End-to-end Unsupervised ASR and Its Application

Presentor: Jiatong Shi

jiatongs@andrew.cmu.edu

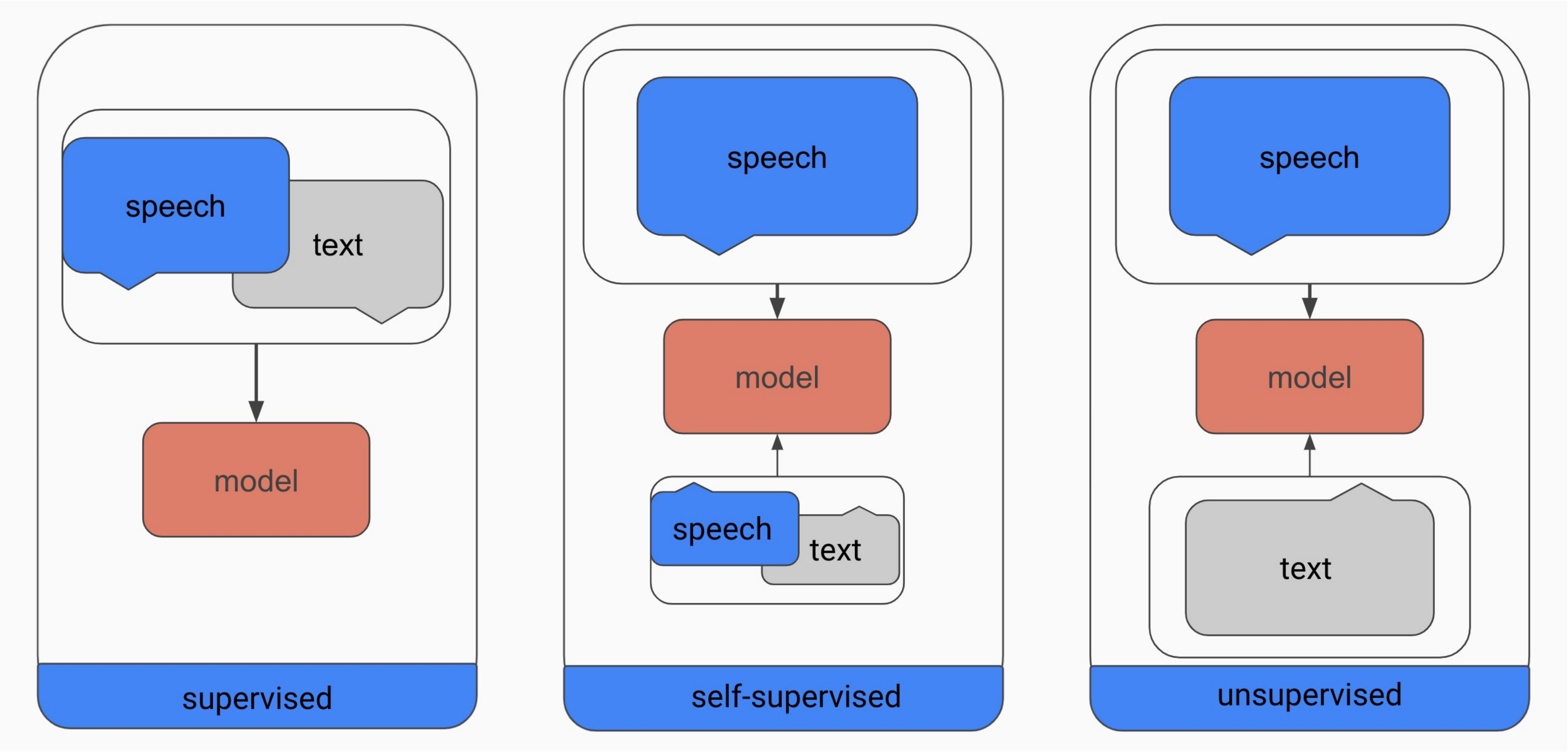
Part of Works from JSALT2022-Pre-training Team

Content

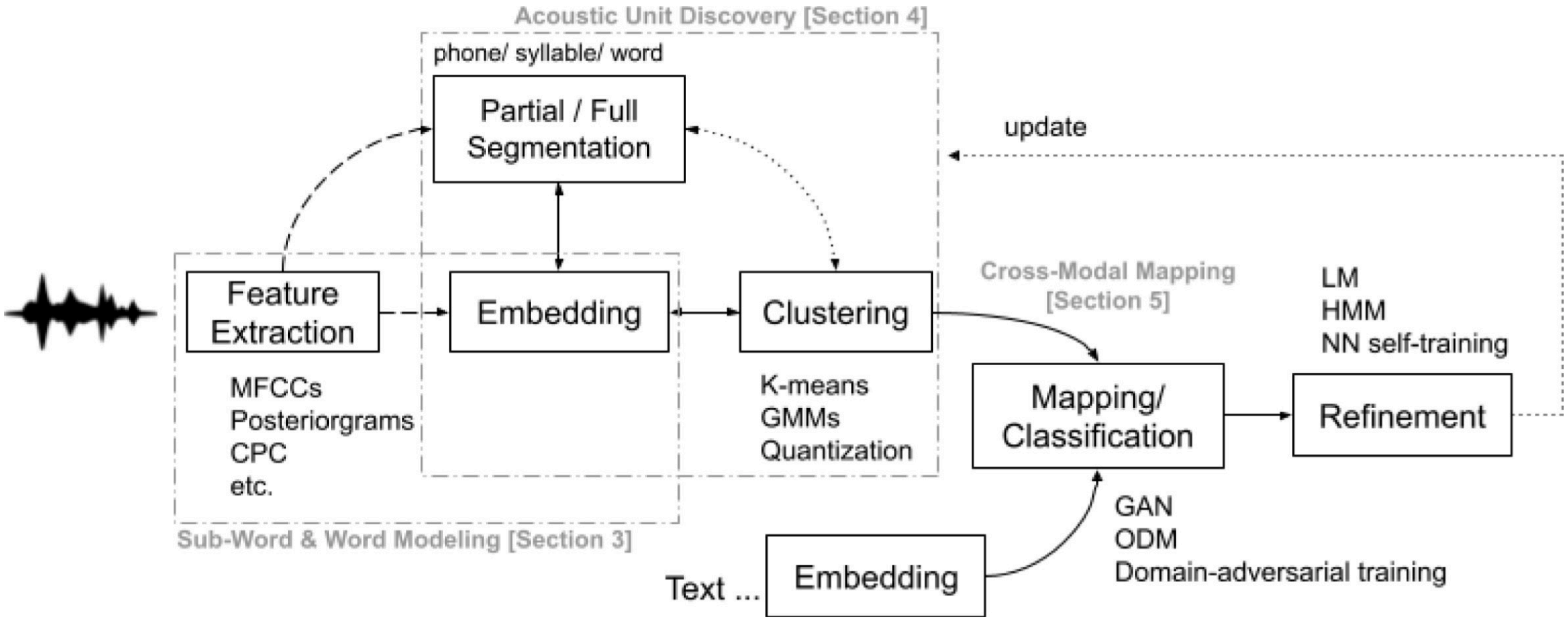
- Unsupervised ASR
  - A bit of the near history
  - Recent works with self-supervised learning
- Empirical results and challenges with unsupervised ASR
- Application of unsupervised ASR
- On-going work (EURO project)

Unsupervised ASR

- Supervision – Self-supervision - Unsupervision

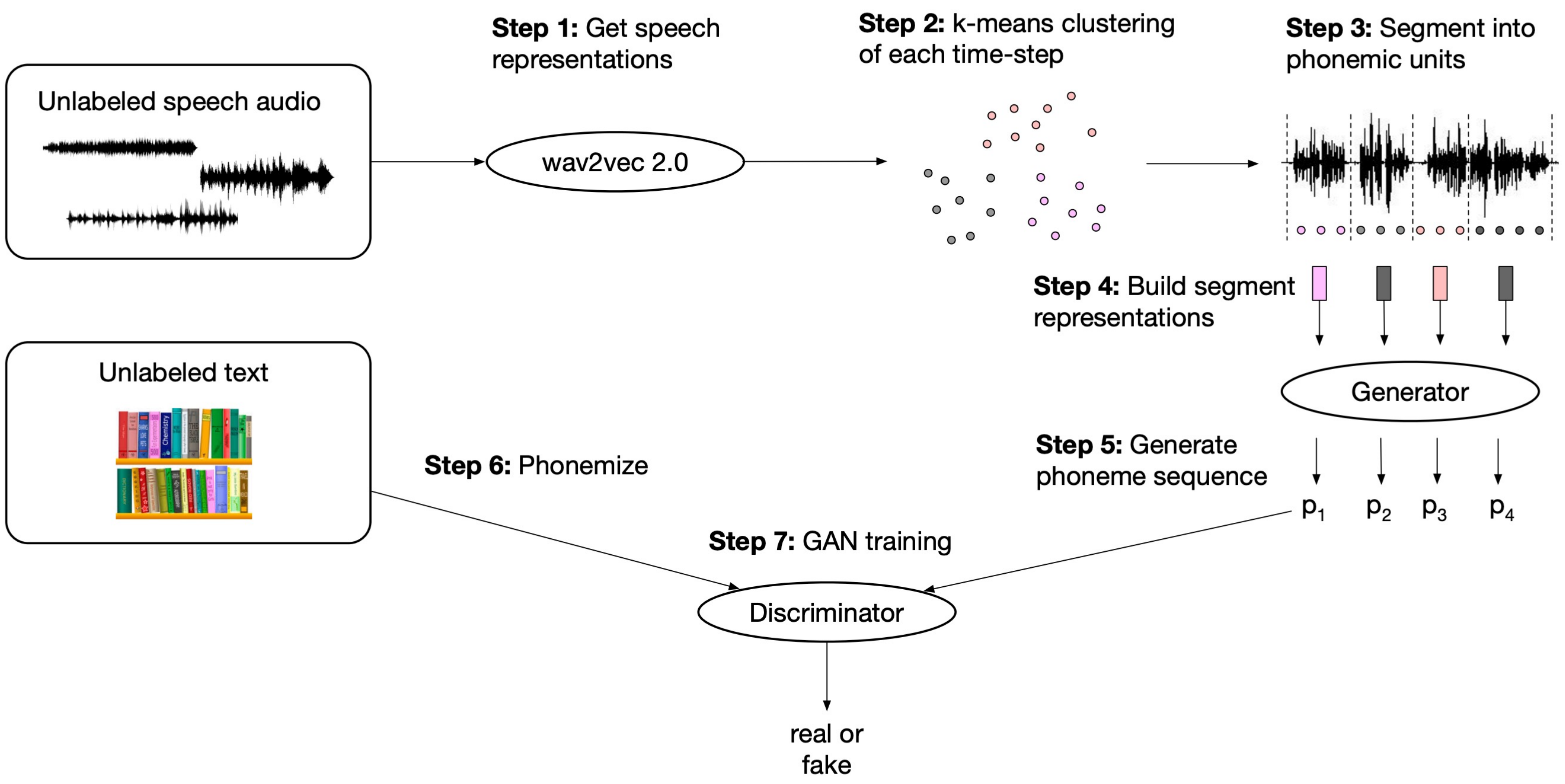


Unsupervised ASR



Aldarmaki, H., Ullah, A., Ram, S., & Zaki, N. (2022). Unsupervised automatic speech recognition: A review. *Speech Communication*.

Recent works with self-supervised model



Baevski, A., Hsu, W. N., Conneau, A., & Auli, M. (2021). Unsupervised speech recognition. *Advances in Neural Information Processing Systems*, *34*, 27826-27839.

Some Issues with the training scheme

- Instability
- Kmeans-segmentation is usually smaller than real phoneme segments
- Possibility to generate trival output

Solution to previous issues

- Gradient penalty loss → reduce drastic changes to discriminator
- Segmentation smoothness loss → encourage similar output between each segments
- Phoneme diversity loss → maximum entropy of prediction distribution to escape from trivial solutions

Wav2vec-u in the paper

Model	Unlabeled data	LM	dev clean	dev other	test clean	test other
960h - Supervised learning						
DeepSpeech 2 (Amodei et al., 2016)	-	5-gram	-	-	5.33	13.25
Fully Conv (Zeghidour et al., 2018)	-	ConvLM	3.08	9.94	3.26	10.47
TDNN+Kaldi (Xu et al., 2018)	-	4-gram	2.71	7.37	3.12	7.63
SpecAugment (Park et al., 2019)	-	-	-	-	2.8	6.8
SpecAugment (Park et al., 2019)	-	RNN	-	-	2.5	5.8
ContextNet (Han et al., 2020)	-	LSTM	1.9	3.9	1.9	4.1
Conformer (Gulati et al., 2020)	-	LSTM	2.1	4.3	1.9	3.9
960h - Self and semi-supervised learning						
Transf. + PL (Synnaeve et al., 2020)	LL-60k	CLM+Transf.	2.00	3.65	2.09	4.11
IPL (Xu et al., 2020b)	LL-60k	4-gram+Transf.	1.85	3.26	2.10	4.01
NST (Park et al., 2020)	LL-60k	LSTM	1.6	3.4	1.7	3.4
wav2vec 2.0 (Baevski et al., 2020c)	LL-60k	Transf.	1.6	3.0	1.8	3.3
wav2vec 2.0 + NST (Zhang et al., 2020b)	LL-60k	LSTM	1.3	2.6	1.4	2.6
Unsupervised learning						
wav2vec-U LARGE	LL-60k	4-gram	13.3	15.1	13.8	18.0

Wav2vec-u Robustness

Speech		Text					
Corpus	Hour	LibriLM	Wiki	NewsCrawl	ImageC	matched*	unmatched*
Full amount of speech							
Librispeech train	960	20.25	26.02	21.83	31.59	N/A	N/A
TED-LIUM v3	452	31.62	35.21	32.05	41.87	28.13	N/A
SwitchBoard	300	92.10	93.08	95.25	80.15	35.80	N/A
SwitchBoard-*w2v2-all*	300	44.38	94.12	43.44	72.10	32.34	N/A
Little amount of speech							
Librispeech train	9.6	22.51	29.03	24.65	105.00	-	-
TED-LIUM v3	10	36.01	88.26	33.52	85.92	29.13	32.44
SwitchBoard	10	95.86	-	-	-	95.13	93.48
SwitchBoard-*w2v2-all*	10	92.10	-	-	-	96.14	93.48

Lin, G. T., Hsu, C. J., Liu, D. R., Lee, H. Y., & Tsao, Y. (2022, May). Analyzing the robustness of unsupervised speech recognition. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 8202-8206). IEEE.

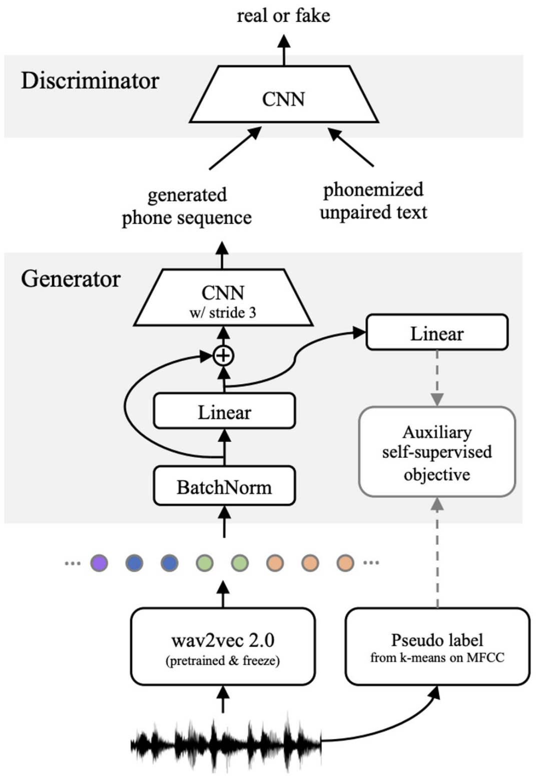
Our Wav2vec-u Experiments (Librispeech-100)

- Key findings:
  - Some factors are **crucial to good convergence**:
    - Layer for feature extraction -> 7, 14 are the best, layer combination cannot always converge
    - Network simplicity -> For example, adding two layer CNN would harm the results (+10-20 PER or not converge); Layer combination sometimes also hurt results (+10 PER)
  - Some factors are **good to tune**
    - Preprocessing parameters: cluster number fo Kmean pooling (K=128, K=256, K=64) and adjacent pooling
    - Training parameters: learning rates, weights for losses (for gradient penalty, phoneme diversity, and others)
- Our best PER results with wav2vec-u after tuning on Librispeech-100 is 24.1%

We want an end-to-end version…

We want an end-to-end version…

- Use an Batchnorm to replace the preprocessing
- Add K-means cluster objectives to stable the results
- Use CNN with stride to conduct downsampling



Liu, A. H., Hsu, W. N., Auli, M., & Baevski, A. (2022). Towards End-to-end Unsupervised Speech Recognition. *arXiv preprint arXiv:2204.02492*.

According to the paper:

	Pre-processing			Generator configuration				Result	
	Adjacent pooling	Cluster pooling	PCA reduction	Batch norm.	Linear proj.	Auxiliary loss	Stride	Freq. (Hz)	Average PER
wav2vec-U	✓	✓	✓	-	-	-	1	14	18.8 ± 0.9
step (i)	-	✓	✓	-	-	-	1	28	> 100
step (ii)	-	✓	✓	-	-	-	2	14	18.5 ± 0.6
step (iii)	-	-	✓	-	-	-	2	25	> 100
step (iv)	-	-	✓	-	-	-	3	16	19.0 ± 0.9
step (v)	-	-	-	-	-	-	3	16	> 100
step (vi)	-	-	-	✓	-	-	3	16	16.4 ± 0.7
step (vii)	-	-	-	✓	✓	-	3	16	15.9 ± 1.1
wav2vec-U 2.0	-	-	-	✓	✓	✓	3	16	$\mathbf{13.6} \pm \mathbf{0.9}$
input wav2vec 2.0 feature								50	-
ground truth phone sequence								$\sim$10	-

Our Wav2vec-U 2.0 Experiments (Librispeech100)

- Key factor for convergence:
  - Batchnorm with scaling factor + large batch size
    - Standard scaling factor 1.0 does not suitable for wav2vec2 feature (might different for other SSLs?) -> get +20PER or non-converge
    - Large batch size is necessary to get reasonable performances -> get non-converge results with small batch size like 10
  - Network simplicity
    - Similar to wav2vec-u 1.0, cannot hold very large network -> e.g., even additional layer of CNN
    - But can be mitigate / even get improvements by adding auxiliary losses (e.g., K-means clustering as prediction target)
- Our best system: 21.3 PER

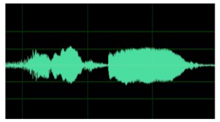
Still a long way to go…

Still a long way to go…

- But we are working towards a more stable system which could be easily trained
- Come back later!

After unsupervised ASR

Of course, use for ASR!



I'm not you

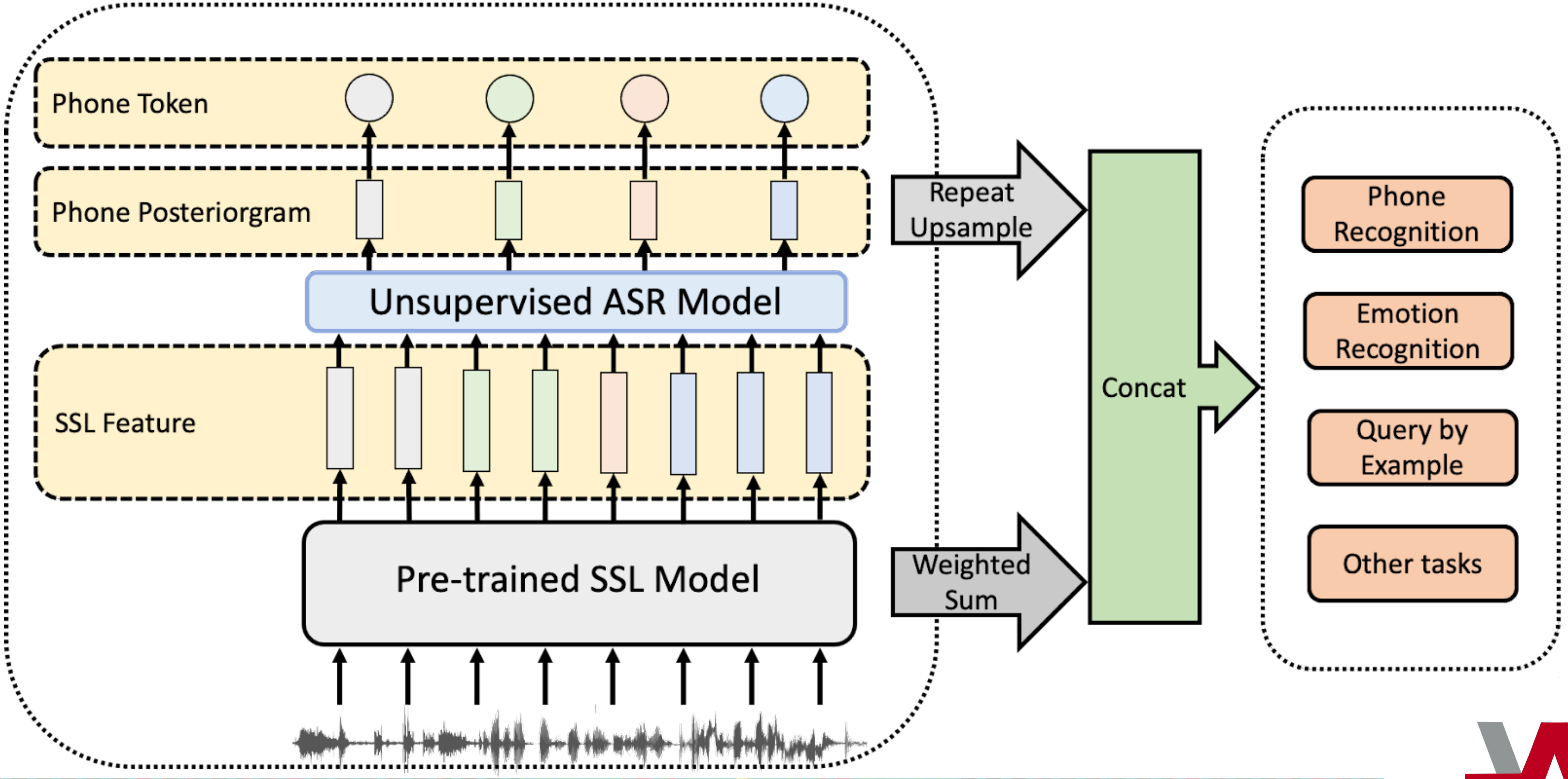
$S = \{s_n \in \mathbb{Z} | n = 1, ..., N\}$

$W = \{w_l \in \mathcal{V} | l = 1, ..., L\}$

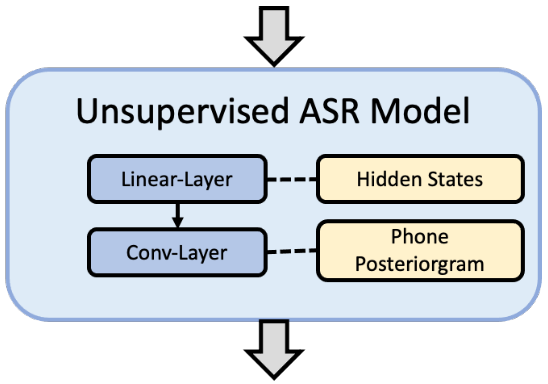
Next

- Use as **a self-supervised model**
  - No supervised data needed
- Use as **a segmenter**
  - Unsupervised phone segmentation
- Use as **a connector**
  - Connecting Speech SSL with Text SSL

Unsupervised ASR as an SSL Model



Unsupervised ASR as an SSL Model (SUPERB Public Leaderboard)



Upstream model		Param (M)	PR (↓)	PR-10h (↓)	ASR (↓)
Wav2vec2 (Large)		317.39	5.51	7.09	3.79
UASR	Hidden states	320.18	4.57	7.50	**3.76**
	Phone posteriorgram (PPG)	320.18	**4.53**	**6.26**	3.83
Hubert (Large)		316.61	3.53	5.15	3.56

- **Better** performances in **PR**
- **Similar** performances in **ASR**
- Still **cannot fill the gap** between **Hubert**

- Phone Recognition (PR) - SUPERB public set (Librispeech-100)
- Phoneme Recognition (PR-10h) - Librilight 10h split
- Automatic speech recognition (ASR) - SUPERB public set (Librispeech-100)

Unsupervised ASR as an SSL Model (SUPERB Hidden-set Leaderboard)

Models	Phone Recognition (↓)	Speech Recognition (↓)	Emotion Recognition (↑)	Query by Example (↓)	SUPERB Score (↑)
Wav2vec2	22.55	23.58	60.99	22.48	902
Hubert	18.22	**22.03**	64.84	33.05	959
UASR (PPG)	**17.22**	23.75	**65.11**	**21.99**	**962**

- **Better** performances in **PR**
- **Similar** performances in **ASR**
- **Outperforms Hubert** on several tasks

- SUPERB Score is a scaled score over 10 superb hidden-set tasks (from 0 - 1000). Calculation is based on https://superbbenchmark.org/challenge-slt2022/metrics
- All numbers are evaluated by SUPERB **hidden sets** (training & evaluation)

Unsupervised ASR as a segmenter

- It has a relative long background

Sequence Compression for SSL

Why sequence compression?

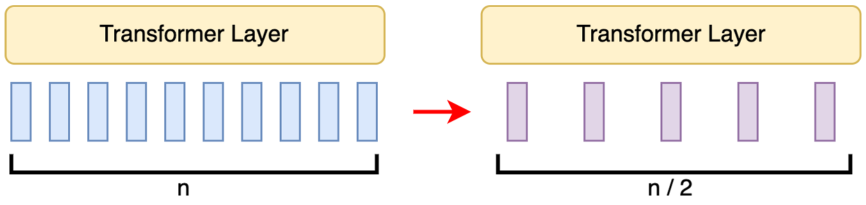
Computational cost reduction

- Faster pre-training/inference speed
- Less operations & memory usage

→ Impact of subsampling on different downstream tasks

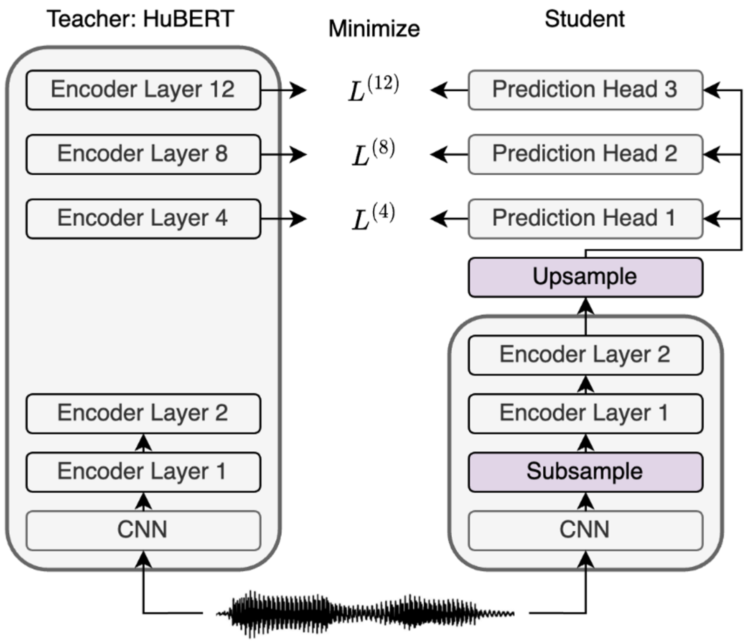
→ How much can the sequence be compressed?

Quadratic complexity

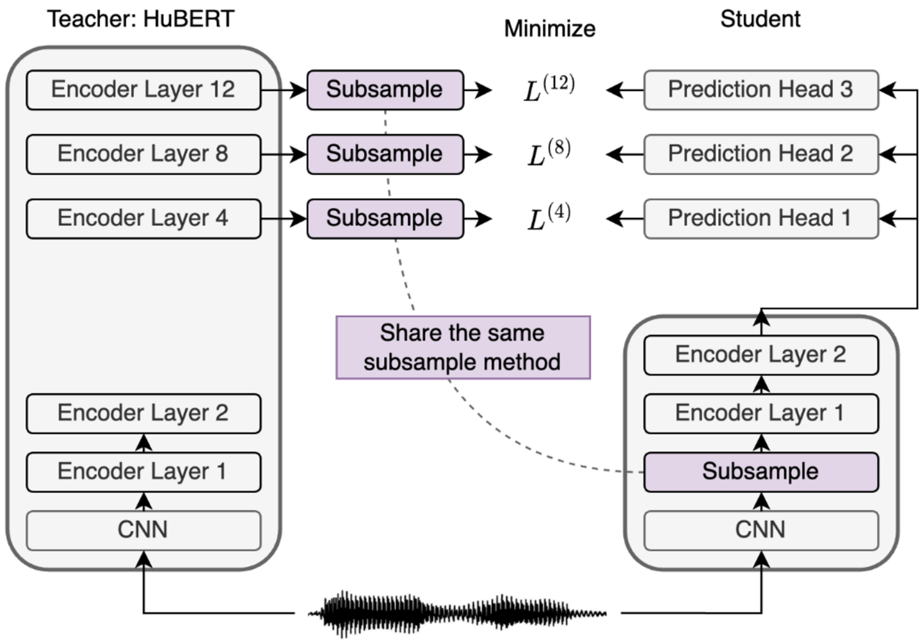


Framework for Sequence Compression

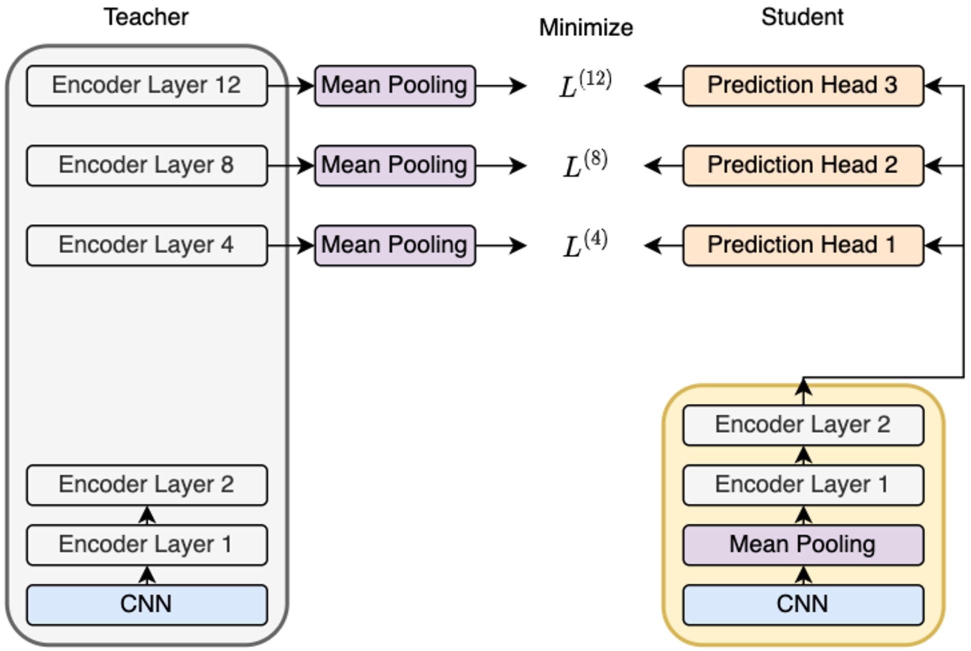
(a) With Upsample (target unchanged)



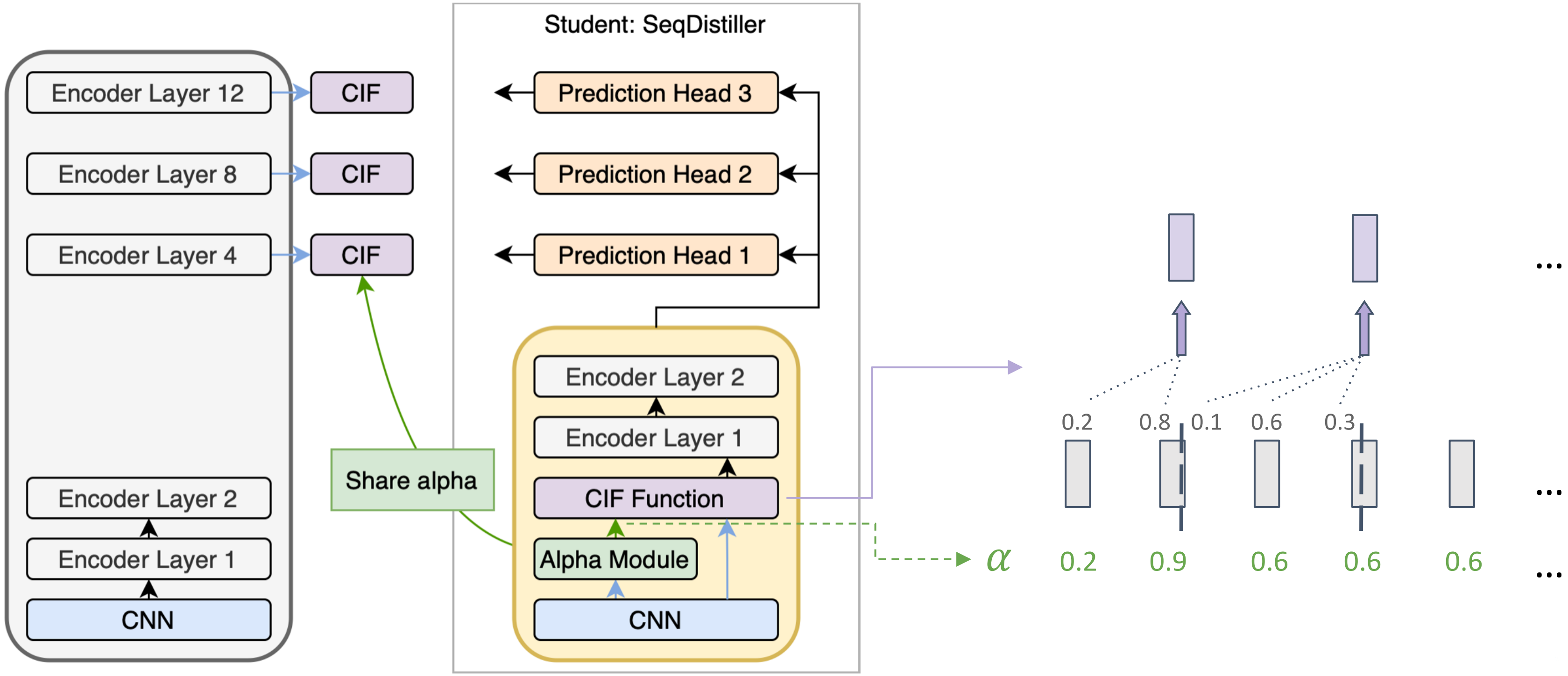
(b) Subsample Target



Choices for subsampling layers– Fixed-length



Choices for subsampling layers– Variable-length

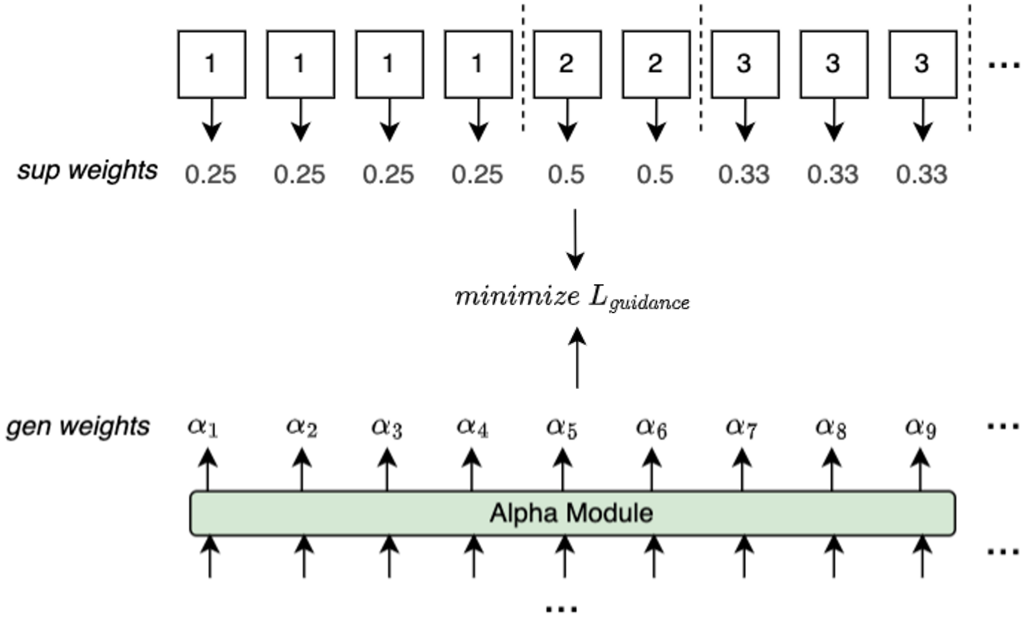


Dong, L., & Xu, B. (2020, May). Cif: Continuous integrate-and-fire for end-to-end speech recognition. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 6079-6083). IEEE.

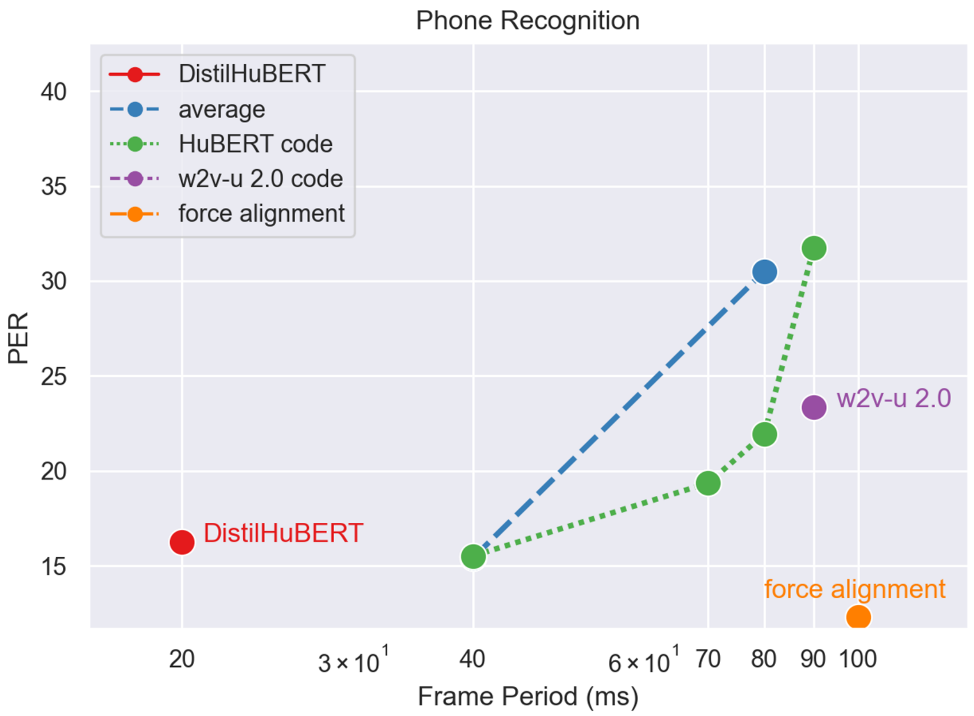
Segmentation guidance

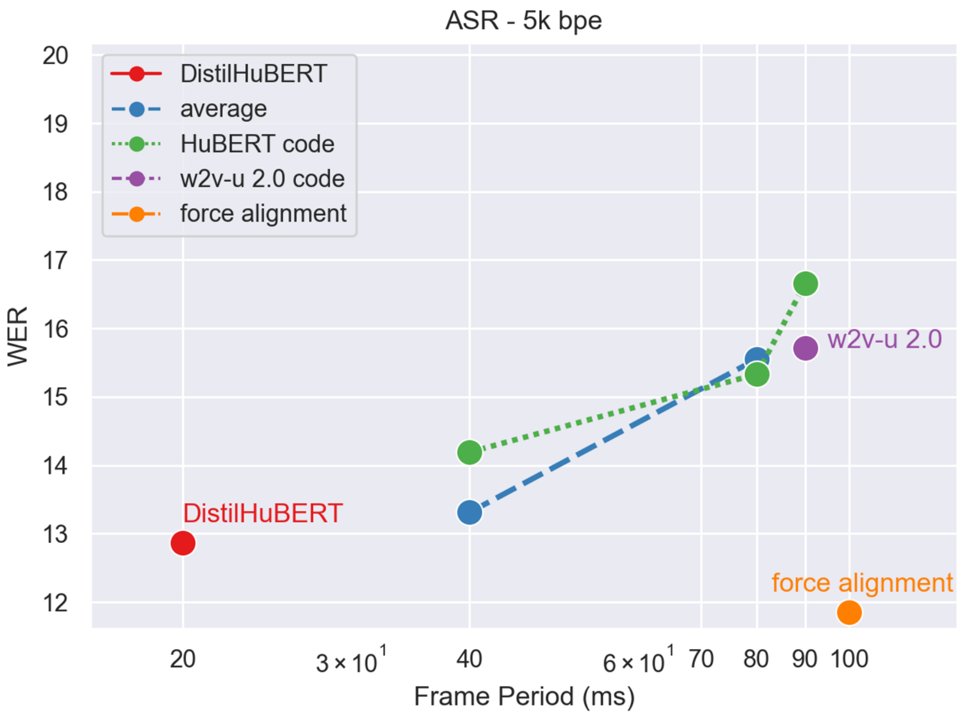
Unsupervised
- Repetition in HuBERT codes
- Repetition in wav2vec-U 2.0 codes

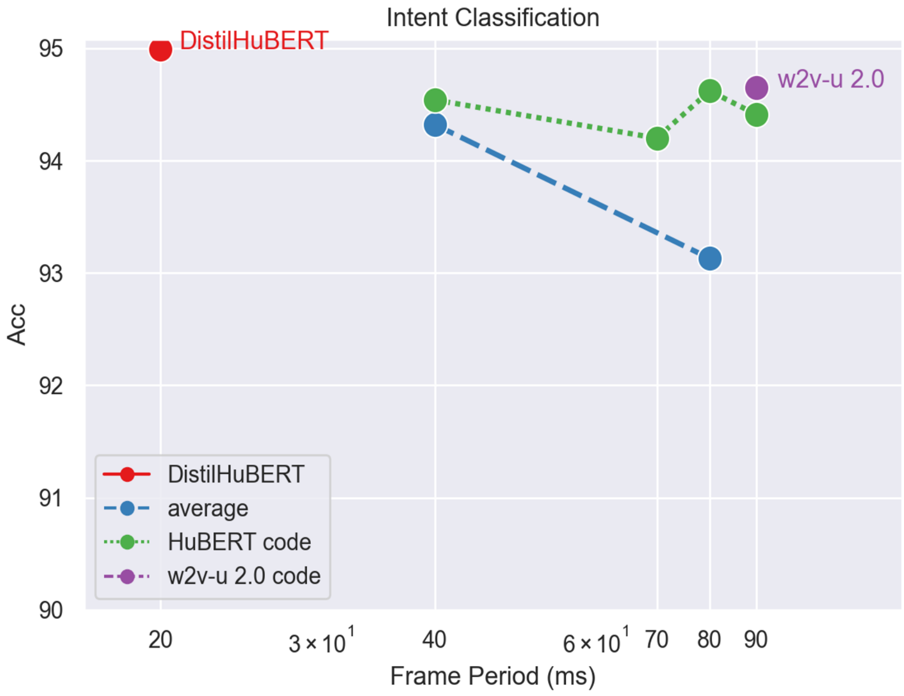
Supervised
- Forced alignments



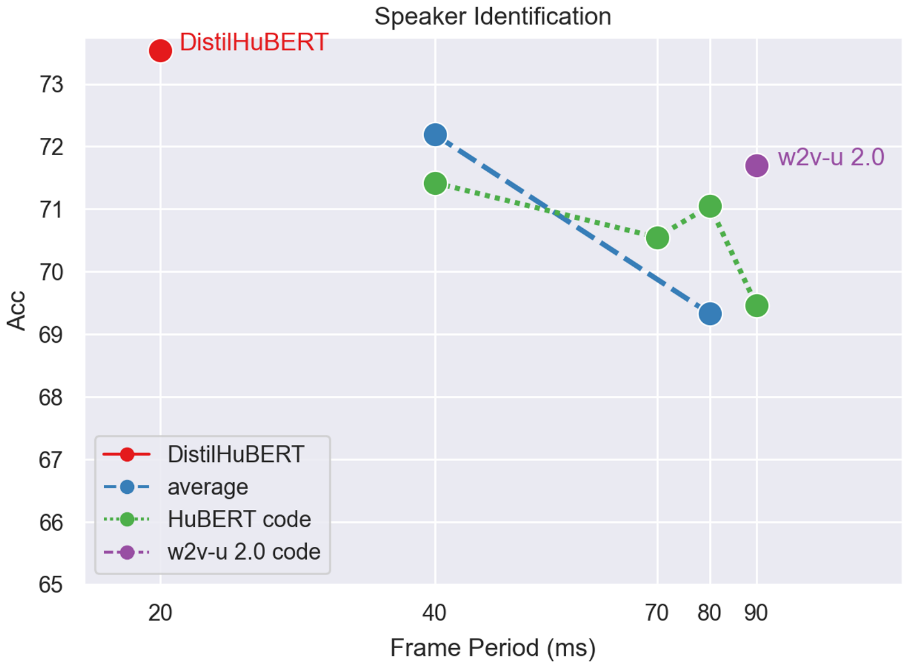
Experiments on SUPERB benchmark





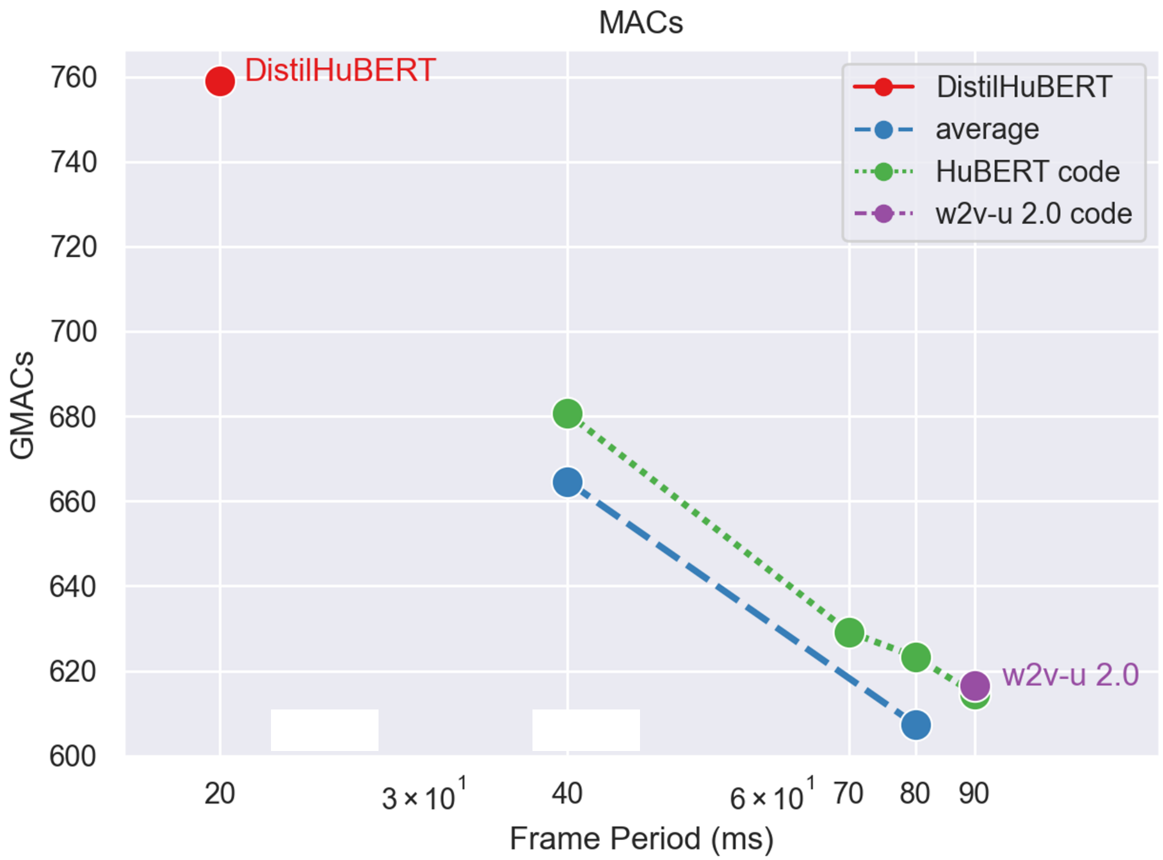


Intent Classification
DistilHuBERT
w2v-u 2.0
95
94
93
92
91
90
Acc
DistilHuBERT
average
HuBERT code
w2v-u 2.0 code
20
3×10^1
40
6×10^1
70
80
90
Frame Period (ms)

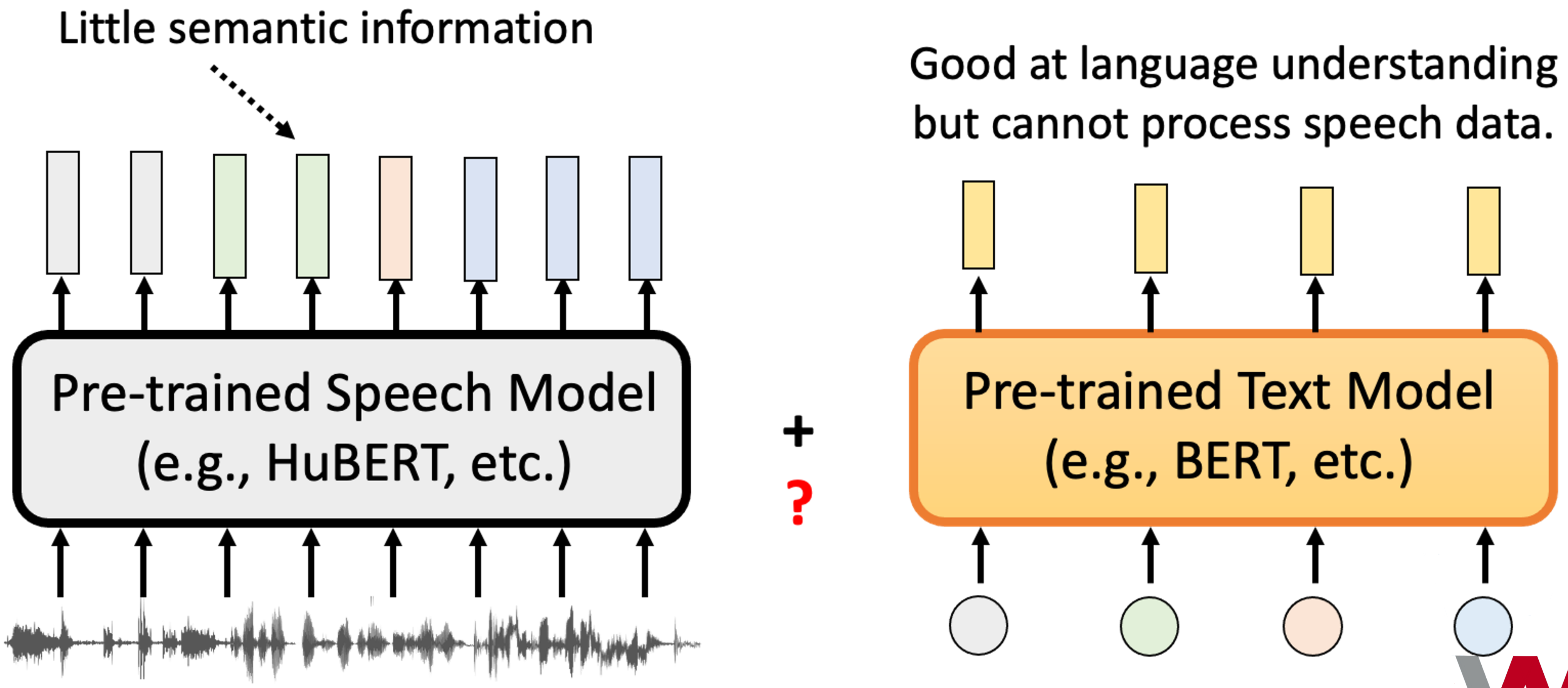


Speaker Identification
DistilHuBERT
w2v-u 2.0
73
72
71
70
69
68
67
66
65
Acc
DistilHuBERT
average
HuBERT code
w2v-u 2.0 code
20
40
70
80
90
Frame Period (ms)

Computational burden?



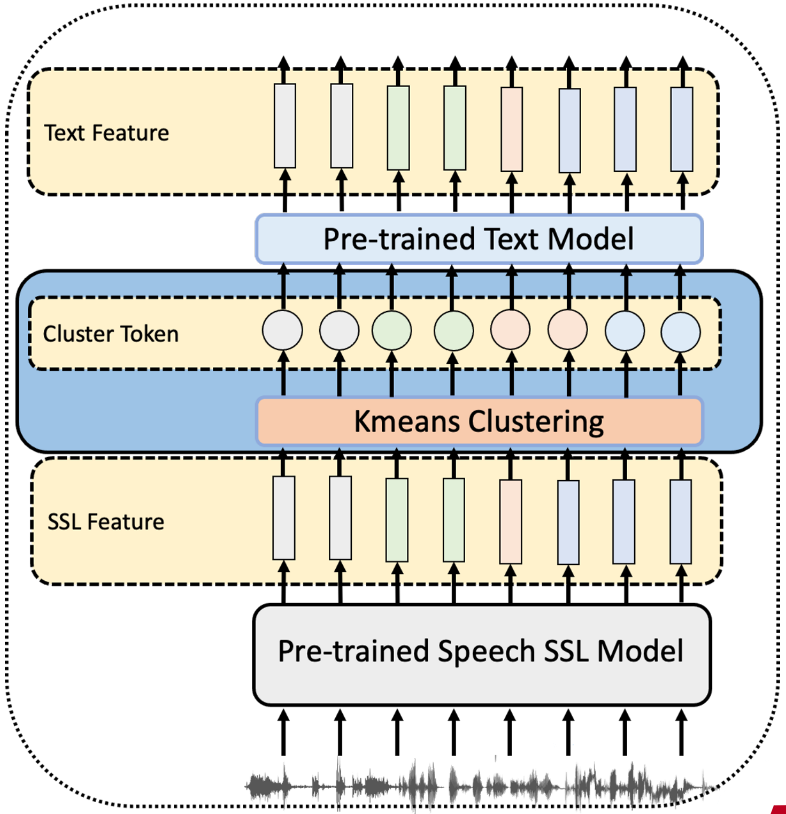
Unsupervised ASR as a Connector



Unsupervised ASR as a Connector (Cont’d)

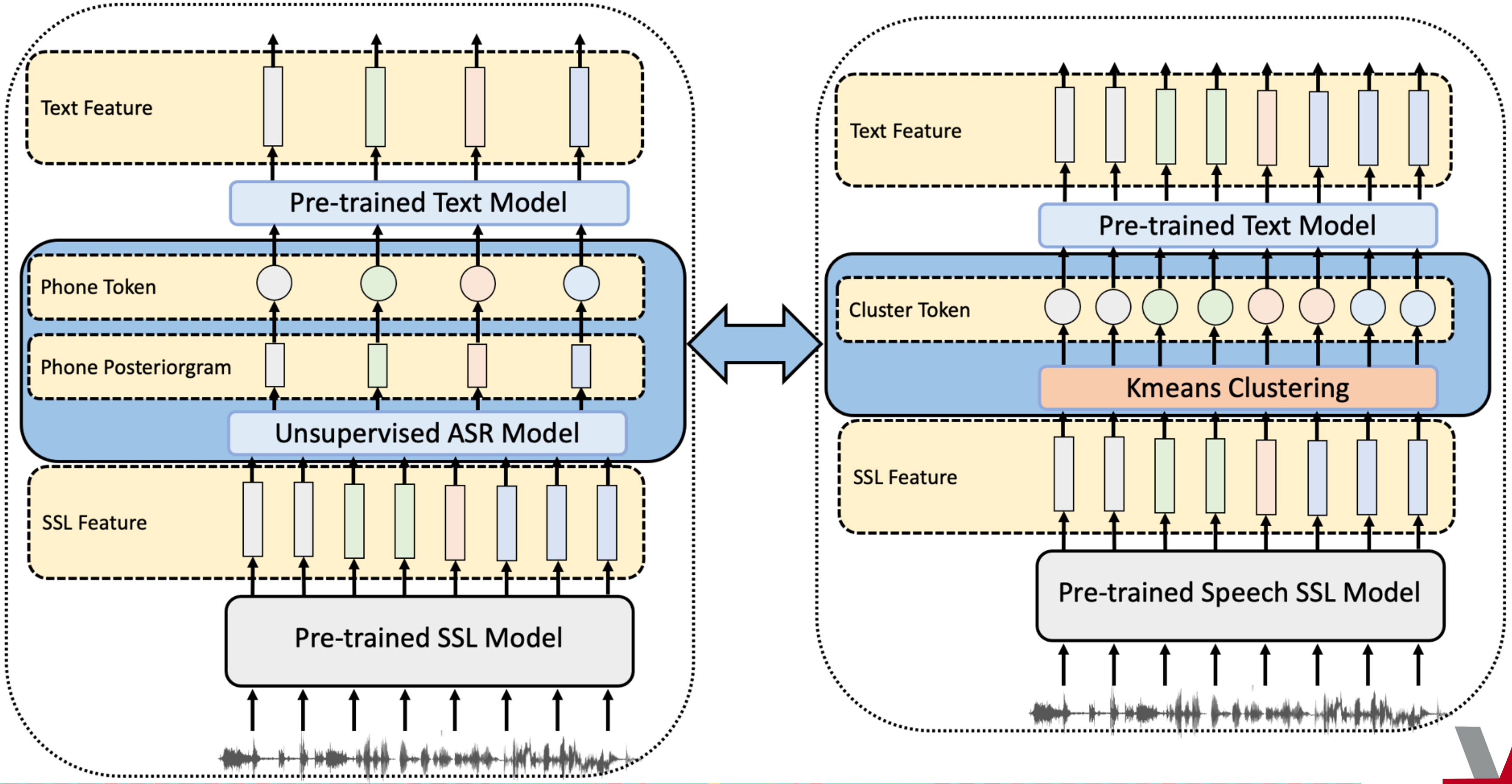
Existing method* to connect speech-SSL and text-SSL

- Method: Use speech-SSL feature clusters
- Domain is still mismatched
    - Acoustic v.s. Semantic



*: Guan-Ting Lin, Yung-Sung Chuang, Ho-Lam Chung, Shu-wen Yang, Hsuan-Jui Chen, Shuyan Dong, Shang-Wen Li, Abdelrahman Mohamed, Hung-yi Lee, Lin-shan Lee. “DUAL: Discrete Spoken Unit Adaptive Learning for Textless Spoken Question Answering” in Interspeech 2022

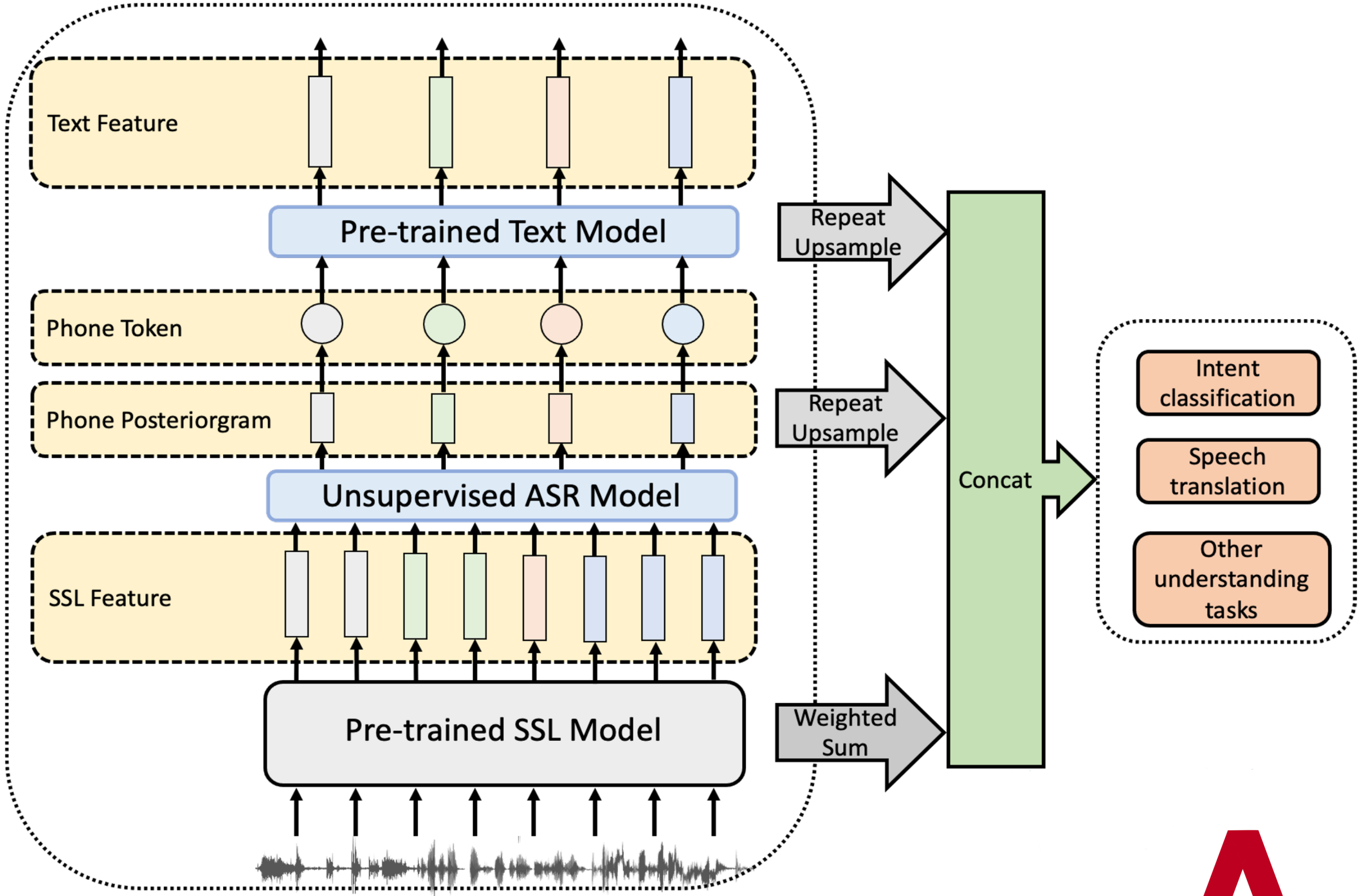
Close the domain mismatch



Unsupervised ASR as a Connector (Cont'd)

Mainly focus on understanding tasks

(e.g., intent classification, speech translation, etc.)



Experimental settings

- Speech SSL models: wav2vec 2.0
- Connector
  - Kmeans pretrained from fairseq
  - Unsupervised ASR
- Pretrained text model
  - Randomized T5
  - Phoneme T5
  - Byte-pair-encoding T5
- Fixed representation vs. Fine-tune text model

Unsupervised ASR as a Connector (Connector Options)

Tasks	Fixed - FSC (↑)	Fine-tuning - SLURP (↑)
Baseline (wav2vec2)	94.38	82.82
KM	93.69	85.31
UASR	**94.88**	**86.14**

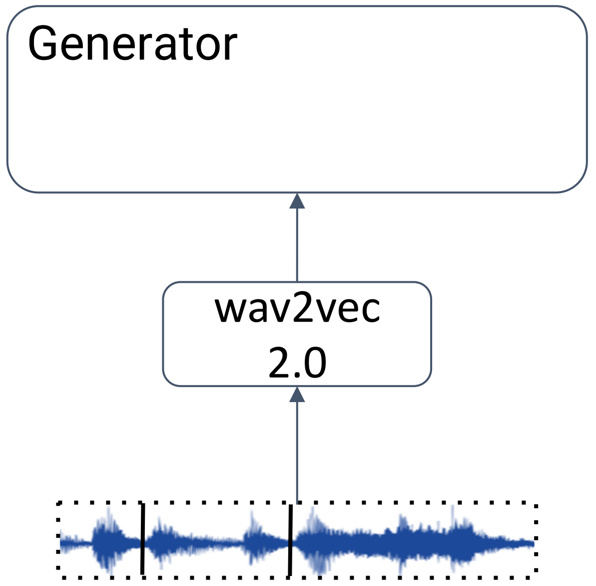
- KM methods **cannot** function well **without fine-tuning**
- **UASR** as a connector **outperforms KM** methods in both **fixed** and **fine-tuning** cases

Ongoing work

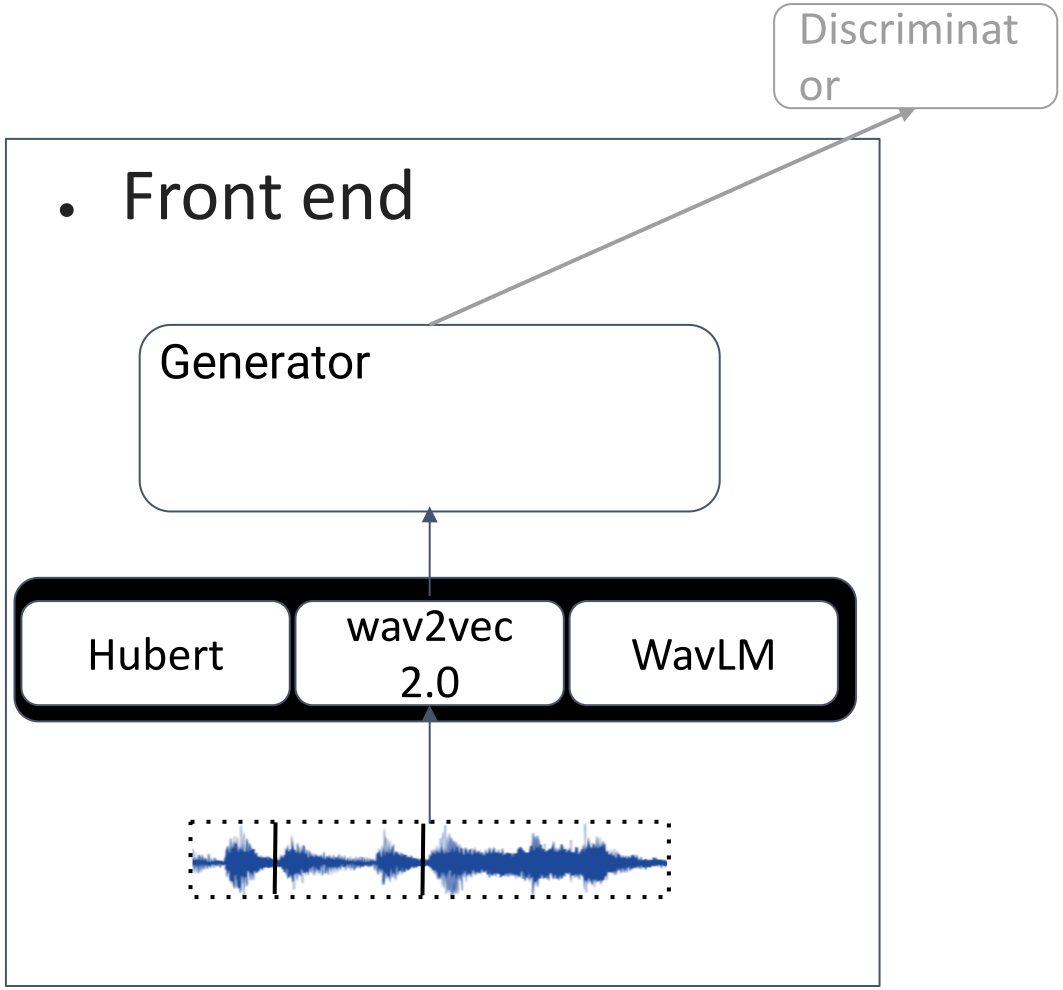
- ESPnet – Unsupervised Recognition – Opensource (EURO) project

EURO project

- Front end

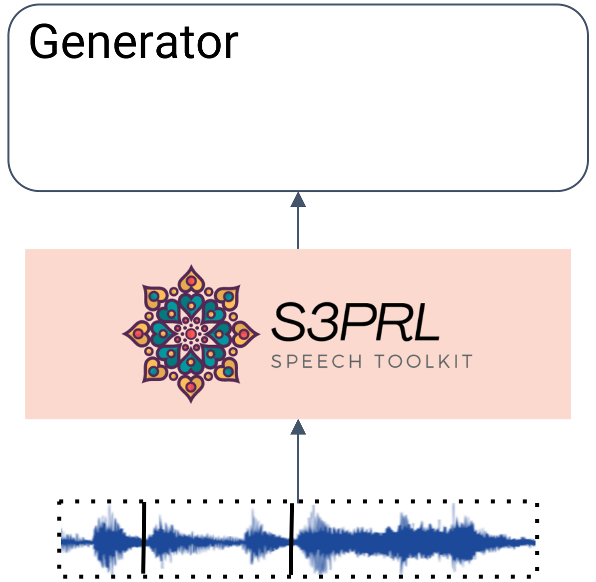


EURO project



EURO project

- Front end



EURO project

- Front end

S3PRL
SPEECH TOOLKIT

- Faster data preprocessing
  - Parallel
    - VAD
    - Remove silence
    - MFCC clustering
  - On-the-fly feature extraction
    - Trainable weighted sum of features from different layer
- Training

Finished

- More on Training
  - Reproducibility
  - Efficiency
  - Performance

Ongoing

Thanks for your attention!