Building far-field speech recognition for Amazon Alexa: Challenges and Solutions

Björn Hoffmeister

Amazon Alexa Speech

amazon alexa

Amazon Alexa Device Family

2014 2015 2016 2017 2018

Outline

Overview

Containing Speech

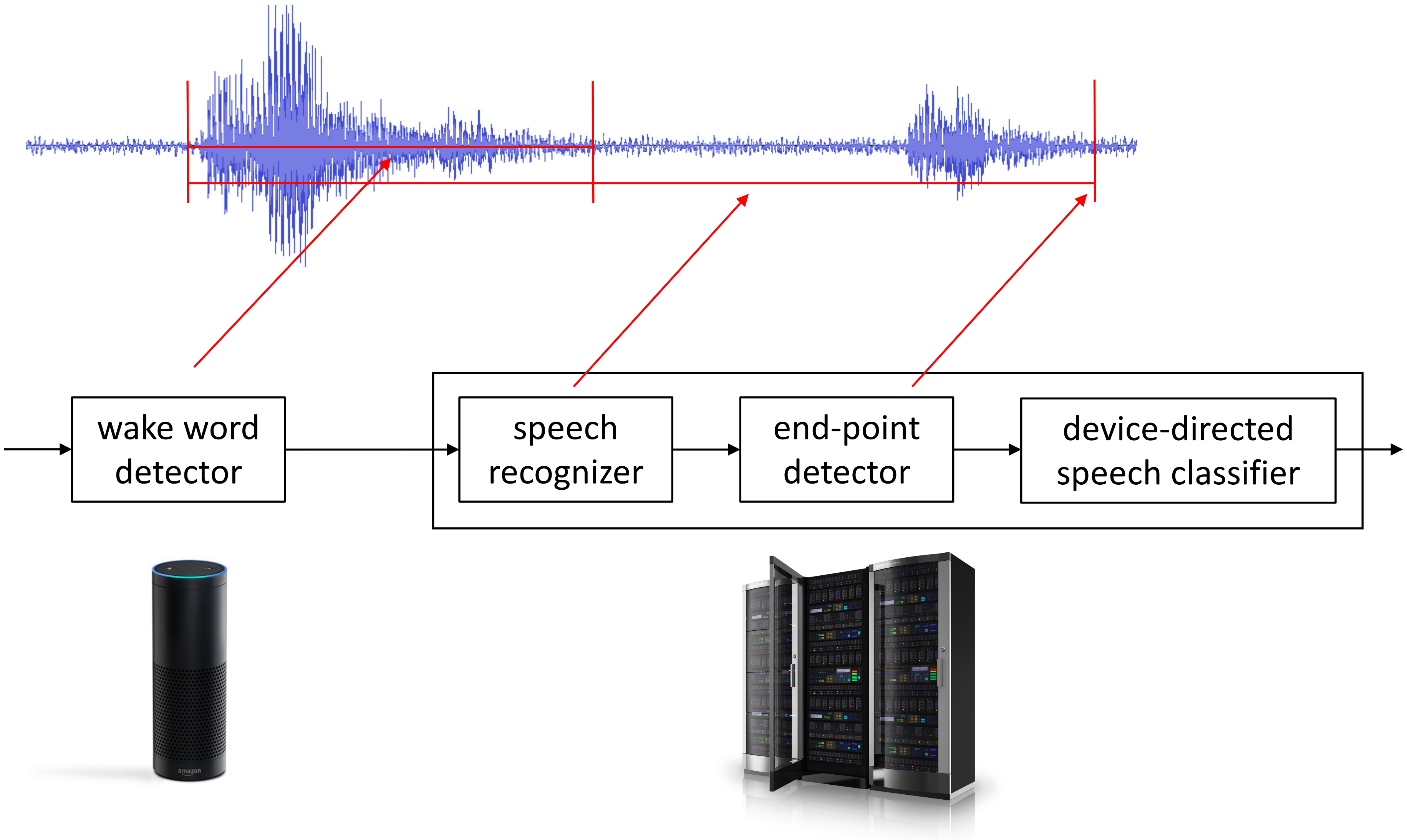
- Wakeword Detection
- End-of-Speech Detection
- Combining Wakeword and End-of-Speech Detection
- Device-Directedness Detection

Recognizing Speech

- Active Learning
- Multi-lingual and low-resource ASR
- Context Modeling

amazon alexa

Overview



Outline

Overview

Containing Speech

- Wakeword Detection
- End-of-Speech Detection
- Combining Wakeword and End-of-Speech Detection
- Device-Directedness Detection

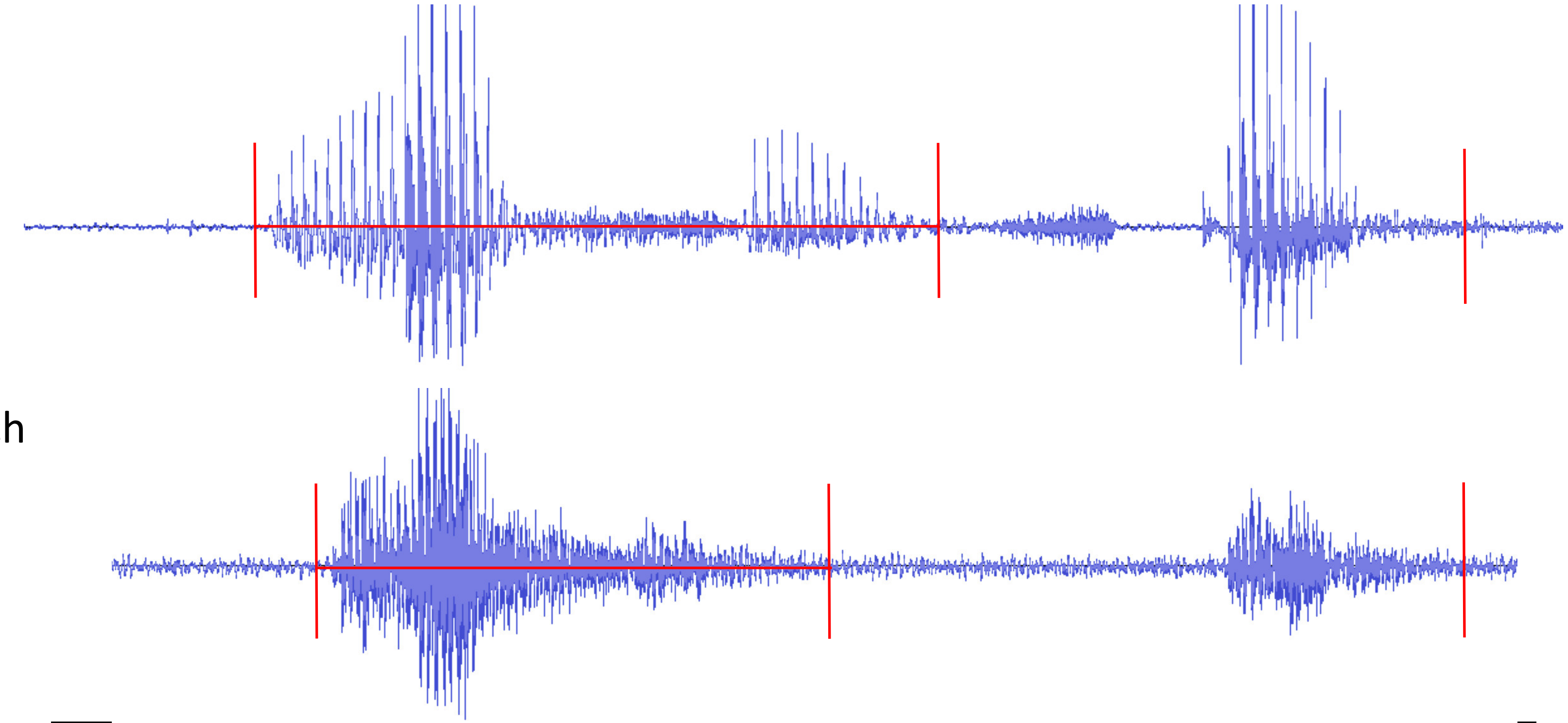
Recognizing Speech

- Active Learning
- Multi-lingual and low-resource ASR
- Context Modeling

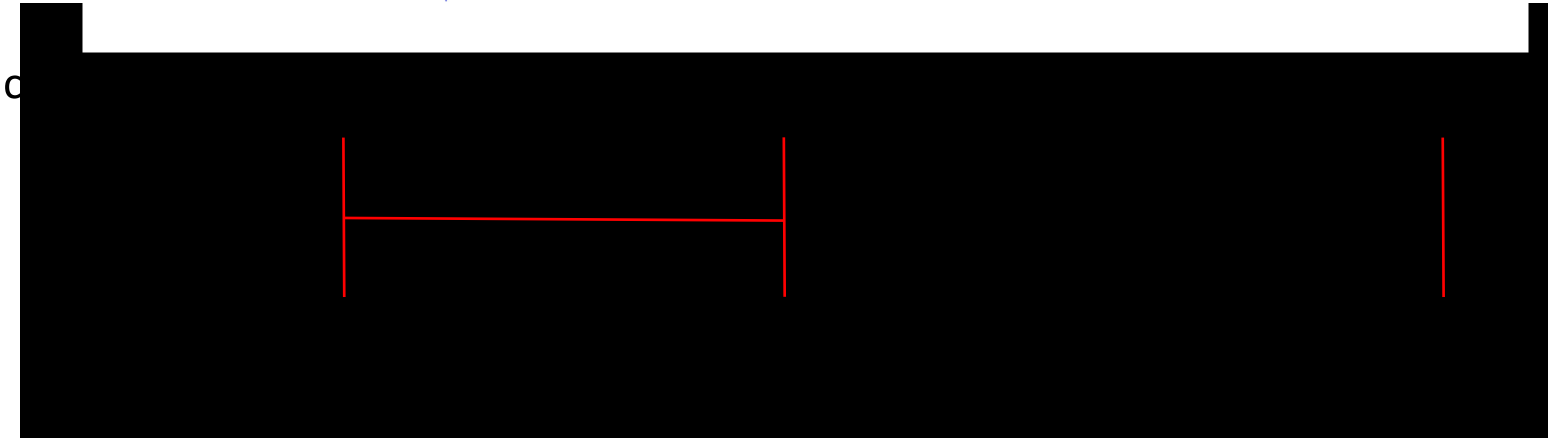
Wakeword Detection

close talk

distant speech

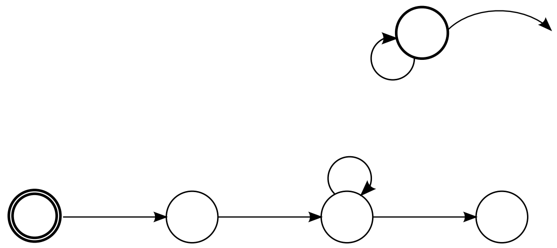


distant with bac[redacted]

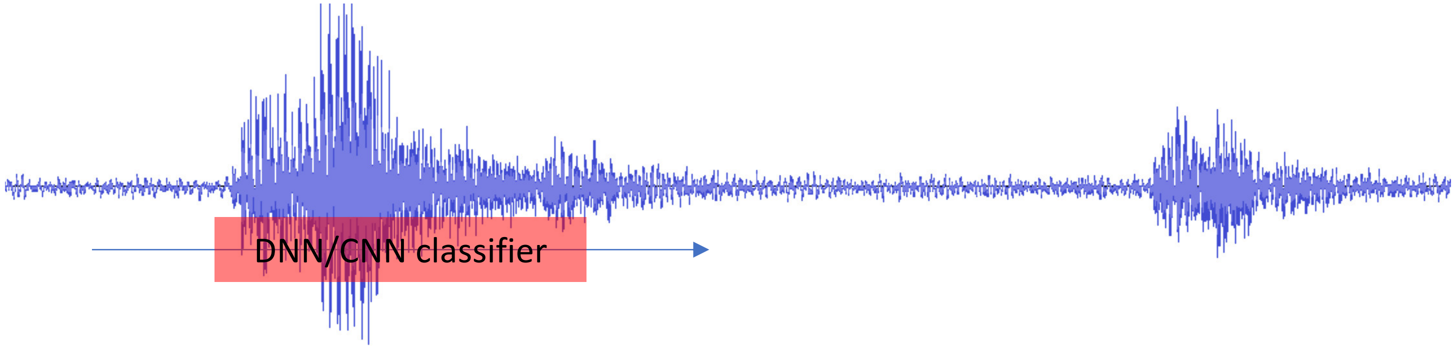


[redacted]

Old-Style Wakeword Detection



DNN/CNN-based Wakeword Detection

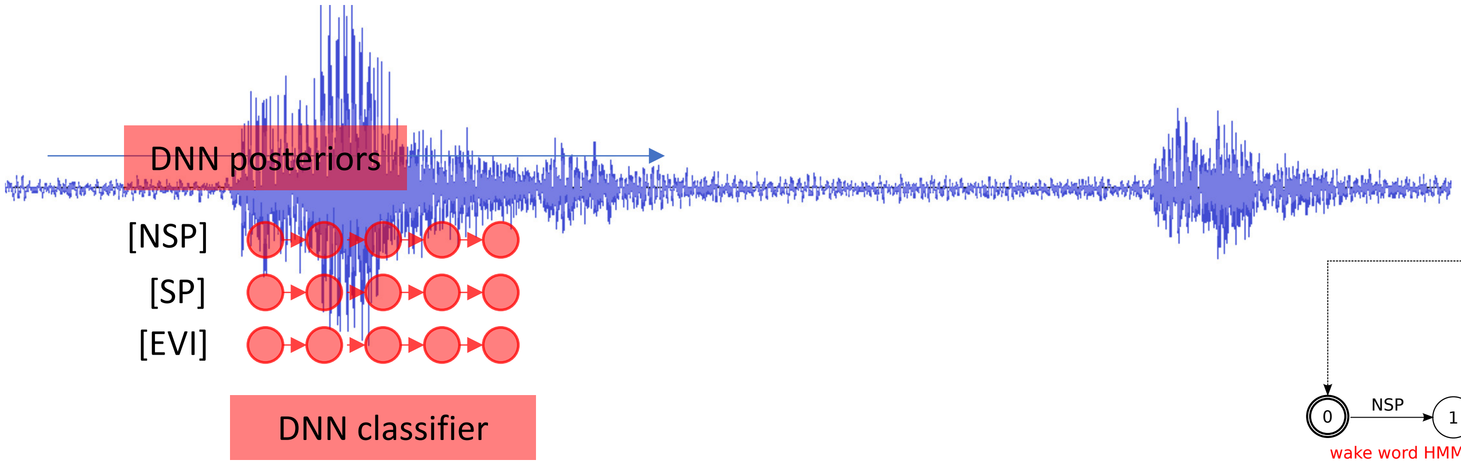


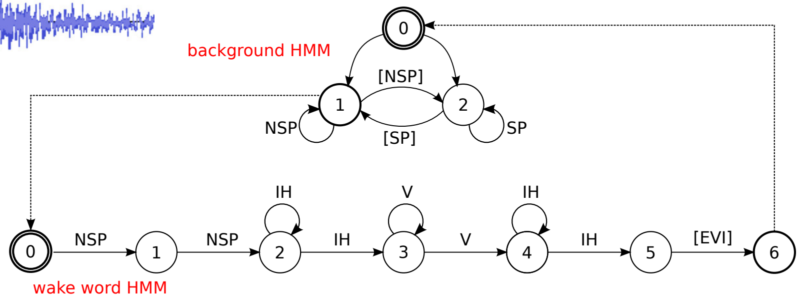
- Move sliding window of DNN/CNN classifier over the acoustic features (25ms analysis window, 10ms shift)
- Train DNN/CNN directly on wakeword instances
- Requires training data with thousands of wakeword instances

[Small-Footprint Keyword Spotting using Deep Neural Networks. *G. Chen et.al., Google, ICASSP 2014*]
- DNN posteriors per acoustic feature => sliding window over posteriors => "max-pooling" over smoothed posteriors
- Whole word modeling, no time warping

[Convolutional Neural Networks for Small-footprint Keyword Spotting. *T. N. Sainath et.al., Google, Interspeech 2015*]
- sliding window over acoustic features => CNN
- Whole word modeling, CNN patches can learn sub-words, limited time warping
- Large accuracy improvements

Hybrid Wakeword Detection





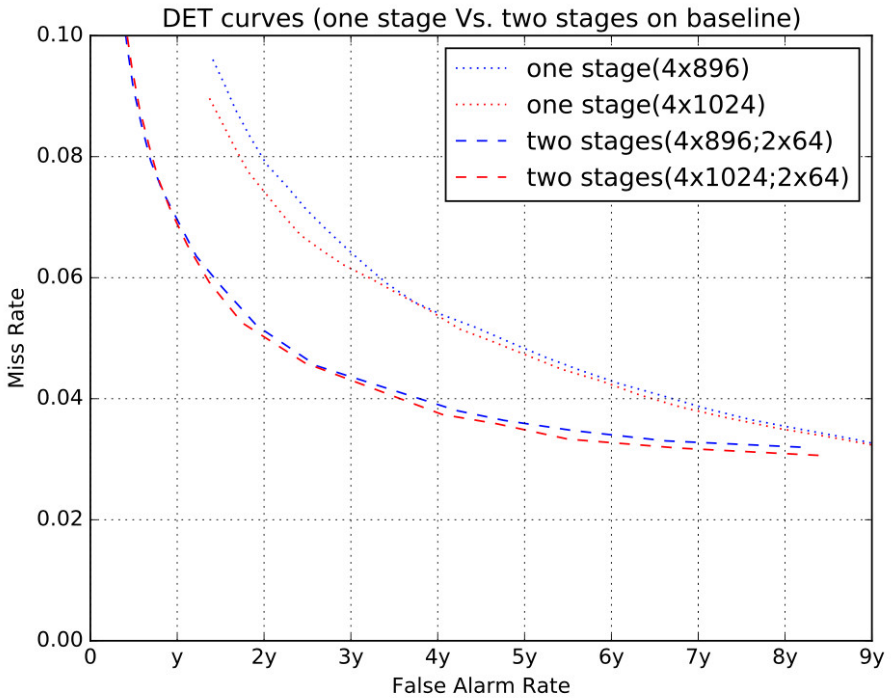
[Monophone-based Background Modeling for Two-Stage On-Device Wake Word Detection. *M. Wu et.al., Amazon, ICASSP 2018*]

- Direct wakeword sub-word unit modeling
- DNN posteriors per acoustic feature => HMM alignment of wakeword and BG model => DNN "sequence" classifier
- BG: speech/non-speech or monophone model

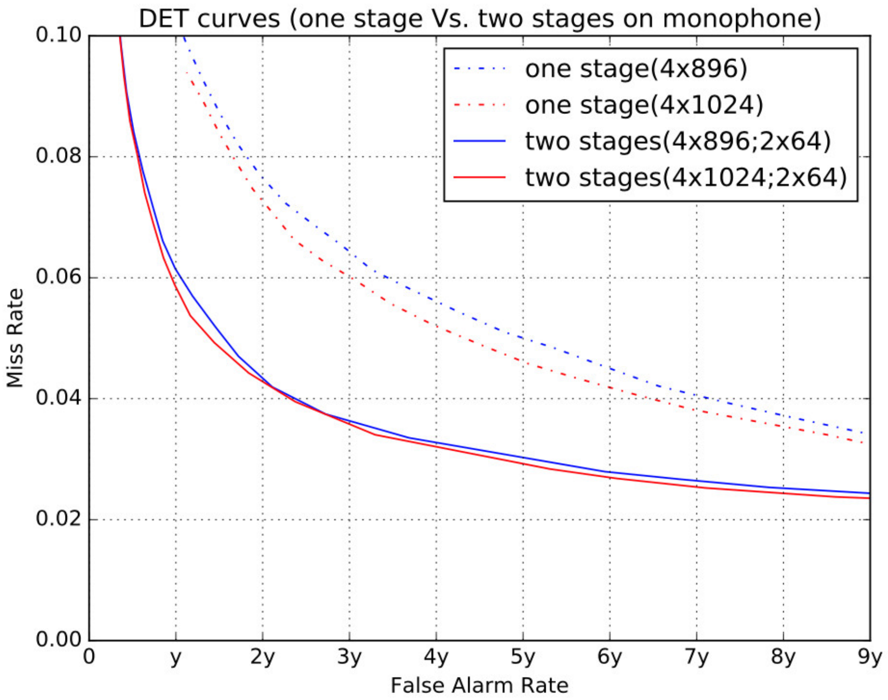
[Direct Modeling of Raw Audio with DNNs for Wake Word Detection. *K. Kumatani et.al., Amazon, ASRU 2017*]

- DNN posteriors directly from audio signal

Hybrid Wakeword Detection



(a) *speech/non-speech*



(b) *monophone*

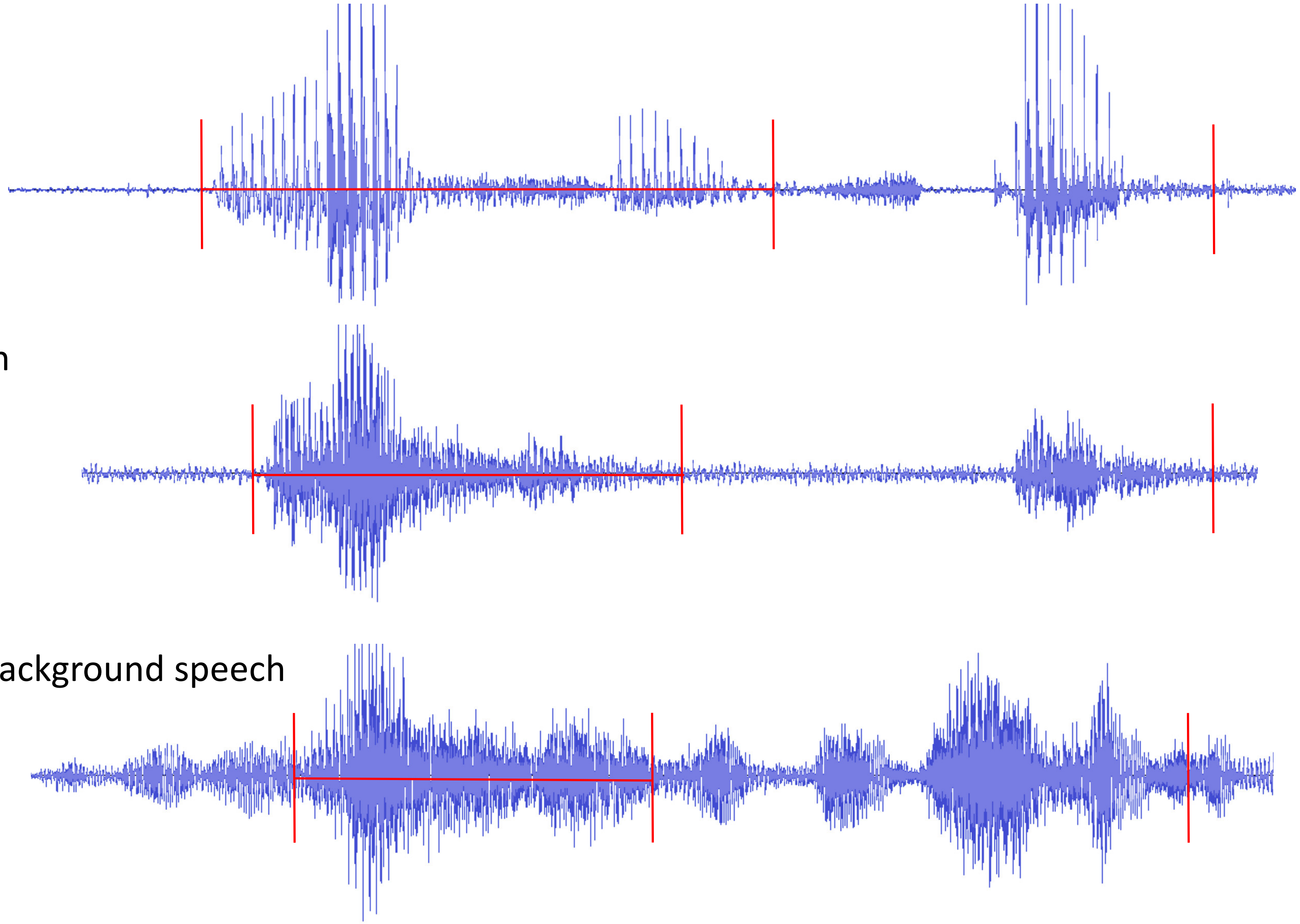
[Monophone-based Background Modeling for Two-Stage On-Device Wake Word Detection. *M. Wu et.al., Amazon, ICASSP 2018*]

End-Point Detection

close talk

distant speech

distant with background speech



Audio-based End-Point Detection

Old-style audio-based

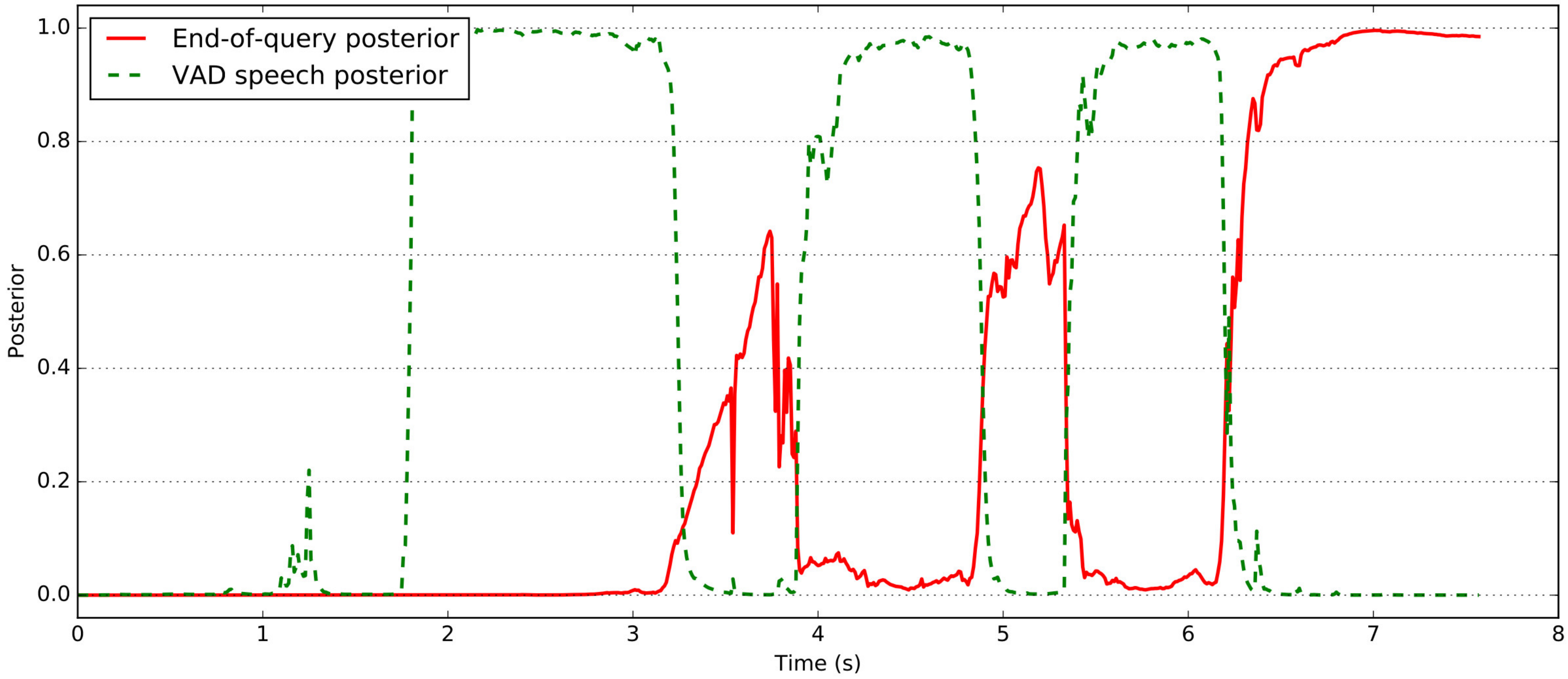
- Energy based + sophisticated thresholding scheme
- DNN/LSTM VAD, frame-wise speech/non-speech classification + thresholding scheme
- Problem: end-of-sentence or within-sentence pause?

What if audio signal contains enough information?

- LSTM/RNN powerful enough to distinguish end-of-sentence vs within-sentence pause?

[Improved End-of-Query Detection for Streaming Speech Recognition. *M. Shannon et.al., Google, Interspeech 2017*]

Audio-based End-Point Detection

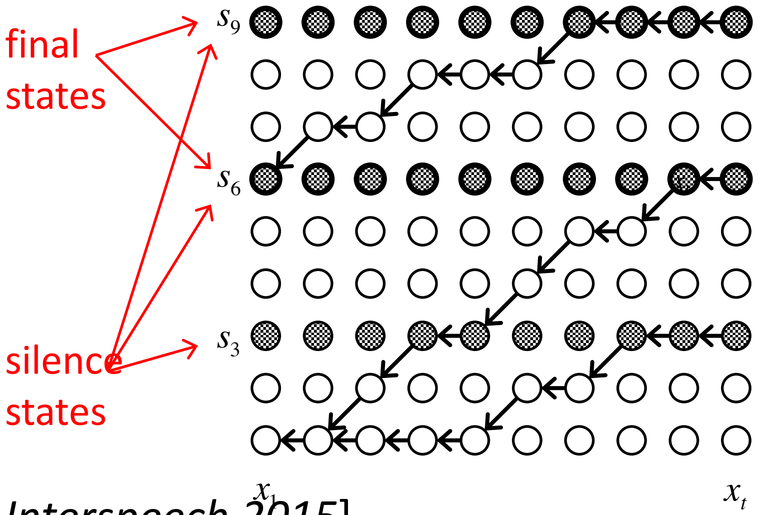


[Improved End-of-Query Detection for Streaming Speech Recognition. *M. Shannon et. al., Google, Interspeech 2017*]

Decoder-based End-Point Detection

Trust the speech recognition system:

- ASR system is the better VAD
- Language model (LM) predicts end-of-sentence, but …
- … limited LM history, typically three or four words
- … sentence end ambiguous “: “What’s the weather -- tomorrow?”
- Combine end-of-sentence prediction with non-speech thresholding
  => “pause duration after sentence end”

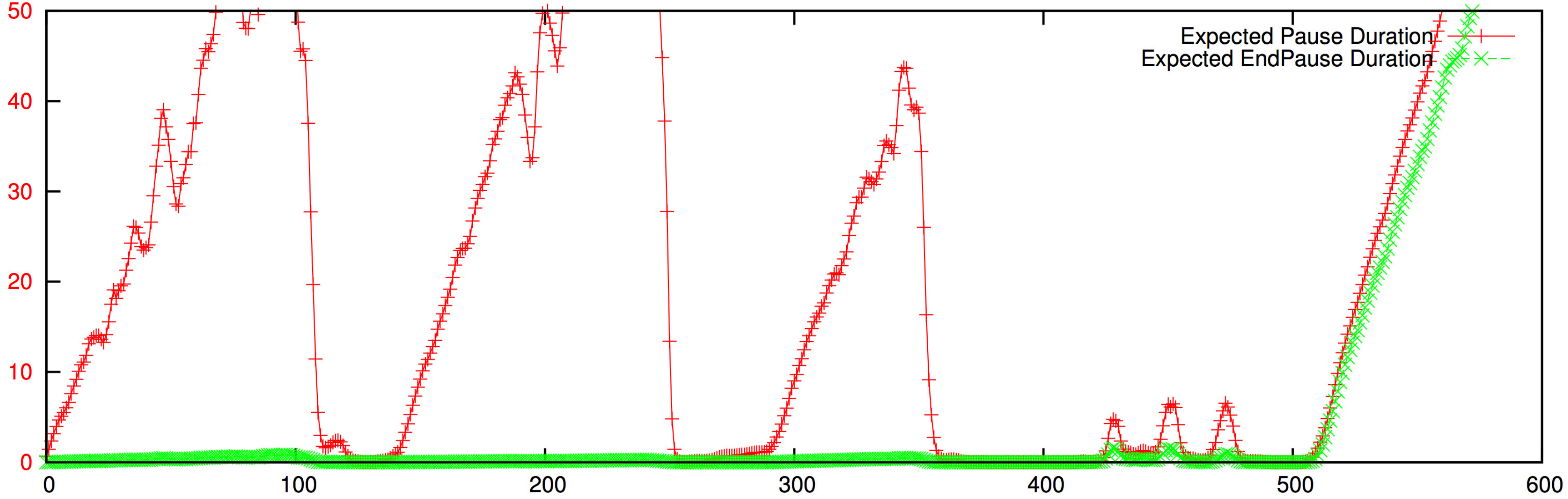


How to handle decoder uncertainty?

- Use expectation over active decoder hypotheses
  => expected “pause duration after sentence end”

[Accurate Endpointing with Expected Pause Duration. *B. Liu et.al., Amazon, Interspeech 2015*]

Decoder-based End-Point Detection

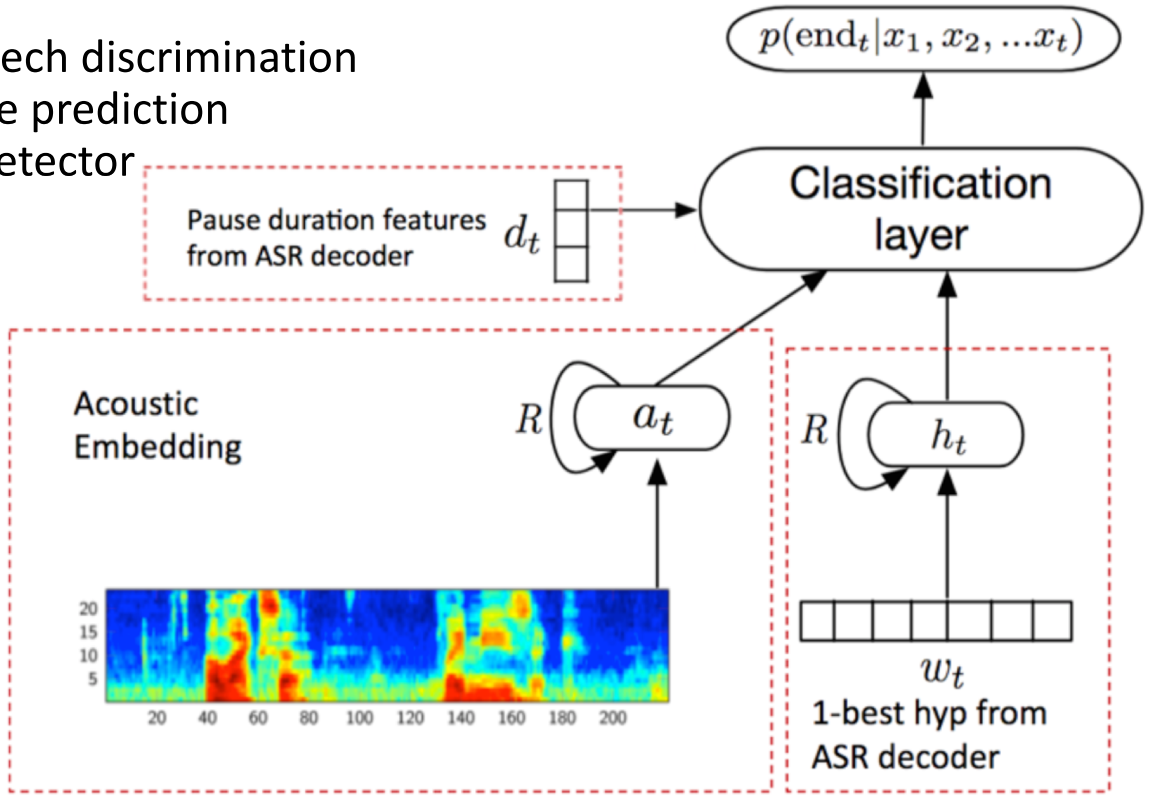


[Accurate Endpointing with Expected Pause Duration. *B. Liu et. al., Amazon, Interspeech 2015*]

Hybrid End-Point Detection

Why hybrid? Why not trusting the decoder?

- Acoustic model (AM) not optimized for speech/non-speech discrimination
- Language model (LM) not optimized for end-of-sentence prediction
- Technical considerations: separate ASR and end-point detector



Hybrid end-point detector

- Features
  - Audio-based end-point detection LSTM
    => acoustic embeddings
  - Lexical sentence-end prediction LSTM
    (based on best decoder hypothesis)
    => lexical embeddings
  - expected “pause duration after sentence end”
- DNN classifier

[Combining Acoustic Embeddings and Decoding Features for End-of-Utterance Detection in Real-Time Far-Field Speech Recognition System. *R. Maas et.al., Amazon, ICASSP 2018*]

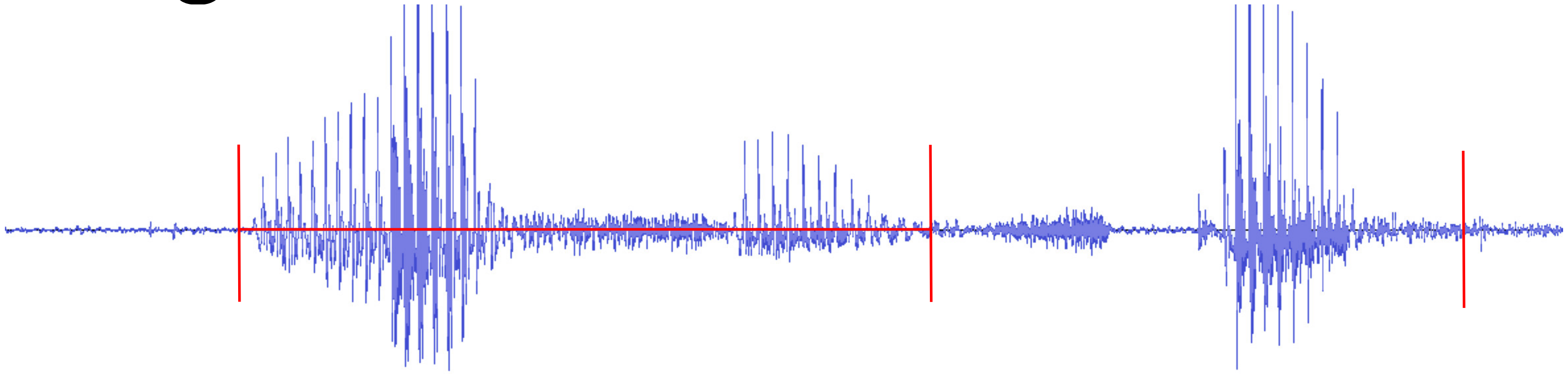
Hybrid End-Point Detection

features	WERR	EEPR	P50	P90	P99
$[T_{\text{min}} = 400, T_{\text{max}} = 1500]$					
$[d_t]$	–	–	380	720	1500
$[h_t]$	-17%	-68%	380	1500	1510
$[a_t]$	-11%	-43%	380	750	1500
$[a_t, d_t]$	-15%	-54%	370	730	1500
$[a_t, h_t]$	-16%	-61%	360	760	1500
$[a_t, h_t, d_t]$	-16%	-59%	360	720	1500

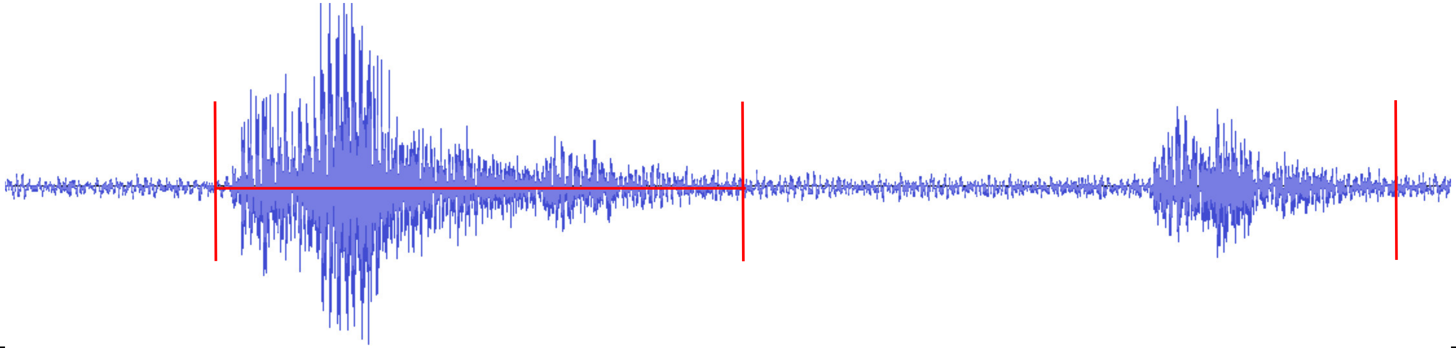
[Combining Acoustic Embeddings and Decoding Features for End-of-Utterance Detection in Real-Time Far-Field Speech Recognition System. *R. Maas et.al., Amazon, ICASSP 2018*]

Combining Wakeword and End-Point Detection

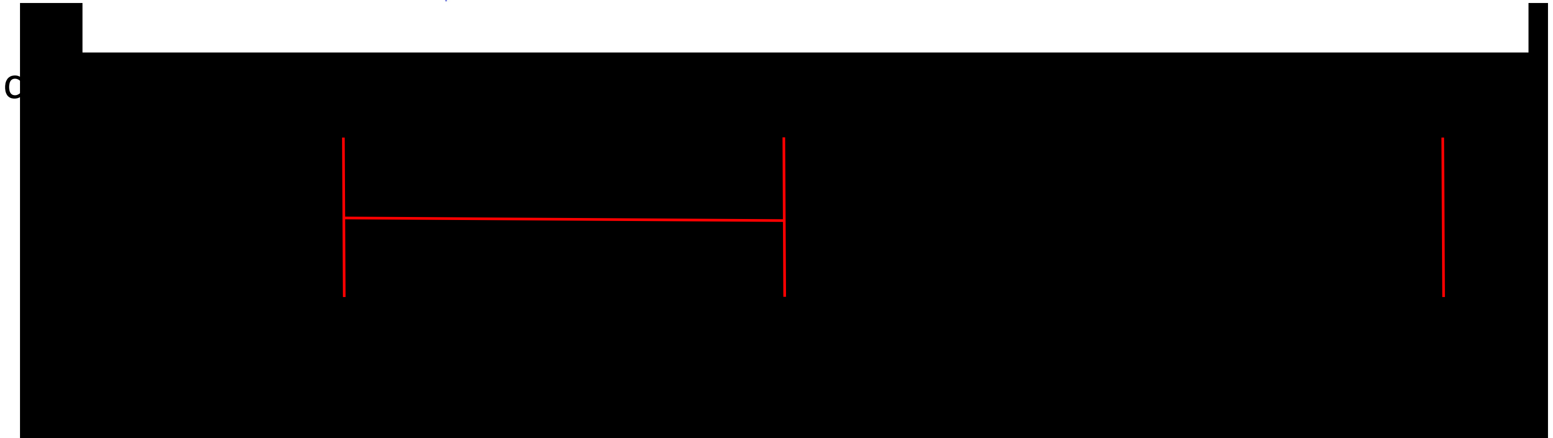
close talk



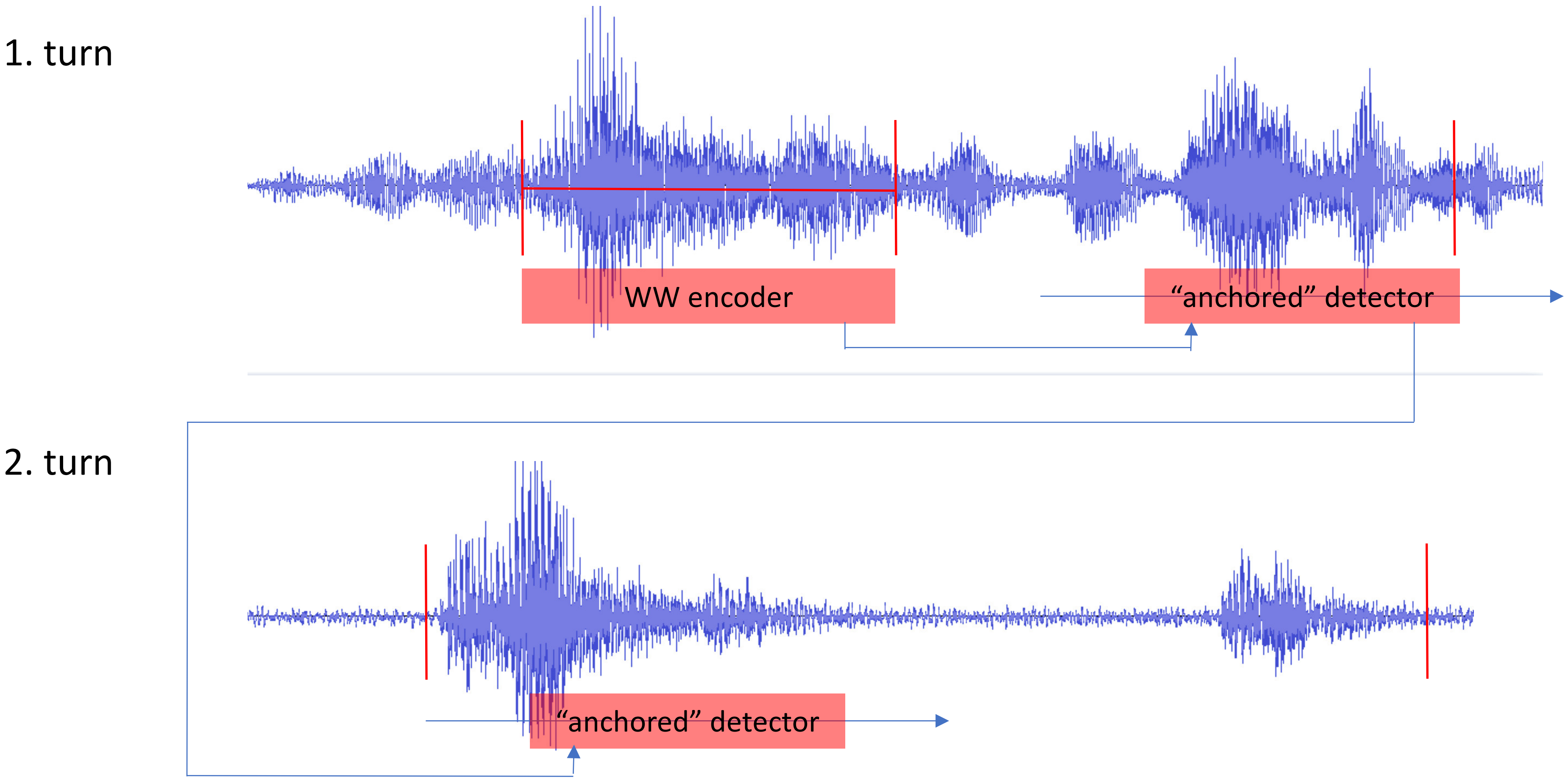
distant speech



distant with bac[redacted]



Anchored End-Point Detection



[Anchored Speech Detection, *R. Maas et.al., Amazon, Interspeech 2016*]

Anchored End-Point Detection

Desired vs interfering speech classification

- Frame error rate [%]
- LFBE: input to encoder/decoder
- LFBE+MS: causal mean subtraction applied to LFBE features
- LFBE+AS: ”anchored” mean subtraction (mean computed over wake word)

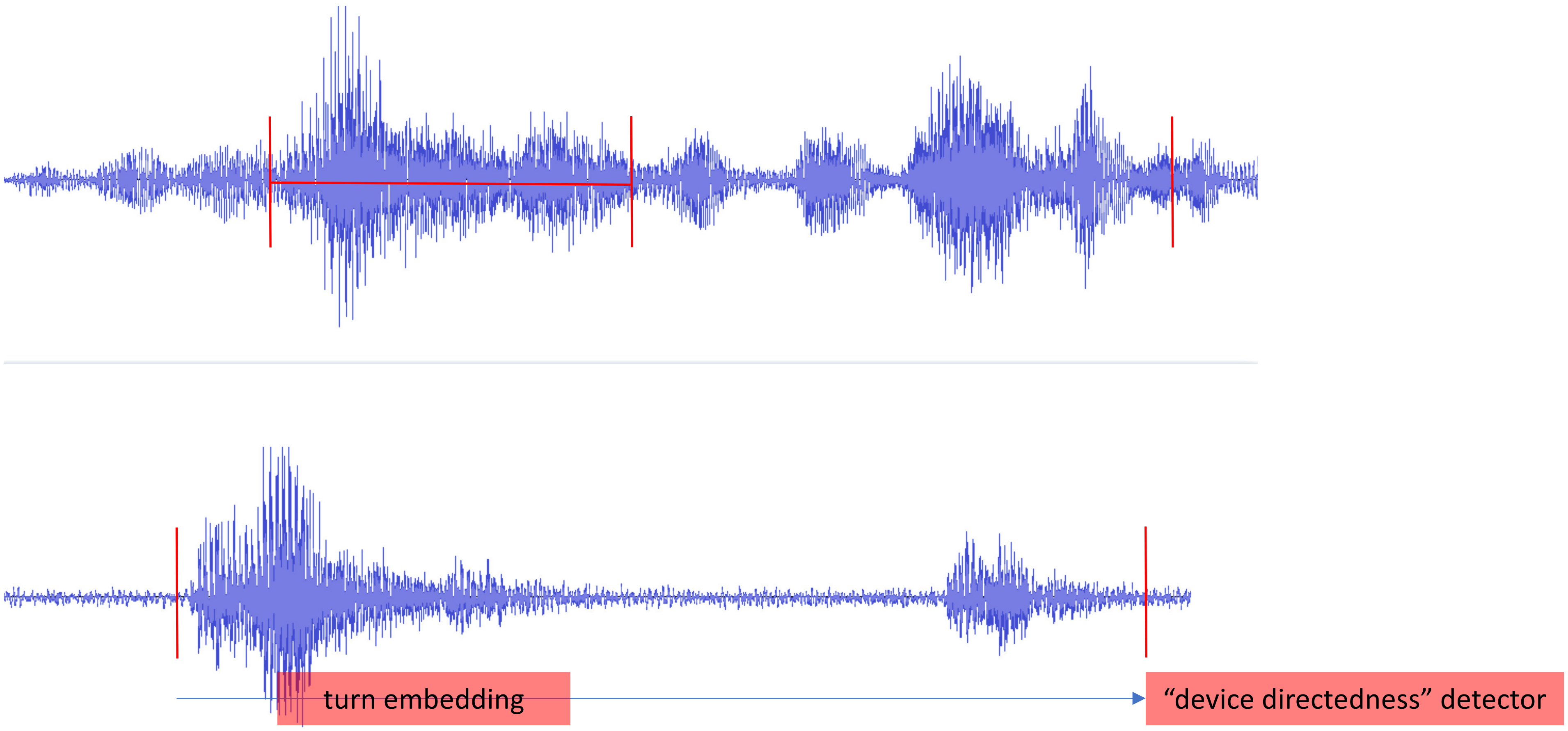
Encoder	Decoder	raw LFBE	LFBE +MS	LFBE +AS
None	FF	19.4	17.2	**15.4**
None	RNN	19.5	17.3	15.5
LSTM	FF	**15.7**	**15.2**	**15.2**
LSTM	RNN	15.8	15.4	15.6

[Anchored Speech Detection, *R. Maas et.al., Amazon, Interspeech 2016*]

2nd Turn Device Directedness Detection

1. turn

2. turn

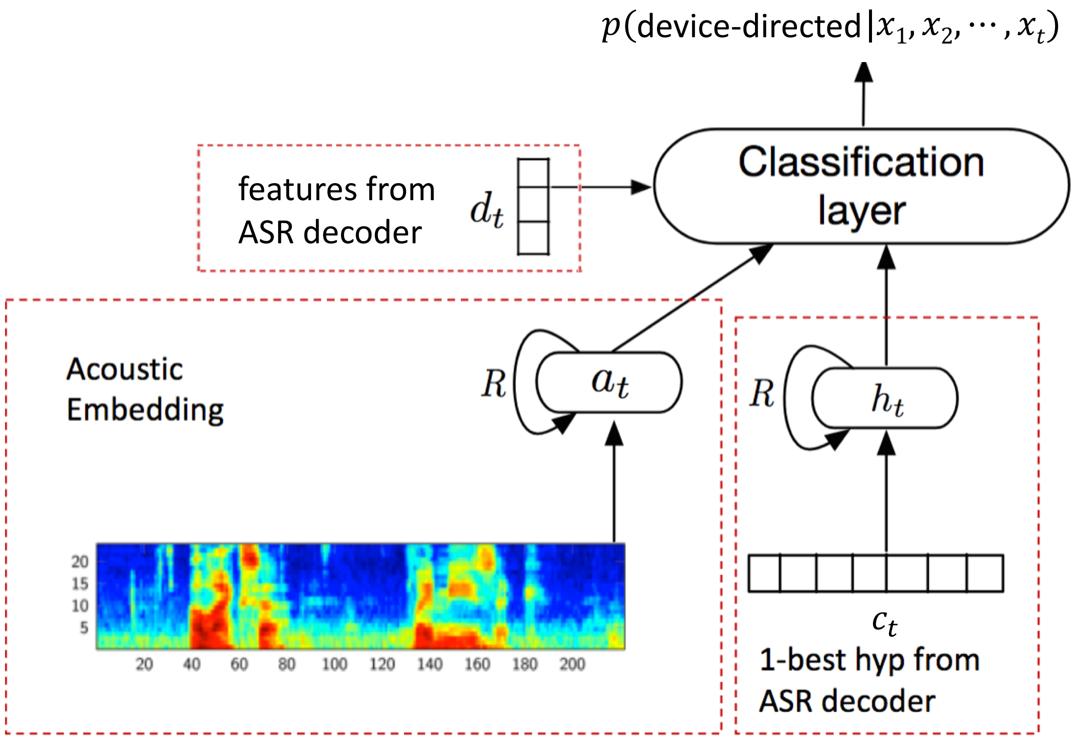


amazon alexa

Hybrid Device Directedness Detection

"Follow-up mode"
- Second interaction without wake word
- Example:
  "Alexa, set alarm for 7am"
  "What's the weather tomorrow?"

Hybrid device-directedness detector
- Features similar to hybrid end-point detector
  - acoustic embedding
  - decoder features
    Viterbi score, avg. token confidence,
    avg. arcs in CN, etc.
  - lexical embedding
    embedding over 1-best character sequence
- DNN classifier



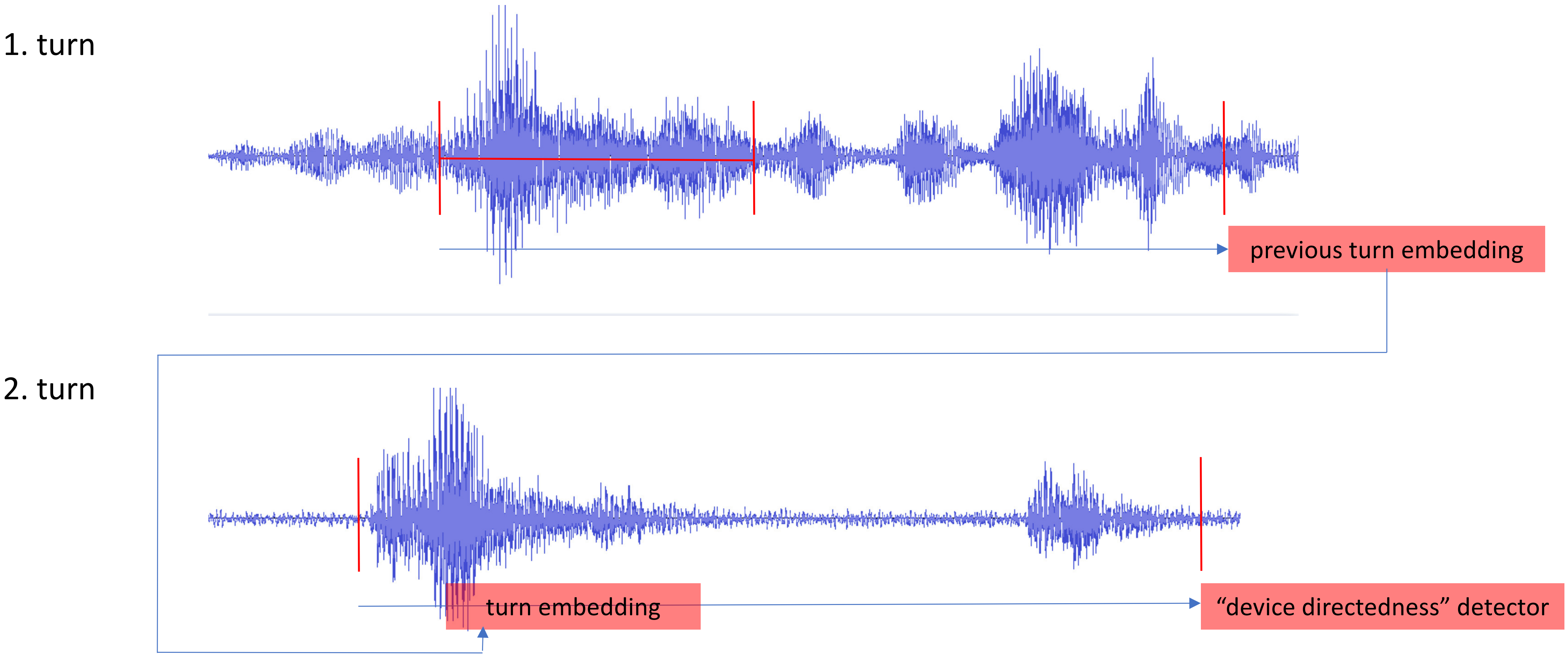
[DeviceDirected Utterance Detection, *S.H. Mallidi et.al., Amazon, Interspeech 2018*]

Hybrid Device Directedness Detection

features	EER(%)
decoder features ($\mathbf{d}$)	9.3
acoustic embedding ($\mathbf{a}$)	10.9
char embedding ($\mathbf{c}$)	20.1
$[\mathbf{a}, \mathbf{d}]$	6.5
$[\mathbf{c}, \mathbf{d}]$	6.9
$[\mathbf{a}, \mathbf{c}]$	8.6
$[\mathbf{a}, \mathbf{c}, \mathbf{d}]$	5.2

[Device Directed Utterance Detection, *S.H. Mallidi et.al., Amazon, Interspeech 2018*]

Anchored Device Directedness Detection



[*unpublished*]

Outline

Overview

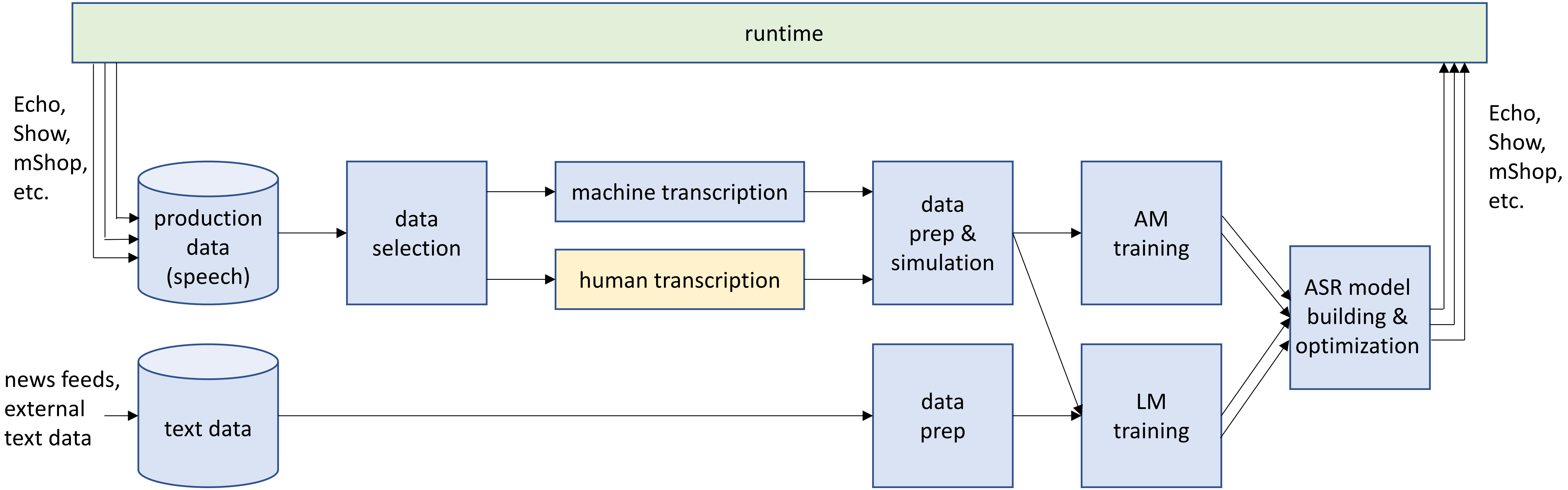
Containing Speech

- Wakeword Detection
- End-of-Speech Detection
- Combining Wakeword and End-of-Speech Detection
- Device-Directedness Detection

Recognizing Speech

- Active Learning
- Multi-lingual and low-resource ASR
- Context Modeling

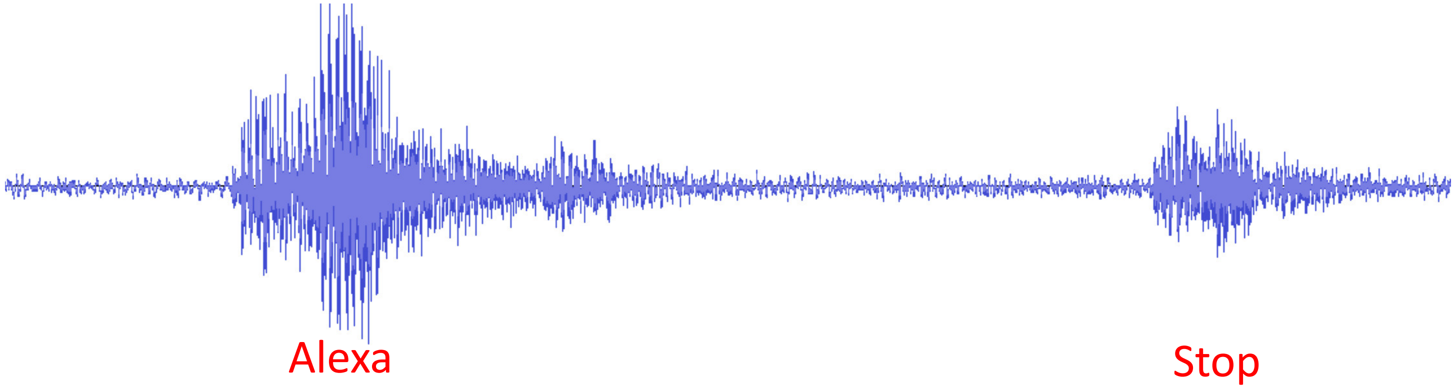
Active Learning for ASR



Active Learning for ASR

Active Learning for ASR
- What are my features (derived from current model)?
- What is the optimal distribution over features?
- What is the optimal distribution over human vs machine transcription?
- How to find the subset yielding the desired distribution?

Utterance features:
- Device type
- Domain/Intent (NLU)
- Phoneme/Triphone distribution
- SNR
- Confidence
- Acoustic embedding (i-vector)
- Transcription occurrence
  (how many "Alexa Stop", etc.)



Uniform Phoneme Distribution

Phoneme distribution:

- Skewed distribution: ”Alexa stop”, “Alexa, what’s the weather”, etc.
- “what’s” vs “watch”, “repeat” vs “reheat”, etc.
- Target distribution?
  => Has to work everywhere (message dictation, contact names, skills, etc.)
  => Uniform distribution (Maximum entropy principle)

Uniform Phoneme Distribution

Data sub-selection

	Random Selection [WERR%]	**Uniform phoneme dist. [WERR%]**
Full (3.8K hours)	-	-
Half (1.9K hours)	-4%	1%
Third (1.15 hours)	-8%	-2%

WERR := relative reduction in WER

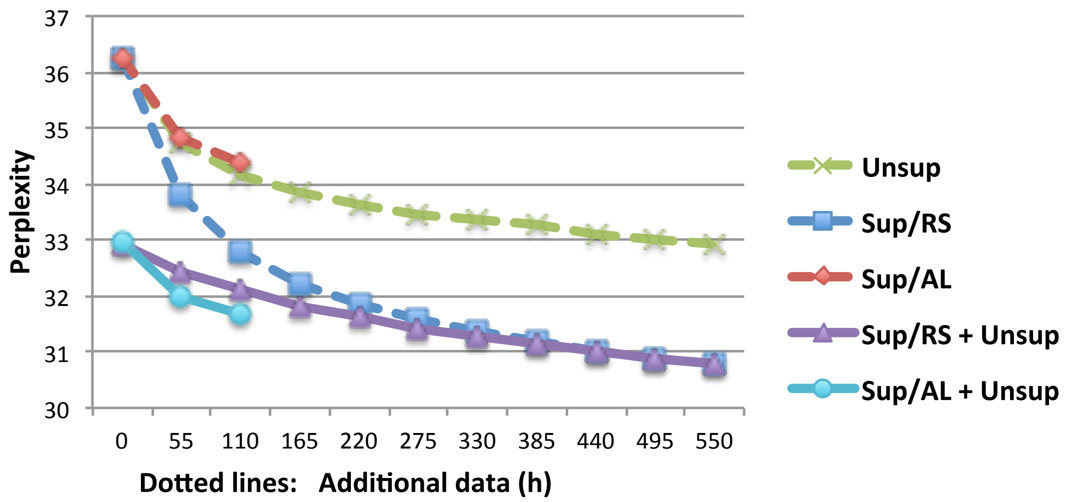
Active Learning:

- Use criterion for selecting data for transcription
- Require only 1/3 of data (need to trust semi-supervised labels)

Active and Semi-Supervised Learning for LM

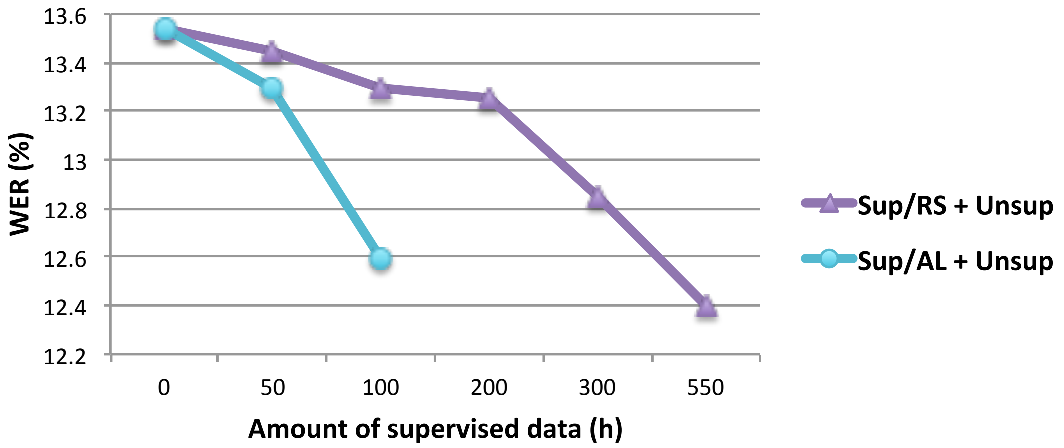
Experimental setup

- Baseline: 50h random selection
  => trainer for semi-supervised learning
  => supervised portion
- AL pool: 100h confidence based selection
- RS pool: 550h random selection



Conclusion

1. Using all data helps
   - combine human and machine transcription
2. Active learning helps
   - "smart" selection what to send to human transcription



[Active and Semi-Supervised Learning in ASR: Benefits on the Acoustic and Language Models, *Th. Drugman et.al., Amazon, Interspeech 2016*]

Active Learning with Sub-modular Functions

Data selection with sub-modular functions

- Function with diminishing return property

$$A \subseteq B \text{ and } v \notin B \Longrightarrow f(A \cup \{v\}) - f(A) \geq f(B \cup \{v\}) - f(B)$$

- Linear greedy algorithm with certain optimality guarantees

$$\tilde{V} := \underset{V \subseteq S, |V| = B}{arg\max}\ f(V) \ \text{ for given budget } B$$

Sub-modular function for feature-based data selection

- Relevance function for feature i

$$r_i(V) \coloneqq \sum_{v \in V} r_i(v)$$

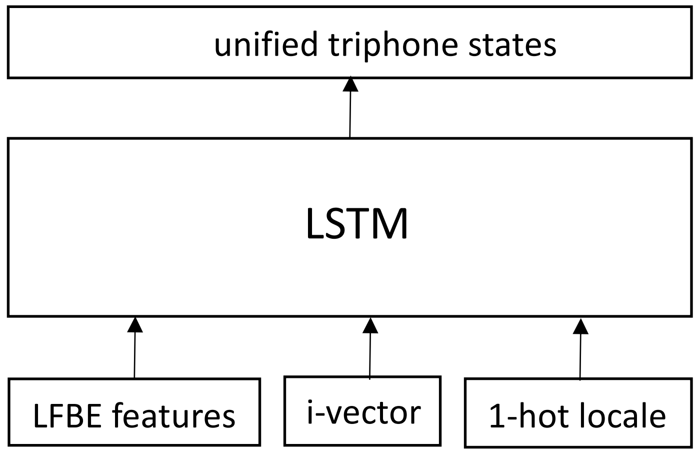
- Sub-modular feature function with feature weights w_i and concave function ϕ

$$f(V) \coloneqq \sum_i w_i \phi(r_i(V))$$

- $\phi \coloneqq log, w_i$:= “target distribution” => entropy-based selection
  Example: $r_\pi(v)$ occurence of phoneme π in utterance v, w_π desired phoneme distribution

[Submodular subset selection for for large scale speech training data, *!"# $ %&#% '"()"*#+ $ *#, - '%. /0%%12 3456*]

Multi-dialect Acoustic Modeling

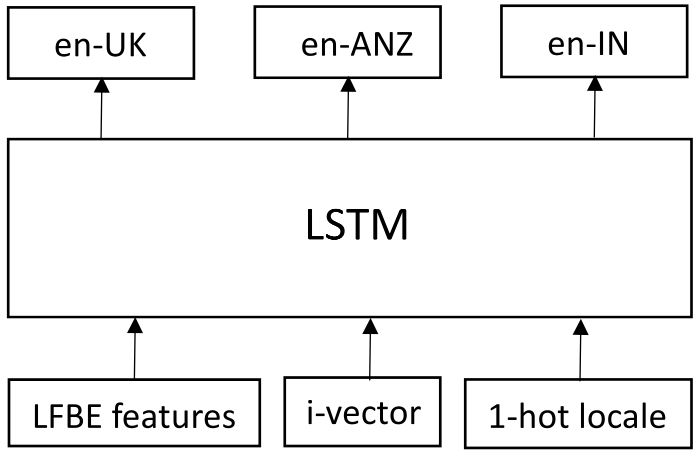


Multiple British-English dialect modeling

- Problem: skewed data distribution

Unified AM

- Pooled AM (LSTM, XENT + bMMI)
- Adapt to locale (bMMI)



Unified AM with locale and speaker embedding

- Additional input to AM
  - speaker embedding: frame-wise updated i-vector
  - locale embedding: one-hot vector
- Build unified AM
  - add locale-specific last layers (work in progress)

Multi-dialect Acoustic Modeling

Training data for British-English locales

	training data [hours]
en-GB	3.2K
en-IN	1.5K
en-ANZ	0.7K
total	**5.4K**

Data pooling for British-English locales

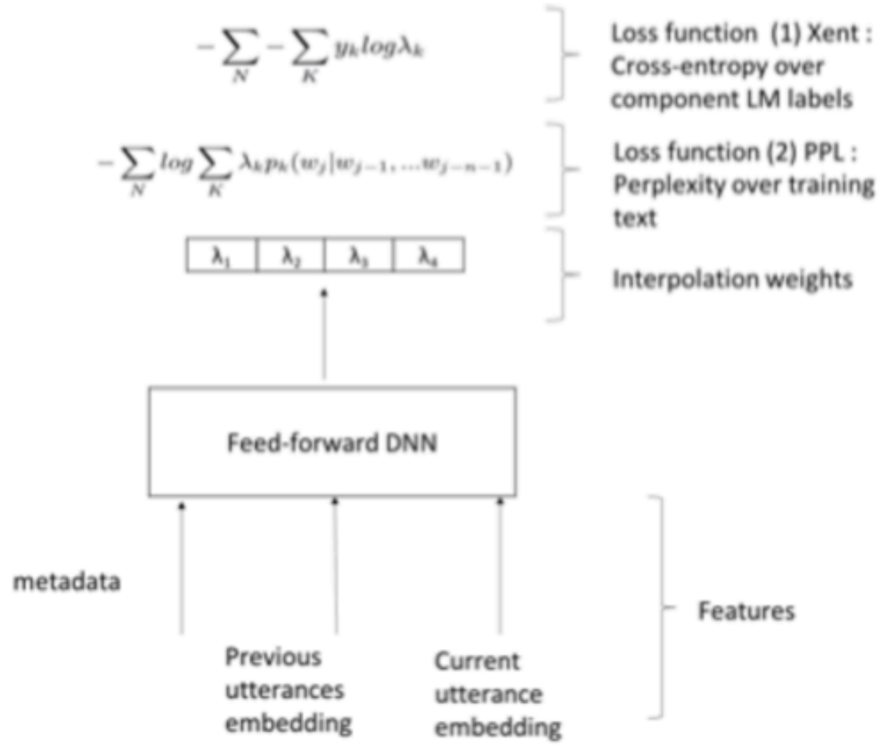
	en-GB [WERR%]	**en-IN [WERR%]**	**en-ANZ [WERR%]**
Pooled AM	-11%	-13%	-16%
+ speaker and locale embedding			-23%

WERR := relative change in WER

Contextual Language Model Adaptation

Contextual LM for a chatbot

- Unsupervised clustering of LM training data
  => 26 “topic” LMs
- Linear interpolation of “topic” LMs
- Predict interpolation weights from
  - previous utterance (1-pass)
  - current utterance (2-pass)
- Optimize predictor MLP (2x200) for
  - unsupervised topic label
  - perplexity



Features

- prev: average word embedding over all past turns
  prev-d: average with decaying weight
- cur: average word embedding over 1-best
- meta: day of week, time of day

[Contextual Language Model Adaptation for Conversational Agents, *A. Raju et.al., Amazon, Interspeech 2018*]

Contextual Language Model Adaptation

Chatbot ASR system on a Chatbot test set (Alexa Prize)

Model	Feats	PPL	WERR(%)	Entity WERR(%)	
		decoder : 1-pass			
No Adapt	-	60.77	-	-	
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
N (PPL)	prev, meta	58.14	-1.61%	-2.98%	D
N (PPL)	prev-d, meta	55.66	-2.76%	-10.92%	D
		decoder: 2-pass			
N (PPL)	prev, cur, meta	42.03	-5.58%	-15.15%	D
N (PPL)	prev-d, cur, meta	42.83	-5.92%	-14.67%	D
N(PPL)	cur, meta	42.72	-5.98%	-15.32%	D
pic model	cur	45.08	-5.52%	-13.14%	To

[Contextual Language Model Adaptation for Conversational Agents, *A. Raju et.al., Amazon, Interspeech 2018*]

Thanks

https://developer.amazon.com/alexa/science