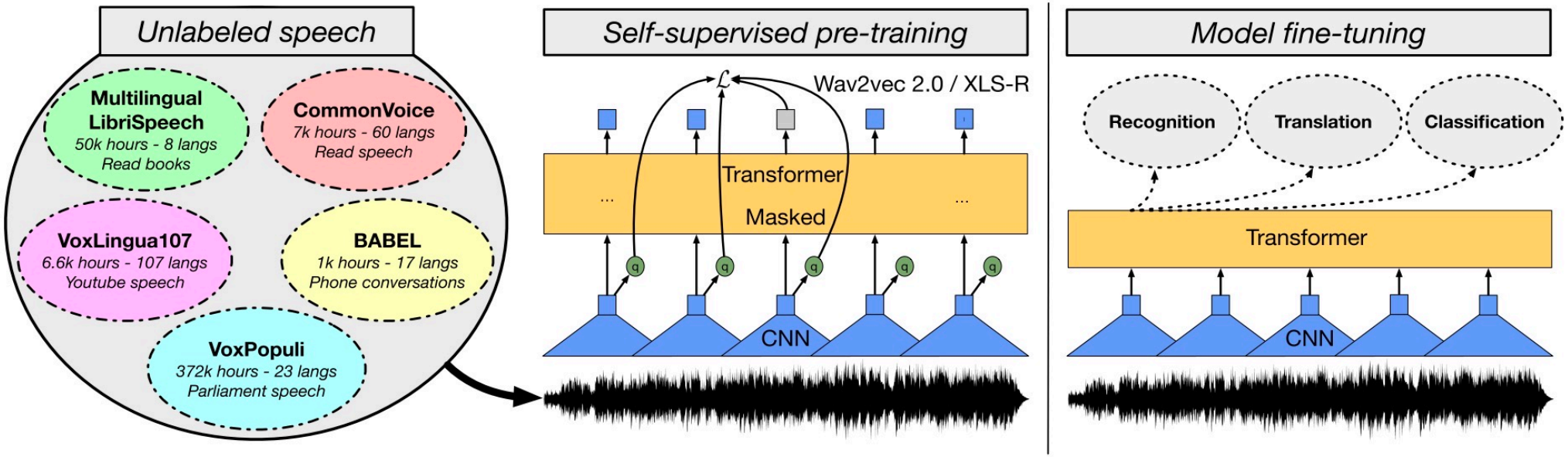


Google的**BigSSL**探索了预训练技术在**超大规模数据集（100万小时无标注数据）**和**超大规模模型参数（8B）**的效果。表格里面的P-model表示只使用pre-training，PS-model表示使用pre-training和self-training

- 第一个表格：随着标注数据量的增大，预训练的优势会越来越小，在34万小时标注数据的情况下相差不大
- 第二个表格：模型参数量越大，效果越好
- 谷歌没有把预训练模型和数据开源

"Bigssl: Exploring the frontier of large-scale semi-supervised learning for automatic speech recognition." ***arXiv preprint arXiv:2109.13226*** **(2021).**

爆发期-数据模型方向-Xlsr



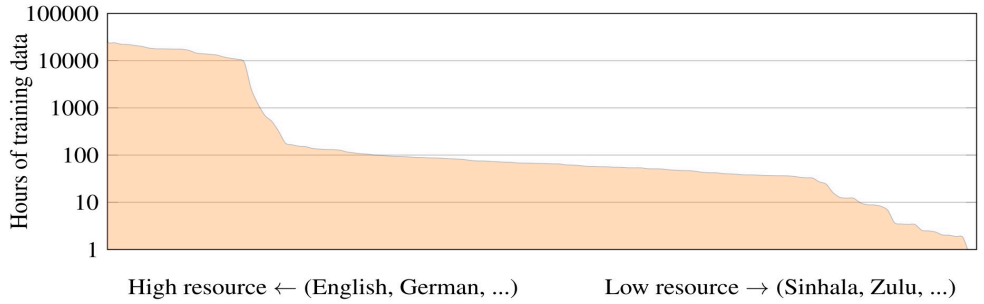


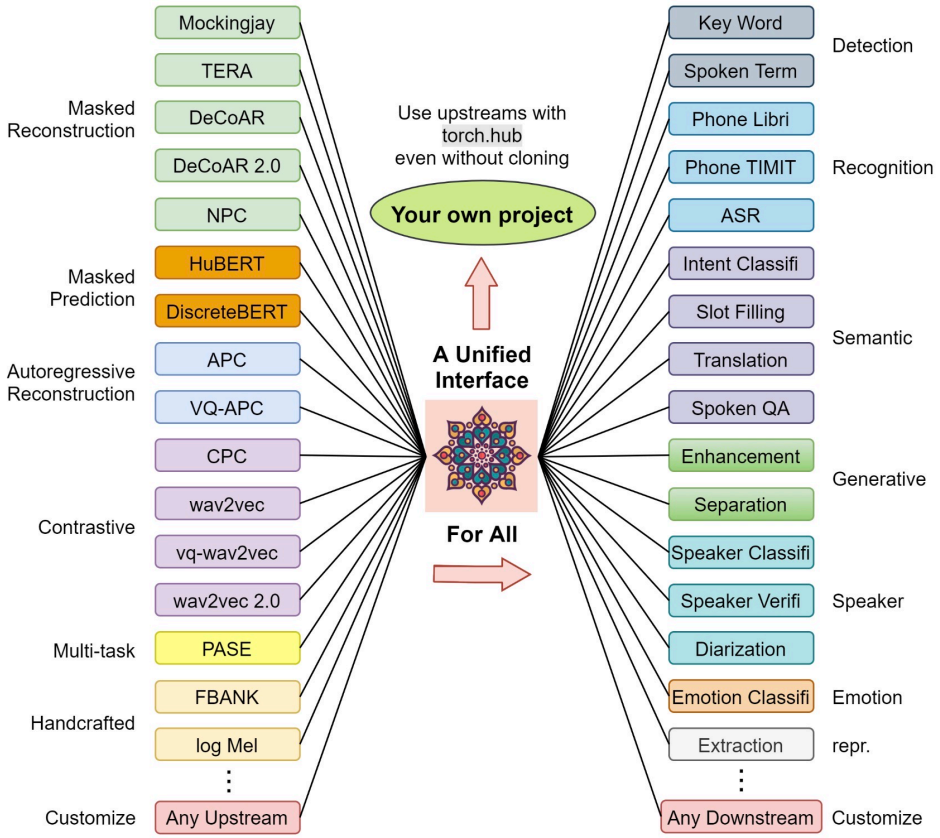
Figure 2: Illustration of the unlabeled training data distribution across the 128 languages of XLS-R.

	es	fr	it	ky	nl	ru	sv	tr	tt	zh-HK	Avg
Labeled data	1h	1h	1h	1h	1h	1h	1h	1h	1h	1h	
Previous work											
m-CPC	38.0	47.1	40.5	41.2	42.5	43.7	47.5	47.3	42.0	55.0	44.5
Fer et al. (2017)	36.6	48.3	39.0	38.7	47.9	45.2	52.6	43.4	42.5	54.3	44.9
XLSR-10	7.9	12.6	11.7	7.0	14.0	9.3	20.6	9.7	7.2	22.8	12.3
XLSR-53	2.9	5.0	5.7	6.1	5.8	8.1	12.2	7.1	5.1	18.3	7.6
This work											
XLS-R (0.3B)	3.1	5.4	4.9	5.1	5.8	6.0	7.2	6.0	4.1	17.0	6.5
XLS-R (1B)	**2.0**	**3.9**	**3.5**	4.1	**4.2**	4.1	5.5	4.4	3.4	15.7	5.1
XLS-R (2B)	2.2	4.0	**3.5**	**4.0**	4.7	**3.7**	**5.0**	**4.0**	**2.9**	**14.8**	**4.9**

Meta AI关于多语种预训练的工作**XLS-R**，预训练数据来自于**128个语种，合计43.6万小时**，在公开数据集common voice上面取得了大幅提升，**代码和模型已经开源：**
https://github.com/facebookresearch/fairseq/blob/main/examples/wav2vec/xlsr/README.md

"Xls-r: Self-supervised cross-lingual speech representation learning at scale." ***arXiv preprint arXiv:2111.09296*** **(2021).**

爆发期-应用方向



* The four columns (1)~(4) correspond to the macs calculated with short, medium, long, longer bucket respectively

Method	Rank ↑	Score ↑	PR ↓	KS ↑	IC ↑	SID ↑	ER ↑	ASR ↓	QbE ↑	SF-F1 ↑	SF-CER ↓	SV ↓	SD ↓
WavLM Large	19.9	1145	3.06	97.86	99.31	95.49	70.62	3.44	8.86	92.21	18.36	3.77	3.24
WavLM Base+	18.7	1106	3.92	97.37	99	89.42	68.65	5.59	9.88	90.58	21.2	4.07	3.5
WavLM Base	16.9	1019	4.84	96.79	98.63	84.51	65.94	6.21	8.7	89.38	22.86	4.69	4.55
HuBERT Large	15.8	919	3.53	95.29	98.76	90.33	67.62	3.62	3.53	89.81	21.76	5.98	5.75
wav2vec 2.0 Large	15.4	914	4.75	96.66	95.28	86.14	65.64	3.75	4.89	87.11	27.31	5.65	5.62
HuBERT Base	15.25	941	5.41	96.3	98.34	81.42	64.92	6.42	7.36	88.53	25.2	5.11	5.88
LightHuBERT Small	13.95	901	6.6	96.07	98.23	69.7	64.12	8.34	7.64	87.58	26.9	5.42	5.85
FaST-VGS+	13.15	809	7.76	97.27	98.97	41.34	62.71	8.83	5.62	88.15	27.12	5.87	6.05
wav2vec 2.0 Base	12.35	818	5.74	96.23	92.35	75.18	63.43	6.43	2.33	88.3	24.77	6.02	6.08
DistilHuBERT	11.2	717	16.27	95.98	94.99	73.54	63.02	13.37	5.11	82.57	35.59	8.55	6.19
DeCoAR 2.0	10.6	722	14.93	94.48	90.8	74.42	62.47	13.02	4.06	83.28	34.73	7.16	6.59
wav2vec	8.9	529	31.58	95.59	84.92	56.56	59.79	15.86	4.85	76.37	43.71	7.99	9.9
vq-wav2vec	7	422	33.48	93.38	85.68	38.8	58.24	17.71	4.1	77.68	41.54	10.38	9.93
APC	5.8	392	41.98	91.01	74.69	60.42	59.33	21.28	3.1	70.46	50.89	8.56	10.53
VQ-APC	5.75	377	41.08	91.11	74.48	60.15	59.66	21.2	2.51	68.53	52.91	8.72	10.45
NPC	5.4	386	43.81	88.96	69.44	55.92	59.08	20.2	2.46	72.79	48.44	9.4	9.34

Benchmark：https://github.com/s3prl/s3prl

Leaderboard：https://superbbenchmark.org/

SUPERB构建了衡量语音预训练模型效果的benchmark，在语音识别、说话人识别、情感识别等十几个下游任务的公开数据集上面均有对应的结果。

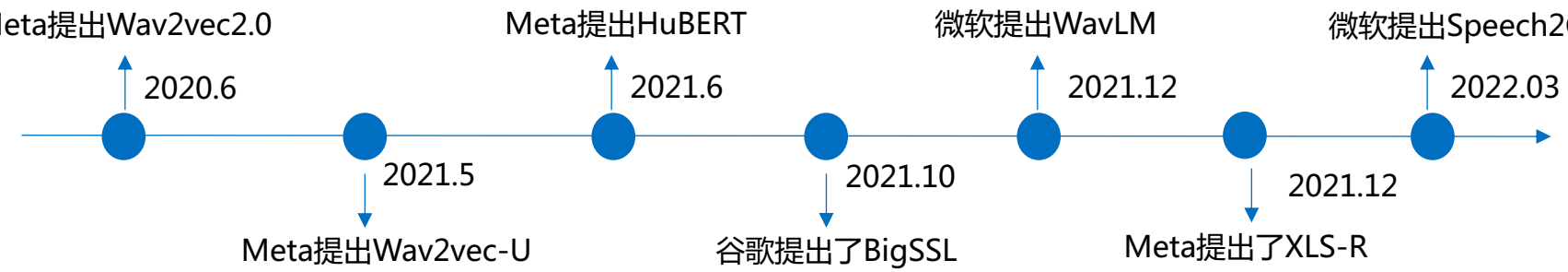
"Superb: Speech processing universal performance benchmark." ***arXiv preprint arXiv:2105.01051*** **(2021).**

语音预训练小结

预训练模型	网络输入	网络结构	loss	Inspired by
APC	fbank	RNN	L1 loss	GPT
MPC	fbank	transformer	MLM loss	BERT
Wav2vec	raw waveform	CNN	Contrastive loss	GPT
Vq-wav2vec	raw waveform	CNN+transformer	Contrastive+MLM loss	BERT
Wav2vec 2.0	raw waveform	CNN+transformer	Contrastive loss	BERT
HuBERT	raw waveform	CNN+transformer	MLM loss	BERT
Speech2C	Raw waveform	CNN+transformer	Contrastive loss+MLE	BART

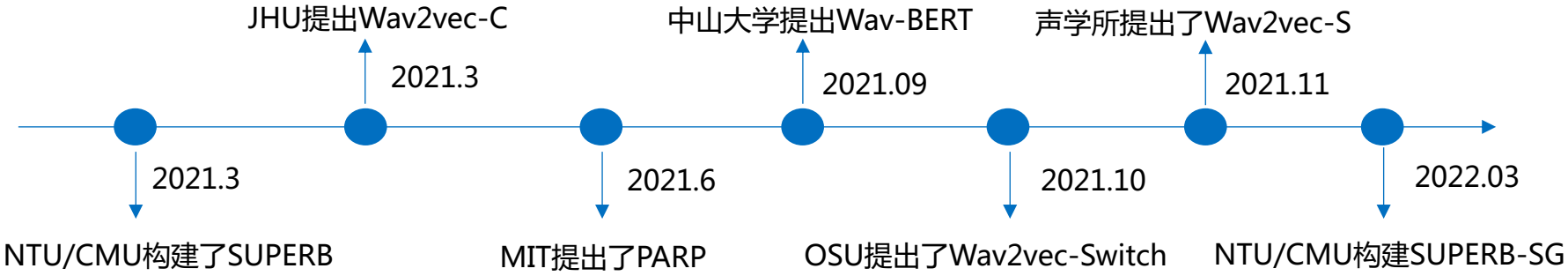
语音预训练小结

工业界



更大模型
更多数据
更难任务

学术界

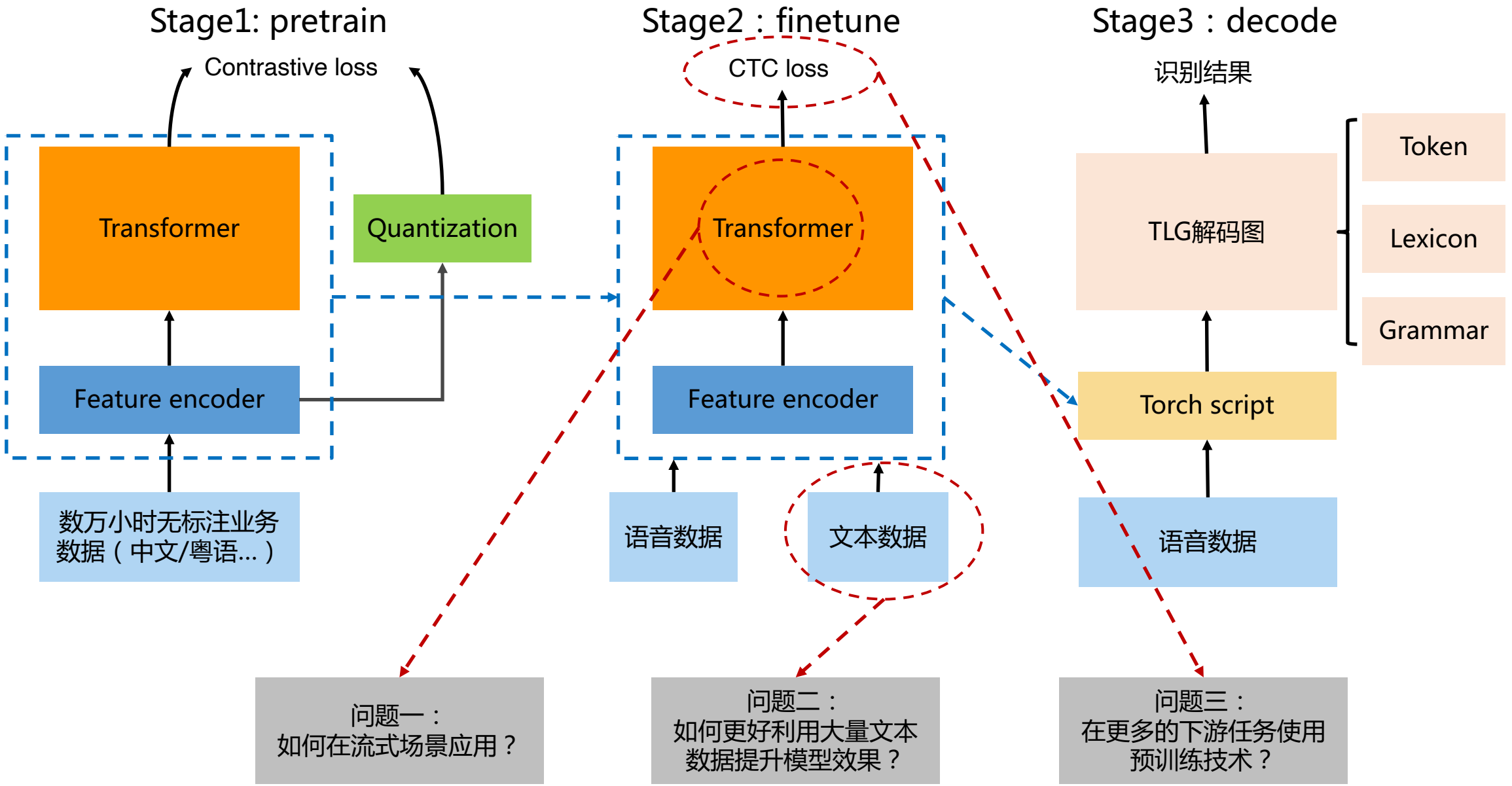


下游任务
应用问题

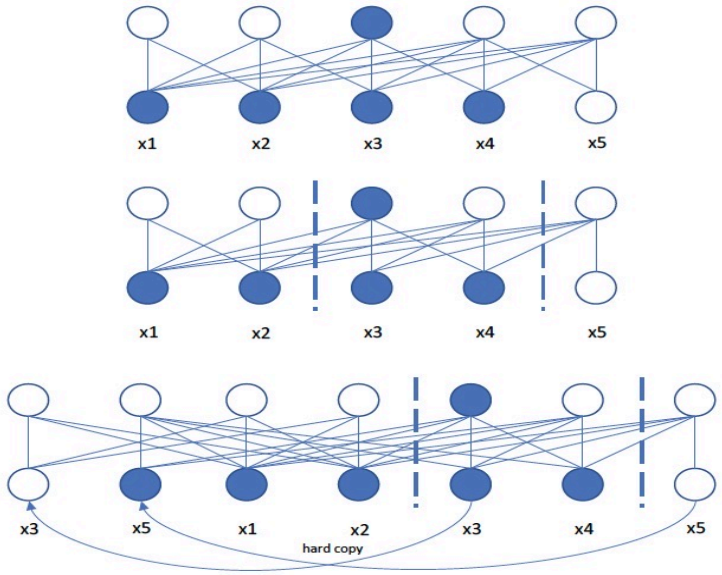
主要内容

- 一、语音识别简介
- 二、语音预训练技术
- 三、我们的进展和应用

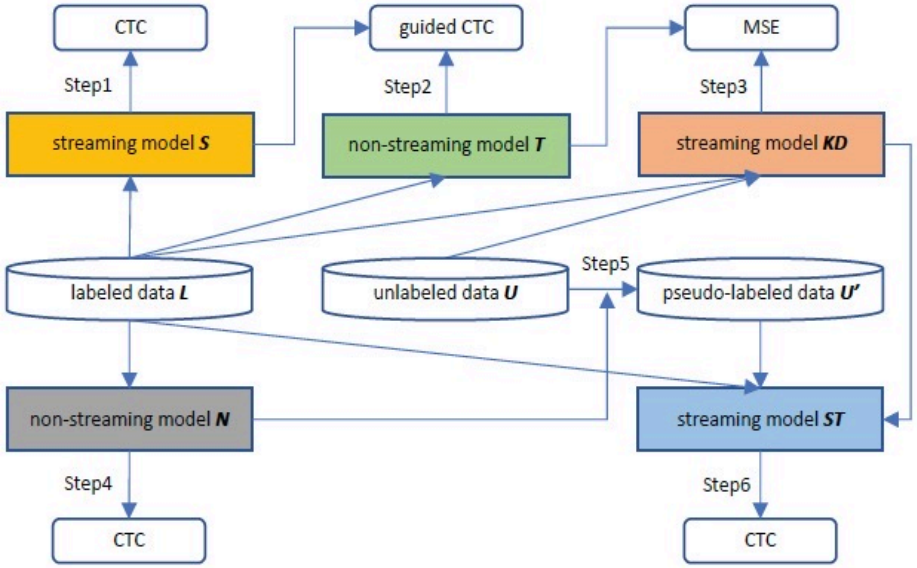
基于预训练的ASR技术方案



如何在流式场景应用？ - 网络流式改造



三种基于不同mask机制的流式方案



two-stage算法流程图

问题背景：

- 非流式语音识别是指模型在用户说完一句话或一段话之后再进行识别（离线转写场景）；流式语音识别是指用户还在说话的时候便同步进行语音识别（助手场景）
- 语音预训练模型一般只适用于非流式语音识别，而无法应用在流式识别场景

算法创新：

- 提出在fine-tune阶段通过引入不同的mask机制来实现流式transformer网络结构
- 提出了基于self-training和knowledge distillation的两阶段的训练算法，其中蒸馏通过非流式模型和流式模型的多个隐层的输出计算MSE loss来实现

"Improving Streaming Transformer Based ASR Under a Framework of Self-supervised Learning." ***Interspeech 2021.***

如何在流式场景应用？ - 网络流式改造

Table 2: *WER results of streaming transformer based models whose EIL is fixed at 480 ms.* ***N3*** *indicates bidirectional transformer,* ***S1*** *indicates Time-restricted Transformer,* ***S2*** *indicates Chunk Transformer,* ***S3*** *and* ***S4*** *indicate Block Transformer.* ***C*** *and* ***F*** *are the chunk size and future size (in millisecond).*

Model	C	F	dev		test	
			clean	other	clean	other
N3	-	-	2.9	8.5	3.5	8.4
S1	-	-	3.9	13.0	4.4	13.0
S2	960	0	3.5	11.4	3.9	11.4
S3	480	240	3.4	10.5	3.9	10.6
S4	240	360	3.5	10.3	3.9	10.4

不同流式方案的结果

Table 3: *WER results of knowledge distillation and self-training.* ***KD*** *indicates knowledge distillation and* ***ST*** *indicates self-training.*

Model	KD	ST	dev		test	
			clean	other	clean	other
N1	-	-	2.9	8.1	3.3	8.1
N4	N	Y	2.9	7.4	3.2	7.6
S4	-	-	3.5	10.3	3.9	10.4
S5	Y	N	3.3	9.6	3.7	9.8
S6	N	Y	3.3	8.6	3.6	8.9
S7	Y	Y	3.2	8.5	3.5	8.7

two-stage算法的效果

实验设置：

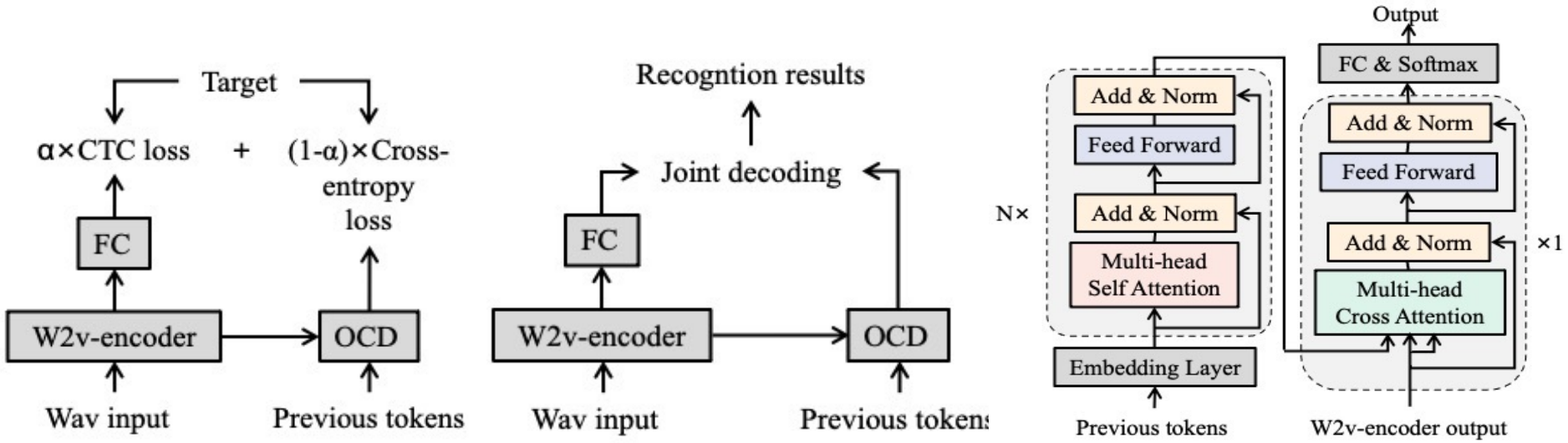
- Pre-train使用Librispeech 960h数据，Fine-tune使用clean-100h数据
- 解码使用开源的Librispeech语言模型

实验结果：

- 不同的流式方案的引入，都会带来比较大的性能下降（N3->S4）
- 相比S4的基线，提出的two-stage算法S7在test-other测试集取得了相对15%的相对提升，大幅缩小了流式场景和非流式场景（N3）的gap

"Improving Streaming Transformer Based ASR Under a Framework of Self-supervised Learning." ***Interspeech 2021.***

如何更好利用文本数据？ - Preformer



训练阶段　　解码阶段　　OCD的详细结构设计

问题背景：

通常把语义预训练模型和语音预训练模型看做两个独立的模块，ASR系统一般会在实际解码的过程中把语义预训练模型（比如BERT/GPT）当做LM来使用，这种shallow fusion的融合方式比较简单，取到的效果相对有限。

算法创新：

我们提出了一种OCD(one-cross decoder)的网络结构，在声学模型的训练阶段成功将两个模态的预训练模型进行了融合，其中encoder部分使用wav2vec2.0，decoder部分使用GPT模型

"Improving Hybrid CTC/Attention End-to-end Speech Recognition with Pretrained Acoustic and Language Model." *ASRU 2021*.

如何更好利用文本数据？ - Preformer

Table 3. The CER (%) of our ASR models with different decoders on the AISHELL-1 corpus.

ASR Model	Encoder	Decoder	CER dev	CER test
baseline	Transformer	Transformer	5.9	6.3
proposed	Transformer	OCD	5.1	5.6
proposed	Transformer	OCD-1	5.2	5.7
proposed	Transformer	OCD-3	5.2	5.7
proposed	Transformer	OCD-all	5.3	5.7
Preformer	w2v-encoder	OCD	**4.3**	**4.6**
Preformer	w2v-encoder	OCD-1	4.5	4.9
Preformer	w2v-encoder	OCD-3	4.7	5.0
Preformer	w2v-encoder	OCD-all	4.7	5.1
proposed	w2v-encoder	TCD	4.3	4.6
proposed	w2v-encoder	OCD-no-init	4.9	5.2

Table 1. Details of the ASR models' parameter number.

ASR Model	Encoder	Decoder	CTC Branch	Total
baseline	101.6M	63.2M	3.3M	168.1M
Preformer	94.4M	61.0M	3.3M	158.7M

Table 5. The CER (%) of our ASR models with different ways of using the DistilGPT2 on the AISHELL-1 corpus.

ASR	Encoder	Decoder	LM	CER dev	CER test
baseline	Transformer	Transformer	vanilla	5.9	6.3
baseline	Transformer	Transformer	DistilGPT2	5.3	5.7
proposed	Transformer	OCD	vanilla	5.1	5.6
proposed	Transformer	OCD	DistilGPT2	4.8	5.2
Preformer	w2v-encoder	OCD	vanilla	4.3	4.6
Preformer	w2v-encoder	OCD	DistilGPT2	**3.9**	**4.2**

实验设置：

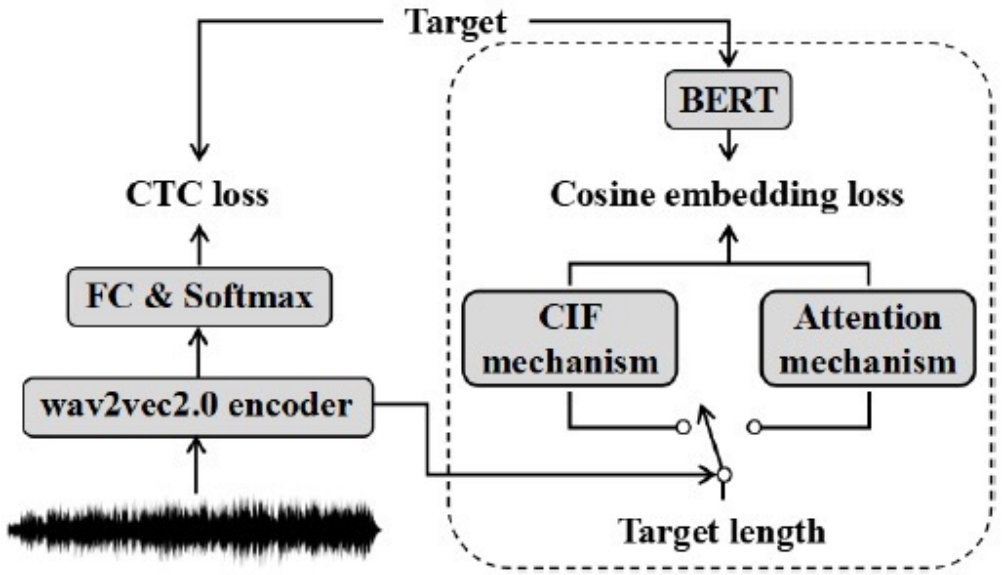
- Pre-train：使用AISHELL-2数据训练wav2vec2.0模型，使用开源的GPT中文模型
- Fine-tune：使用AISHELL-1数据集

实验结果：

- Table1：在参数量不增加的情况下，Preformer相比baseline获得20%相对提升
- Table3：OCD-n表示GPT最后n层可以训练，从结果来看全部固定结果最好
- Table5：在Preformer基础上，在解码阶段引入GPT，还可以进一步提升最终的识别效果

"Improving Hybrid CTC/Attention End-to-end Speech Recognition with Pretrained Acoustic and Language Model." *ASRU 2021*.

如何更好利用文本数据？- KT-RL-CIF



KT-RL-CIF算法示意图

$$w_m = \text{sigmoid}(\max(\text{FC}(\mathbf{h}_m))),$$

$$\hat{w}_m = \frac{w_m}{\sum_{m=1}^{M} w_m} N,$$

$$\mathbf{l}_n = \sum_{m=t_n}^{t_{n+1}} \hat{w}_m \cdot \mathbf{h}_m,$$

$$\mathcal{L}_{cos} = k \cdot \sum_{n=0}^{N} (1 - \cos(\mathbf{l}_n, \mathbf{e}_n)),$$

$\mathbf{h_m}$：模型encoder输出的隐向量

$\mathbf{l}_n$：CIF机制获取的语音模态向量

$\mathbf{e}_n$：BERT获取的语义模态向量

N：语音对应文本的长度

CIF算法计算流程

问题背景：

- 前面提出的Preformer模型虽然取得了比较明显的提升，但是在解码的过程中存在参数量比较大的问题，不方便实际落地

算法创新：

- KT-RL-CIF只在模型训练阶段引入BERT模型，通过CIF机制进行连接，而在实际解码过程把BERT去掉
- 训练过程中通过cosine embedding loss把BERT的语义表征能力向声学模型迁移

“ Improving CTC-based speech recognition via knowledge transferring from pre-trained language models.” *ICASSP 2022*.

如何更好利用文本数据？- KT-RL-CIF

Table 2. The CERs (%) of our ASR system with different structures for knowledge transferring based on representation learning on the AISHELL-1 corpus.

ASR Model	Aux Loss	With LM dev	With LM test	No LM dev	No LM test
Vanilla w2v2.0 CTC	–	4.5	4.9	5.1	5.6
KT-RL-CIF based on w2v2.0	Cosine	**4.1**	**4.2**	**4.3**	**4.7**
KT-RL-CIF based on w2v2.0	MSE	4.4	4.7	4.8	5.1
KT-RL-ATT based on w2v2.0	Cosine	4.2	4.5	4.6	4.8

Table 5. The CER (%) of our ASR system using different pre-trained LM on the AISHELL-1 corpus, which is decoded with external LM.

ASR Model	dev	test
Vanilla w2v2.0 CTC	4.5	4.9
KT-RL-CIF based on w2v2.0 (using BERT)	**4.1**	**4.2**
KT-RL-CIF based on w2v2.0 (using GPT2)	4.2	4.5
KT-CL based on w2v2.0 (using GPT2)	4.4	4.6
KT-CL based on w2v2.0 (using unidirectional BERT)	4.4	4.7

实验设置：

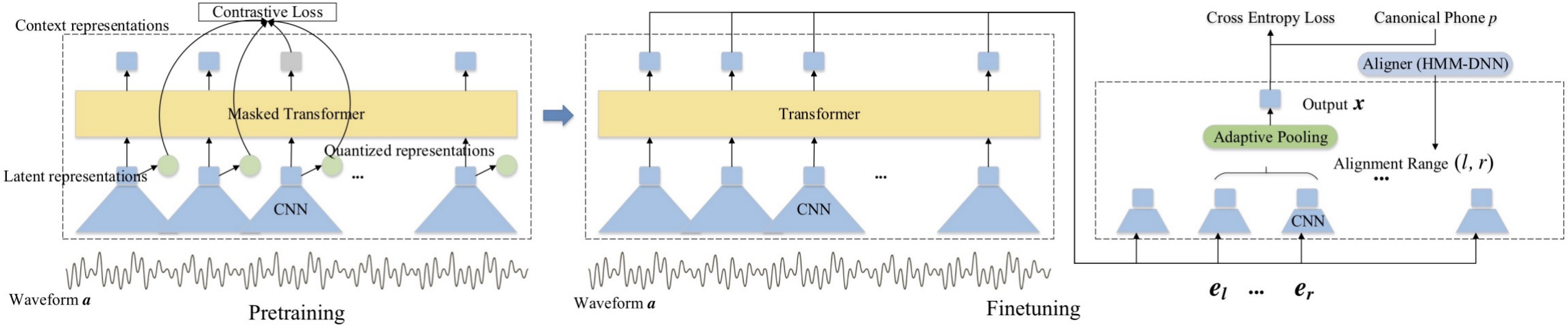
- Pre-train：使用AISHELL-2数据训练wav2vec2.0模型，使用开源的GPT/BERT中文模型
- Fine-tune：使用AISHELL-1数据集

算法结果：

- KT-RL-CIF模型大幅优于base模型，不仅识别效果优于前面的Preformer，而且参数量更少
- Table2对比发现Cosine loss优于MSE，一种解释是对于字向量的相似度评价而言夹角更为重要
- Table2对比发现连接机制使用CIF（KT-RL-CIF），相比传统的Attention（KT-RL-ATT）更优
- Table5给出了使用不同的语义预训练模型的差异，在KT-RL-CIF框架下面BERT优于GPT

“ Improving CTC-based speech recognition via knowledge transferring from pre-trained language models.” *ICASSP 2022*.

更多的下游任务？ - 口语评测



问题背景：

- MD（Mispronunciation Detection）任务一般应用于口语考试等业务场景，存在标注数据难以获取、数据标注困难等问题。

算法创新：

- 我们首先尝试将语音预训练技术应用到该任务，pre-train阶段使用公开英文语音数据，fine-tune阶段使用很少量的标注好的non-native语音数据
- Fine-tune阶段新增CNN和Adaptive Pooling层，同时将传统MD任务的音素识别任务转化为二分类问题（发音错误-1，发音正确-0）

“Explore Wav2vec 2.0 for Mispronunciation Detection Improving.” *Interspeech 2021*.

更多的下游任务？ - 口语评测

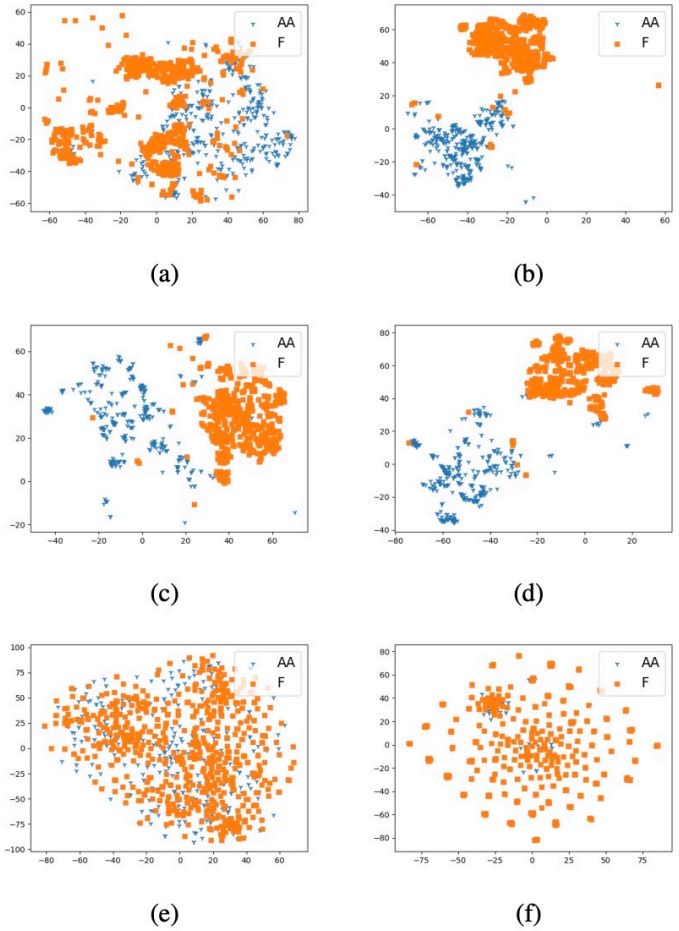


Figure 3: *Visulizations of the representations. (a), (c) and (e) refer to the visualizations of Proposed-SS, Proposed-ASR and Proposed-DL respectively before the finetuning, while (b), (d) and (f) are the visualizations of Proposed-SS, Proposed-ASR and Proposed-DL respectively after the finetuning.*

Table 2: *Performance on different methods on L2-ARCTIC test set.*

	Precision	Recall	F1	FAR	FRR
LPP	0.566	0.602	0.583	0.398	0.080
Trans-GOP	0.464	0.583	0.517	0.417	0.116
Proposed-ASR	0.538	**0.681**	0.602	**0.319**	0.101
Proposed-SS	**0.580**	0.643	**0.610**	0.357	**0.080**
Proposed-CTC	0.514	0.550	0.531	0.450	0.083

实验设置：

- Pre-train使用960小时的Librispeech数据集，Fine-tune使3小时的L2-ARCTIC数据集
- Proposed-SS：使用预训练模型初始化
  Proposed-ASR：使用ASR模型初始化
  Proposed-DL/CTC：没有初始化

实验结果：

- Table2：预训练方案优于传统的技术方案LPP/Trans-GOP
- Figure3：预训练模型对于phone具有明显的区分度

“Explore Wav2vec 2.0 for Mispronunciation Detection Improving.” *Interspeech 2021.*

更多的下游任务？ - 口音语音识别

口音种类	British	America	China	Japan	Russia	India	Portugal	Korea	ALL
官方 baseline	10.06	9.96	11.77	6.79	5.26	10.05	7.45	7.69	8.63
语音预训练技术方案	4.81	4.06	7.09	4.51	4.44	4.22	3.73	2.55	4.42

任务背景：

- INTERSPEECH2020举办英文口音语音识别比赛，共有8种不同类型英文口音，每种口音对应的训练数据只有20小时，合计160小时

技术方案：

- 首次将自监督语音识别技术方案应用到这个低资源任务上面，pre-train阶段使用了960小时的Librispeech英文数据，fine-tune阶段使用160小时的口音英文数据

算法结果：

- 我们在该任务取得了第一名的成绩，同时相关算法成果发表在INTERSPEECH2021会议上面

“Improving Accent Identification and Accented Speech Recognition Under a Framework of Self-supervised Learning.” *Interspeech 2021*.

更多的下游任务？ - 方言识别

模型	Pre-train	Fine-tune	测试集（CER）
竞品1	-	-	16.0
竞品2	-	-	17.3
From-scratch	-	100小时标注	23.2
预训练方案	40000小时无标注	100小时标注	15.2

任务背景：

- 业务侧有方言识别的需求，我们需要从零到一建设这种方言识别的能力，面临最大的挑战就是缺少对应的标注数据。

技术方案：

- 我们尝试将已有的预训练模型方案用在该任务上，只用了100小时的标注数据就取得了超过竞品的效果，相比于之前动辄几千小时标注数据的任务，大幅降低了数据标注的成本。

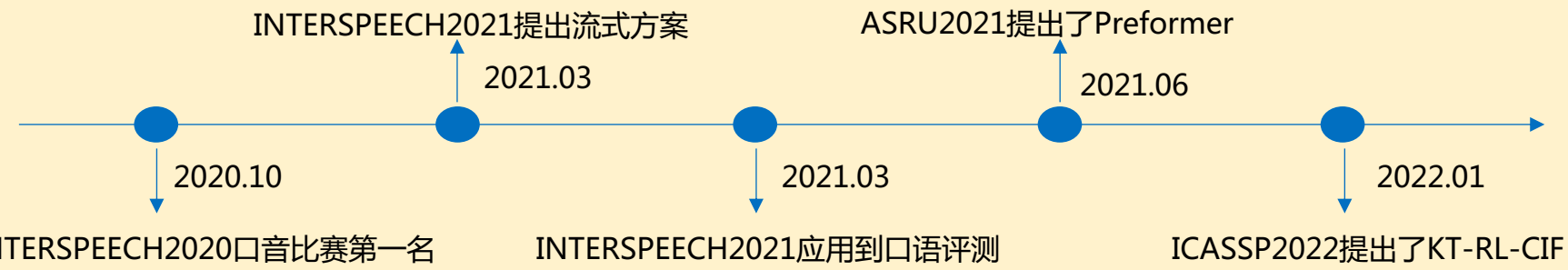
应用启发：

- 结合预训练技术，可以大幅减少对标注数据的依赖，降低数据标注的成本，可以快速实现技术能力的落地。

语音预训练总结

业务支持	车载语音助手	智能客服	口语考试	
技术能力	口音语音识别	方言识别	口语评测	小语种识别

技术底座



Q&A