语音表征学习与应用
从连续到离散的视角

史嘉彤

WAVLab, Language Technologies Institute, CMU

jiatongs@cs.cmu.edu

Self-intro

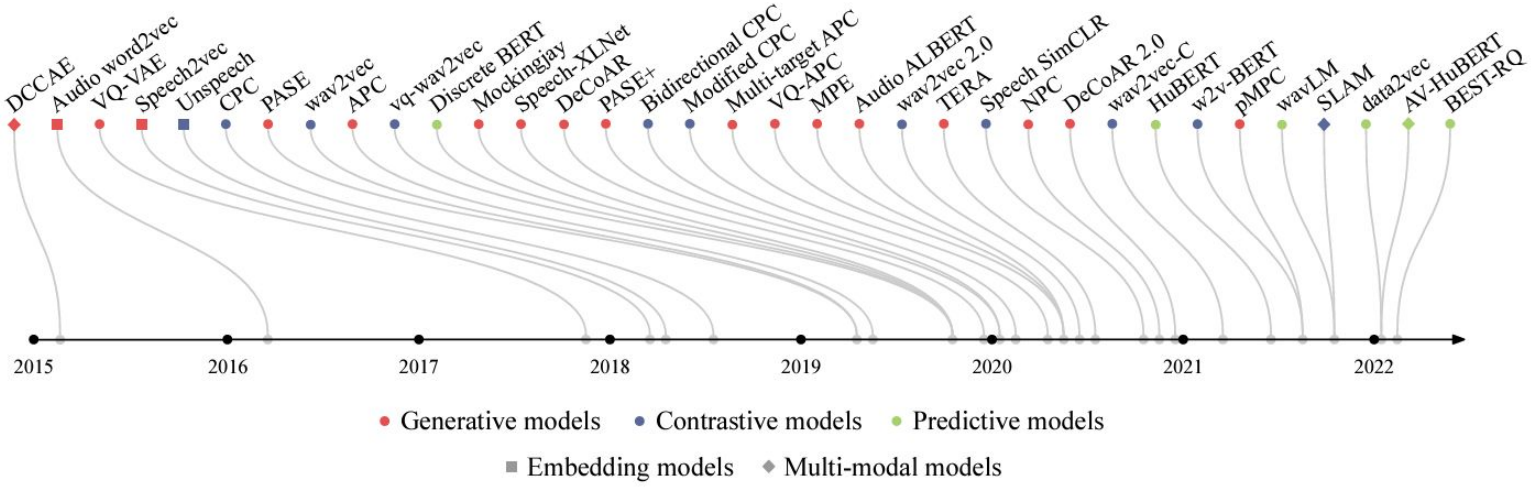
- 三年级博士研究生 (advised by Dr. Shinji Watanabe)
- 开源框架 ESPnet maintainer
- 主要研究兴趣:
  - 语音表征学习以及其在语音识别、语音翻译、语音及歌声合成、语音理解等任务的应用

Content

- 连续语音表征学习
  - Multi-resolution的角度
- 离散语音表征学习
  - 离散语音表征的背景
  - 离散语音表征的应用

Continuous Speech Representation Learning

- 连续语音表征学习--> 自监督学习在语音上的延续



Mohamed, Abdelrahman, et al. "Self-supervised speech representation learning: A review." *IEEE Journal of Selected Topics in Signal Processing* 16.6 (2022): 1179-1210.

More recently:

- Data2vec 2.0 (Baevski et al. 2022)
- SPIRAL (Huang et al. 2022)
- Lipsound2 (Qu et al. 2022)
- Byol-s (Elbanna et al. 2022)
- Wav2vec2-aug (Sriram et al. 2022)
- ccc-wav2vec 2 (Lodagala et al. 2022)
- DinoSR (Liu et al. 2023)
- MT4SSL (Ma et al. 2023)
- AV-data2vec (Lian et al. 2023)
- Large-scale WavLM (Chen et al. 2023)
- cTSTM (Wu et al. 2023)
- FastHuBERT (Yang et al. 2023)
- MERT (Li et al. 2023)
- WavlabLM (Chen et al. 2023)
- MR-HuBERT (Shi et al. 2023)
- Etc.

Speech Self-supervised learning

- Wav2vec2 (Contrastive-learning-based)
  - 优异的效果
    - 在Librispeech 10min、1h等子集中简单finetune就可以快速得到一个不错的识别模型
  - 广泛的应用
    - 后续在multilingual中的应用 -> XLS-R、MMS
    - 相对完善的open-source环境（大量相关的pre-trained models）
- HuBERT (Prediction-based)
  - 更好的效果
    - SUPERB评测中在基本所有语音任务上比Wav2vec2更好
  - 相对容易的应用难度
    - 复现更加容易（predictive training vs. contrastive learning）
    - 更多角度的引入：离散语音特征

Speech and its resolutions

- 语音信号通常是一个长序列信号，包含有较高的采样率
- 由于短时平稳的特性：
  - 我们倾向于从小窗口去考虑 local 语音信号
  - 大部分的特征提取过程也都是从一个“**降频**”的角度去考虑这种和位置有关的信息
    - 传统的特征：语音频谱，Mel频谱，Mel倒谱…
    - 可学习的滤波角度：卷积神经网络（在具体的下游任务中或者从自监督学习模型中产生）

Jiatong Shi[1], *Yun Tang*[2], *Hirofumi Inaguma*[2], *Hongyu Gong*[2], *Juan Pino*[2], *Shinji Watanabe*[1]

[1]Language Technologies Institute, Carnegie Mellon University [2]Meta AI

Does resolution matters?

- Utilizing HuBERT with **different** resolutions

Table 1: *Configurations of the pre-trained HuBERT with multi-resolutions. The convolution (Conv.) module is represented in [(kernel-size, stride)* layer-number].*

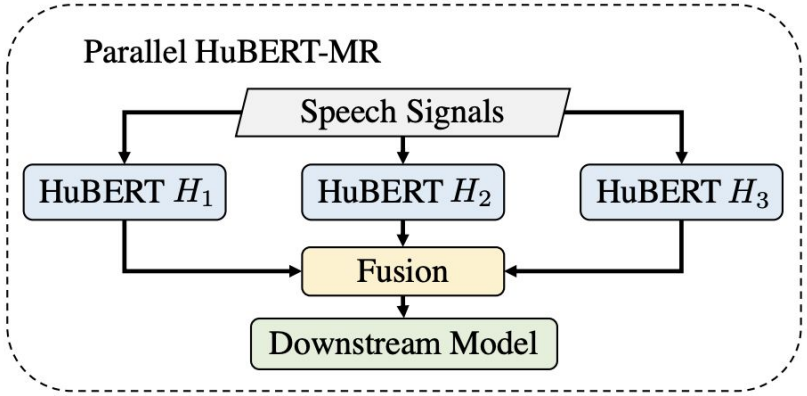
ID	Res.(ms)	Param.	Conv. Module
A	20	94.7	(10,5)*1 + (3,2)*4 + (2,2)*2
B	40	95.2	(10,5)*1 + (3,2)*4 + (2,2)*3
C	100	97.3	(10,5)*2 + (3,2)*4 + (2,2)*2

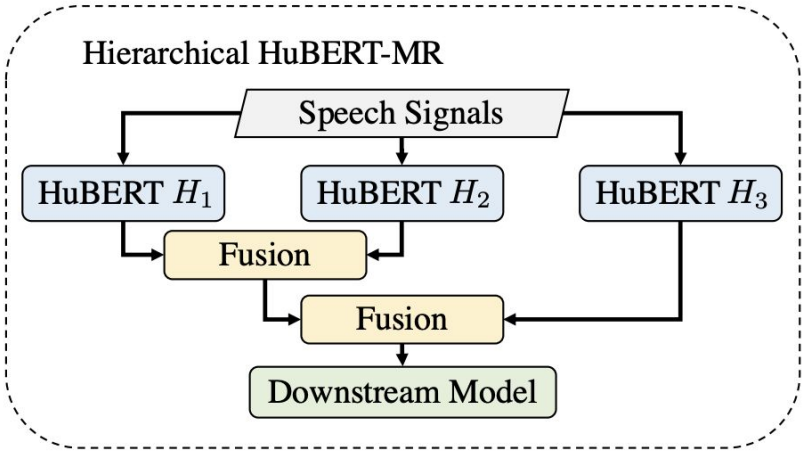
Does resolution matters?

- Utilize HuBERT with **different** resolutions
- Use **simple** method to combine the features

Table 1: *Configurations of the pre-trained HuBERT with multi-resolutions. The convolution (Conv.) module is represented in [(kernel-size, stride)* layer-number].*

ID	Res.(ms)	Param.	Conv. Module
A	20	94.7	(10,5)*1 + (3,2)*4 + (2,2)*2
B	40	95.2	(10,5)*1 + (3,2)*4 + (2,2)*3
C	100	97.3	(10,5)*2 + (3,2)*4 + (2,2)*2





Does resolution matters?

- Get **better performance over (7/9)** tasks than HuBERT in the frozen setting in SUPERB

Model	Res.(ms)	PR(↓)	ASR(↓)	ER(↑)	IC(↑)	SID(↑)	SD(↓)	SV(↓)	SE(↑)	ST(↑)
HuBERT	20	5.41	6.42	**64.92**	98.34	81.42	5.88	5.11	**2.58**	15.53
wav2vec2	20	5.74	6.43	63.43	92.35	75.18	6.08	6.02	2.55	14.81
HuBERT-MR-P	(100,40,20)	**4.83**	**5.48**	63.76	**98.51**	**83.23**	**5.75**	**5.10**	2.55	**16.18**

Does resolution matters?

- With fine-tuning, still **better**

Model	Res.(ms)	WER(↓)
HuBERT	20	7.73(3.81)
HuBERT	40	12.38(4.90)
HuBERT	100	98.37(97.87)
HuBERT-MR-P	(100,20)	6.99(3.70)
HuBERT-MR-P	(40,20)	7.13(3.75)
HuBERT-MR-P	(100,40,20)	6.53(3.61)
HuBERT-MR-H	(100,20)	6.59(3.59)
HuBERT-MR-H	(40,20)	7.01(3.71)
HuBERT-MR-H	(100,40,20)	**6.11**(**3.31**)

Does resolution matters?

- Reduced computation as well:

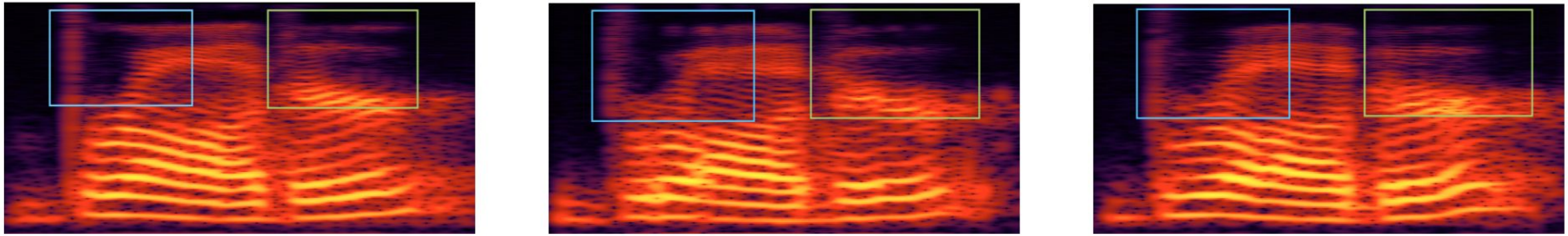
Table 5: *Fine-tuning results on Librispeech-100h (comparison with other models). Results with language model rescoring are in brackets. * indicates the large model setting. The unlabeled column shows the number of hours used for SSL pre-training. HuBERT-MR-H is discussed in Sec 2.2. Noted that the HuBERT base model with 20ms is our baseline.*

Model	Res.(ms)	Unlabeled(h)	Param.(M)	MACs(G)	FLOPs(T)	WER(↓)
HuBERT	20	960	94.7	1669	3.34	7.73(3.81)
wav2vec2	20	960	95.0	1669	3.34	6.54(4.33)
wav2vec2*	20	60K	317.4	4326	8.66	5.90(3.45)
HuBERT*	20	60K	316.6	4324	8.66	**5.40(2.82)**
HuBERT-MR-H	(100,40,20)	960	298.4	3454	6.91	6.11(3.31)

Does resolution matters?

https://arxiv.org/pdf/2306.01084.pdf for more information

- Surprisingly, good for resynthesis directly with its features (without clustering)



(a) 20ms (b) 40ms (c) 100ms

Figure 2: *Speech re-synthesis using features from HuBERT base models at different resolutions. The high-resolution features capture better envelope information in the time domain (shown in the blue box), while the low-resolution features provide more detailed information in the frequency domain (shown in the green box). See Sec. 3.4 for experimental details and discussion.*

Multiresolution in Speech and Language

- 序列信息中不同的分辨率下通常有不同的信息
- 文本信息
  - 字、词、短语、句子、段落、分章
- 语音信息
  - 单独的采样点，加窗信息，音素信息，词信息，句子信息，长语音信息

Multi-resolution in Speech and Language

- 大部分主流的时序模型
  - 隐式的考虑了不同分辨率的信息 通过直接喂高分信息
- 一些具体的语音生成任务也考虑了多分辨率的问题
  - 语音增强、分离 (Multi-resolution convolutional networks in Wan et al. 2023; Multi-resolution frequency encoder/decodr in Shi et al. 2023)
  - 语音合成 (Multi-scale discriminator in Kong et al. 2020)

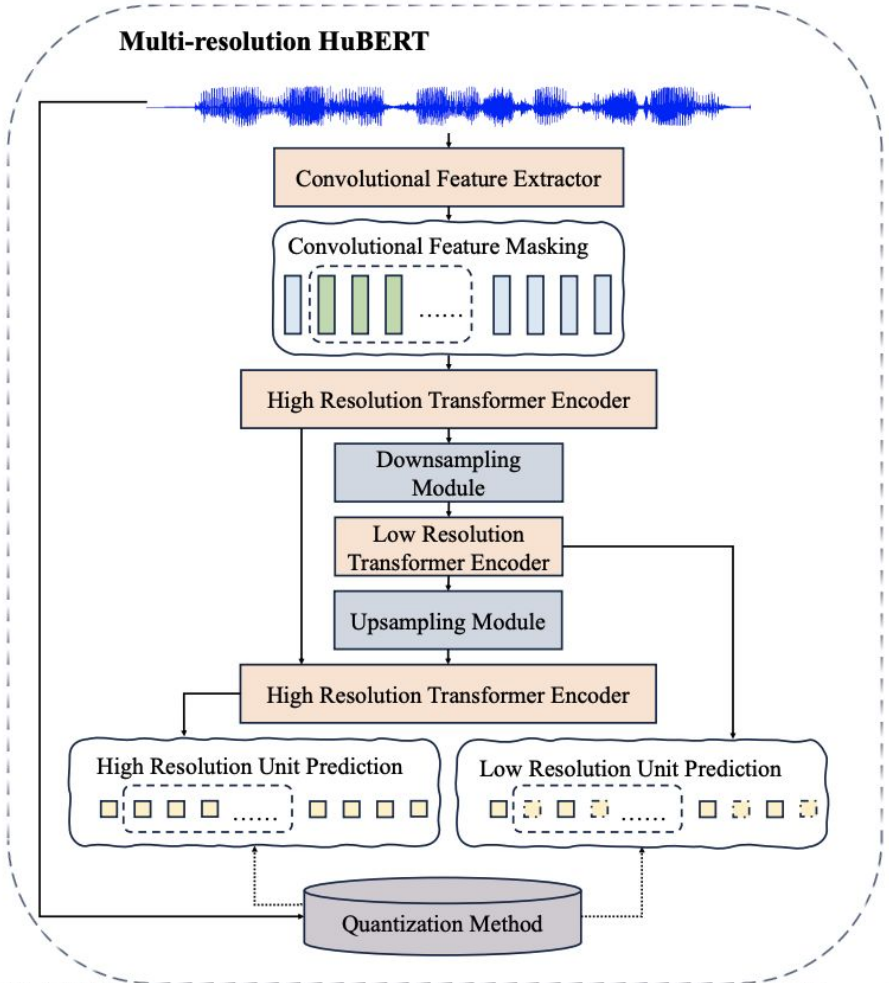
- 语音中，我们输入20ms或者10ms shift的特征帧
- 文本中我们直接喂字或者字组（BPE token etc）

More than HuBERT with multi-resolution?

- 考虑到不同resolutio对HuBERT的影响
  - 我们是否可以在SSL pre-training的时候就考虑到这个问题？
  - 可否把整个multi-resolution的逻辑放到一个framework中呢？

Multi-resolution HuBERT

- Residual connection
- Downsampling and Upsampling
  - Originally used conv+deconv
  - Then found simple skip-subsampling and linear-upsampling works the best
    - 1 2 3 4 -> 1 3
    - 1 3 –linear-> [1, 2] [3, 4] –reshape-> 1 2 3 4



Simple concept, what about performance?

10h librispeech ASR training (with fine-tuning) (no LM rescoring)

Model	Param	Dev-clean	Test-clean	Inference Speech (token per second)
HuBERT-base	95M	9.62	9.71	5871
HuBERT-base+ (one more iteration)	95M	9.51	9.72	5833
MR-HuBERT-base (mono-base)	97M	8.51	8.46	6332

Performance significantly better!

Simple concept, what about performance?

10h librispeech ASR training (with fine-tuning) (no LM rescoring)

Model	Param	Dev-clean	Test-clean	Inference Speech (token per second)
HuBERT-base	95M	9.62	9.71	5871
HuBERT-base+ (one more iteration)	95M	9.51	9.72	5833
MR-HuBERT-base (mono-base)	97M	8.51	8.46	6332

Speed significantly faster!

Simple concept, what about performance?

10h librispeech ASR training (with fine-tuning) (no LM rescoring)

Model	Param	Dev-clean	Test-clean	Inference Speech (token per second)
HuBERT-base	95M	9.62	9.71	5871
HuBERT-base+ (one more iteration)	95M	9.51	9.72	5833
MR-HuBERT-small (3+3+3)	76M	9.13	9.36	7096
MR-HuBERT-base (mono-base)	97M	8.51	8.46	6332

We can also reduce the parameter size, with better performance and faster speed

Other scenarios

- SUPERB benchmark (understanding tasks and enhancement tasks)

Model	Understanding							Enhancement		
	PR(↓)	ASR(↓)	IC(↑)	KS(↑)	SF-F1(↑)	SF-CER(↓)	ST(↑)	SE-STOI(↑)	SE-PESQ(↑)	SS(↑)
HuBERT-base	5.40	6.42	98.34	96.30	88.53	25.20	15.53	**0.94**	2.58	9.36
HuBERT-base^{+}	4.56	6.34	98.39	96.46	89.12	23.10	16.33	0.93	2.55	9.72
HuBERT-large	3.54	3.62	98.76	95.29	89.81	21.76	20.01	**0.94**	2.64	10.45
HuBERT-large*	3.59	**3.53**	98.73	97.70	89.88	22.51	20.02	**0.94**	2.65	10.61
mono-base	4.16	5.76	98.68	96.49	88.96	23.59	16.94	**0.94**	2.55	9.92
mono-large	**3.15**	3.78	**98.76**	**97.76**	**90.57**	**20.60**	**21.05**	**0.94**	**2.67**	**10.97**

Other scenarios

- ML-SUPERB benchmark (on multilingual tasks)

SSL	Monolingual ASR	Multilingual ASR		LID	Multilingual ASR + LID			SUPERB$_s$(↑)
		Normal	Few-shot	Normal	Normal		Few-shot	
	CER/PER(↓)	CER(↓)	CER(↓)	ACC(↑)	ACC(↑)	CER(↓)	CER(↓)	
HuBERT-base	42.8 / 35.3	39.8 / 31.4	44.5 / 42.7	61.2 / 86.1	**71.5** / 86.0	39.2 / 30.9	43.8 / 41.8	831.9 / 884.9
HuBERT-base$^+$	42.9 / 35.3	41.5 / 31.2	45.8 / 42.8	63.8 / 81.9	70.1 / 85.8	39.6 / 31.3	44.6 / 40.7	819.1 / 875.8
HuBERT-large	**38.2** / 32.2	44.4 / 37.7	48.2 / 43.5	46.5 / 64.1	55.4 / 77.7	45.6 / 35.1	49.3 / 42.2	678.7 / 783.6
HuBERT-large*	41.2 / 32.6	42.8 / 32.8	45.6 / 42.5	42.3 / 58.9	59.2 / 84.7	42.3 / 29.8	44.1 / 41.4	704.5 / 817.6
mHuBERT-base	41.0 / 33.0	40.5 / 33.4	45.6 / 43.6	52.4 / 72.5	46.6 / 70.9	36.8 / 29.7	44.2 / 43.1	746.2 / 812.7
mHuBERT-base*	40.1 / 32.3	36.3 / **27.3**	**38.6** / 39.0	**64.0** / 82.0	70.4 / 84.6	35.4 / **27.1**	**39.0** / 37.0	950.8 / 964.5
mono-base	42.8 / 34.6	40.2 / 30.6	45.0 / 42.2	67.2 / **86.3**	68.7 / **86.9**	40.3 / 30.6	44.1 / 41.6	843.5 / 899.9
mono-large	40.5 / 32.0	38.9 / 29.4	42.7 / 40.5	45.1 / 75.4	67.6 / 85.9	39.0 / 29.7	43.8 / 40.8	785.2 / 905.4
multi-base	38.3 / **30.6**	**34.1** / 27.5	39.6 / **38.9**	**64.0** / 85.1	69.9 / 84.4	**34.4** / 28.0	40.9 / **36.6**	**957.2 / 986.8**

Other scenarios

SSL	Monolingual ASR	Multilingual ASR		LID	Multilingual ASR + LID			SUPERB$_s$
		Normal	Few-shot	Normal	Normal		Few-shot	
	CER/PER	CER	CER	ACC	ACC	CER	CER	
ML-SUPERB Benchmark (Shi et al., 2023a)								
FBank	72.1 / 63.7	62.4 / 59.3	58.3 / 57.4	11.1 / 9.3	35.9 / 43.5	62.0 / 58.6	58.9 / 58.1	0 / 0
wav2vec2-base	44.2 / 35.9	43.0 / 35.5	45.7 / 44.3	54.4 / 80.8	66.9 / 83.6	40.6 / 32.1	44.2 / 42.6	755.2 / 827.2
wav2vec2-large	42.0 / 35.4	42.6 / 35.7	45.8 / 43.9	30.9 / 8.0	54.6 / 78.2	45.5 / 34.7	50.3 / 42.2	598.3 / 586.9
robust-wav2vec2-large	44.4 / 35.7	40.1 / 31.1	45.4 / 42.2	50.8 / 72.1	33.1 / 62.9	38.6 / 33.7	44.9 / 46.0	680.3 / 768.6
wav2vec2-base-23	49.2 / 35.1	37.7 / 32.0	43.4 / 42.2	58.7 / 71.9	45.1 / 66.3	37.2 / 30.9	44.3 / 43.0	735.7 / 798.0
wav2vec2-large-23	42.0 / 34.2	42.1 / 35.3	44.3 / 42.4	1.1 / 64.2	21.8 / 49.7	43.4 / 35.2	46.1 / 43.1	433.8 / 724.9
XLSR-53	49.5 / 34.9	33.9 / 26.9	43.6 / 40.6	6.6 / 87.1	45.6 / 76.9	33.4 / 28.6	43.2 / 44.6	528.8 / 894.0
XLSR-128	39.7 / **30.6**	**29.2** / **22.0**	40.9 / 39.3	**66.9** / **87.9**	55.6 / 85.6	**28.4** / **22.9**	42.1 / 42.4	947.5 / **996.0**
HuBERT-base	42.8 / 35.3	39.8 / 31.4	44.5 / 42.7	61.2 / 86.1	71.5 / 86.0	39.2 / 30.9	43.8 / 41.8	831.9 / 884.9
HuBERT-large	**38.2** / 32.2	44.4 / 37.7	48.2 / 43.5	46.5 / 64.1	55.4 / 77.7	45.6 / 35.1	49.3 / 42.2	678.7 / 783.6
HuBERT-base-cmn	43.1 / 35.3	40.8 / 31.4	45.4 / 42.7	49.3 / 86.1	**75.1** / **86.1**	37.7 / 30.9	43.5 / 41.8	779.0 / 810.2
HuBERT-large-cmn	39.4 / 32.2	42.6 / 37.7	45.8 / 43.5	39.5 / 64.1	66.4 / 77.7	41.9 / 35.1	45.2 / 42.2	715.4 / 783.6
mHuBERT-base	41.0 / 33.0	40.5 / 33.4	45.6 / 43.6	52.4 / 72.5	46.6 / 70.9	36.8 / 29.7	44.2 / 43.1	746.2 / 812.7
Additional Baseline (Section 4.1)								
HuBERT-base$^+$	42.9 / 35.3	41.5 / 31.2	45.8 / 42.8	63.8 / 81.9	70.1 / 85.8	39.6 / 31.3	44.6 / 40.7	819.1 / 875.8
HuBERT-large*	41.2 / 32.6	42.8 / 32.8	45.6 / 42.5	42.3 / 58.9	59.2 / 84.7	42.3 / 29.8	44.1 / 41.4	704.5 / 817.6
mHuBERT-base*	40.1 / 32.3	36.3 / 27.3	**38.6** / 39.0	64.0 / 82.0	70.4 / 84.6	35.4 / 27.1	**39.0** / 37.0	950.8 / 964.5
Proposed Method (Section 4.1)								
mono-base	42.8 / 34.6	40.2 / 30.6	45.0 / 42.2	67.2 / 86.3	68.7 / 86.9	40.3 / 30.6	44.1 / 41.6	843.5 / 899.9
mono-large	40.5 / 32.0	38.9 / 29.4	42.7 / 40.5	45.1 / 75.4	67.6 / 85.9	39.0 / 29.7	43.8 / 40.8	785.2 / 905.4
multi-base	38.3 / **30.6**	34.1 / 27.5	39.6 / **38.9**	64.0 / 85.1	69.9 / 84.4	34.4 / 28.0	40.9 / **36.6**	**957.2** / 986.8

Compared to XLSR-128, multi-base can reach comparable performance (slightly better in 10min, slightly worse in 1h), with

- **⅓ model size**
- **¼ pre-trained data**
- **⅙ language coverage**

Potential Extensions and Future Works

- Integration with Other Frameworks
  - Wav2vec2, WavLM, W2V-BERT, Data2vec
- Diverse Resolution
  - Higher resolution for detailed acoustic analysis
  - Lower resolution for environmental information
- Richer Representation
  - Semantic vs. Acoustics
- Speech Disentanglement
  - Adversarial elements for speech disentanglement

Discrete Speech Representation

- 从去年的VALL-E开始，逐渐出现大量关于离散语音表征的工作
  - (GSLM & pGSLM) https://aclanthology.org/2021.tacl-1.79/ and https://aclanthology.org/2022.acl-long.593/
  - (Vall-E）https://arxiv.org/abs/2301.02111
  - (AudioLM) https://ieeexplore.ieee.org/abstract/document/10158503
  - (AudioGen) https://openreview.net/forum?id=CYK7RfcOzQ4
  - (MusicLM) https://arxiv.org/abs/2301.11325
  - (SPAER-TTS) https://arxiv.org/abs/2302.03540
  - (Vall-E X) https://arxiv.org/abs/2303.03926
  - (Soundstorm) https://arxiv.org/abs/2305.09636
  - (NaturalSpeech2) https://arxiv.org/abs/2304.09116
  - (VioLA) https://arxiv.org/abs/2305.16107
  - (Make-A-Voice) https://arxiv.org/abs/2305.19269
  - (MusicGEN) https://arxiv.org/abs/2306.05284
  - (AudioPaLM) https://arxiv.org/abs/2306.12925
  - (VoiceBox) https://arxiv.org/abs/2306.15687
  - (SpeechX) https://arxiv.org/abs/2308.06873
  - (VoxtLM）https://arxiv.org/abs/2309.07937

Discrete Speech Representation

- 不直接从语音特征中提取连续表征，而是离散化的表征的过程

- 离散化表征仍然是在“降频”的主旨下进行，是更进一步的压缩

Benefits from discrete representation

- 离散语音表征的好处？
- 更简单的语音表示
  - 更快的训练速度
  - 更大的训练规模
  - 更有效率的语音传播（low-bitrate）
  - 更容易预测（regression vs classification）
- 更好的和大语言模型结合
  - 更好在模型层面和大语言模型结合
  - 更好的应用大语言模型的一些训练和应用方法（e.g., prompt tuning, decoder-only architecture）

How to extract?

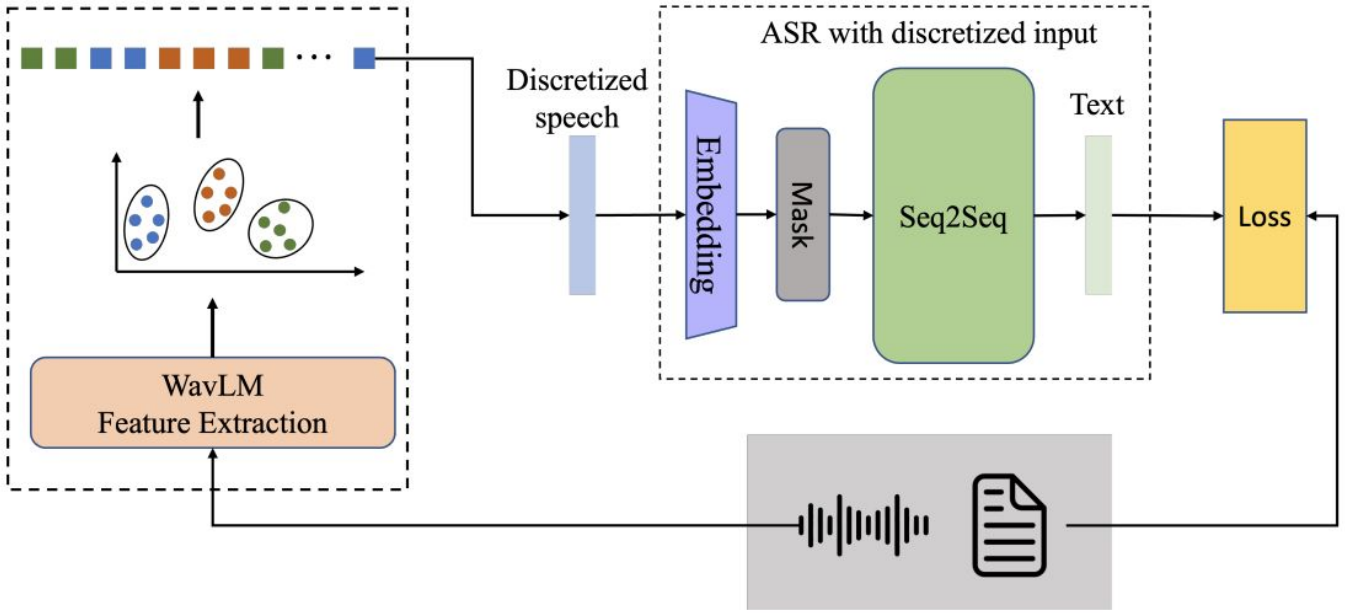
- 目前的主流方法：
- HuBERT-style：
  - 使用自监督语音特征（SSL）进行聚类（clustering）
  - 利用聚类获得的类别作为离散的语音表征
  - 一般是single-stream（一段语音对应一个离散序列）
- Neural-codec style
  - 使用语音重建任务进行学习
  - 利用量化层来达到中间信息的重构
  - 一般是multi-stream（一段语音对应多个并列的离散序列）

Application of discrete speech representation

- 输入信号
  - 语音识别的输入（espnet asr2）
  - 其他语音理解任务的输入（espnet asr2）
  - 语音重建的输入（github parallel-wavegan）
- 输出信号
  - 语音到语音的翻译输出（espnet s2st）
  - 作为辅助学习手段（espnet-uasr、HuBERT-style SSL）
  - 语音合成输出（espnet tts2）
  - 歌唱合成输出（espnet-muskits (espnet-svs2)）

Input for Speech Recognition

- Directly use discrete speech representation as the input to the speech recognition ("machine translation") system



(Chang et al. 2023, https://arxiv.org/pdf/2305.18108.pdf)

Input for Speech Recognition

- Performance-wise -> **better** than spectrogram, **worse** than original SSL features

Feature	dev		test	
	clean	other	clean	other
ASR-FBank	2.5	6.3	2.6	6.2
ASR-WavLM last-layer	1.9	3.9	2.0	4.0
Discrete tokens (dedup)	2.9	6.8	3.0	7.0
Discrete tokens (dedup+1D_Conv)	2.9	6.8	3.1	6.9

Librispeech 960hour

Input for Speech Recognition

- Benefits?
  - A balance between speed, storage, and performance

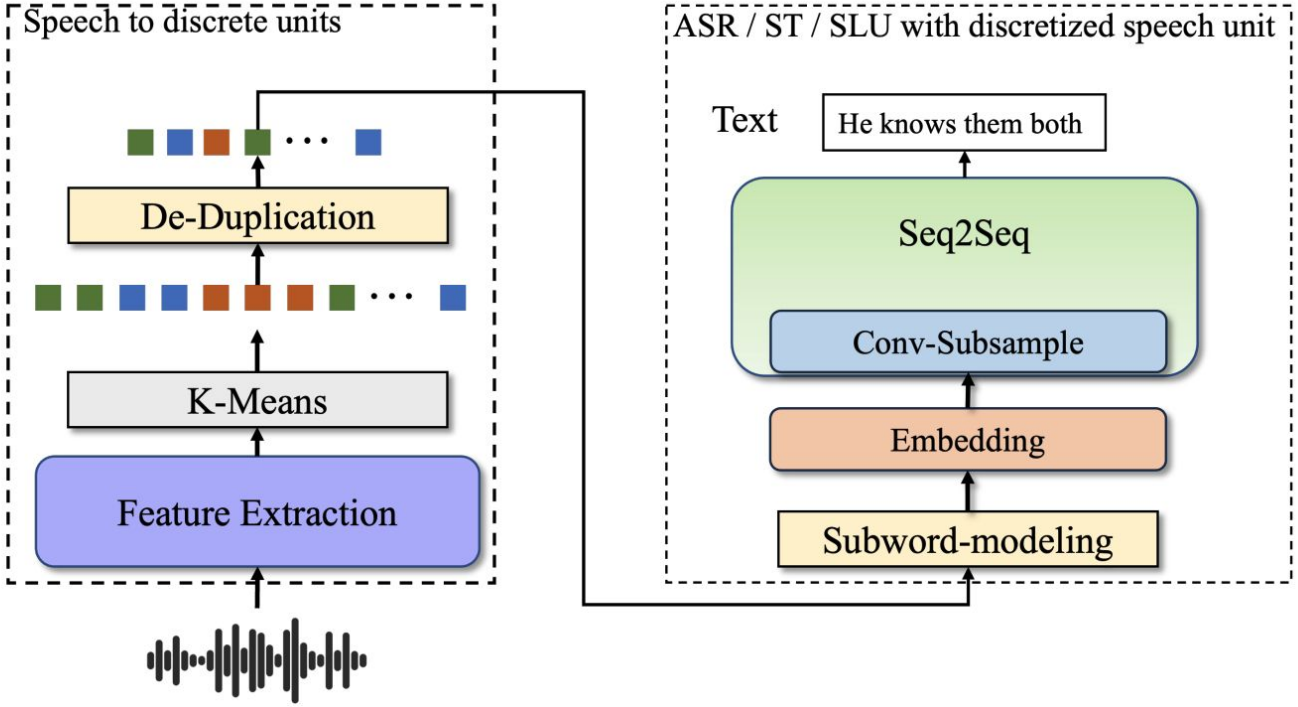
Length Reduction	Avg. Input Length (train / dev)	dev		test	
		clean	other	clean	other
vq-wav2vec [26] + token-BERT + 4-gram LM	- / -	4.0	10.9	4.5	12.1
WavLM-token	633.8 / 358.1	5.6	12.5	5.9	12.8
+SW_6000	349.9 / 200.1	5.5	11.8	5.8	12.0
WavLM-token dedup	468.7 / 270.9	6.4	12.7	6.4	13.2
+SW_6000	**249.4 / 144.5**	5.8	12.3	6.0	12.5

Huge length reduction □ **faster** training/decoding speed, **less** storage constraints

Input for Speech Recognition

- Benefits?
  - A balance between speed, storage, and performance
    - 60G 960-hour audio -> 0.3G 172M tokens
    - Model sequence length (key bottleneck in sequence-to-sequence modeling)
      - Previous setting: around 600 frames per sequence
      - Current setting: around 250 frames per sequence
    - Training speed on a single GPU (V100-16G)
      - Previous training -> 7 days
      - Current training -> 0.5 day

Input for Other Speech Tasks



(Chang et al. 2023, https://arxiv.org/pdf/2309.15800.pdf)

Input for Other Speech Tasks

BLEU score with MuST-C (Speech Translation task))

Dataset	FBank	Discrete tokens	SSL (top line)
MuST-C En-De	26.7	**28.6**	29.7
MuST-C En-Es	31.2	**33.0**	33.7
MuST-C En-Fr	37.1	**38.7**	40.2

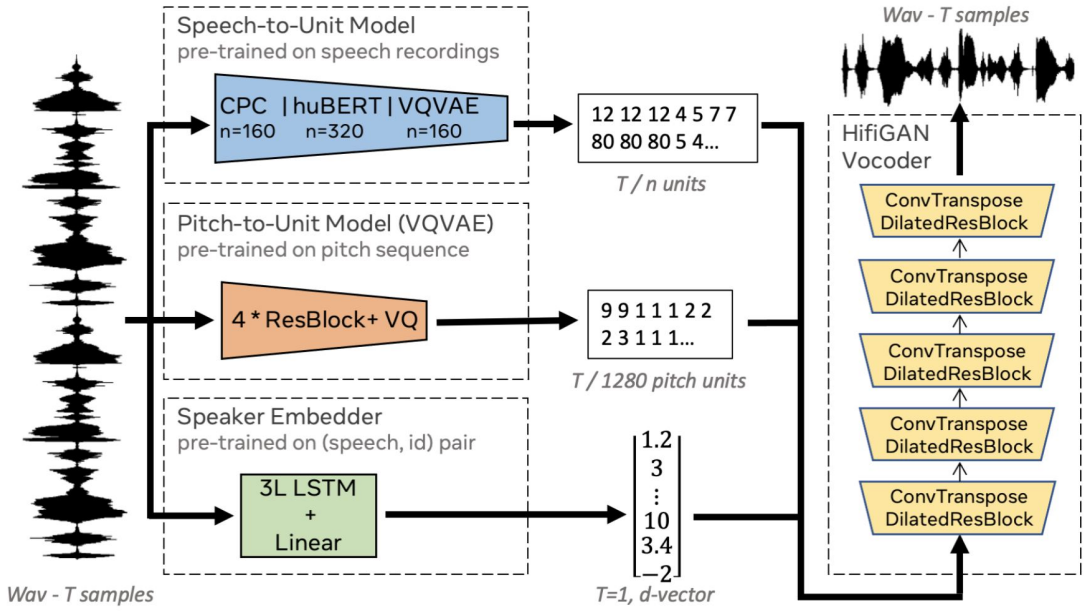
Librispeech (960h) performance with different discrete representations

Dataset	SSL	OWSM	EnCodec
LibriSpeech	**2.2 / 4.5 / 2.4 / 4.6**	3.9 / 10.1 / 4.2 / 10.2	3.2 / 8.5 / 3.0 / 8.5

(Chang et al. 2023, https://arxiv.org/pdf/2309.15800.pdf)

Input for Speech Reconstruction

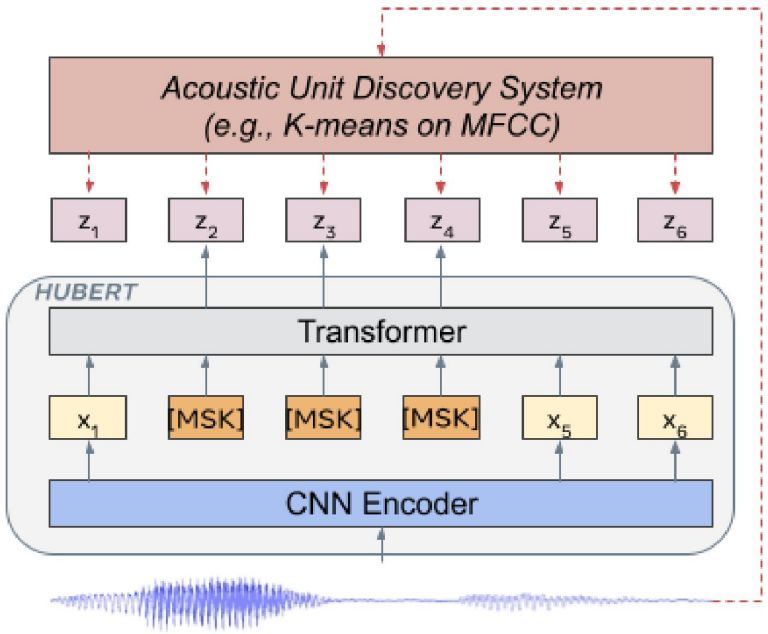
- First usage of SSL-based discrete resynthesis



(Polyak et al. 2021, https://arxiv.org/pdf/2104.00355.pdf)

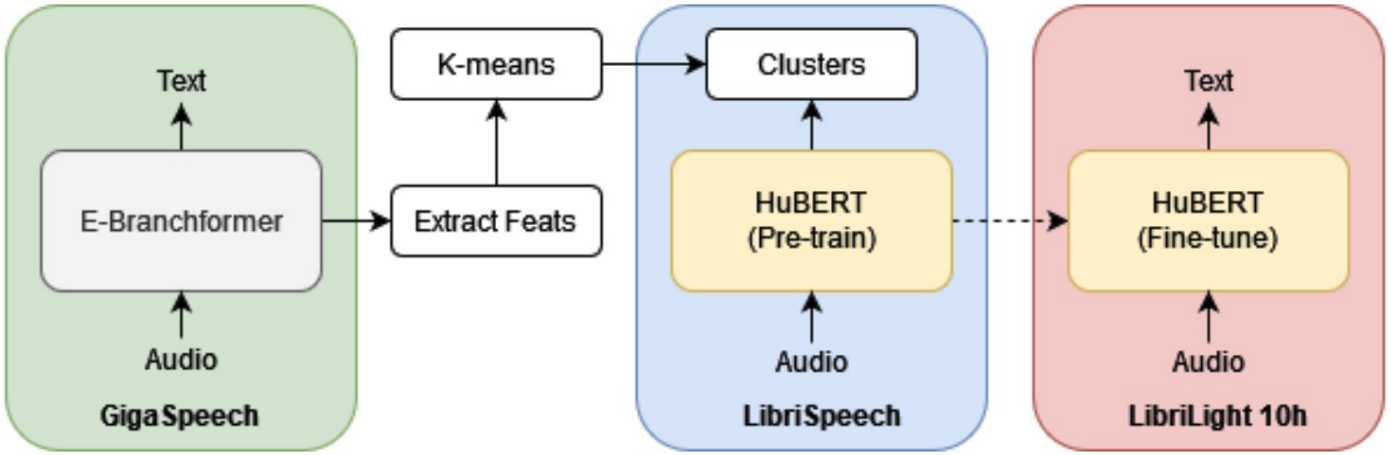
Support for Representation Learning

- HuBERT-style training
  - The start of discrete-representation usage
- Iterative refining the SSL pre-trained modeling



Hsu et al. 2021, https://arxiv.org/pdf/2106.07447.pdf

Use Supervised Discrete Tokens to Help HuBERT pre-training

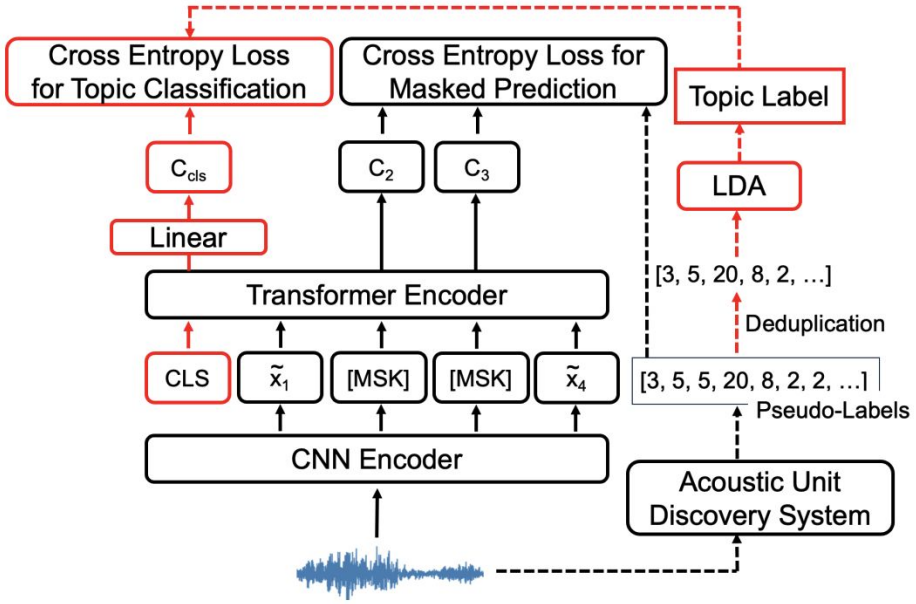


model	dev		test	
	clean	other	clean	other
Prior Work (Iteration 1)				
HuBERT Base[25]	10.9	17.5	10.9	17.8
Iteration 0				
HuBERT Base	15.5	23.9	15.9	24.1
GS-8	12.2	19.4	12.3	19.9
GS-16	**9.6**	**15.7**	**9.7**	**16.3**
Iteration 1				
HuBERT Base	10.4	17.4	10.5	17.6
GS-16 (Base)	10.2	16.0	10.3	16.4
GS-16 (Large)	**8.4**	**13.8**	**8.2**	**13.8**
Iteration 2				
HuBERT Large	9.5	14.9	9.6	14.9

Chen et al. 2023, https://arxiv.org/pdf/2306.06672.pdf

Understand and re-use

- 辅助HuBERT-style training □ 从离散序列角度建模

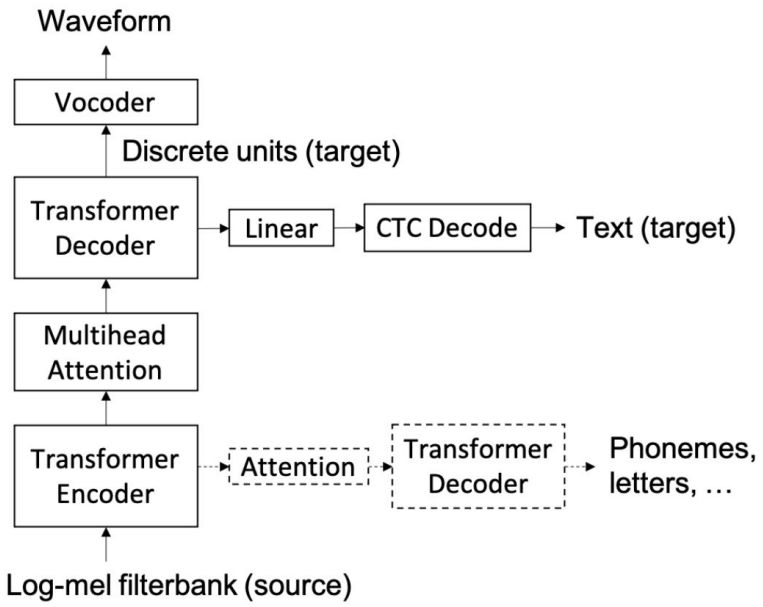


Training Data	Model	PR(↓)	ER(↑)	IC(↑)	SID(↑)	SD(↓)	SF(↑)	KS(↑)	SE(↑)
LS-100h	HuBERT	13.89	60.24	88.72	60.48	8.86	80.62	94.22	2.48
	HuBERTopic	**12.97**	**60.92**	**90.64**	**61.82**	**8.59**	**81.05**	**94.87**	**2.50**
LS-960h	HuBERT	5.04	**64.12**	97.57	**79.34**	7.49	88.61	**96.04**	2.53
	HuBERTopic (30-30)	**4.83**	64.10	97.68	78.98	**6.93**	88.76	95.26	2.53
	HuBERTopic (150-1000)	4.84	63.61	**98.10**	79.21	7.07	**88.79**	95.81	**2.55**

Maekaku et al. 2023 https://arxiv.org/pdf/2310.03975.pdf

Outputs for Speech-to-speech Translation

- Initial successful application speech reconstruction:
  - Speech-to-speech translation

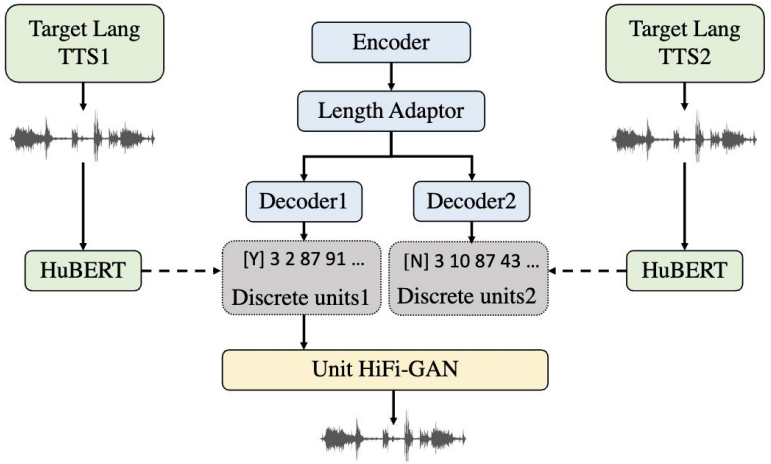


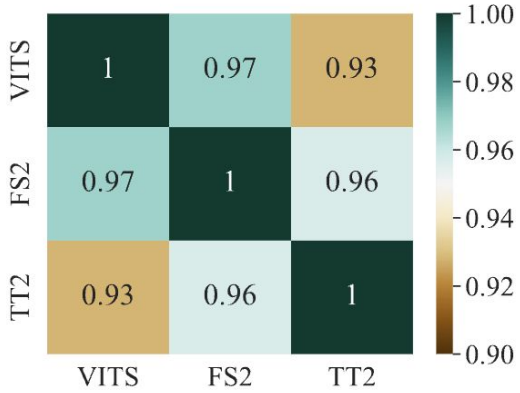
(Lee et al. 2022, https://arxiv.org/pdf/2107.05604.pdf)

Outputs for Speech-to-speech Translation

Noise in discrete representation -> Does discrete representation from SSL only have semantic information?

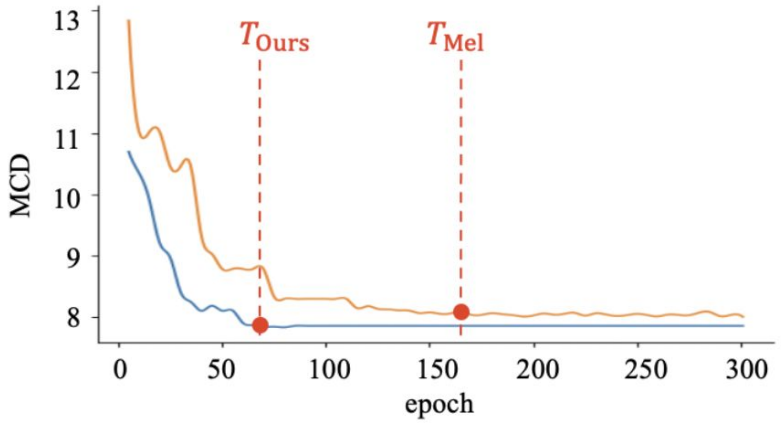
- Noise exists in discrete units (Lee et al. 2023)
- Several augmentation methods shown to be work well in discrete-based s2st
  - **More data helps** (Popuri et al. 2022)
  - **Multiple TTS targets helps** – (Shi et al. 2023)





So why going with SSL-based discrete units in such generation tasks?

- Similar benefits as in ASR-side
  - Training speed (convergence speed is **faster**)
    - **Better** loss indicator as well
  - Performance balance
    - **Less challenging** for understanding
    - **More challenging** for vocoder
    - (Similar debate happens around the use of SSL-based discrete tokens vs. neural codec-based discrete tokens)



Yan et al. 2023, https://arxiv.org/abs/2304.04596

Discrete Speech Challenge

https://www.wavlab.org/activities/2024/Interspeech2024-Discrete-Speech-Unit-Challenge/

- Interspeech 2024 challenge
- Three Tasks
  - ASR Task
    - Multilingual ASR Performance -> conducted on ML-SUPERB dataset
  - TTS Task
    - Single-speaker TTS (Acoustic+Vocoder) track on LJSpeech
      - 1-hour training data
      - Full training data
    - Vocoder track on Expresso
      - LowSR track
      - HighSR track
  - SVS Task
    - Single-singer SVS track on SVS

Initial Findings

- Leaderboard
  - https://huggingface.co/discrete-speech
- Discrete representation can be strong

Model	EN_LibriSpeech CER	ML_SUPERB CER	Bitrate	Rank
CTAIT/char	2.21	17.32	262.64	1
NTU_CMU/Fusion	1.85	11.65	665.13	2
CTAIT/SystemA	2.23	17.99	296.67	3
CTAIT/system_e_branchformer_fusion3	1.91	16.03	946.77	4
JHU/fuse_char	1.98	20.23	599.2	5

Initial Findings

- Leaderboard
  - https://huggingface.co/discrete-speech
- Discrete representation can be strong

Model	MCD Eval	Log F0 RMSE Eval	WER Eval	UTMOS Eval	Bitrate	Rank
UTokyo-sarulab/DiscreteTTS-v2.2	4.36	0.29	7.2	4.3597	313.8	1
UTokyo-sarulab/DiscreteTTS-v3	6.96	0.29	7.7	4.3344	277.6	2
UTokyo-sarulab/DiscreteTTS-v2	7.03	0.29	7.8	4.3537	318	3
VQTTS/VQTTS_2	7.7	0.29	6.8	4.4214	727.5	4
VQTTS/VQTTS	8.01	0.29	8	4.3638	727.5	5

Initial Findings

- Leaderboard
  - https://huggingface.co/discrete-speech
- Discrete representation can be strong

Model	MCD Eval	Log F0 RMSE Eval	MOS Eval	Bitrate	Rank
AI·M3-Music/TokSing3	7.43	0.17	3.76	2100	1
DOVSinger/DOVSinger	11.48	0.21	2.65	725.9	2
AI·M3-Music/TokSing1	7.72	0.18	3.57	2100	3
DOVSinger/DOVSinger2	10.75	0.19	2.63	725.9	4
RUC-AIM3-t/MutliResolutionSVS	7.72	0.19	3.4	875	5

Thank you for listening!