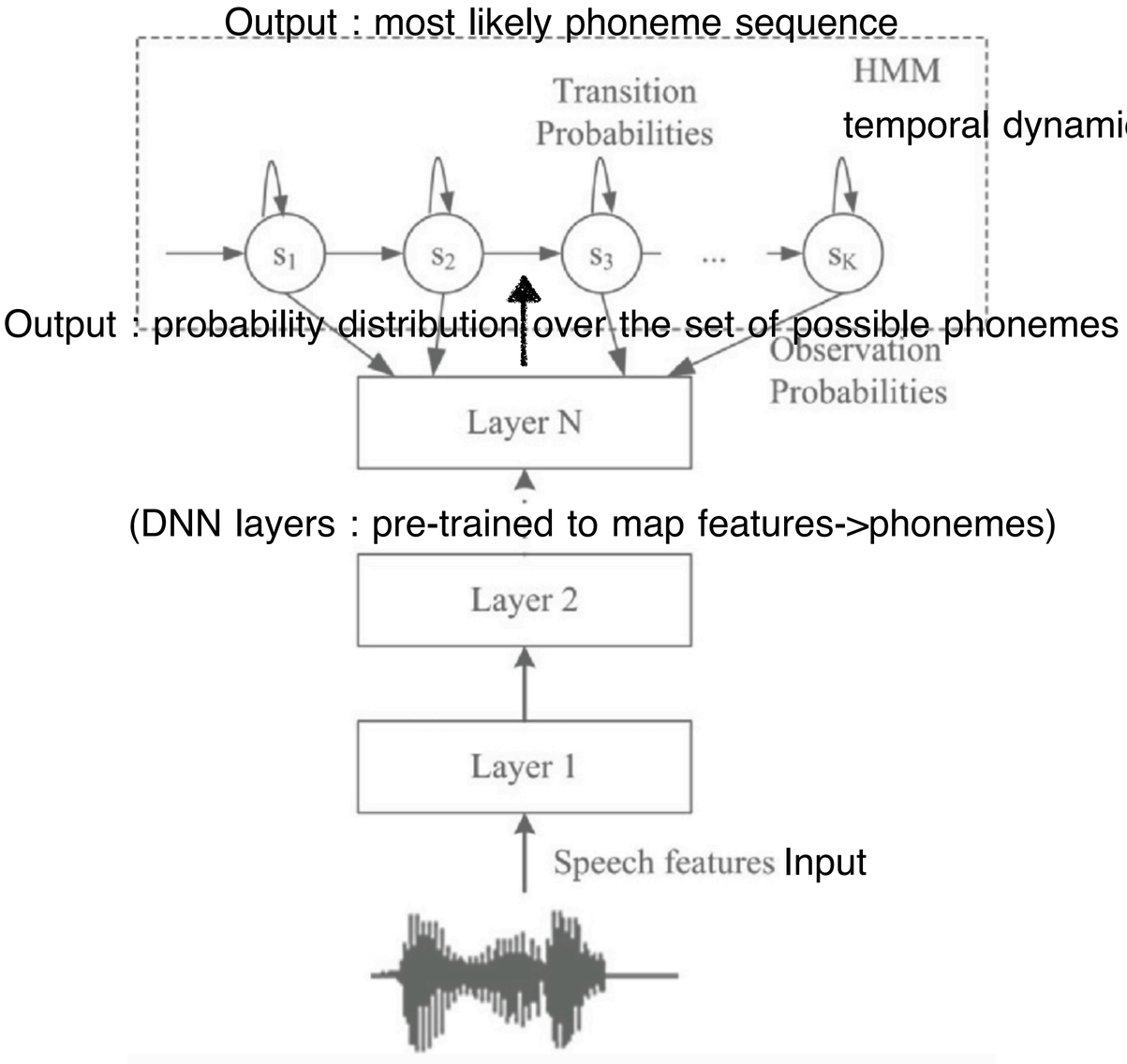
Spoken Term Detection
&
Voice Search

용어정리

- Keyword Spotting (KWS) : 짧은 발화 속 특정 키워드의 발화 여부를 판단하는 작업
  - Spoken Term Detection (STD) : 비교적 긴 발화 속 특정 키워드의 발화 여부(들)을 판단하는 작업 (ex. 방송 모니터링)
  - Voice Trigger Detection (VTD) : 기기, 시스템이 작동하도록 하는 특정 키워드를 인식하는 작업 (ex. Siri, Bixby)

=> 조금씩 차이가 있지만 핵심은 keyword detection

1. Deep LSTM Spoken Term Detection using Wav2Vec 2.0 Recognizer

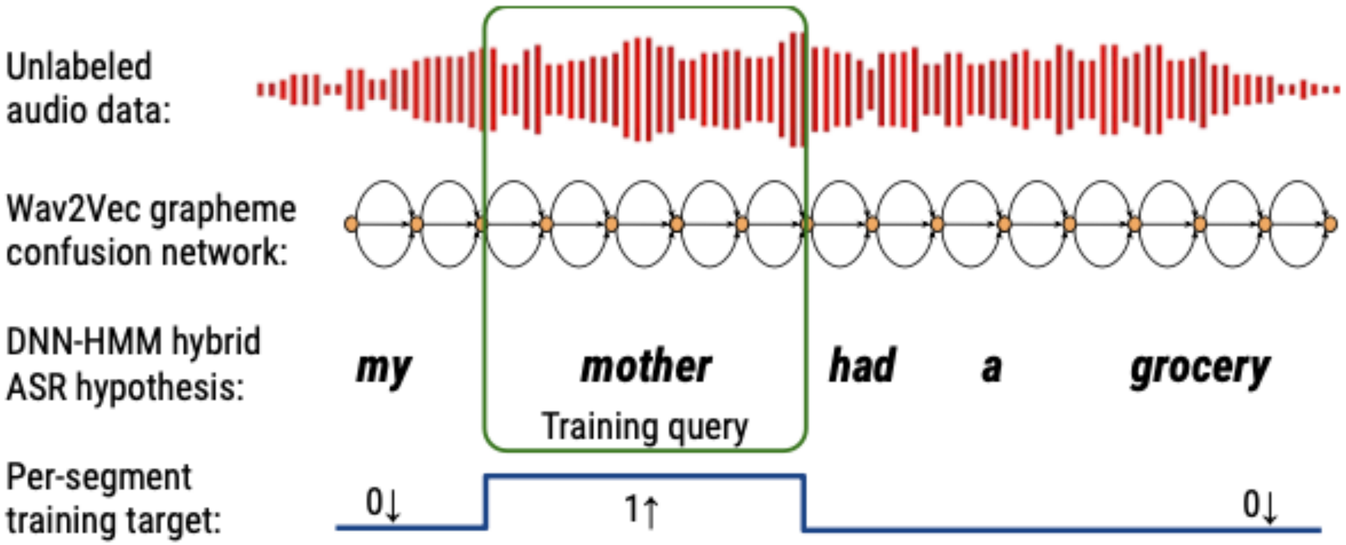


<DNN-HMM hybrid speech recognizer>

- 단점 : OOV에 대한 대처 능력 부족

  => Wav2Vec (Self-supervised learning) 활용

1. Deep LSTM Spoken Term Detection using Wav2Vec 2.0 Recognizer



1.input audio가 Wav2Vec 모델 통과 ⇒ “자소 혼합 네트워크” (grapheme confusion network)

2.input audio가 DNN-HMM 음성인식기 통과 ⇒ target word 생성

3.**Target word**와 이에 해당하는 혼합 네트워크 부분을 training query로 지정,

training target(0/1)과 비교 통해 LSTM STD 훈련에 사용됨.

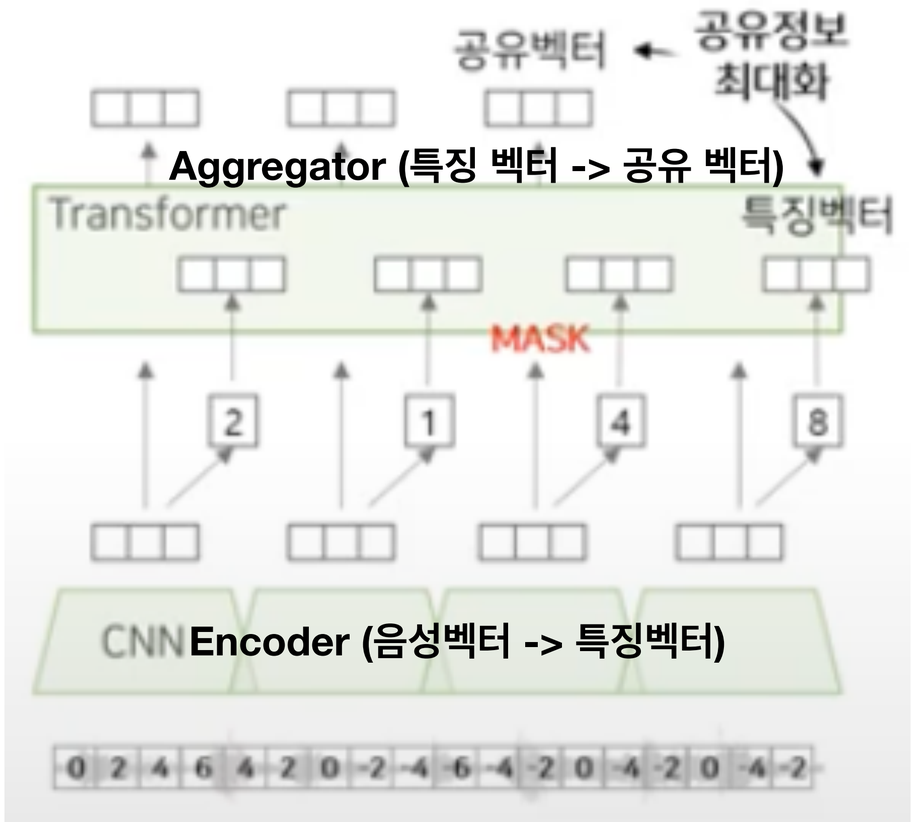
=> AM/LM 지식을 모두 학습한 LSTM STD 네트워크

1. Deep LSTM Spoken Term Detection using Wav2Vec 2.0 Recognizer

CTC Classification Layer

Fine-tune : 정답 비교 통한 추가학습

+



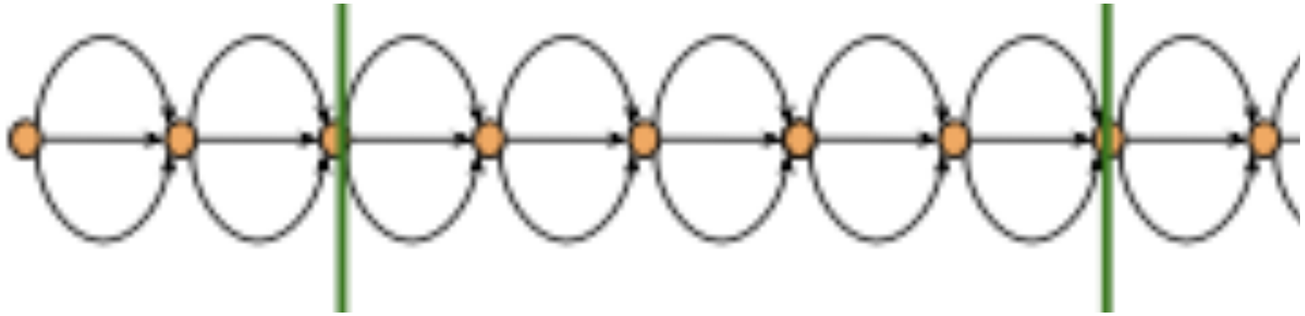
Pre-train : how to best represent raw audio data

Output: frame별 (row) 자소 확률분포 (column)으로 구성된 matrix

1. frame별 가장 높은 확률**(1-best)**의 자소 선택 (b-b-ε-o-o-ε-o-k-k)
2. 중복되는 자소 및 공백 삭제 (b, o, o, k)
3. 합치기 (book)

각 노드는 frame별 자소의 확률 분포로 구성

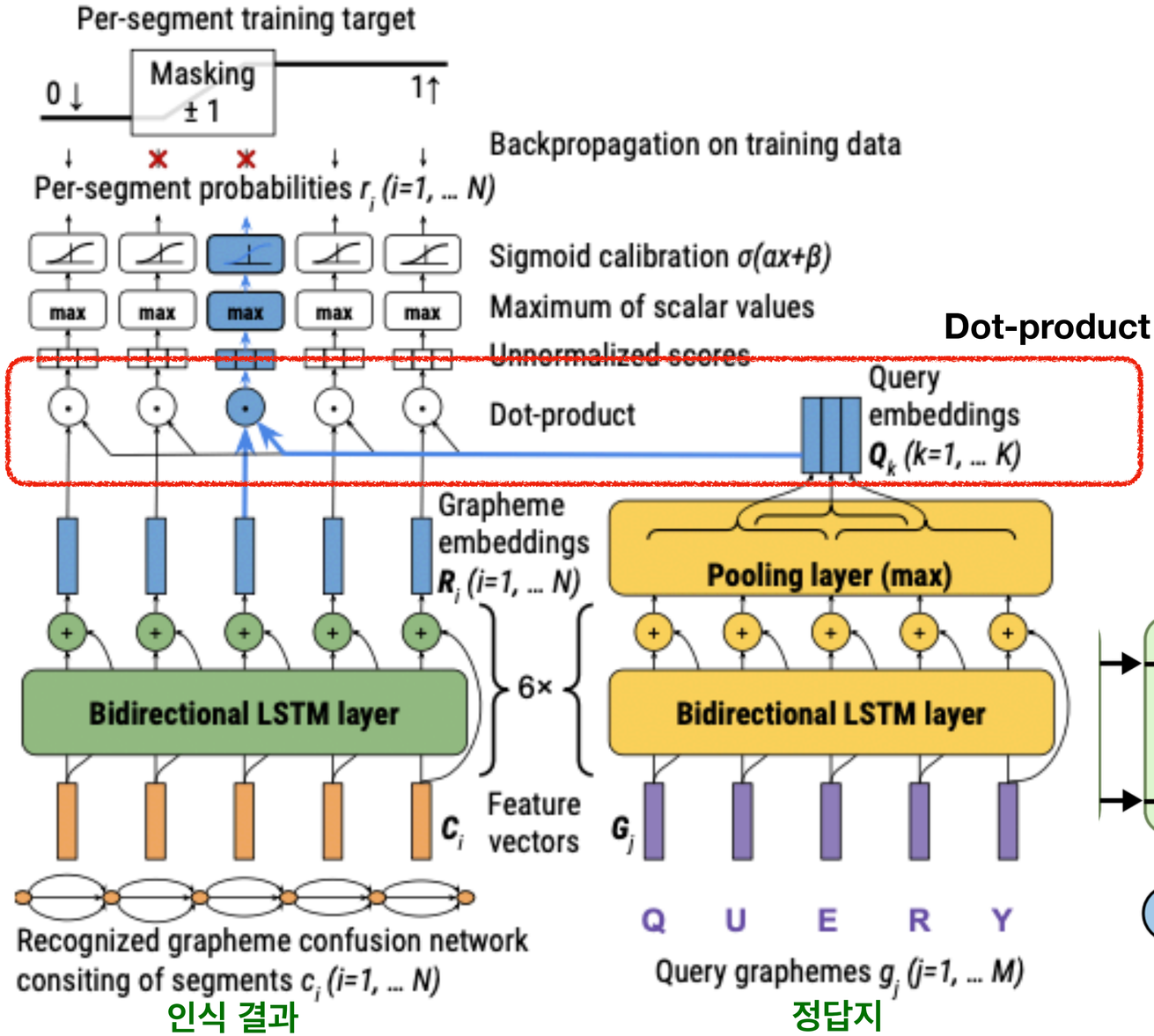
len(node)=num. of graphemes



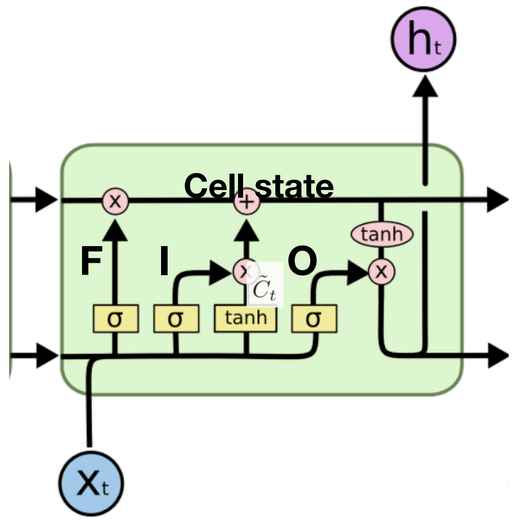
ex. [0.01, 0.005, 0.48, …]

=> 1-best가 아닌 가능한 모든 자소들의 확률 정보를 사용하려는 시도.

1. Deep LSTM Spoken Term Detection using Wav2Vec 2.0 Recognizer



Dot-product into single matrix



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$
$$h_t = o_t * \tanh(C_t)$$

I grew up in France
…
I met my friend in England
…
I speak fluent *French*

1. Deep LSTM Spoken Term Detection using Wav2Vec 2.0 Recognizer

In-vocabulary terms
(LM 없는 raw wav2vec2)

	English	Czech
DNN-HMM hybrid ASR	0.7899	0.8760
Wav2Vec 2.0 grapheme recognizer	0.6178	0.7248

Total result

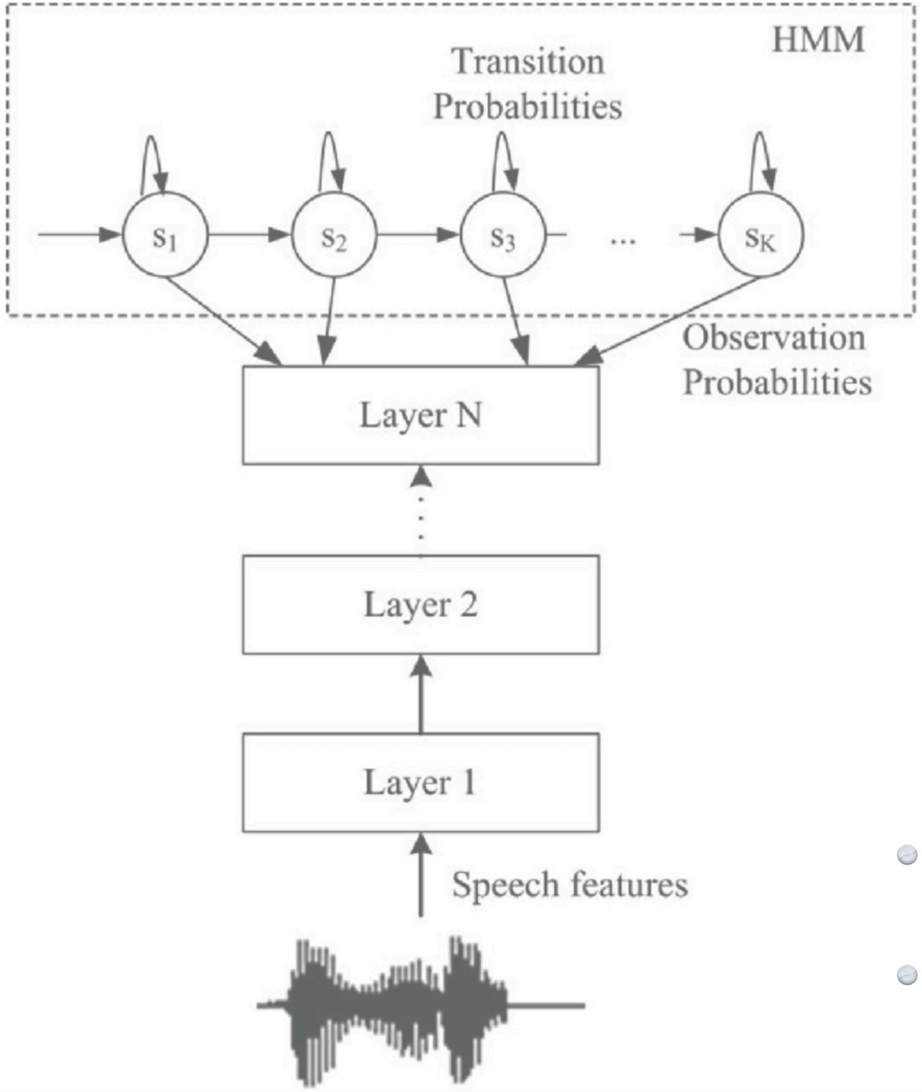
	English	Czech
Empirical method [4]	0.4636	0.6225
Siamese neural network [18]	0.5012	0.6547
Deep LSTM / DNN-HMM [13]	0.6703	0.7723
Deep LSTM / Wav2Vec 2.0 (proposed method)		
no masking	0.8198	0.8823
masking ± 1	**0.8308**	0.8888
masking ± 2	0.7873	**0.8987**
masking ± 3	0.7293	0.8790
In-vocabulary terms	0.8398	0.9117
Out-of-vocabulary terms	0.7027	0.8784

Data

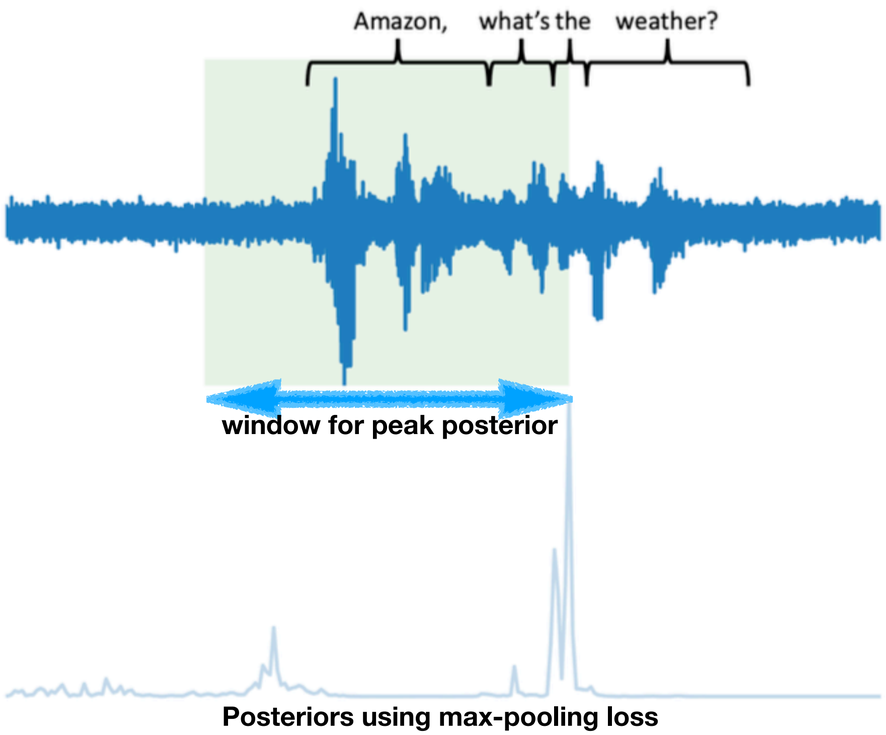
- wav2vec 2.0 base + MALACH dataset (역사, 교육 등)
- Deep LSTM STD : Visual History Archive

2. Latency Control for Keyword Spotting

키워드 시작-끝시간 계산 등 -> computationally expensive => CNN (using max-pooling loss)

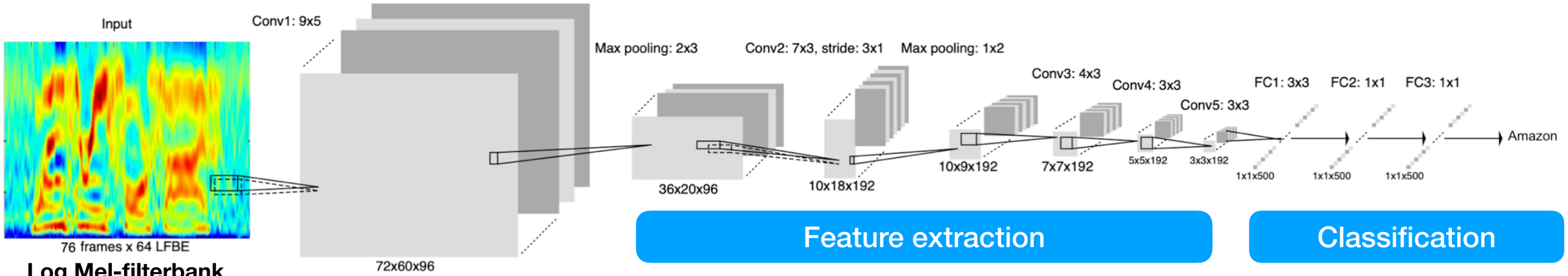


<DNN-HMM hybrid speech recognizer>



- Max-pooling loss : target class와 유사도 점수 산출, 가장 높은 점수의 frame만 선택
- 문장이 길어질 경우,
  모델이 **keyword 구간**에 확신을 갖지 못해 응답이 늦어진다.
  -> keyword 포착 구간을 앞당기자!

2. Latency Control for Keyword Spotting

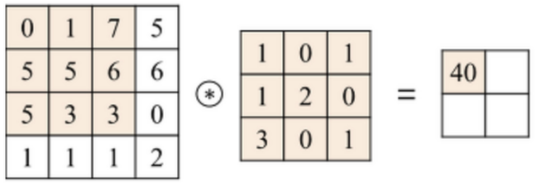


Log Mel-filterbank energy features (3D)

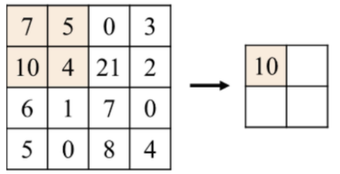
- CNN : 사람이 여러 사진을 보고 무엇인지 맞추는 사고에서 착안
  - Fully-connected layer : 3차원 이미지를 평면화 -> 정보 유실

- Keyword “Amazon”을 포함한 약 1400시간 데이터로 훈련
  - 기존 모델과의 성능 비교 실험은 기술되어 있지 않다.

- **일정 확률 b**로, 최고점수 frame의 **바로 이전 frame**에 손실 함수를 적용한다.
  => 이전 frame의 점수도 높게 측정되어, keyword의 탐색 지점을 앞당긴다.

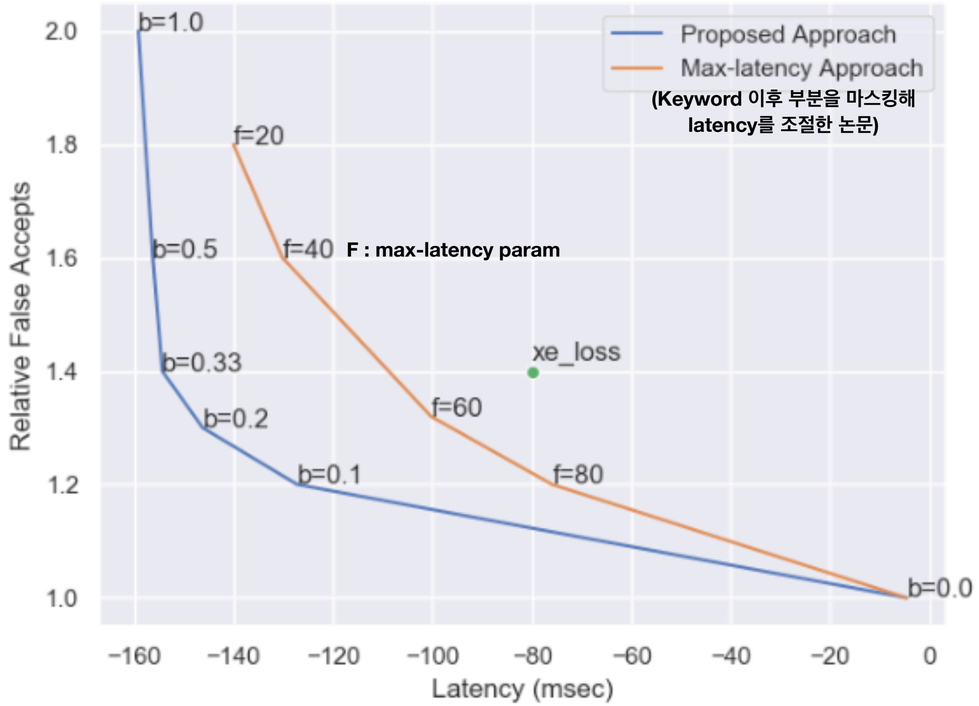


Convolution layer

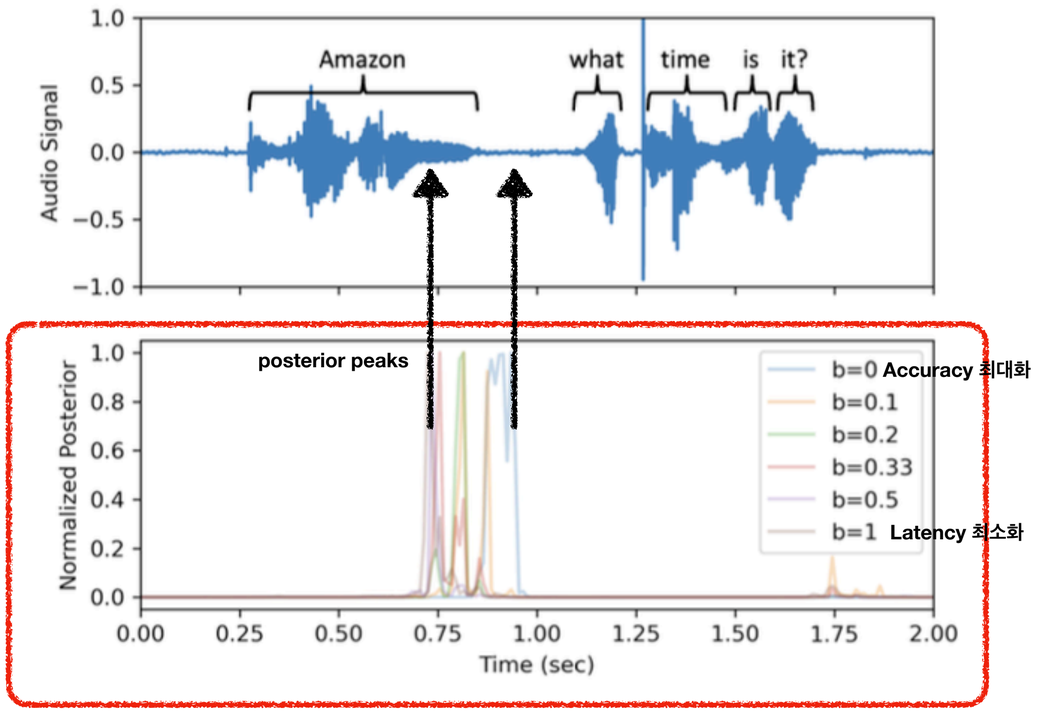


Pooling layer (-> max pooling)

2. Latency Control for Keyword Spotting



Latency parameter (b, f)와 정확도는 반비례 !



- b={0.0, 0.1, 0.2, 0.33, 0.5, 1.0} 총 6개 모델을 비교 분석
- 목적에 따라(latency, accuracy) 모델을 직접 조정할 수 있다.

3. Improving Voice Trigger Detection with Metric Learning

Conventional Voice Trigger Detectors..

Trained w/ **Speaker-independent audio data**

Verified by a separate Speaker Recognition Model

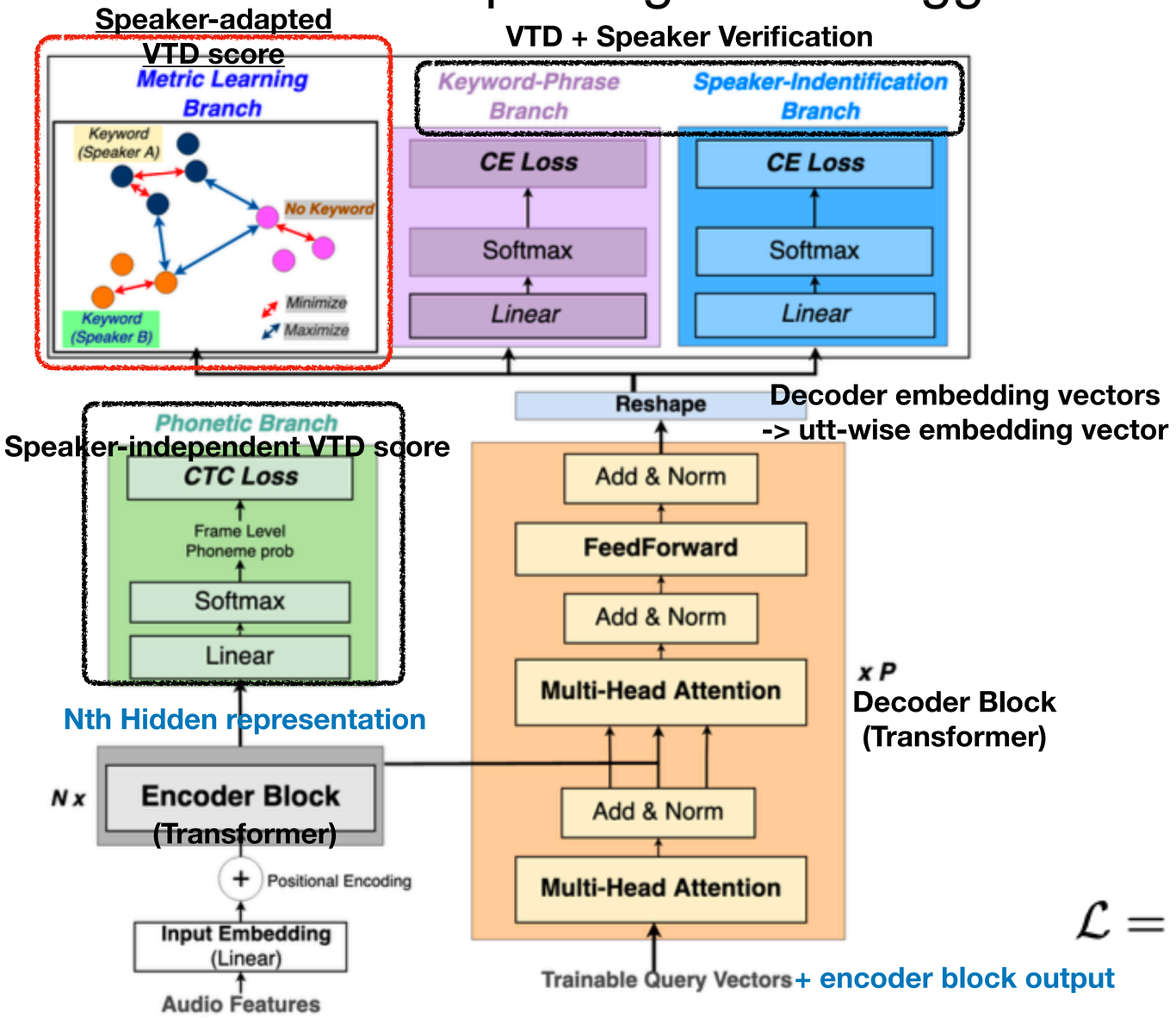
Weakness for under-represented speakers

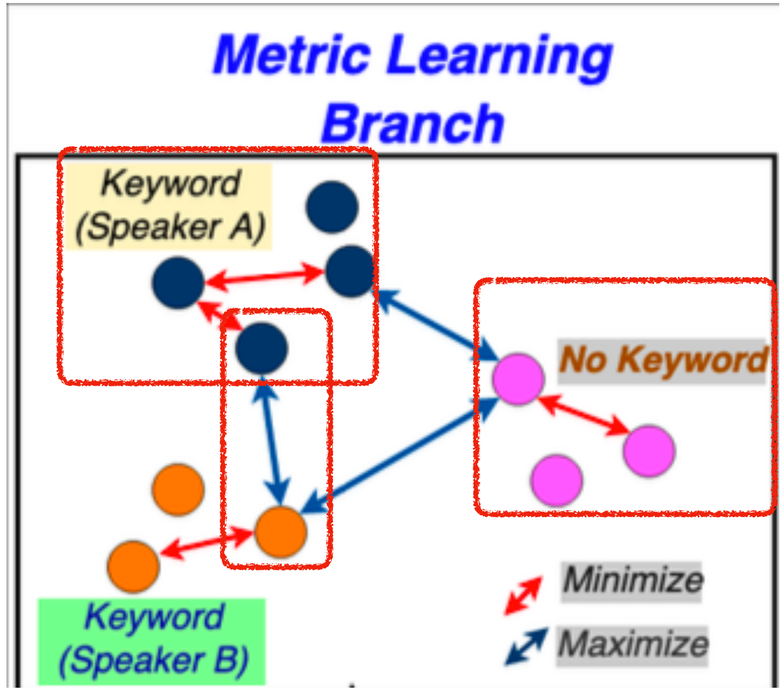
Unable to train w/ little data

(encoder : speaker-independent VTD)
+
(decoder : speaker-dependent VTD)

=> Integrated Voice Trigger Detector

3. Improving Voice Trigger Detection with Metric Learning





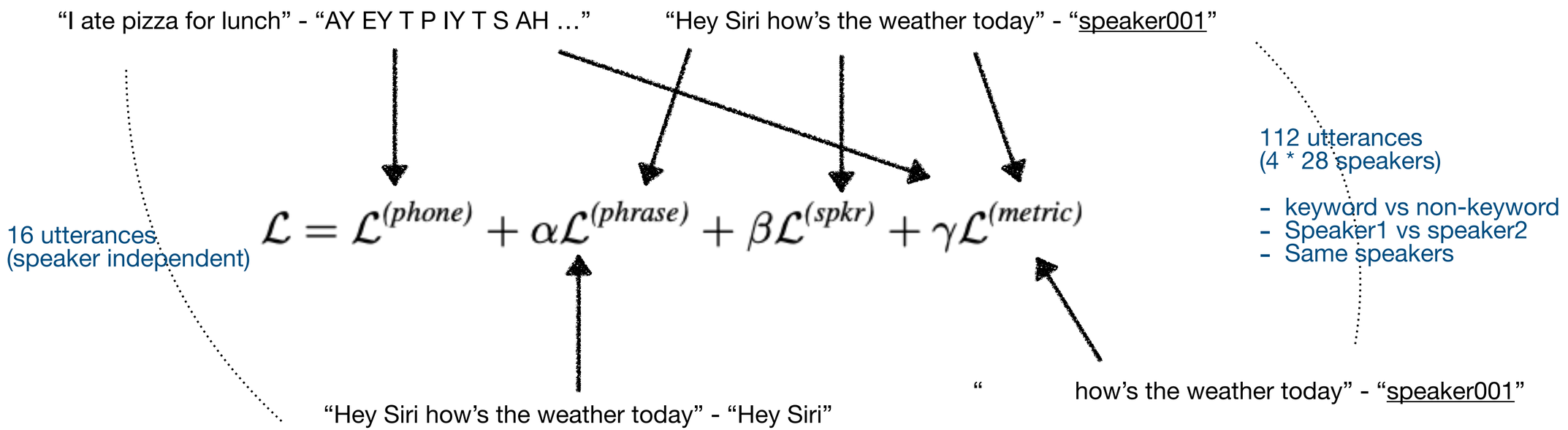
$$\mathcal{L} = \mathcal{L}^{(phone)} + \alpha\mathcal{L}^{(phrase)} + \beta\mathcal{L}^{(spkr)} + \gamma\mathcal{L}^{(metric)}$$

<Multi-Task Learning, customized>

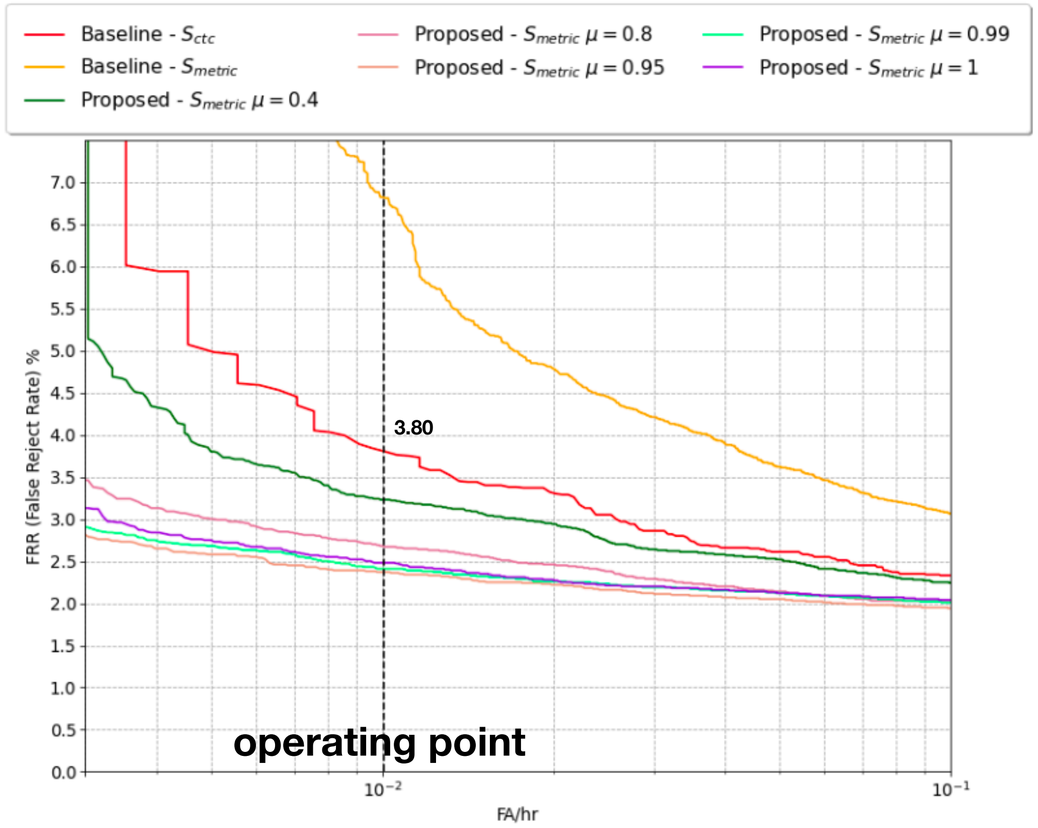
3. Improving Voice Trigger Detection with Metric Learning

data : 9 mil. Anonymized utterances (1,000h) + 15 mil. speaker-identified utterances

For every batch_size=128 …



3. Improving Voice Trigger Detection with Metric Learning



Detection Error Trade-off (DET)

	Branch	FRRs	
Baseline [33]	S_{ctc}	3.80	**Speaker-independent score**
	S_{phrase}	7.73	
	S_{metric} (μ=1)	6.82	
+ fine-tuning w/ spk-ID data	S_{metric} (μ=1)	8.89	
Proposed	S_{ctc}	3.80	
	S_{metric} (μ=1)	**2.48**	
Proposed	S_{ctc} and S_{metric} (μ=0.4)	3.23	
	S_{ctc} and S_{metric} (μ=0.8)	2.67	
	S_{ctc} and S_{metric} (μ=0.95)	**2.37**	**Speaker-adapted score**
	S_{ctc} and S_{metric} (μ=0.99)	2.41	

Scaling factors

38% reduction