CS 224S / Linguist 285
Spoken Language Processing

Andrew Maas | Stanford University | Spring 2025

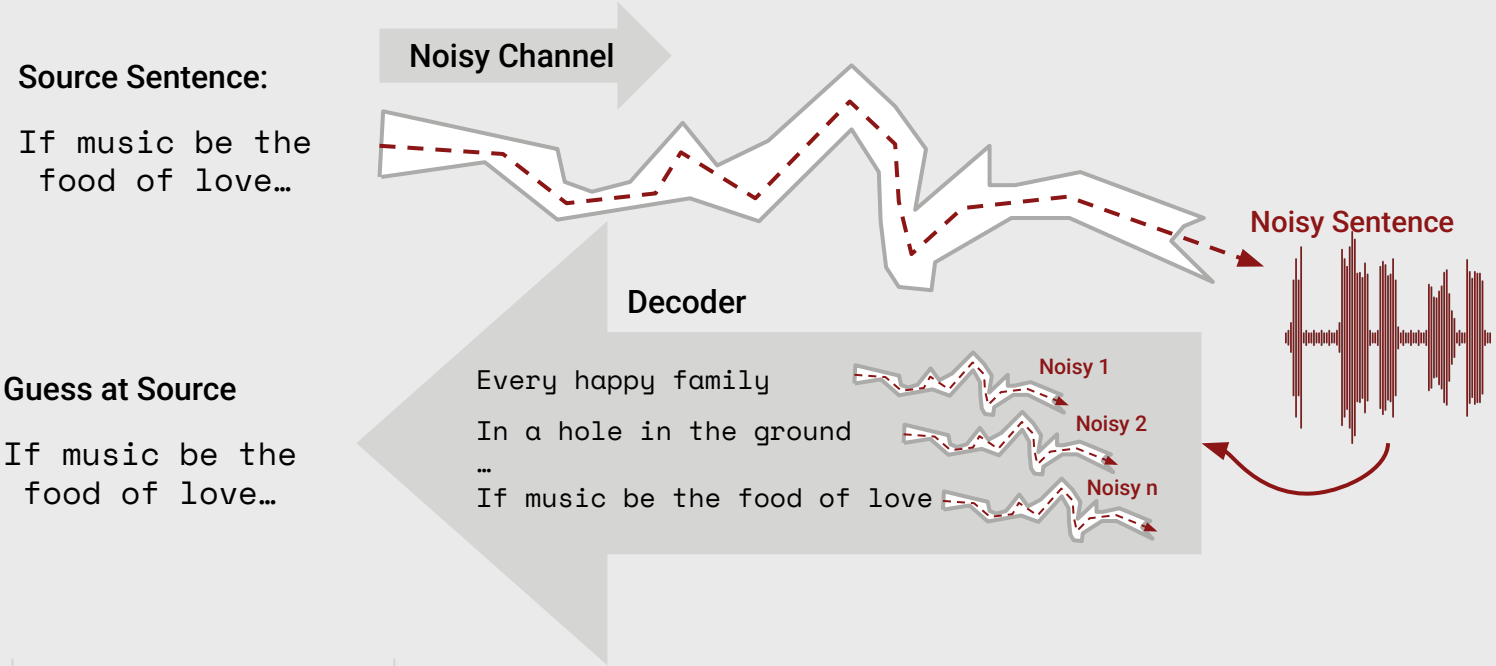
Lecture 10: Speech recognition with neural networks. HMM-DNN and CTC loss function

Outline

- **Deep neural network acoustic models and HMM-DNN ASR**
- **Connectionist temporal classification (CTC) - based systems**
  - CTC loss function & inference computation
  - Lexicon-free CTC
- RNN-Transducer loss+model as an improvement over CTC assumptions

The Noisy Channel Model

- Search through space of all possible sentences.
- Pick the one that is most probable given the waveform



The Noisy Channel Model

- Probabilistic implication: Pick the highest prob word sequence W:

$$\hat{W} = \arg\max_{W \in L} P(W \mid O)$$

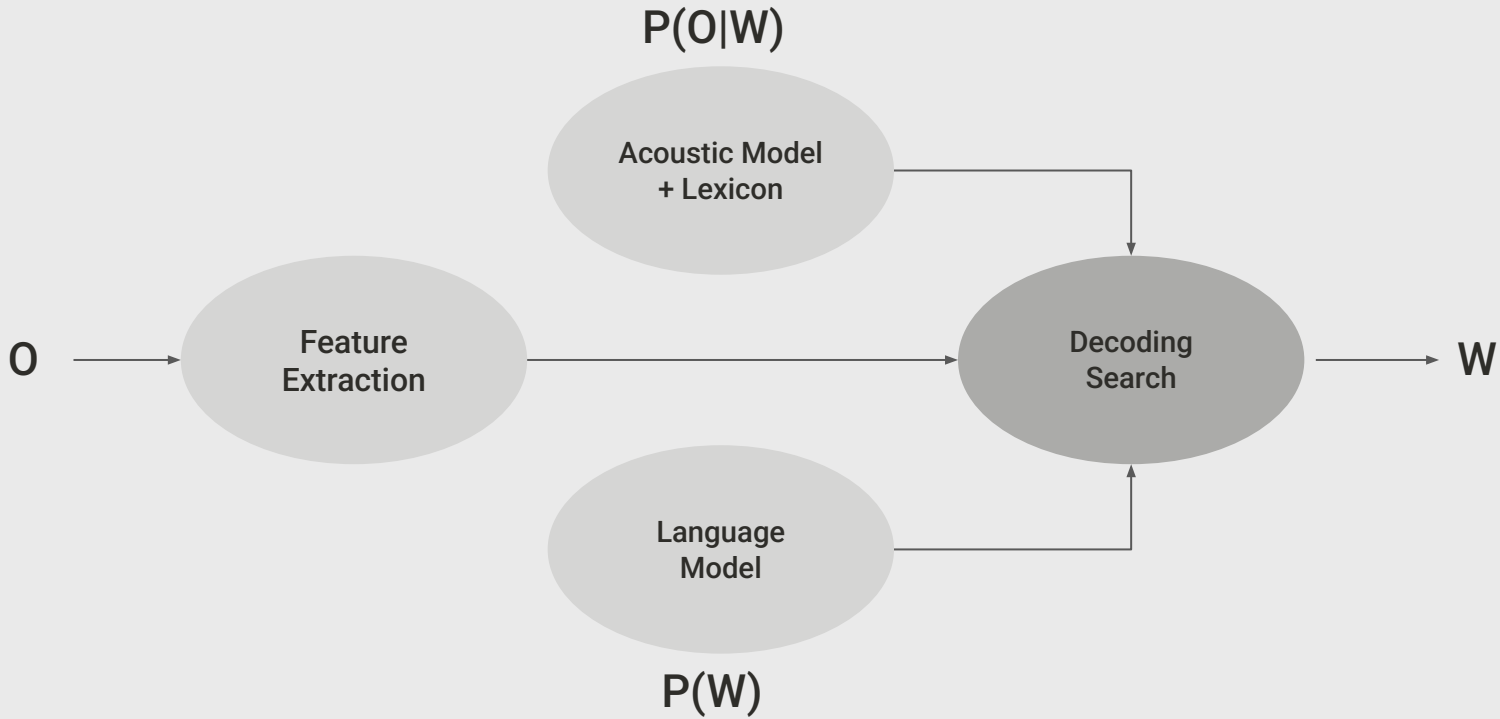
- We can use Bayes rule to rewrite this:

$$\hat{W} = \arg\max_{W \in L} \frac{P(O \mid W)P(W)}{P(O)}$$

- Since denominator is the same for each candidate sentence W, we can ignore it for the argmax:

$$\hat{W} = \arg\max_{W \in L} P(O \mid W)P(W)$$

Speech Architecture Meets Noisy Channel

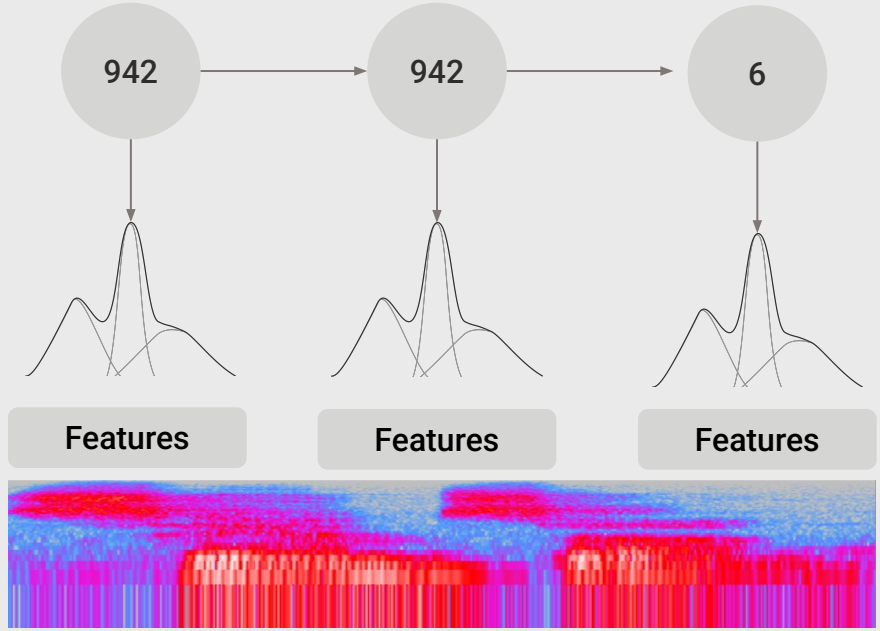


HMM-GMM ASR Model

Transcription: **Samson**

Pronunciation: **S – AE – M – S –AH – N**

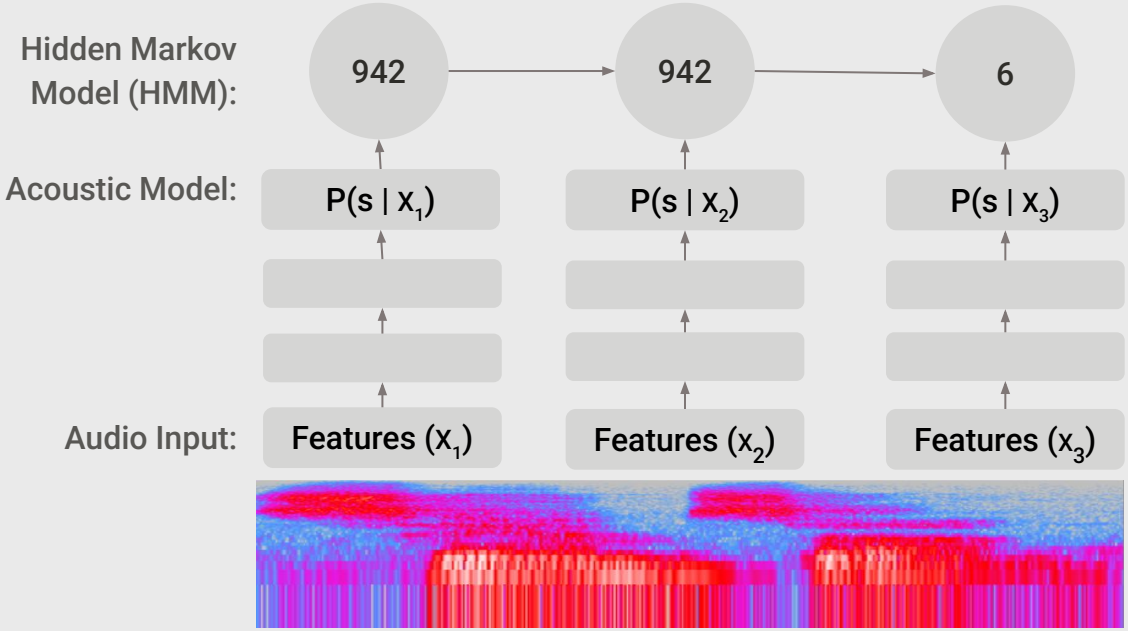
Sub-phones: **942 – 6 – 37 – 8006 – 4422 …**



GMM models:
P(o|q)
o: input features
q: HMM state

DNN Hybrid Acoustic Models

Transcription: Samson
Pronunciation: S – AE – M – S –AH – N
Sub-phones: 942 – 6 – 37 – 8006 – 4422 …



Use a DNN to approximate:
P(s|x)

Apply Bayes' Rule:
P(x|s) = P(s|x) * P(x) / P(s)

DNN * Constant / State prior

HMM-DNN ASR

Objective function for DNN training: per-frame classification

- Supervised learning, minimize our classification errors
- Standard choice: Cross entropy loss function
  - Straightforward extension of logistic loss for binary

$$Loss(x, y; W, b) = -\sum_{k=1}^{K} (y = k) \log f(x)_k$$

- This is a **frame-wise** loss. We use a label for each frame from a forced alignment
- Other loss functions possible. Can get deeper integration with the HMM or word error rate

Neural nets for acoustic modeling goes back to early speech

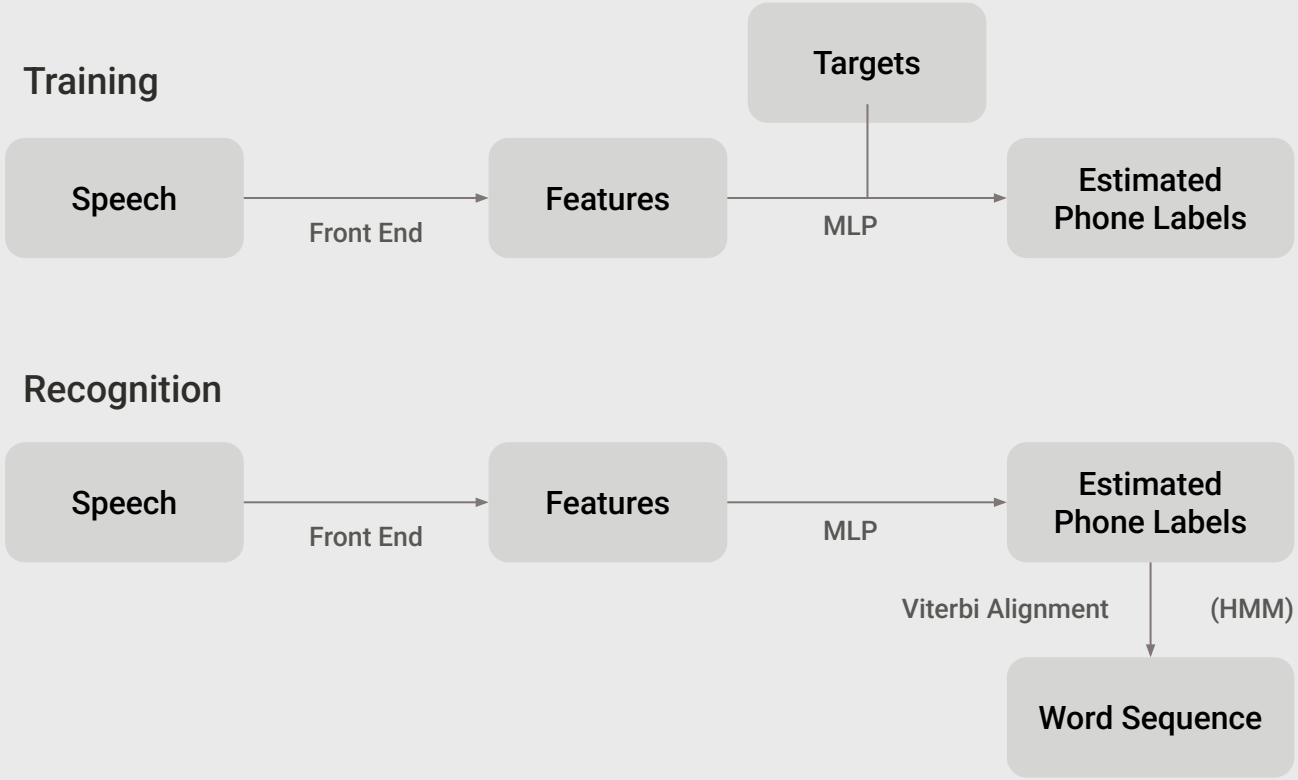
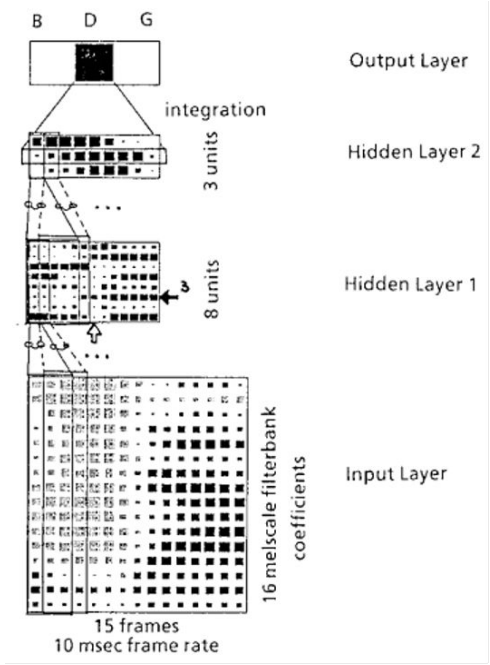


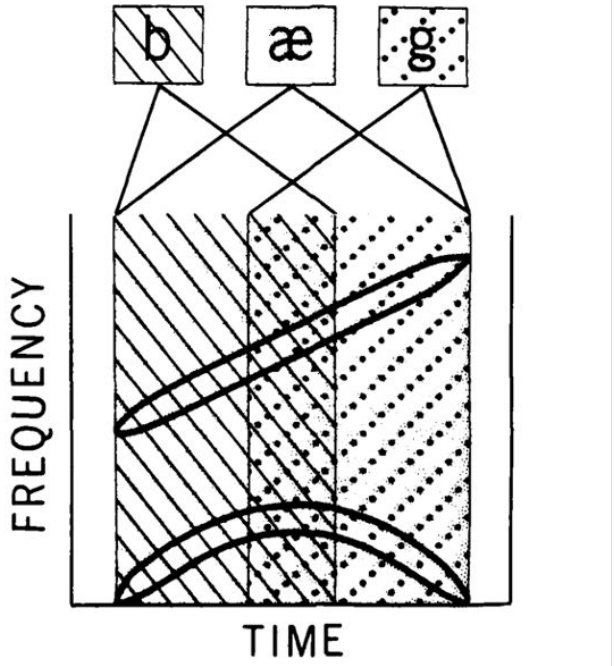
Figure: Renals, Morgan, Bourland, Cohen, & Franco. (1994)

Early neural network approaches

Speech tasks motivated some of the original backpropagation work



Waibel, Hanazawa, Hinton, Shikano, & Lang. 1988



McClelland, & Elman. 1985

Hybrid MLPs on Resource Management dataset in 1994

	% error		
Test Set	CI-MLP	CD-HMM	MIX
Feb 91	5.8	3.8	3.2
Sep 92a	10.9	10.1	7.7
Sep 92b	9.5	7.0	5.7

TABLE II
RESULTS USING THE THREE TEST SETS
USING NO GRAMMAR (PERLPEXITY 991)

	% error		
Test Set	CI-MLP	CD-HMM	MIX
Feb 91	24.7	19.3	15.9
Sep 92a	31.5	29.2	25.4
Sep 92b	30.9	26.6	21.5

Table: Results using the three test sets with the perplexity 60 wordpair grammar. (CI-MLP is the context-independent MLP-HMM hybrid system, CD-HMM is the full context-dependent Decipher system, and the MIX system is a simple interpolation between the CD-HMM and the CI-MLP.)

(Renals, Morgan, Bourland, Cohen, & Franco. 1994)

Hybrid systems took over ASR around 2012-2015

[TABLE 3] A COMPARISON OF THE PERCENTAGE WERs USING DNN-HMMs AND GMM-HMMs ON FIVE DIFFERENT LARGE VOCABULARY TASKS.

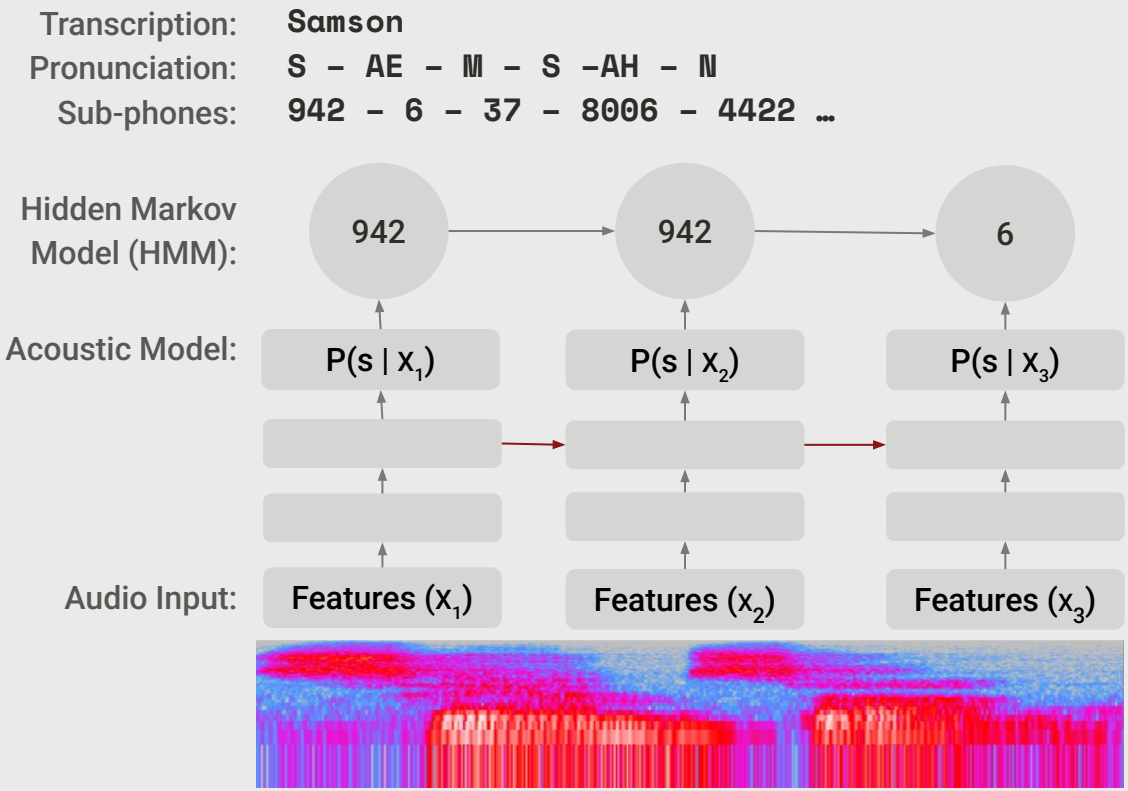
TASK	HOURS OF TRAINING DATA	DNN-HMM	GMM-HMM WITH SAME DATA	GMM-HMM WITH MORE DATA
SWITCHBOARD (TEST SET 1)	309	18.5	27.4	18.6 (2,000 H)
SWITCHBOARD (TEST SET 2)	309	16.1	23.6	17.1 (2,000 H)
ENGLISH BROADCAST NEWS	50	17.5	18.8	
BING VOICE SEARCH (SENTENCE ERROR RATES)	24	30.4	36.2	
GOOGLE VOICE INPUT	5,870	12.3		16.0 (>> 5,870 H)
YOUTUBE	1,400	47.6	52.3	

Hinton et al. 2012

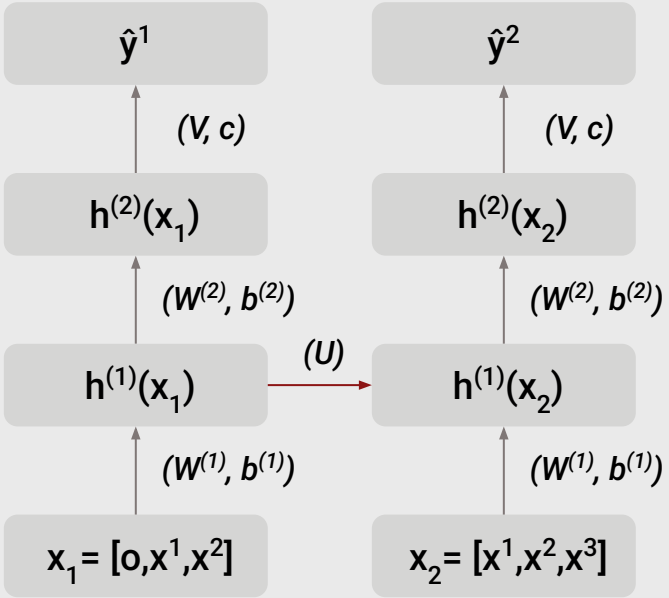
What's made modern DNNs successful?

- Context-dependent HMM states
- Deeper nets improve on single hidden layer nets
- Hidden unit nonlinearity
- Many more model parameters (scaling up) and large datasets
- Specific depth (e.g. 3 vs 7 hidden layers)
- Fast computers = run many experiments
- Architecture choices (easiest is replacing sigmoid)
- Pre-training does not matter*
  - Initially we thought this was the new trick that made things work

Recurrent DNN hybrid acoustic models



Deep recurrent networks



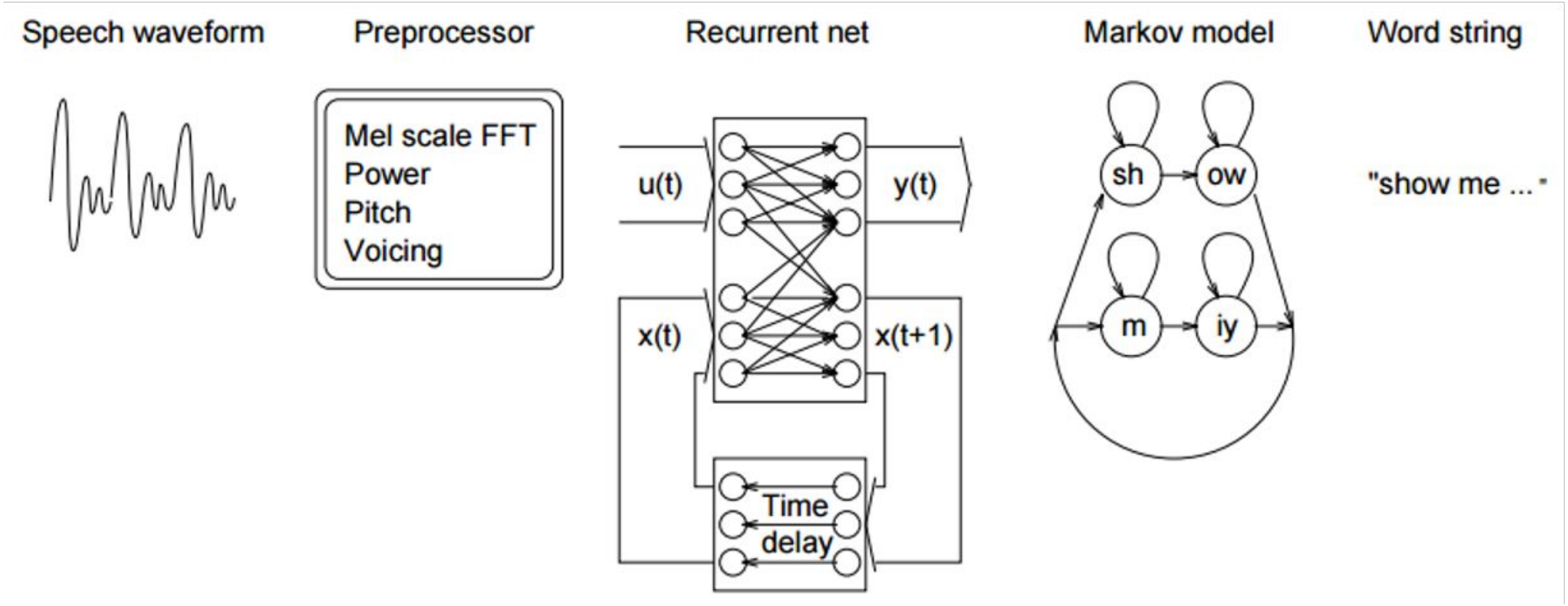
Output layer: $\hat{y}^2 = Vh^{(1)}(x_2) + c$

Hidden layer: $h^{(2)}(x_2) = \sigma\,(W^{(2)}h^{(1)}(x_2) + b^{(3)})$

Hidden layer: $h^{(1)}(x_2) = \sigma\,(W^{(1)}x_2 + b^{(1)} + Uh^{(1)}(x_1))$

Input: x

RNNs for acoustic modeling in 1996



(Robinson, Hochberg, & Renals, 1996)

Adding more parameters 25 years ago

Size matters: An empirical study of neural network training for LVCSR. Ellis & Morgan. ICASSP. 1999.

Hybrid NN. 1 hidden layer. 54 HMM states.

74hr broadcast news task

"…improvements are almost always obtained by increasing either or both of the amount of training data or the number of network parameters … We are now planning to train an 8000 hidden unit net on 150 hours of data … this training will require over three weeks of computation."

Scaling Total Parameters on Switchboard in 2015

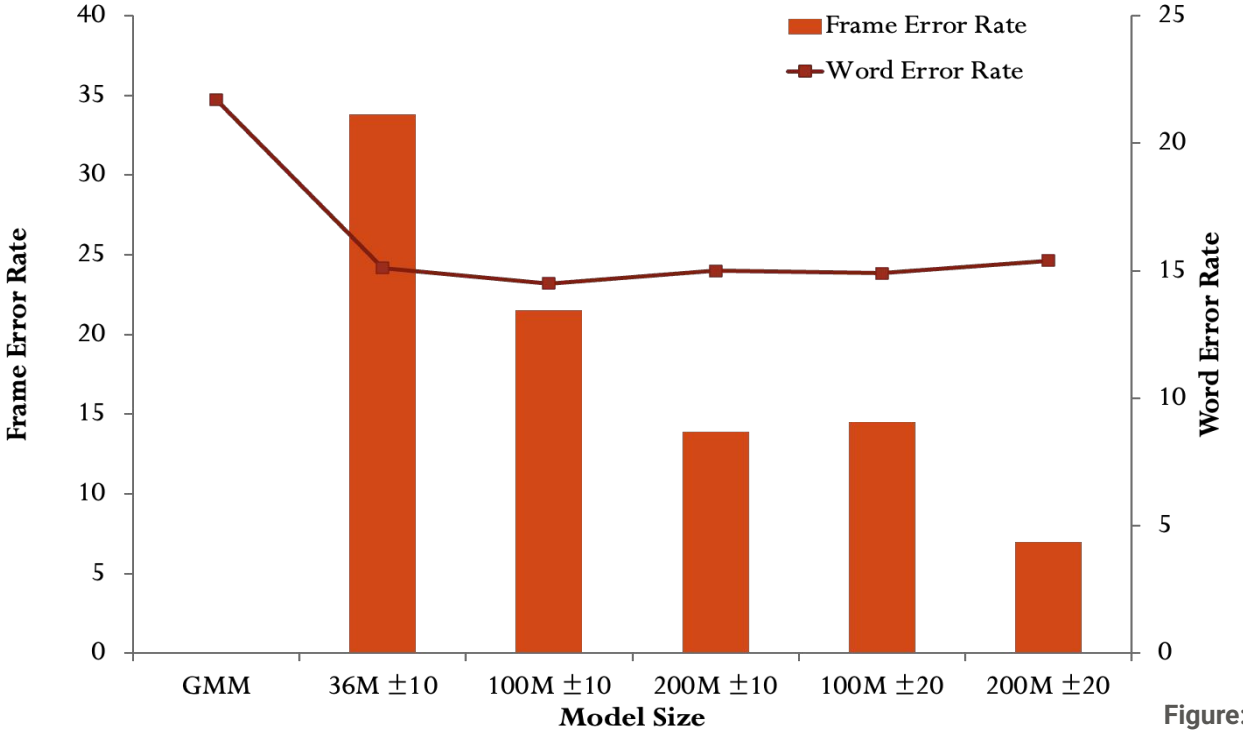
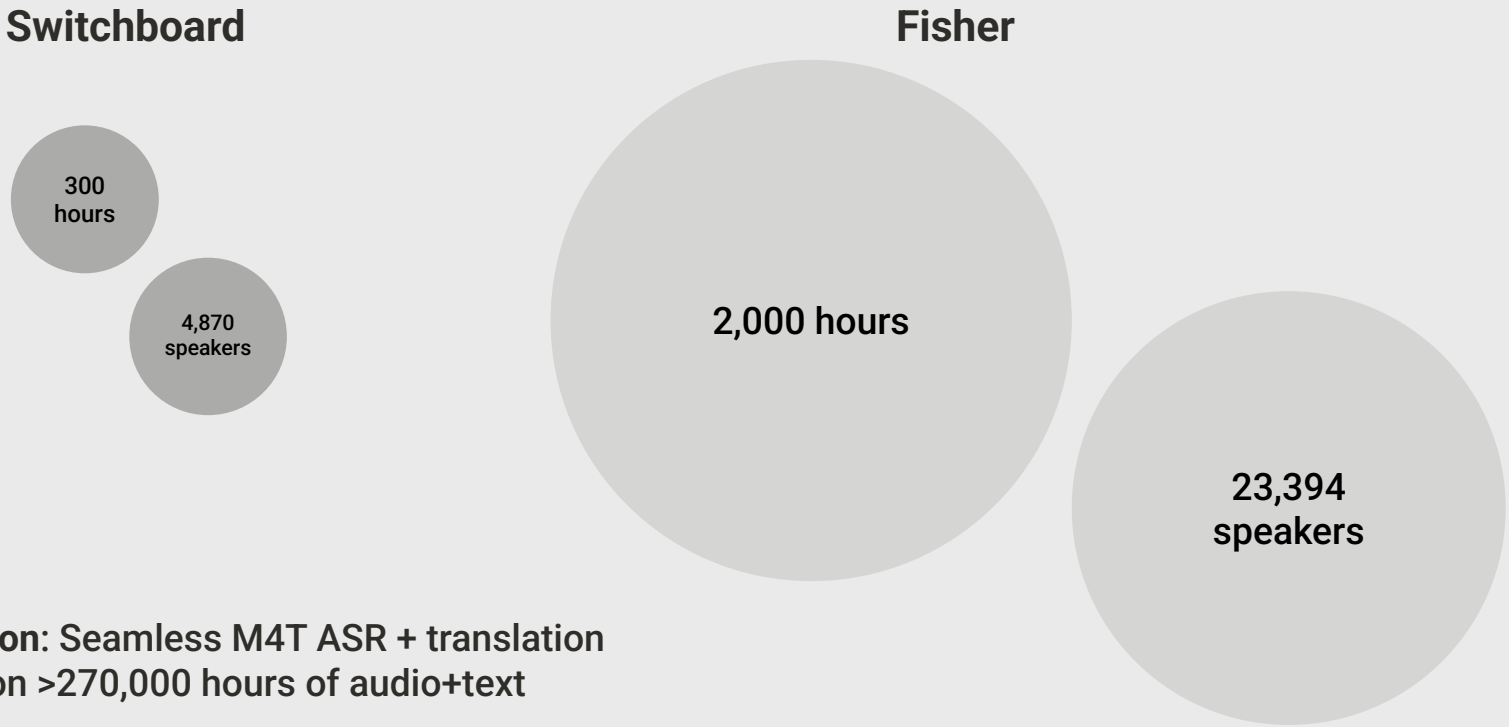


Figure: Maas, Qi, Xie, Hannun, Lengerich, Jurafsky, & Ng. (2017)

Combining speech training corpora in 2015



2025 comparison: Seamless M4T ASR + translation model trained on >270,000 hours of audio+text

HMM-DNN framework + per-frame training limitations

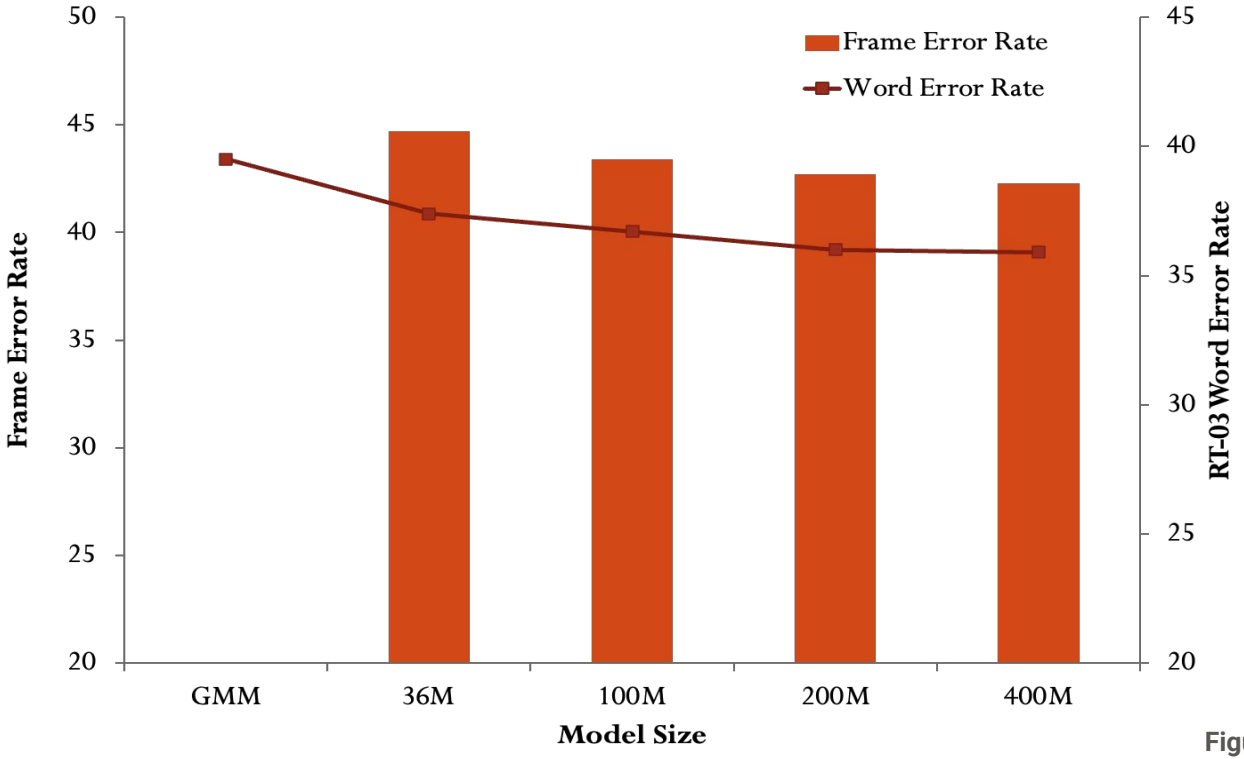
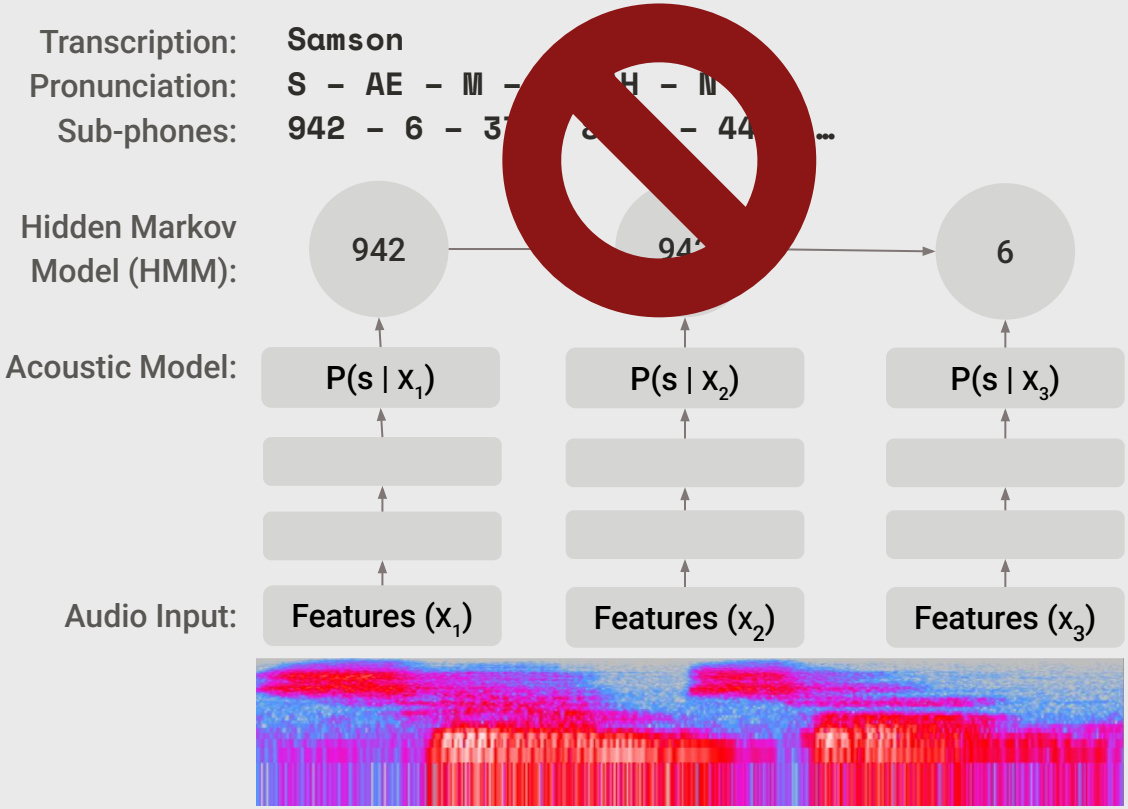


Figure: Maas, Qi, Xie, Hannun, Lengerich, Jurafsky, & Ng. (2017)

How can we improve upon HMM-DNN?

- Lexicon introduces too many pronunciation assumptions. Requires hand engineering
- Iteratively building HMM systems requires complex “recipes” to progressively improve alignments + acoustic models
- HMM-DNN systems perform better, but still require the above
  - … can we use deep learning approaches to replace the HMM-based approaches so far?

HMM-Free Recognition



Use a DNN to approximate:
P(s|x)

Apply Bayes' Rule:
P(x|s) = P(s|x) * P(x) / P(s)

DNN * Constant / State prior

Connectionist Temporal Classification (CTC)

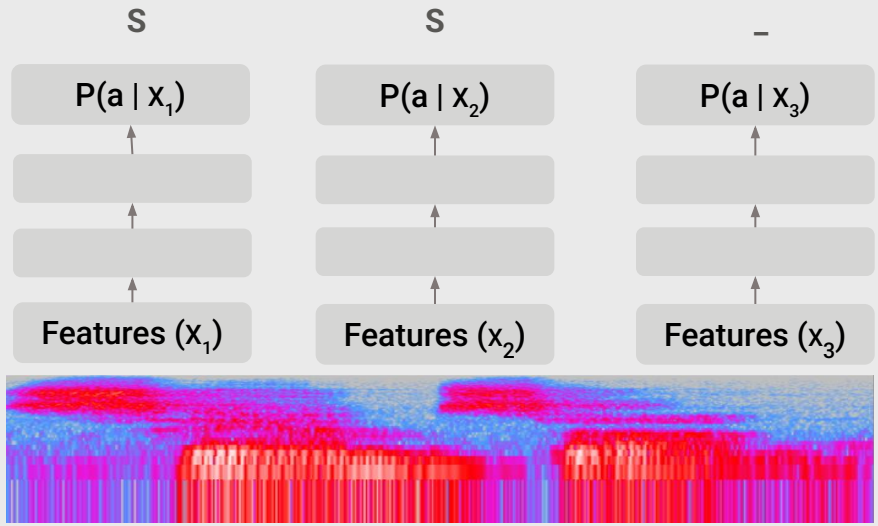
Directly estimating transcript token outputs from audio inputs

HMM-Free Recognition with CTC

Transcription: Samson

Characters: S A M S O N

Collapsing function: SS___AA_M_S___O___NNNN



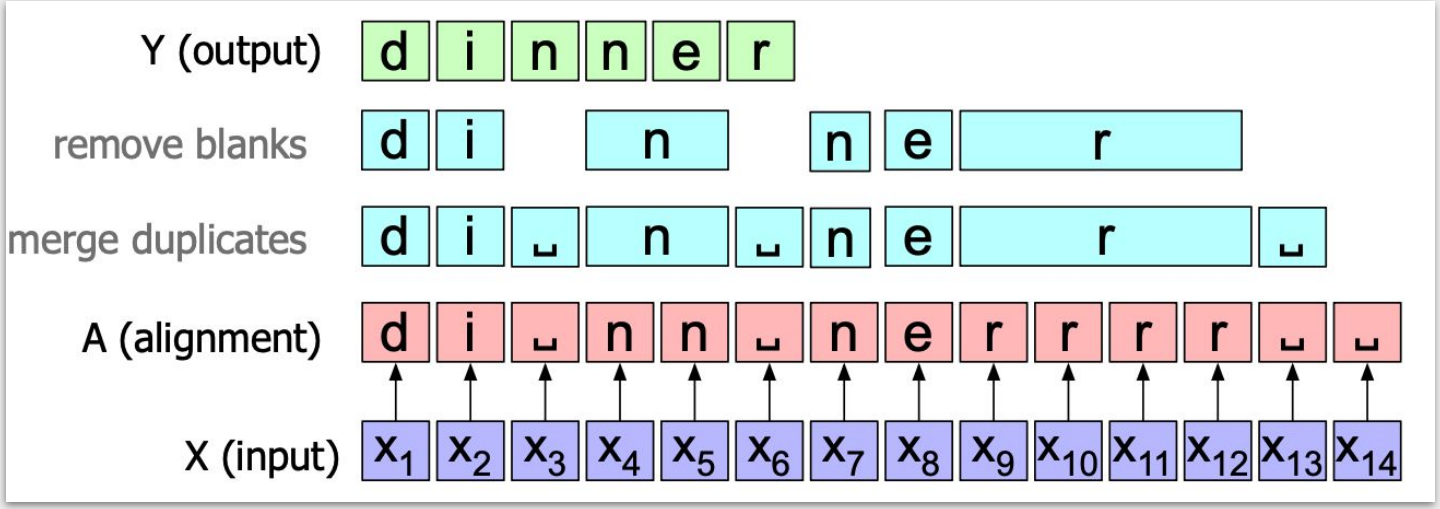
Use a DNN to approximate: P(a|x)

The distribution over *characters*

CTC output assumptions

Each timestep *t* the model outputs $P(a \mid x_1)$ which can either be an output token (character, phoneme) or "blank"

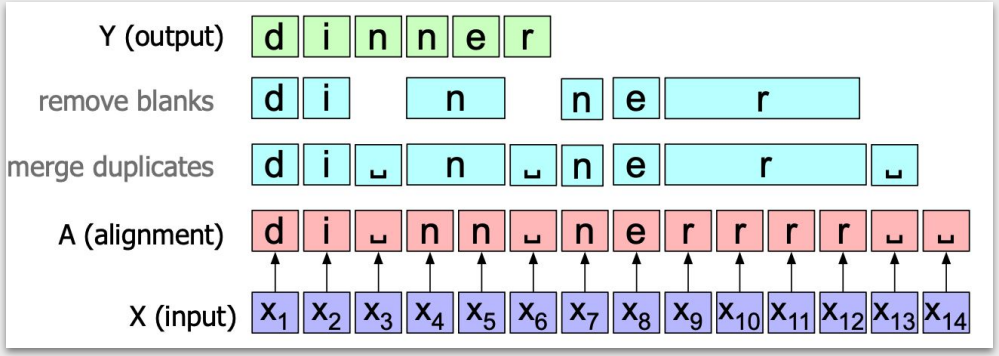
Assume: input sequence (length T) >= output sequence length (safe assumption in ASR)
Collapsing rule: Ignore all blank tokens, and ignore repeated outputs of the same token



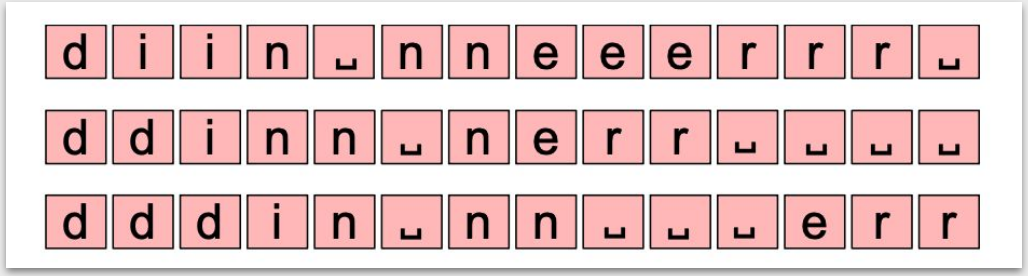
CTC collapsing. Sum over all possible alignments

Many alignments produce the same transcript output.

P(transcript) = sum over P(alignment) for all alignments consistent with transcript



Additional alignments for ‘dinner’



… (more alignments possible)

Real example. Full utterance with blank token (_)

Per-frame argmax:

```
yy__ee_________tt_ _____________________________________________a_____
_rr__e_________hh_________b___ii_______lll__i_____tt_____aa_____tt_______iio__n___
___cc_____rrr_u_____________________ _________ii___ss
_____________o_________nn_____________hhh_a_____________nnddd _____________i__n___
__thh_e_____ _____________________________________bb_uuii_______llllldd____ii____nng_____
_____________________________l___o___o_g__g___ii____nng______
____b___rr_ii_______ck__s_____________________________________p___ll__a_______sstt_______eerr__
______a___nnd_ ___b___lll_uu____ee__pp___r___i_______nnss_
_____________f_____oou_____________rrr_____ _____________f_____oo__rrr__tt_y____
_____t____www_oo_____ ____nn___ew_____________
_____________________________________b___e_____t_____i____n___
____e_____pp_____aa___rr___tt____mm_ee___nnntss
```

After collapsing:

YET A REHBILITATION CRU IS ONHAND IN THE BUILDING LOOGGING BRICKS PLASTER AND BLUEPRINS FOUR FORTY TWO NEW BETIN EPARTMENTS

Reference

YET A REHABILITATION CREW IS ON HAND IN THE BUILDING LUGGING BRICKS PLASTER AND BLUEPRINTS FOR FORTY TWO NEW BEDROOM APARTMENTS

Example CTC results. No word lexicon or LM (WSJ corpus)

YET A REHBILITATION CRU IS ONHAND IN THE BUILDING LOOGGING BRICKS PLASTER AND BLUEPRINS FOUR FORTY TWO NEW BETIN EPARTMENTS

YET A REHABILITATION CREW IS ON HAND IN THE BUILDING LUGGING BRICKS PLASTER AND BLUEPRINTS FOR FORTY TWO NEW BEDROOM APARTMENTS

THIS PARCLE GUNA COME BACK ON THIS ILAND SOM DAY SOO

THE SPARKLE GONNA COME BACK ON THIS ISLAND SOMEDAY SOON

TRADE REPRESENTIGD JUIDER WARANTS THAT THE U S WONT BACKCOFF ITS PUSH FOR TRADE BARIOR REDUCTIONS

TRADE REPRESENTATIVE YEUTTER WARNS THAT THE U S WONT BACK OFF ITS PUSH FOR TRADE BARRIER REDUCTIONS

TREASURY SECRETARY BAGER AT ROHIE WOS IN AUGGRAL PRESSED FOUR ARISE IN THE VALUE OF KOREAS CURRENCY

TREASURY SECRETARY BAKER AT ROH TAE WOO'S INAUGURAL PRESSED FOR A RISE IN THE VALUE OF KOREAS CURRENCY

CTC loss function

- Labels at each time index are **conditionally independent** (like HMMs)

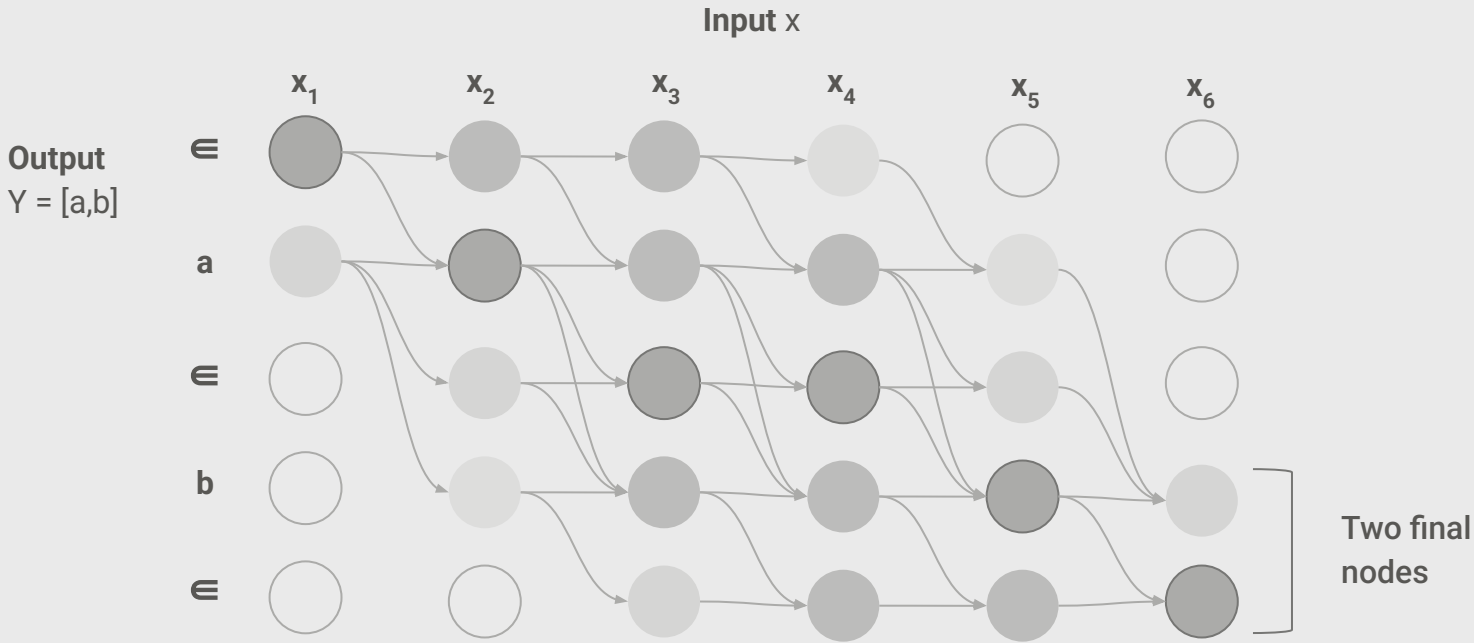
$$CTC(x) = -\log \Pr(y^*|x) \qquad \Pr(\mathrm{a}|x) = \prod_{t=1}^{T} \Pr(a_t, t|x)$$

- Sum over all time-level labelings consistent with the output label

$$\Pr(y|x) = \sum_{a \in \mathcal{B}^{-1}(y)} \Pr(a|x)$$

- T=3, transcript: HI
- All possible pre-collapsed / time-level labelings: HI_, H_I, _HI , HHI, HII
- Final loss maximizes probability of transcript y*
- **Insights**: Conditionally independent means we can compute outputs for each *t* in parallel
  Choice of blank+collapsing allows efficient, exact summation over all possible alignments

CTC loss function computation: Dynamic programming



Node *(s,t)* in the diagram represents $a_{s,t}$ - the CTC score of the subsequence $Z_{1:s}$ after input *t* steps

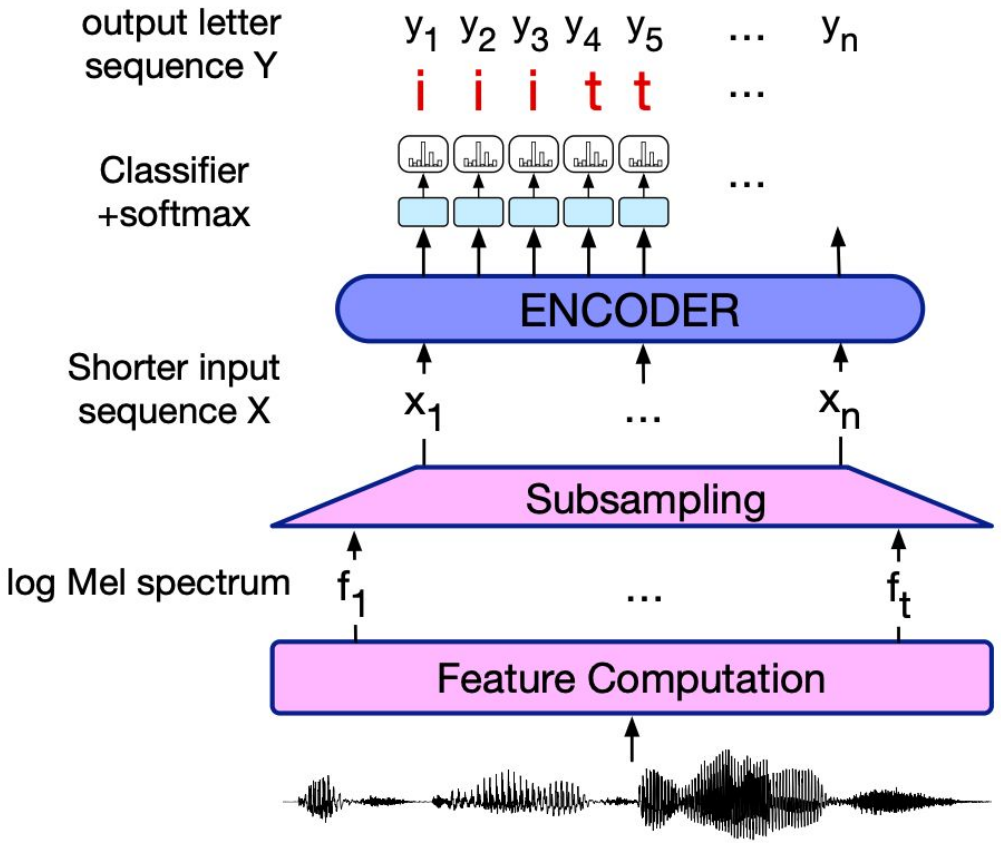
CTC inference: Most likely alignment

Output model assumes conditional independence

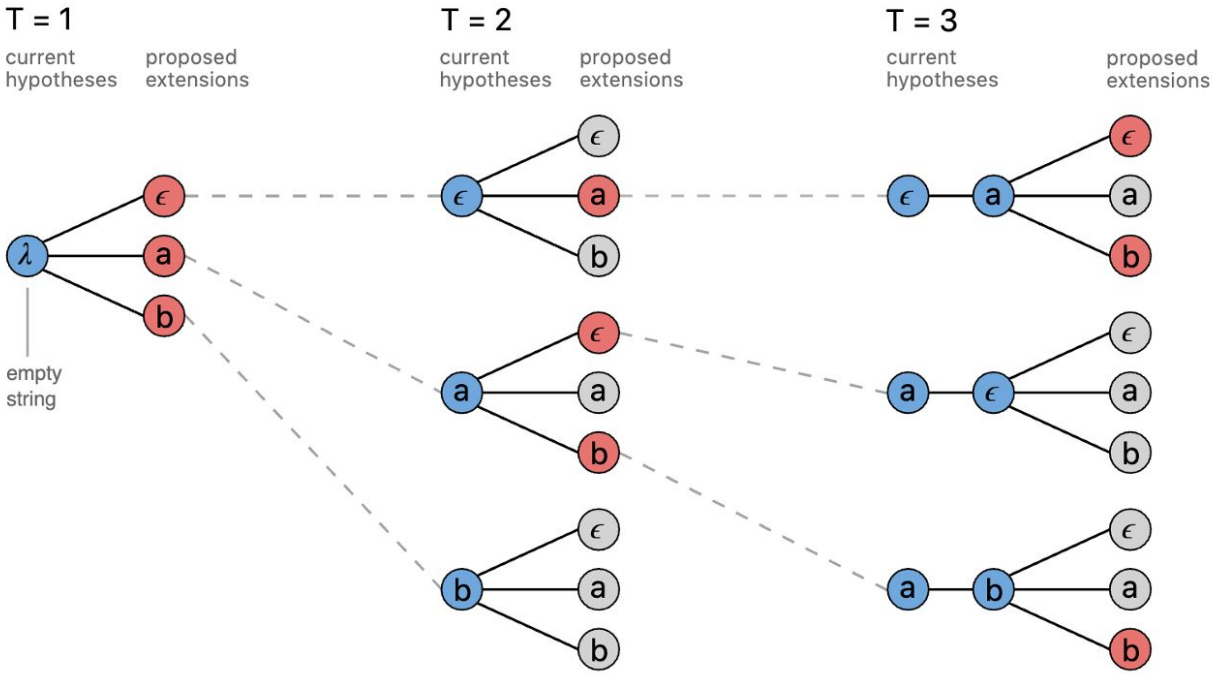
Each *t* most likely output depends only on inputs (not auto-regressive)

Most likely pre-collapsed output is simply the sequence of argmax outputs for each *t*

Run CTC collapsing to obtain transcript.



CTC inference: Simple beam search with blanks in prefix



A standard beam search algorithm with an alphabet of $\{\epsilon, a, b\}$ and a beam size of three.

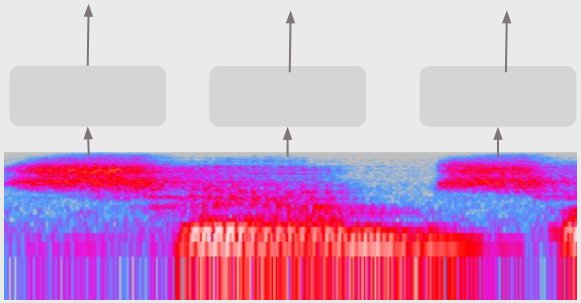
Search the space of alignments (pre-collapsed) and keeps top *k*(=3 here). Then sum over alignments in the final hypothesis set to better estimate most likely transcript

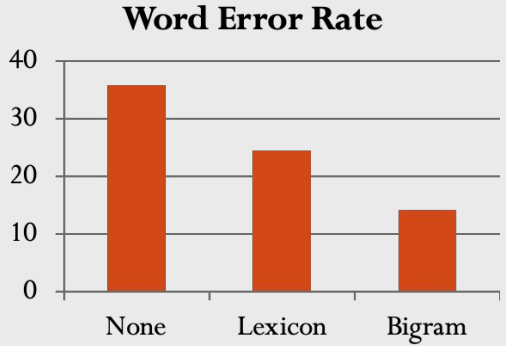
Decoding CTC with a lexicon-based language model

Lexicon: `[a, …, zebra]`

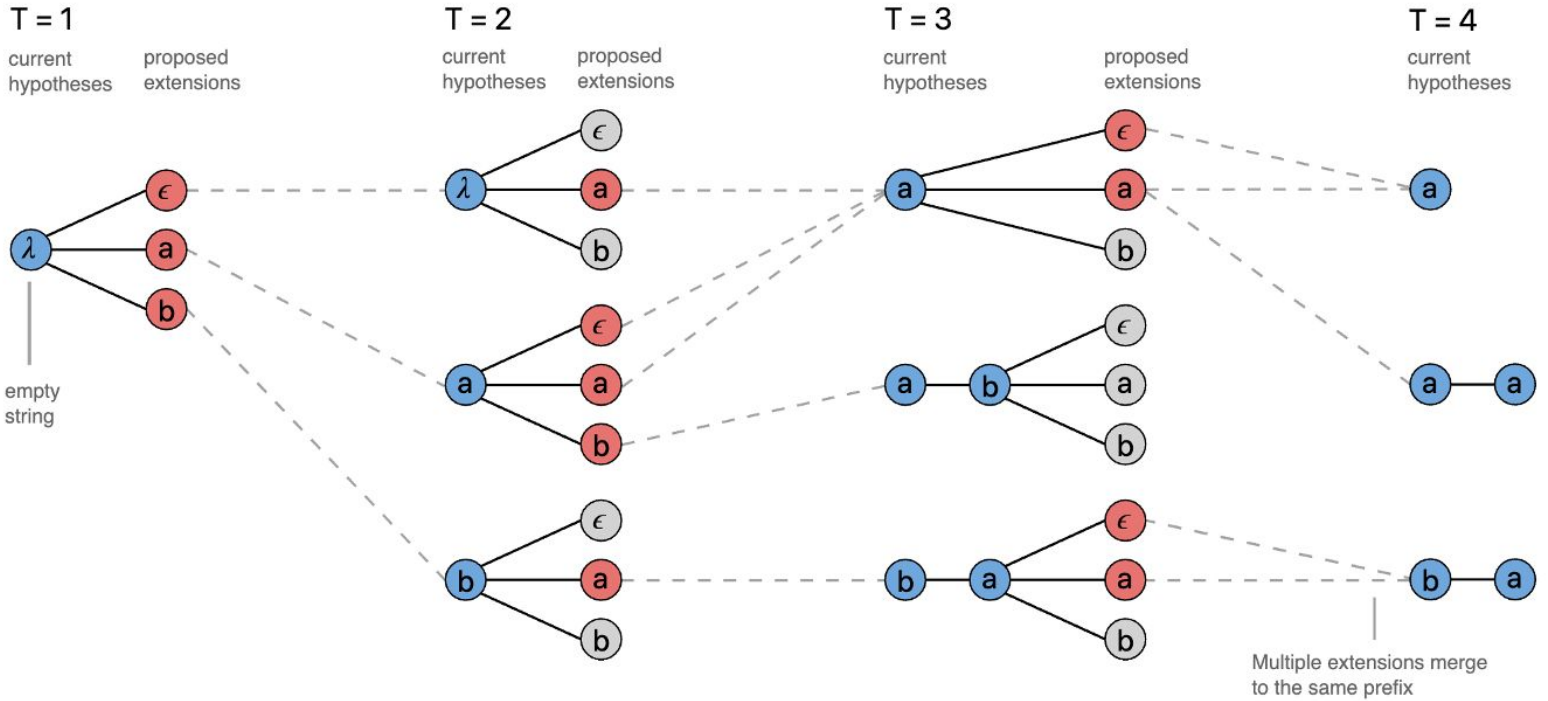
Language model: `p("yeah" | "oh")`

Character probabilities: `__oo_h__y_e_aa_h`





CTC inference: Beam search over collapsed outputs + LM



The CTC beam search algorithm with an output alphabet $\{\epsilon, a, b\}$ and a beam size of three.

Language model probabilities applied to each hypothesis

Lexicon-free CTC with character LM

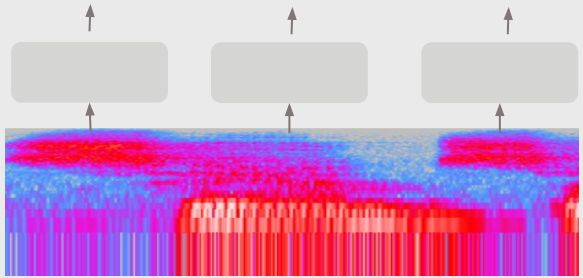
Out of Vocabulary Words

syriza abo- rona fru– schmidhuber

Lexi… zel…

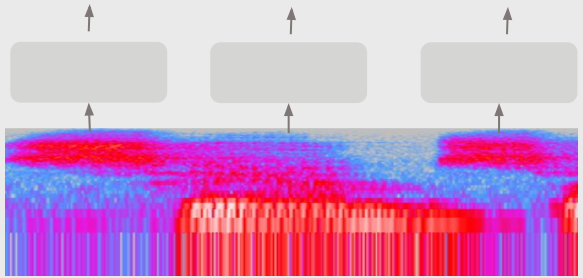
Charac… language mode… (“yeah” … “oh”)

Character probabilities: __oo_h__y_e_aa_h



Character language model: p(“yeah” | “oh”)

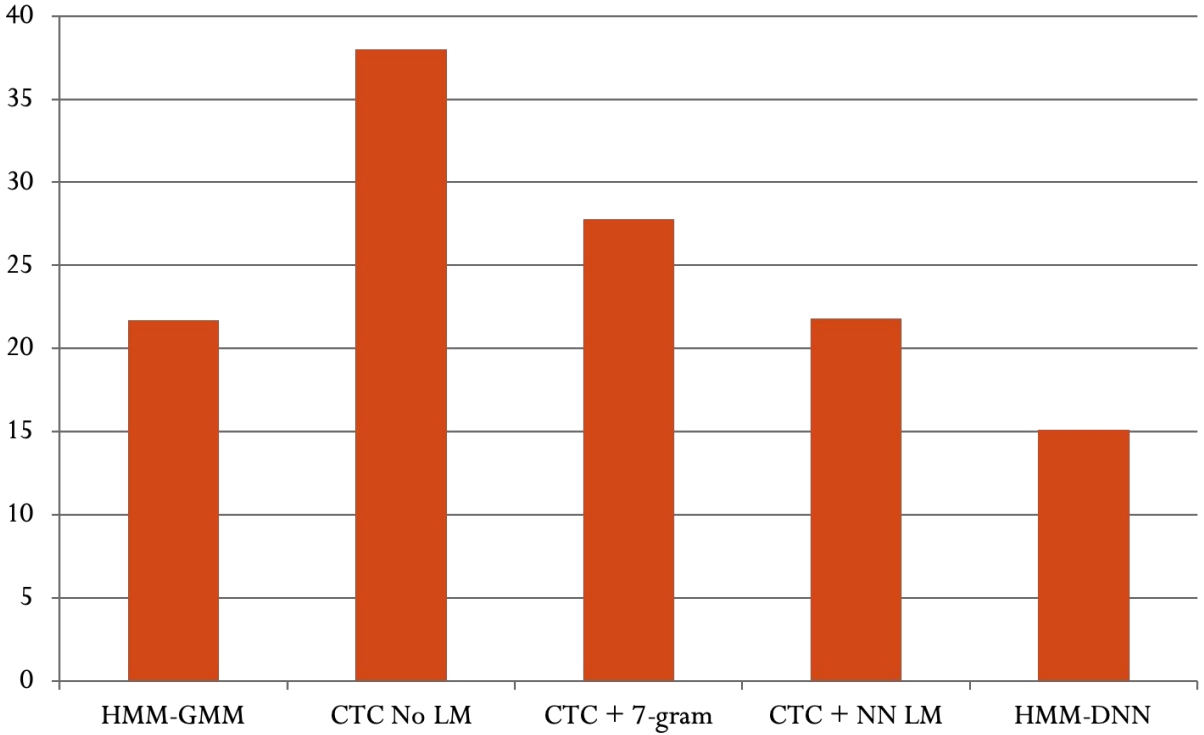
Character probabilities: __oo_h__y_e_aa_h



Beam search decoding for CTC with character LM

Inputs CTC likelihoods $p_{\text{ctc}}(c|x_t)$, character language model $p_{\text{clm}}(c|s)$
Parameters language model weight α, insertion bonus β, beam width k
Initialize $Z_0 \leftarrow \{\varnothing\}$, $p_{\text{b}}(\varnothing|x_{1:0}) \leftarrow 1$, $p_{\text{nb}}(\varnothing|x_{1:0}) \leftarrow 0$
for $t = 1, \ldots, T$ **do**
  $Z_t \leftarrow \{\}$
  for s **in** Z_{t-1} **do**
    $p_{\text{b}}(s|x_{1:t}) \leftarrow p_{\text{ctc}}(\_|x_t)p_{\text{tot}}(s|x_{1:t-1})$ ▷ Handle blanks
    $p_{\text{nb}}(s|x_{1:t}) \leftarrow p_{\text{ctc}}(c|x_t)p_{\text{nb}}(s|x_{1:t-1})$ ▷ Handle repeat character collapsing
    Add s to Z_t
    for c **in** ζ' **do**
      $s^+ \leftarrow s + c$
      if $c \neq s_{t-1}$ **then**
        $p_{\text{nb}}(s^+|x_{1:t}) \leftarrow p_{\text{ctc}}(c|x_t)p_{\text{clm}}(c|s)^\alpha p_{\text{tot}}(c|x_{1:t-1})$
      else
        $p_{\text{nb}}(s^+|x_{1:t}) \leftarrow p_{\text{ctc}}(c|x_t)p_{\text{clm}}(c|s)^\alpha p_{\text{b}}(c|x_{1:t-1})$ ▷ Repeat characters have "_" between
      end if
      Add s^+ to Z_t
    end for
  end for
  $Z_t \leftarrow k$ most probable s by $p_{\text{tot}}(s|x_{1:t})|s|^\beta$ in Z_t ▷ Apply beam
end for
Return $\arg\max_{s \in Z_t} p_{\text{tot}}(s|x_{1:T})|s|^\beta$

Lexicon-Free & HMM-Free WER on Switchboard



Example results (Switchboard) ~19% CER

i i don'tknow i don't know what the rain force have to do with it but you know their chop a those down af the tr minusrat everyday

i- i don't kn- i don't know what the rain forests have to do with it but you know they're chopping those down at a tremendous rate everyday

come home and get back in to regular cloos aga

come home and get back into regular clothes again

i guess down't here u we just recently move to texas so my wor op has change quite a bit muh we ook from colorado were and i have a cloveful of sweatterso tuth

i guess down here uh we just recently moved to texas so my wardrobe has changed quite a bit um we moved from colorado where and i have a closet full of sweaters that

i don't know whether state lit state hood whold itprove there a conomy i don't i don't know that to that the actove being a state

i don't know whether state woul- statehood would improve their economy i don't i don't know that the ve- the act of being a state

Transcribing out of vocabulary (OOV) words

Truth: yeah i went into the i do not know what you think of ***fidelity*** but

HMM-GMM: yeah when the i don't know what you think of **fidel it even** them

CTC-CLM: yeah i went to i don't know what you think of **fidelity** but um

Truth: no no speaking of weather do you carry a ***altimeter*** slash ***barometer***

HMM-GMM: no i'm not all being the weather do you uh carry a uh helped **emitters** last **brahms her**

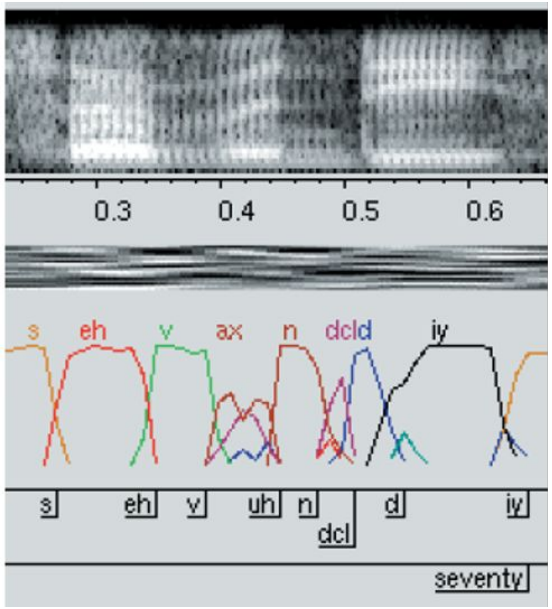
CTC-CLM: no no beating of whether do you uh carry a uh **a time or** less **barometer**

Truth: i would **ima-** well yeah it is i know you are able to stay home with them

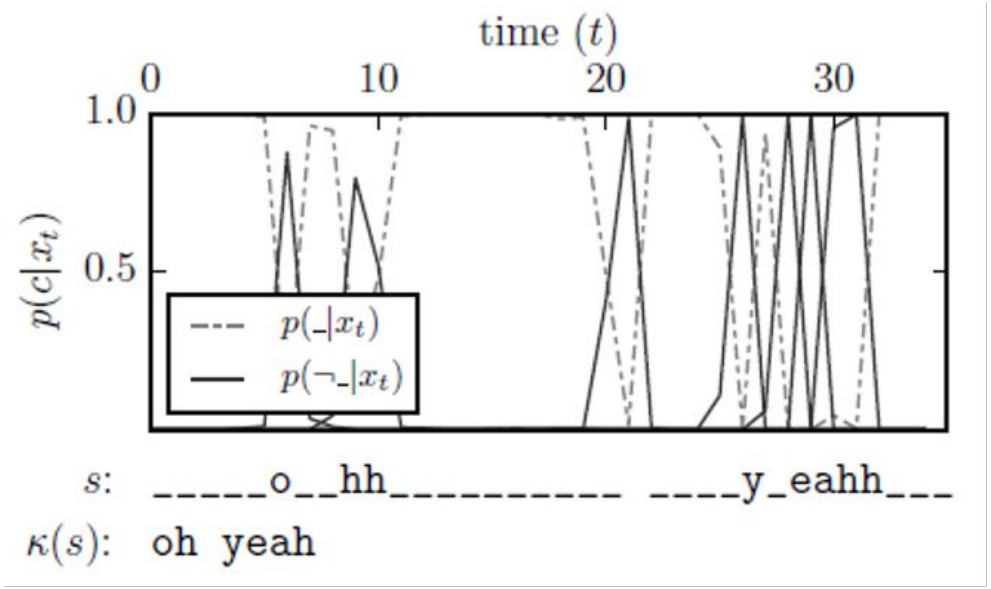
HMM-GMM: i would **amount** well yeah it is i know um you're able to stay home with them

CTC-CLM: i would **ima-** well yeah it is i know uh you're able to stay home with them

Comparing alignments



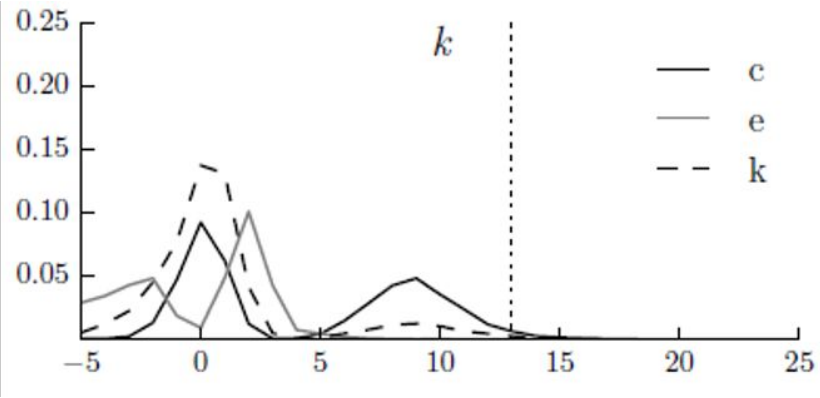
HMM-GMM phone probabilities



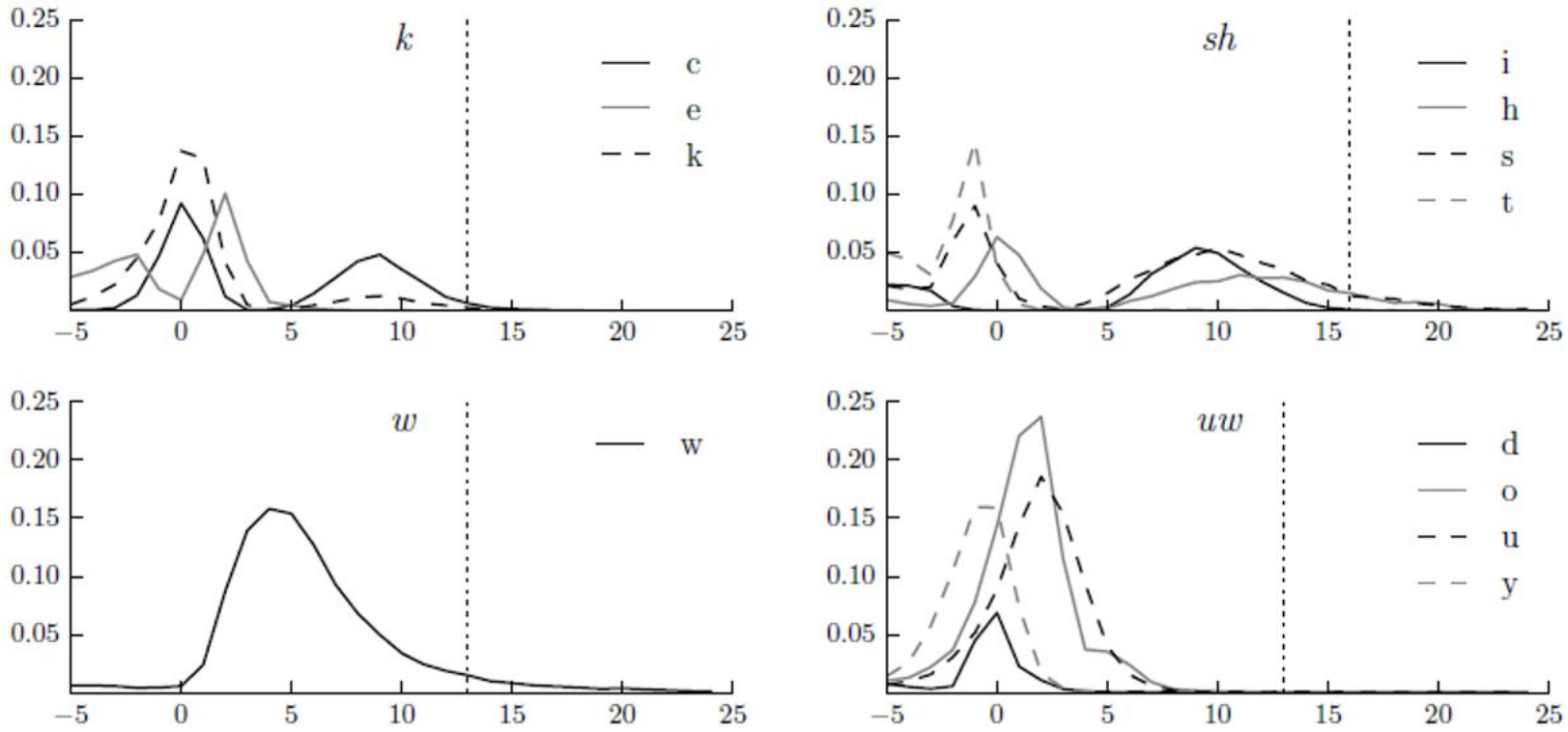
CTC character probabilities

Learning phonemes and timing

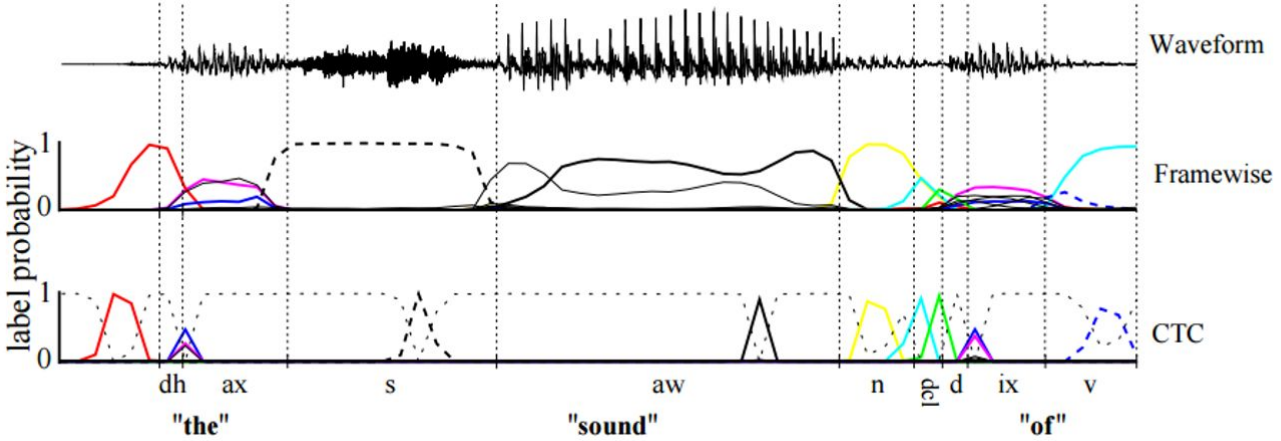
- Take all phone segments from HMM-GMM alignments (k)
- Align all segments to start at the same time = 0
- Compute the average CTC character probabilities during the segment (c, e, k)
- Vertical line shows median end time of phone segment from HMM-GMM alignments



Learning phonemes and timing



Earlier work on CTC with phonemes



System	LER
Context-independent HMM	38.85 %
Context-dependent HMM	35.21 %
BLSTM/HMM	33.84 ± 0.06 %
Weighted error BLSTM/HMM	31.57 ± 0.06 %
CTC (best path)	31.47 ± 0.21 %
CTC (prefix search)	30.51 ± 0.19 %

Table: Label Error Rate (LER) on TIMIT. CTC and hybrid results are means over 5 runs, ‡ standard error.All differences were significant ($p < 0.01$), except between weighted error BLSTM/HMM and CTC (best path).

(Graves, Fernández, Gomez, & Schmidhuber. 2006)

CTC loss during training

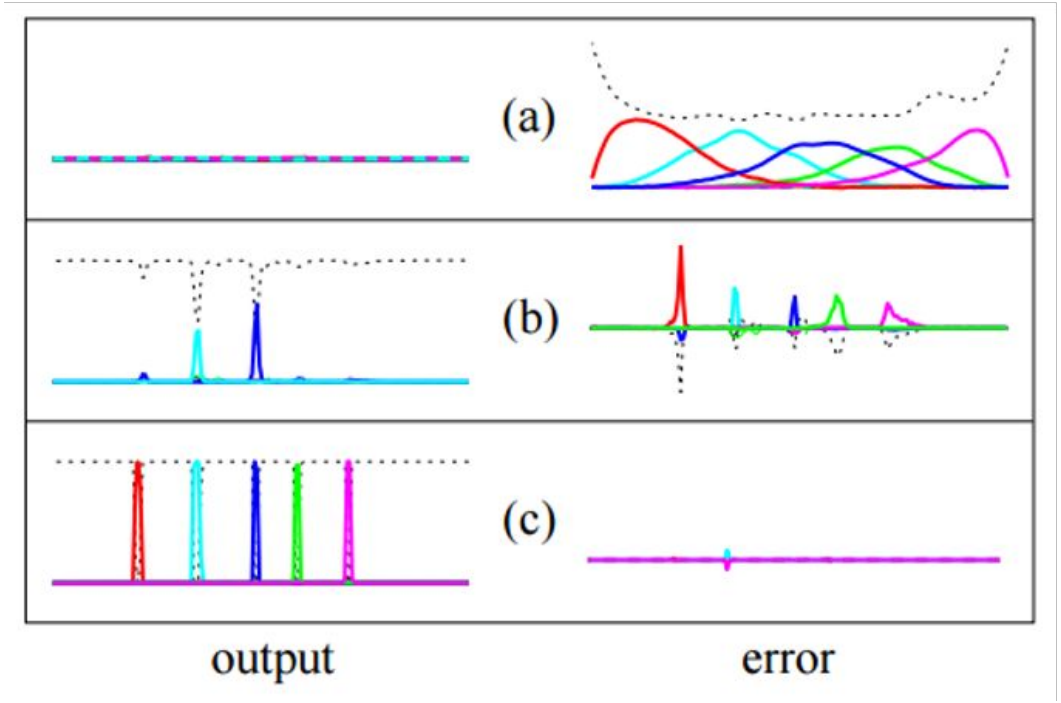


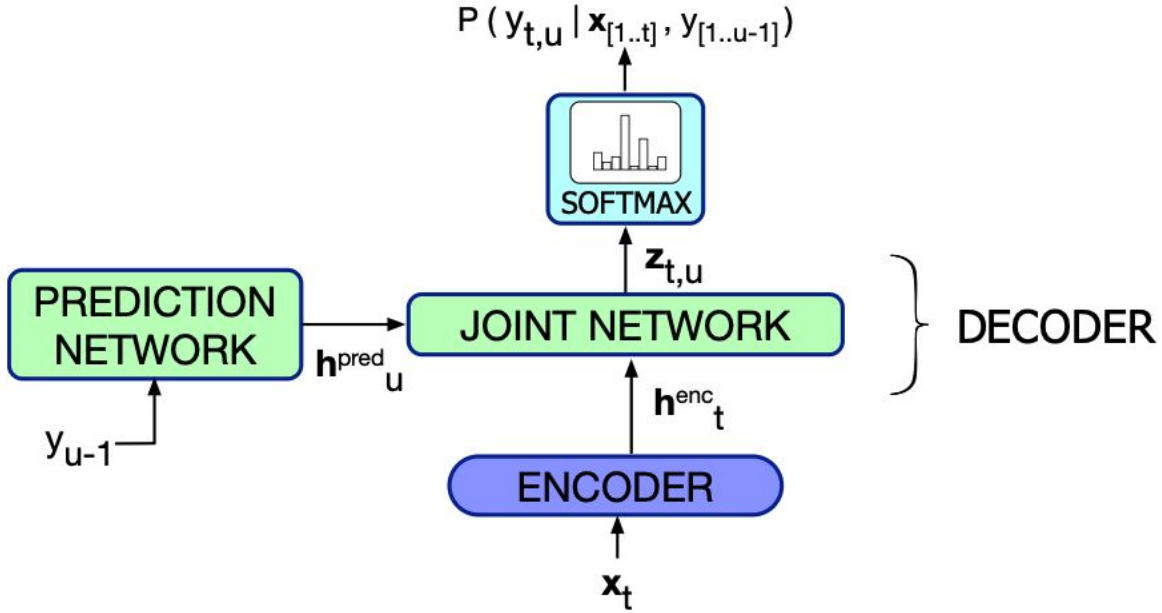
Figure: Evolution of the CTC Error Signal During Training. The left column shows the output activations for the same sequence at various stages of training (the dashed line is the 'blank' unit); the right column shows the corresponding error signals. Errors above the horizontal axis act to increase the corresponding output activation and those below act to decrease it. (a) Initially the network has small random weights, and the error is determined by the target sequence only. (b) The network begins to make predictions and the error localises around them. (c) The network strongly predicts the correct labelling and the error virtually disappears.

(Graves, Fernández, Gomez, & Schmidhuber. 2006)

Improving on CTC: RNN-Transducer output model

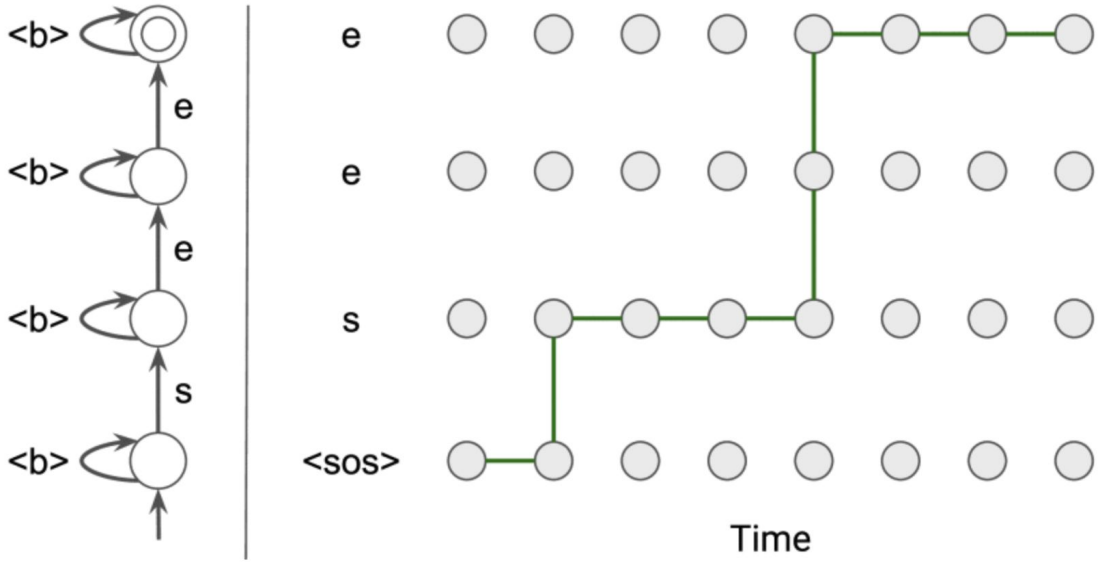
Improving on CTC: RNN-Transducer output model

- Add network that reasons over sequence of non-blank characters/tokens and input sequence
- Decoding outputs each transcript token only once. Optional blanks between. Variation on CTC
- Prediction network often pre-trained with CTC. Encoder provides transformed features for each *t*



Loss computation in RNN-Transducer output model

- Output each transcript token *only once*. Optional blanks between.
- Must reach top right (end time & end transcript output)
  Vertical move emits transcript token
  Right move emits blank <b>
  Start at <sos> tag



(Prabhavalkar *et al*, 2023)

Improving on CTC: RNN-Transducer output model

- Add network that reasons over sequence of non-blank characters/tokens (c)

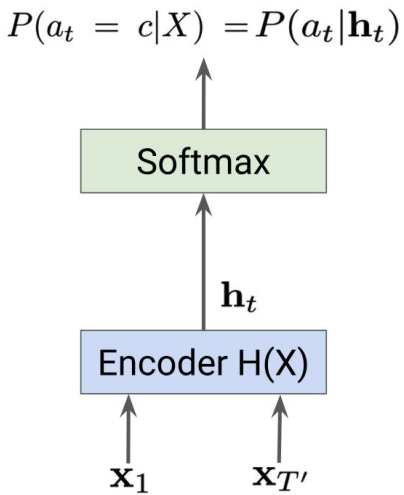


Fig. 2. Representation of the CTC model consisting of an encoder which maps the input speech into a higher-level representation, and a softmax layer which predicts frame-level probabilities over the set of output labels and blank.

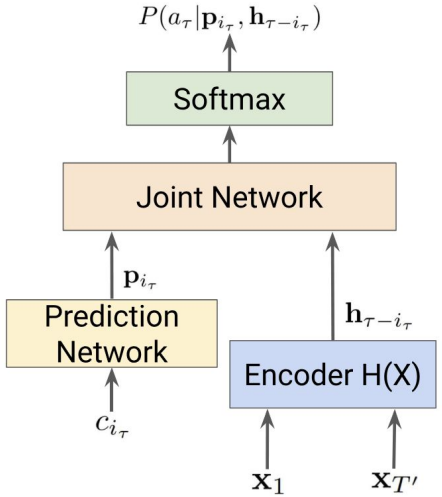


Fig. 3. RNN-T Model [14], [48] consists of an encoder which transforms the input speech frames into a high-level representation, and a prediction-network which models the sequence of non-blank labels that have been output previously. The prediction network output, p_{i_t}, represents the output after producing the previous non-blank label sequence $c_1, \ldots, c_{i_t}$. The joint network produces a probability distribution over the output symbols (augmented with blank) given the prediction network state and a specific encoded frame.

(Prabhavalkar *et al*, 2023)

Extending CTC: RNN-Transducer output model

- Add network that reasons over sequence of non-blank characters/tokens (c)

We may then define the posterior probability $P(C|X)$ as before:

$$P_{\text{RNNT}}(C|X) = \sum_{A \in \mathcal{A}^{\text{RNNT}}_{(X,C)}} P(A|H(X))$$

$$= \sum_{A \in \mathcal{A}^{\text{RNNT}}_{(X,C)}} \prod_{\tau=1}^{T+L} P(a_\tau|a_{\tau-1}, \ldots, a_1, H(X))$$

$$= \sum_{A \in \mathcal{A}^{\text{RNNT}}_{(X,C)}} \prod_{\tau=1}^{T+L} P(a_\tau|c_{i_\tau}, c_{i_\tau-1}, \ldots, c_0, \mathbf{h}_{\tau-i_\tau})$$

$$= \sum_{A \in \mathcal{A}^{\text{RNNT}}_{(X,C)}} \prod_{\tau=1}^{T+L} P(a_\tau|\mathbf{p}_{i_\tau}, \mathbf{h}_{\tau-i_\tau}) \quad (3)$$

where, $P = (\mathbf{p}_1, \ldots, \mathbf{p}_L)$ represents the output of the *prediction network* depicted in Fig. 3 which summarizes the sequence of previously predicted non-blank labels, implemented as another

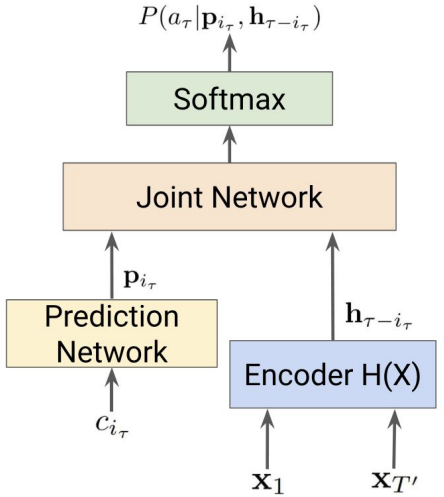


Fig. 3. RNN-T Model [14], [48] consists of an encoder which transforms the input speech frames into a high-level representation, and a prediction-network which models the sequence of non-blank labels that have been output previously. The prediction network output, p_{i_t}, represents the output after producing the previous non-blank label sequence $c_1, \ldots, c_{i_t}$. The joint network produces a probability distribution over the output symbols (augmented with blank) given the prediction network state and a specific encoded frame.

(Prabhavalkar *et al*, 2023)

Questions?

Appendix: Listen, Attend, & Spell

Listen, Attend, and Spell

- Discriminative, character-based encoder-decoder
- Unlike CTC:
  - Outputs also condition on previous outputs so far
  - No blank/epsilon. LAS just outputs characters
- Attention-based decoder. Precursor to modern encoder-decoder and transformer approaches

$$P(\mathbf{y}|\mathbf{x}) = \prod_i P(y_i|\mathbf{x}, y_{<i})$$

$$\mathbf{h} = \text{Listen}(\mathbf{x})$$
$$P(\mathbf{y}|\mathbf{x}) = \text{AttendAndSpell}(\mathbf{h}, \mathbf{y})$$

From: Chan, Jaitly, Le, & Vinyals. 2015

Listen, Attend, and Spell

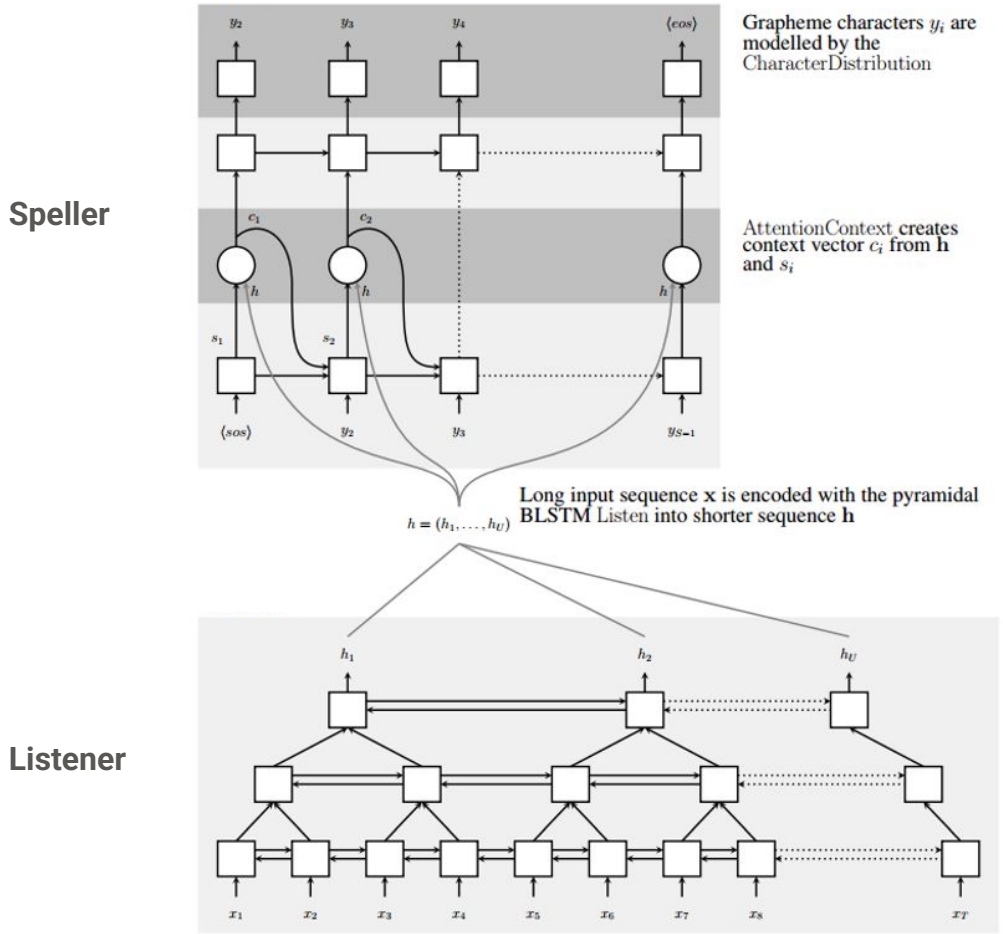


Figure: Listen, Attend and Spell (LAS) model: the listener is a pyramidal BLSTM encoding our input sequence x into high level features h, the speller is an attention-based decoder generating the y characters from h.

(Chan, Jaitly, Le, & Vinyals. 2015)

Listen, Attend, and Spell

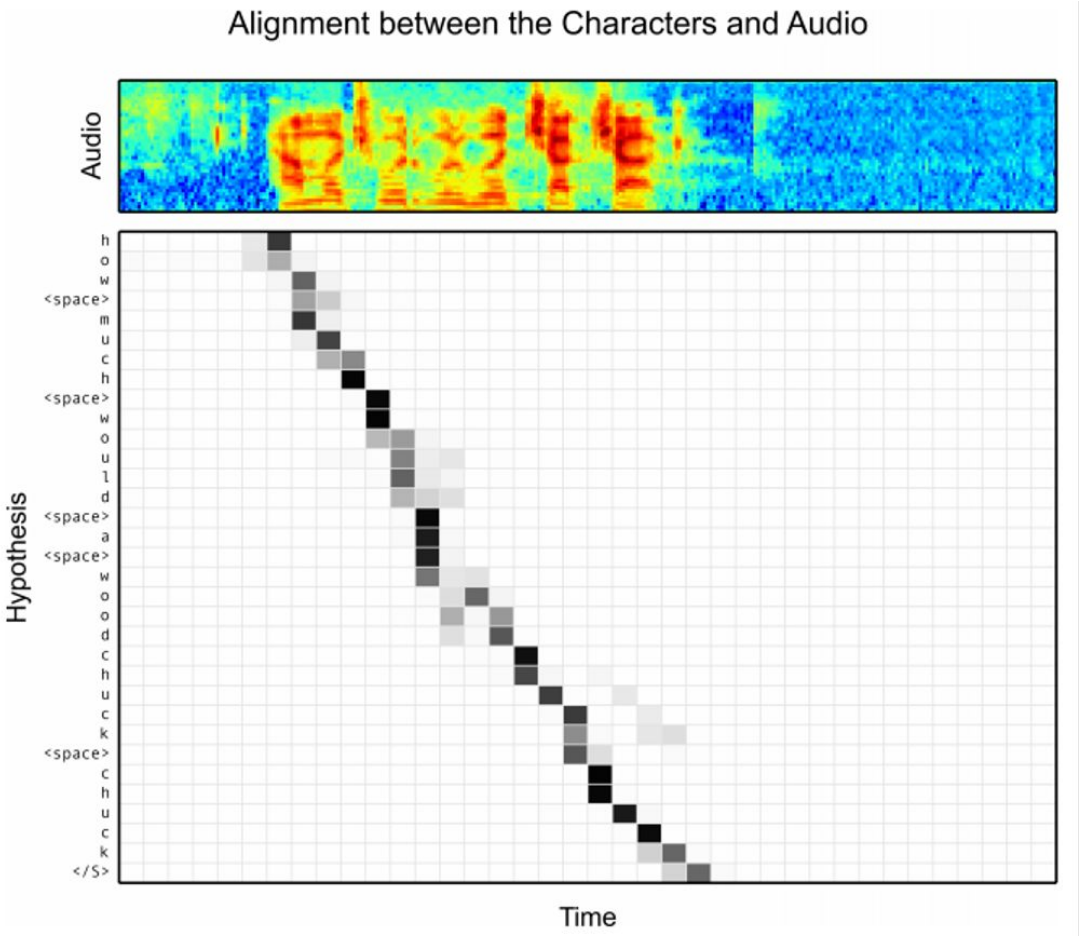


Figure: Chan, Jaitly, Le, & Vinyals. 2015

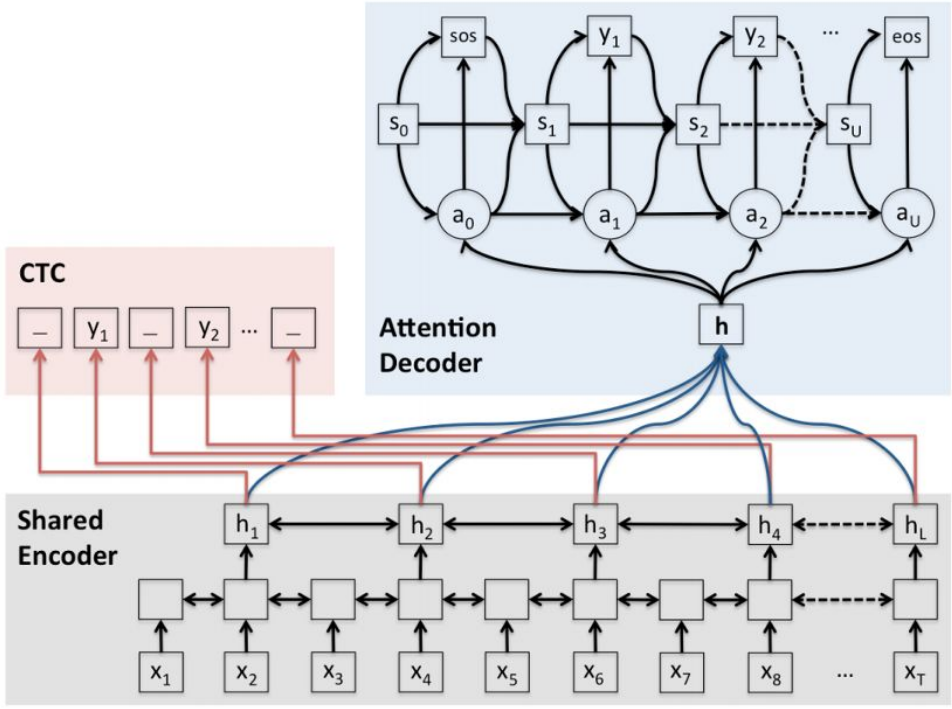
Listen, Attend, and Spell

Model	Clean WER	Noisy WER
CLDNN-HMM [20]	8.0	8.9
LAS	16.2	19.0
LAS + LM Rescoring	12.6	14.7
LAS + Sampling	14.1	16.5
LAS + Sampling + LM Rescoring	10.3	12.0

Table: WER comparison on the clean and noisy Google voice search task. The CLDNN-HMM system is the state-of-the-art system, the Listen, Attend and Spell (LAS) models are decoded with a beam size of 32. Language Model (LM) rescoring was applied to our beams, and a sampling trick was applied to bridge the gap between training and inference.

(Chan, Jaitly, Le, & Vinyals. 2015)

CTC + LAS Multi-Task Approach



$$\mathcal{L}_{\text{MTL}} = \lambda \mathcal{L}_{\text{CTC}} + (1-\lambda)\mathcal{L}_{\text{Attention}}$$

Figure: Our proposed Joint CTC-attention based end-to-end framework: the shared encoder is trained by both CTC and attention model objectives simultaneously. The shared encoder transforms our input sequence x into high level features h, the location-based attention decoder generates the character sequence y

(Kim, Hori, & Watanabe. 2017)

CTC + LAS Multi-Task Approach

Model(train)	CER(valid)	CER(eval)
WSJ-train_si284 (80hrs)	dev93	eval92
CTC	11.48	8.97
Attention(content-based)	13.68	11.08
Attention(location-based)	11.98	8.17
MTL($\lambda = 0.2$)	**11.27**	**7.36**
MTL($\lambda = 0.5$)	12.00	8.31
MTL($\lambda = 0.8$)	11.71	8.45
WSJ-train_si84 (15hrs)	dev93	eval92
CTC	27.41	20.34
Attention(content-based)	28.02	20.06
Attention(location-based)	24.98	17.01
MTL($\lambda = 0.2$)	**23.03**	**14.53**
MTL($\lambda = 0.5$)	26.28	16.24
MTL($\lambda = 0.8$)	32.21	21.30
CHiME-4-tr05_multi (18hrs)	dt05_real	et05_real
CTC	37.56	48.79
Attention(content-based)	43.45	54.25
Attention(location-based)	35.01	47.58
MTL($\lambda = 0.2$)	**32.08**	**44.99**
MTL($\lambda = 0.5$)	34.56	46.49
MTL($\lambda = 0.8$)	35.41	48.34

Table: Character Error Rate (CER) on clean corpora WSJ1 (80 hours) and WSJO (15 hours), and a noisy corpus CHiME-4 (18 hours). None of our experiments used any language model or lexicon information. (Word Error Rate (WER) of our model MTL(1 = 0.2) was 18.2% and WER of [7] was 18.6% on WSJ1.Note that this is not an exact comparison because the hyper parameters were not completely same as [7].)

(Kim, Hori, & Watanabe. 2017)

$$e_{u,l} = \begin{cases} \text{content-based:} \\ \quad w^T \tanh(W s_{u-1} + V h_l + b) \\ \text{location-based:} \\ \quad f_u = F * a_{u-1} \\ \quad w^T \tanh(W s_{u-1} + V h_l + U f_{u,l} + b) \end{cases}$$

$$a_{u,l} = \frac{\exp(\gamma e_{u,l})}{\sum_l \exp(\gamma e_{u,l})}$$

$$\mathcal{L}_{\text{MTL}} = \lambda \mathcal{L}_{\text{CTC}} + (1 - \lambda) \mathcal{L}_{\text{Attention}}$$

CTC + LAS Multi-Task Approach

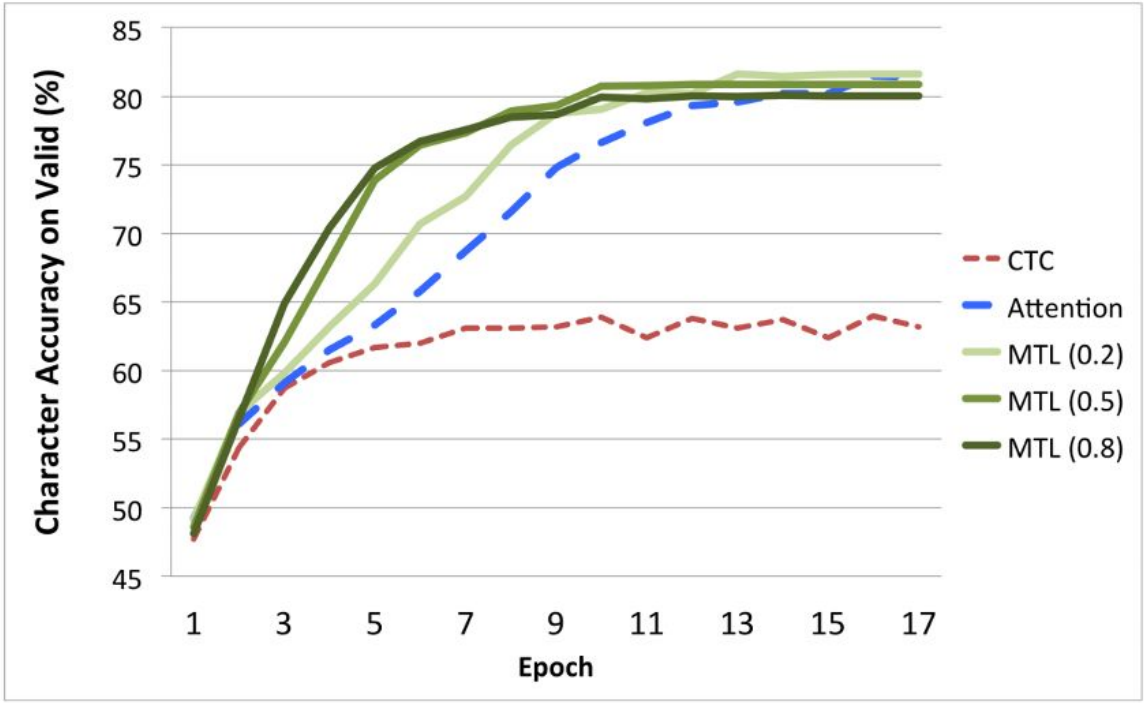


Figure: Comparison of learning curves: CTC, location-based attention model, and MTL with (1 = 0.2, 0.5, 0.8). The character accuracy on the validation set of CHiME-4 is calculated by edit distance between hypothesis and reference. Note that the reference history were used in the attention and our MTL models.

(Kim, Hori, & Watanabe. 2017)

CTC + LAS Multi-Task Approach

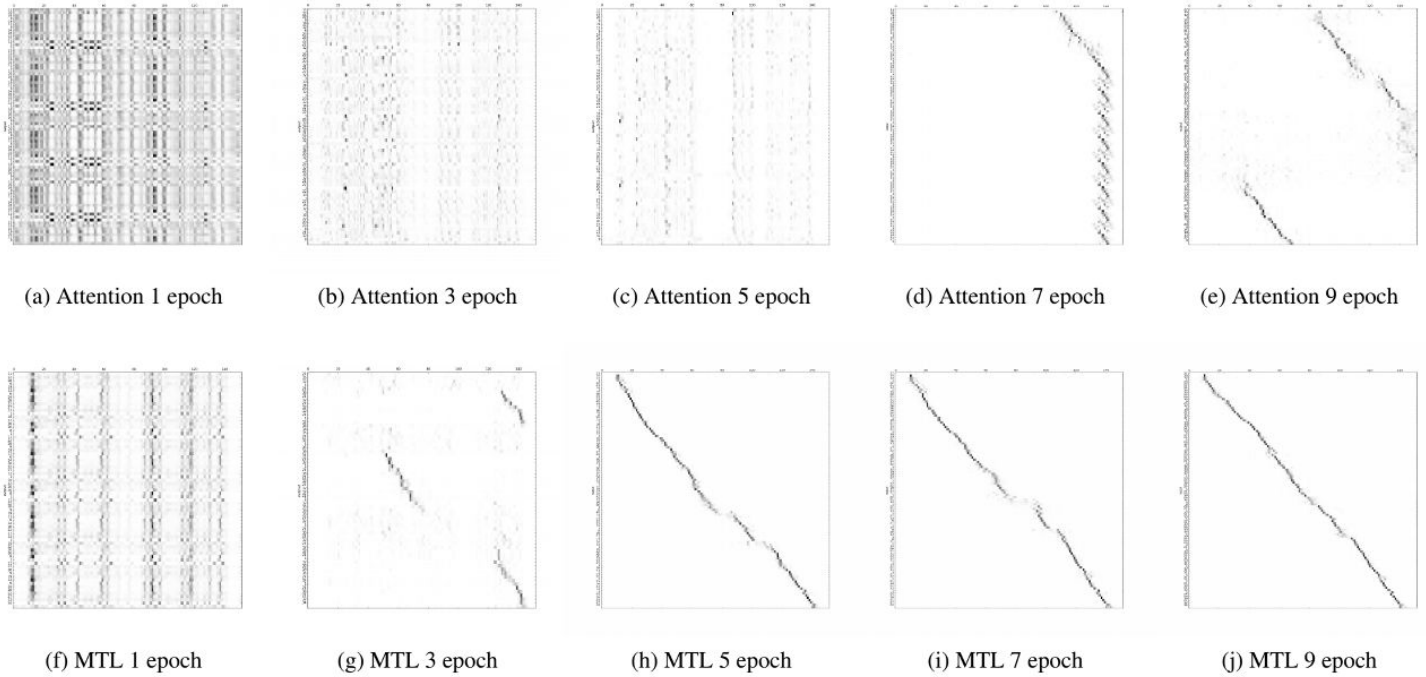


Figure: Comparison of speed in learning alignments between characters (y-axis) and acoustic frames (x-axis) between the location-based attention model (1st row) and our model MTL (2nd row) over training epoch (1,3,5,7, and 9). All alignments are for one manually chosen utterance F05_442C020U_CAF_REAL - "THE ONE HUNDRED SHARE INDEX CLOSED SIX POINT EIGHT POINTS LOWER AT ONE THOUSAND SEVEN HUNDRED FIFTY NINE POINT NINE") in the noisy CHiME-4 evaluation set. (Kim, Hori, & Watanabe. 2017)

CTC + LAS Multi-Task Approach

Model	Hour	Task1	Task2	Task3
Attention	581	11.4	7.9	9.0
MTL	581	10.5	7.6	8.3
MTL + joint decoding (rescoring)	581	10.1	7.1	7.8
MTL + joint decoding (one pass)	581	10.0	7.1	7.6
MTL-large + joint decoding (rescoring)	581	**8.4**	6.2	**6.9**
MTL-large + joint decoding (one pass)	581	**8.4**	**6.1**	**6.9**
GMM-discr. (Moriya et al., 2015)	236 for AM, 581 for LM	11.2	9.2	12.1
DNN/HMM (Moriya et al., 2015)	236 for AM, 581 for LM	9.0	7.2	9.6
CTC-syllable (Kanda et al., 2016)	581	9.4	7.3	7.5

Table: Character error rate (CER) for conventional attention and hybrid CTC/attention end-to-end ASR. Corpus of Spontaneous Japanese speech recognition (CSJ) task.

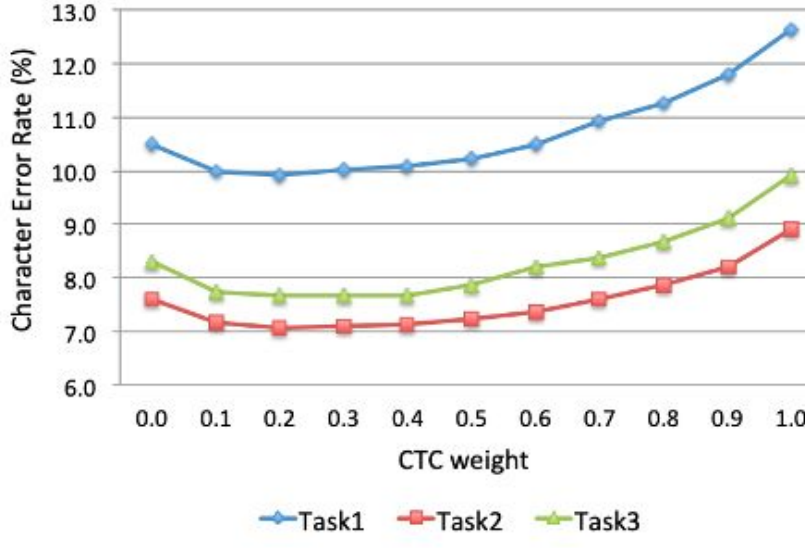


Figure: The effect of weight parameter \ in Eq. (14) on the CSJ evaluation tasks (The CERs were obtained by one-pass decoding).

(Hori, Watanabe, Zhang, & Chan. 2017)

Data Augmentation

Training Augmentation: SpecAugment

1. Time warping (image warp)
2. Frequency masking
3. Time masking

Mix using different policies

Implementations available for PyTorch

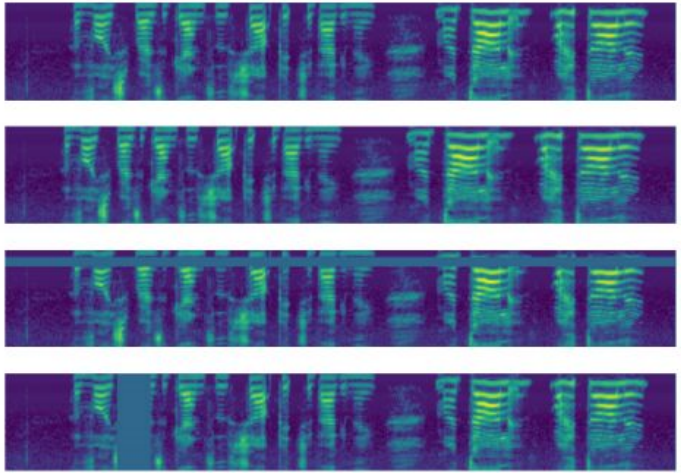


Figure 1: Augmentations applied to the base input, given at the top. From top to bottom, the figures depict the log mel spectrogram of the base input with no augmentation, time warp, frequency masking and time masking applied.

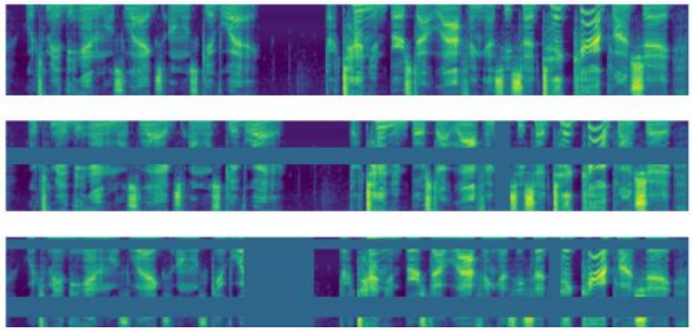


Figure 2: Augmentation policies applied to the base input. From top to bottom, the figures depict the log mel spectrogram of the base input with policies None, LB and LD applied.

(*Park et al*, ICASSP 2020)

Training Augmentation: SpecAugment

Method	No LM		With LM	
	SWBD	CH	SWBD	CH
HMM				
Veselý et al., (2013) [41]			12.9	24.5
Povey et al., (2016) [30]			9.6	19.3
Hadian et al., (2018) [42]			9.3	18.9
Zeyer et al., (2018) [24]			8.3	17.3
CTC				
Zweig et al., (2017) [43]	24.7	37.1	14.0	25.3
Audhkhasi et al., (2018) [44]	20.8	30.4		
Audhkhasi et al., (2018) [45]	14.6	23.6		
LAS				
Lu et al., (2016) [46]	26.8	48.2	25.8	46.0
Toshniwal et al., (2017) [47]	23.1	40.8		
Zeyer et al., (2018) [24]	13.1	26.1	11.8	25.7
Weng et al., (2018) [48]	12.2	23.3		
Zeyer et al., (2018) [38]	11.9	23.7	11.0	23.1
Our Work				
LAS	11.2	21.6	10.9	19.4
LAS + SpecAugment (SM)	**7.2**	14.6	**6.8**	14.1
LAS + SpecAugment (SS)	7.3	**14.4**	7.1	**14.0**

Table: Switchboard 300h WER (%)

(*Park et al*, ICASSP 2020)