Automatic Speech Recognition (CS753)

Lecture 21: End-to-End ASR Systems

Instructor: Preethi Jyothi

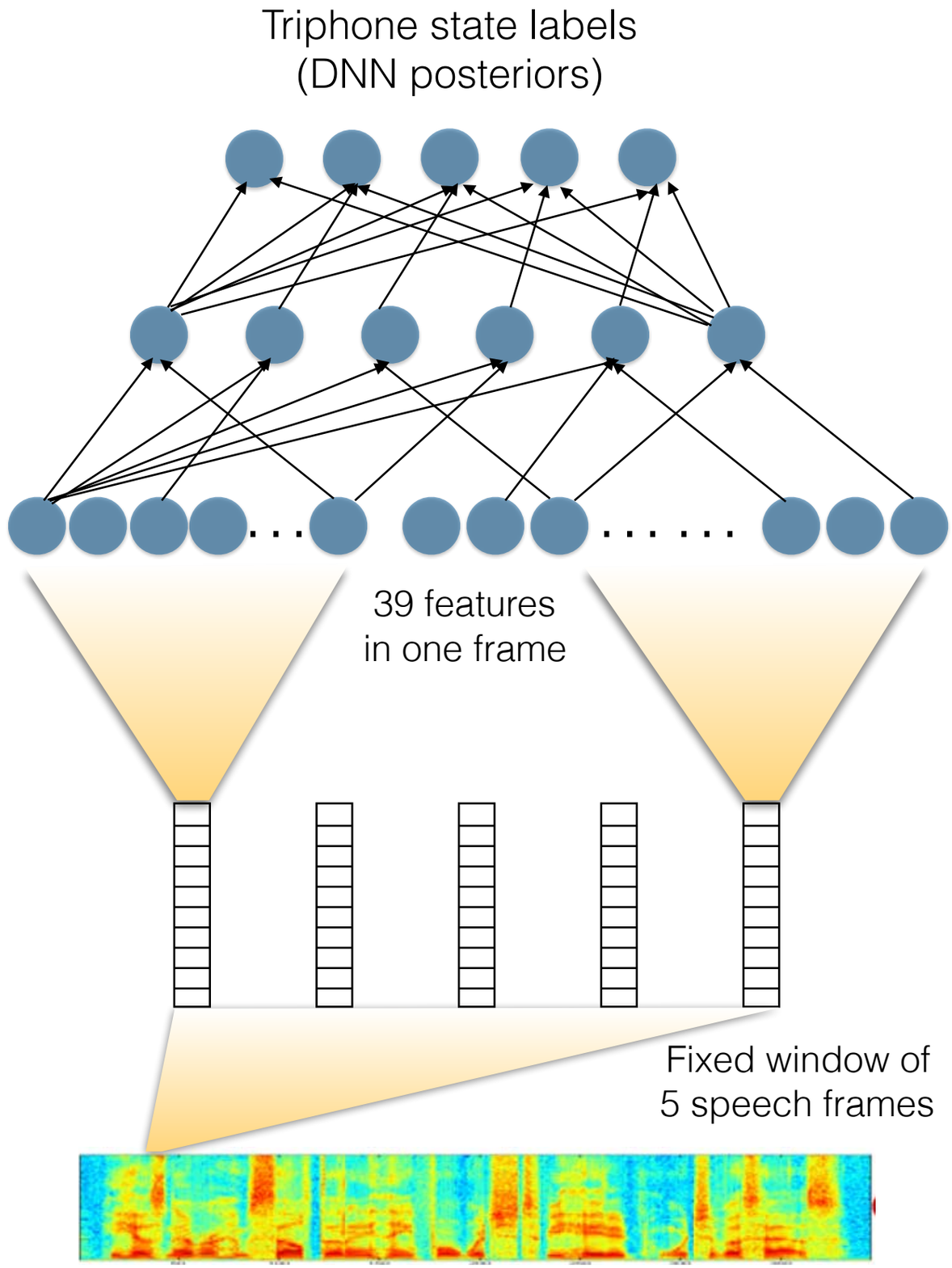
Apr 6, 2017

Recall: Hybrid DNN-HMM acoustic models

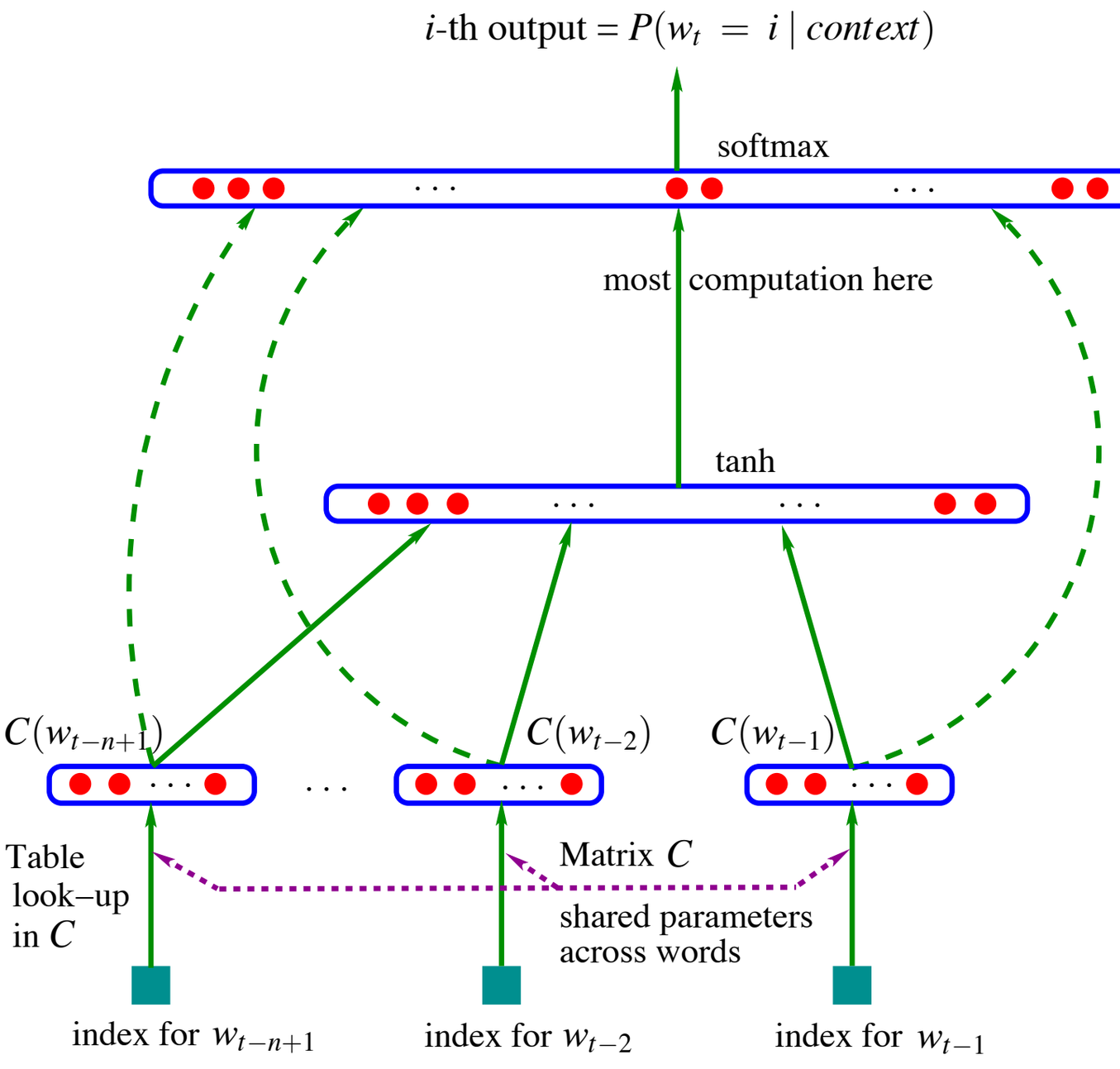
- DNNs trained using triphone labels derived from a forced alignment "Viterbi" step.

- DNNs give posteriors $\Pr(q_t|o_t)$ where o_t is the acoustic vector at time t and q_t is a triphone HMM state

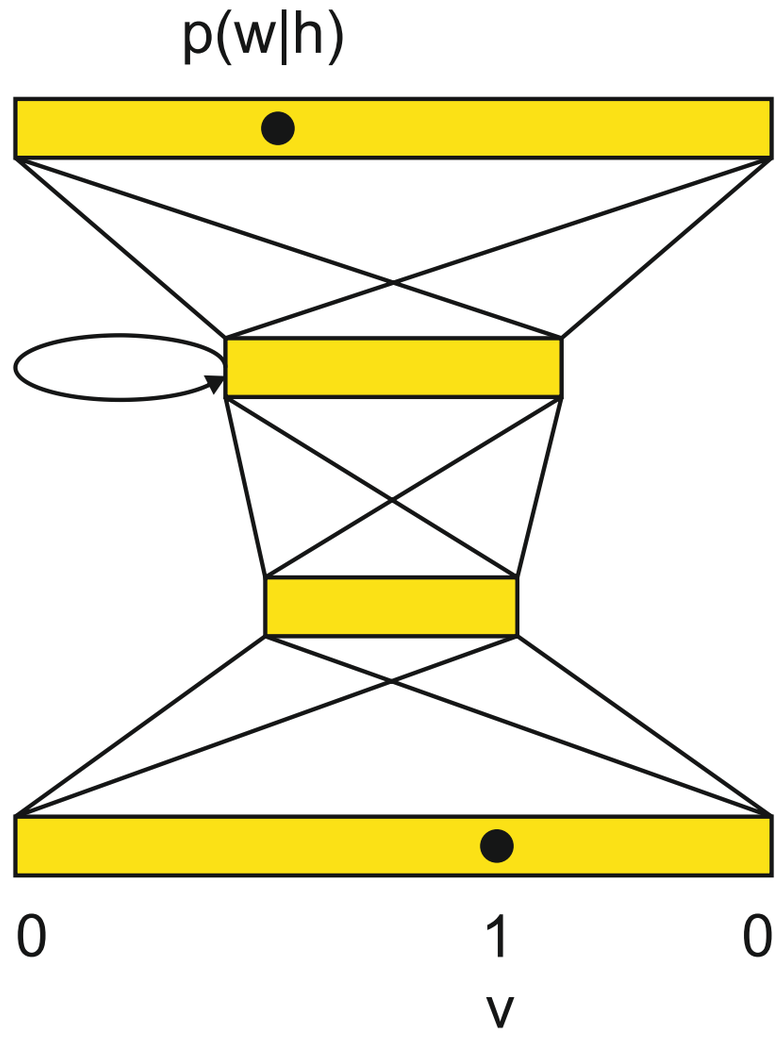
- Compute scaled posteriors $\Pr(o_t|q_t)$ which are used as emission probabilities for an HMM



Recall: (R)NN-based language models



NN Language models



RNN Language models

Image from: Bengio et al., “A neural probabilistic language model”, JMLR, 03

Neural network-based ASR components

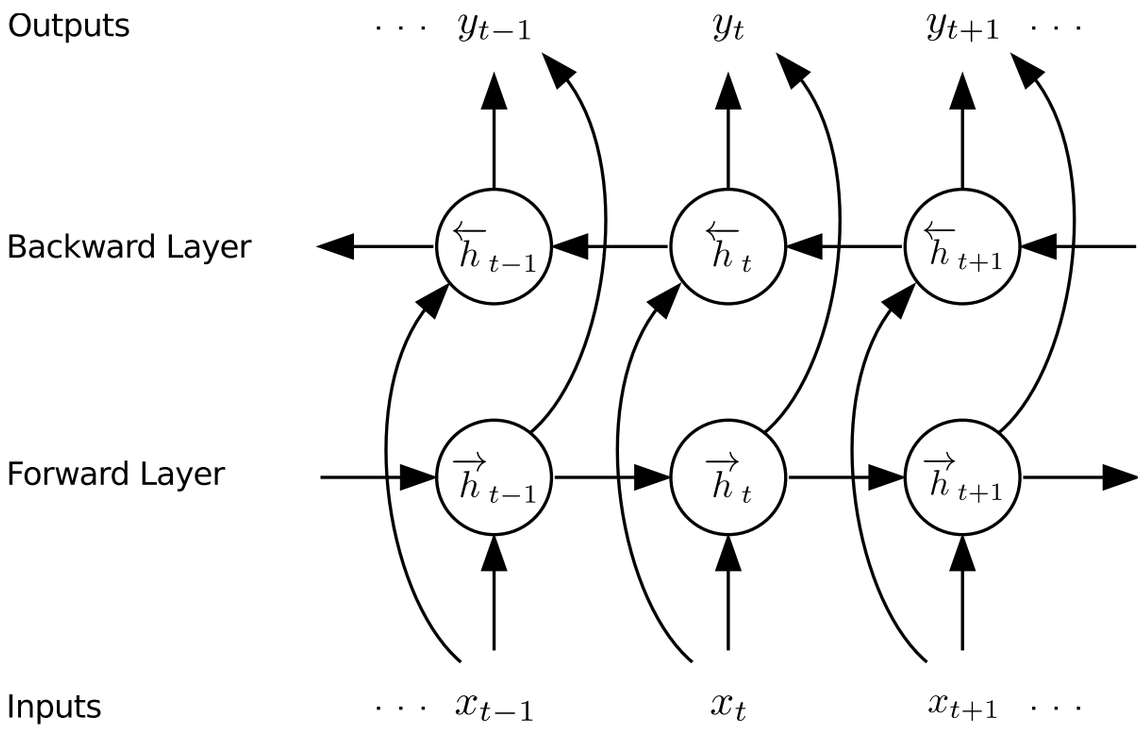
- Significant improvements in ASR performance by using neural models for both these components within the ASR pipeline
- However, there are limitations to using neural networks for a single component within such a complex pipeline

Motivation for end-to-end ASR systems

- Limitations:
  - Objective function optimized in neural networks very different from final evaluation metric (i.e. word transcription accuracy)
  - Additionally, frame-level training targets derived from HMM-based alignments
  - Pronunciation dictionaries are used to map from words to phonemes; expensive resource to create
- Can we build a single RNN architecture that represents the entire ASR pipeline?

End-to-End ASR Systems

Network Architecture



$$\overrightarrow{h}_t = \mathcal{H}\left(W_{x\overrightarrow{h}}x_t + W_{\overrightarrow{h}\overrightarrow{h}}\overrightarrow{h}_{t-1} + b_{\overrightarrow{h}}\right)$$

$$\overleftarrow{h}_t = \mathcal{H}\left(W_{x\overleftarrow{h}}x_t + W_{\overleftarrow{h}\overleftarrow{h}}\overleftarrow{h}_{t+1} + b_{\overleftarrow{h}}\right)$$

$$y_t = W_{\overrightarrow{h}y}\overrightarrow{h}_t + W_{\overleftarrow{h}y}\overleftarrow{h}_t + b_o$$

- Input: Acoustic feature vectors. Output: Characters
- Long Short-Term Memory (LSTM) units (with in-built memory cells) are used to implement $\mathcal{H}$ (in eqns above)
- Deep bidirectional LSTMs: Stack multiple bidirectional LSTM layers

---

Image from: Graves & Jaitley, Towards End-to-End Speech Recognition with Recurrent Neural Networks, ICML 14

Connectionist Temporal Classification (CTC)

- RNNs in ASR are trained at the frame-level and typically require alignments between the acoustics and the word sequence during training telling you which label (e.g. triphone state) should be output at each timestep

- CTC tries to get around this!

- This is an objective function that allows RNN training without this explicit alignment step: CTC considers all possible alignments

CTC: Pre-requisites

- Augment the output vocabulary with an additional “blank” (_) label

- For a given label sequence, there can be multiple alignments: (x, y, z) could correspond to (x, _, y, _, _, z) or (_, x, x, _, y, z)

- Define a 2-step operator B that reduces a label sequence by first, removing repeating labels and second, removing blanks. B(“x, _, y, _, _, z”) = B(“_, x, x, _, y, z”) = “x, y, z”

CTC Objective Function

- CTC objective function is the probability of an output label sequence y given an utterance x

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

- Here, we sum over all possible alignments for y, enumerated by $B^{-1}(y)$

- CTC assumes that Pr(a|x) can be computed as $\prod_{t=1}^{T} \Pr(a_t|x)$

  - i.e. CTC assumes that outputs at each time-step are conditionally independent given the input

- Efficient dynamic programming algorithm to compute this loss function and its gradients [GJ14]

[GJ14] Towards End-to-End Speech Recognition with Recurrent Neural Networks, ICML 14

Decoding

- Pick the single most probable output at every time step

$$\arg\max_{y} \Pr(y|x) \approx B(\arg\max_{a} Pr(a|x))$$

- Decoding is at the word level: Use a beam search algorithm to integrate a dictionary and a language model

- Different algorithm from the one used with HMM-based systems

Experimental Results

Table 1. **Wall Street Journal Results.** All scores are word error rate/character error rate (where known) on the evaluation set. 'LM' is the Language model used for decoding. '14 Hr' and '81 Hr' refer to the amount of data used for training.

SYSTEM	LM	14 HR	81 HR
RNN-CTC	NONE	74.2/30.9	30.1/9.2
RNN-CTC	DICTIONARY	69.2/30.0	24.0/8.0
RNN-CTC	MONOGRAM	25.8	15.8
RNN-CTC	BIGRAM	15.5	10.4
RNN-CTC	TRIGRAM	13.5	8.7
BASELINE	NONE	—	—
BASELINE	DICTIONARY	56.1	51.1
BASELINE	MONOGRAM	23.4	19.9
BASELINE	BIGRAM	11.6	9.4
BASELINE	TRIGRAM	9.4	7.8
COMBINATION	TRIGRAM	—	6.7

Table from: Graves & Jaitley, Towards End-to-End Speech Recognition with Recurrent Neural Networks, ICML 14

Sample char-level transcripts

target: *TO ILLUSTRATE THE POINT A PROMINENT MIDDLE EAST ANALYST IN WASHINGTON RECOUNTS A CALL FROM ONE CAMPAIGN*
output: *TWO ALSTRAIT THE POINT A PROMINENT MIDILLE EAST ANALYST IM WASHINGTON RECOUNCACALL FROM ONE CAMPAIGN*

target: *T. W. A. ALSO PLANS TO HANG ITS BOUTIQUE SHINGLE IN AIRPORTS AT LAMBERT SAINT*
output: *T. W. A. ALSO PLANS TOHING ITS BOOTIK SINGLE IN AIRPORTS AT LAMBERT SAINT*

target: *ALL THE EQUITY RAISING IN MILAN GAVE THAT STOCK MARKET INDIGESTION LAST YEAR*
output: *ALL THE EQUITY RAISING IN MULONG GAVE THAT STACRK MARKET IN TO JUSTIAN LAST YEAR*

target: *THERE'S UNREST BUT WE'RE NOT GOING TO LOSE THEM TO DUKAKIS*
output: *THERE'S UNREST BUT WERE NOT GOING TO LOSE THEM TO DEKAKIS*

Another end-to-end system

- Decoding is still at the word level. Out-of-vocabulary (OOV) words cannot be handled.

- Build a system that is trained and decoded entirely at the character-level.

  - This would enable the transcription of OOV words, disfluencies, etc.

- [M et al.]: Shows results on the Switchboard task. Matches a GMM-HMM baseline system but underperforms compared to an HMM-DNN baseline.

[M et al.]:Maas et al., “Lexicon Free Conversational Speech Recognition with Neural Networks”, NAACL 15

Model Specifics

- Approach consists of two neural models:
  - A deep bidirectional RNN (DBRNN) mapping acoustic features to character sequences (Trained using CTC.)
  - A neural network character language model

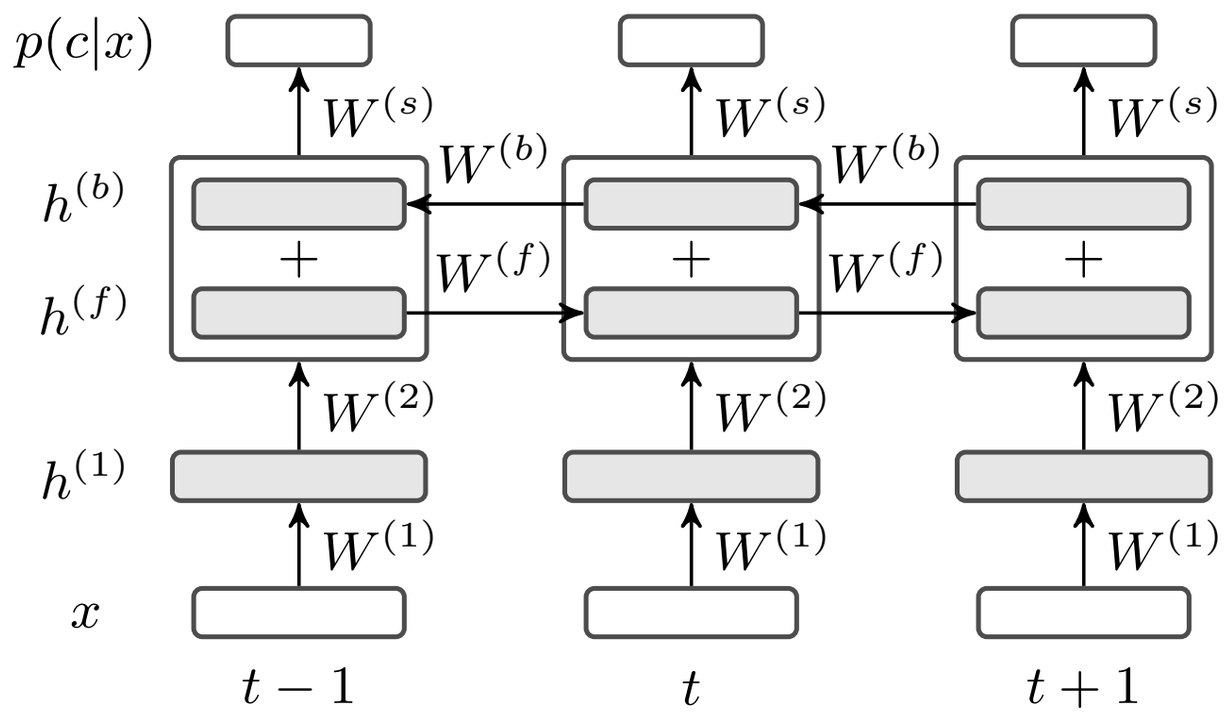


Image from Maas et al., “Lexicon Free Conversational Speech Recognition with Neural Networks”, NAACL 15

Decoding

- Simplest form: Decode without any language model
- Beam Search decoding:
  - Combine DBRNN outputs with a char-level language model
  - Char-level language model applied at every time step (unlike word models)
  - Circumvents the issue of handling OOV words during decoding

Experimental Results

Method	CER	EV	CH	SWBD
HMM-GMM	23.0	29.0	36.1	21.7
HMM-DNN	17.6	21.2	27.1	15.1
HMM-SHF	NR	NR	NR	12.4
CTC no LM	27.7	47.1	56.1	38.0
CTC+5-gram	25.7	39.0	47.0	30.8
CTC+7-gram	24.7	35.9	43.8	27.8
CTC+NN-1	24.5	32.3	41.1	23.4
CTC+NN-3	24.0	30.9	39.9	21.8
CTC+RNN	24.9	33.0	41.7	24.2
CTC+RNN-3	24.7	30.8	40.2	21.4

Table 1: Character error rate (CER) and word error rate results on the Eval2000 test set. We report word error rates on the full test set (EV) which consists of the Switchboard (SWBD) and CallHome (CH) subsets. As baseline systems we use an HMM-GMM system and HMM-DNN system. We evaluate our DBRNN trained using CTC by decoding with several character-level language models: 5-gram, 7-gram, densely connected neural networks with 1 and 3 hidden layers (NN-1, and NN-3), as well as recurrent neural networks s with 1 and 3 hidden layers. We additionally include results from a state-of-the-art HMM-based system (HMM-DNN-SHF) which does not report performance on all metrics we evaluate (NR).

Image from Maas et al., “Lexicon Free Conversational Speech Recognition with Neural Networks”, NAACL 15

Sample Test Utterances

#	Method	Transcription
(1)	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
(2)	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
(3)	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

Image from Maas et al., "Lexicon Free Conversational Speech Recognition with Neural Networks", NAACL 15

Analysis

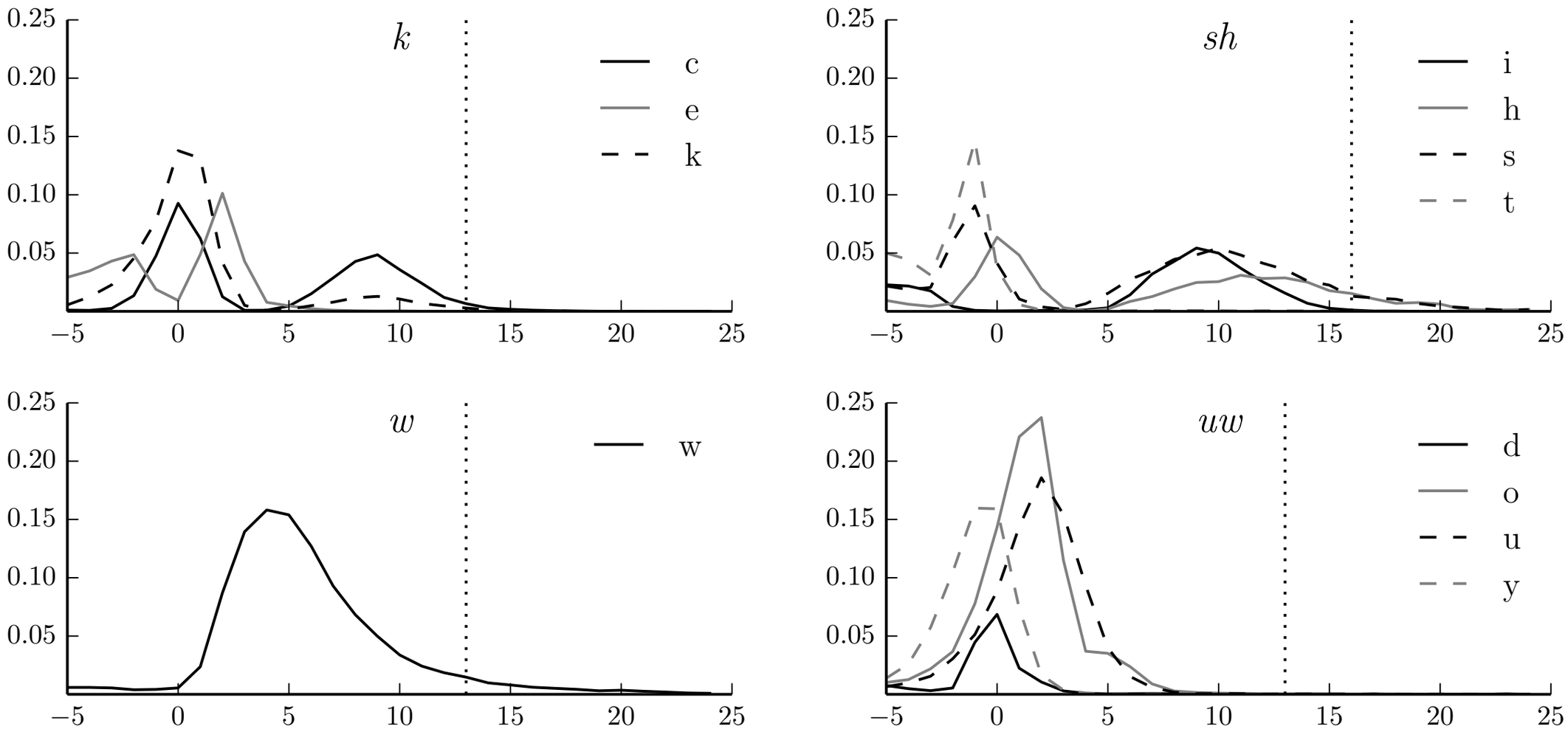


Image from Maas et al., “Lexicon Free Conversational Speech Recognition with Neural Networks”, NAACL 15

A truly end-to-end system

- Build a truly end-to-end model that subsumes all the standard ASR components (ideally, without any additional language model during decoding)
- Listen, Attend and Spell [LAS)]: Makes *no independence assumptions* (unlike the CTC models) about the prob. distribution of the output sequences given the input

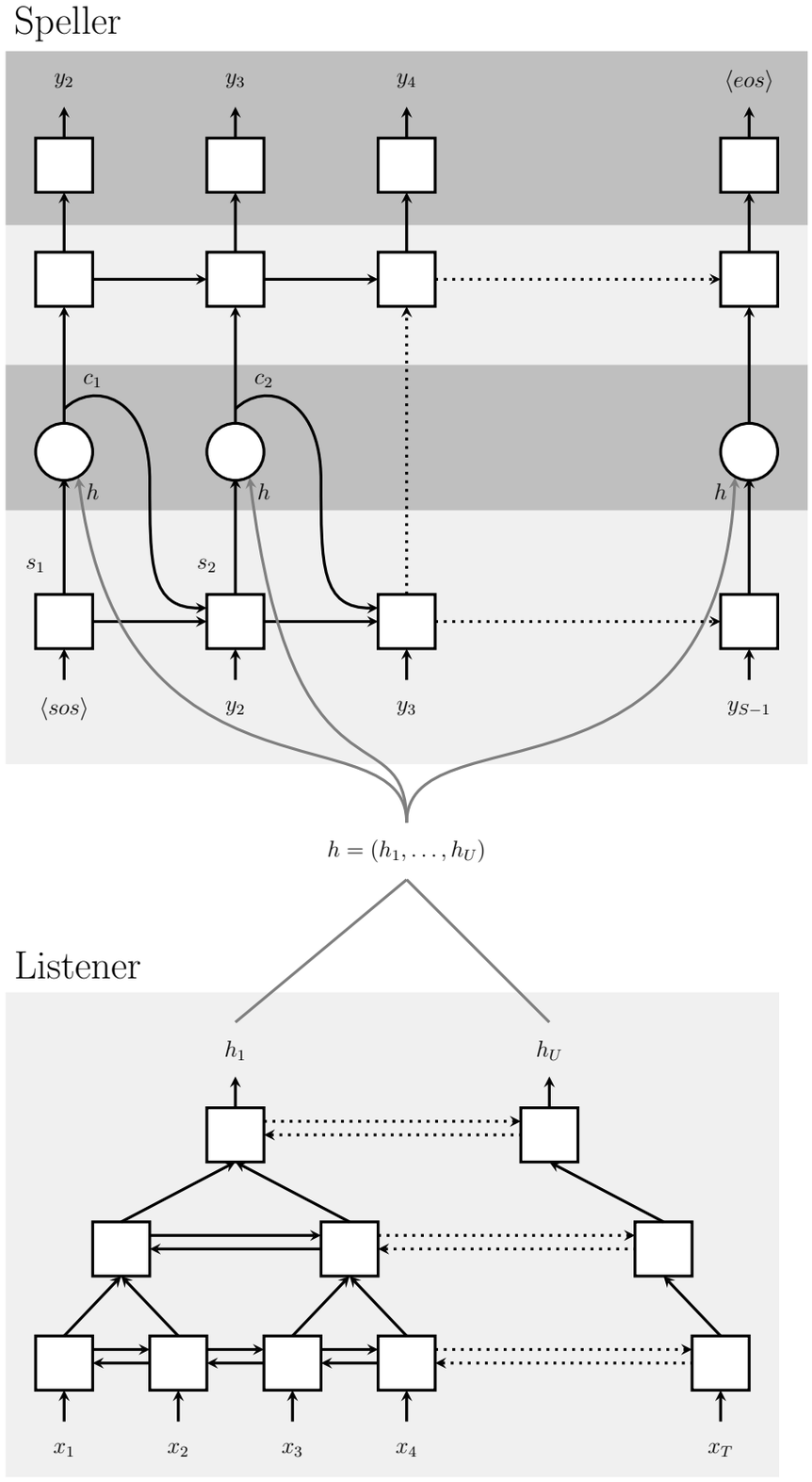
$$P(\mathbf{y}|\mathbf{x}) = \prod_i P(y_i|\mathbf{x}, y_{<i})$$

- Based on the sequence-to-sequence learning framework with attention

---

[LAS]: Chan et al., Listen, Attend and Spell: A NN for LVCSR, ICASSP 2016

The Model



- The Listen, Attend & Spell (LAS) architectures consists of
  - the listener (**Listen**): an acoustic model encoder. Deep BLSTMs with a pyramidal structure: reduces the time resolution by a factor of 2 in each layer
  - the speller (**AttendAndSpell**): an attention-based decoder. Consumes ***h*** and produces a prob. distr. over characters

$$\mathbf{h} = \text{Listen}(\mathbf{x})$$

$$P(y_i|\mathbf{x}, y_{<i}) = \text{AttendAndSpell}(y_{<i}, \mathbf{h})$$

Image from: Chan et al., Listen, Attend and Spell: A NN for LVCSR, ICASSP 2016

Attend and spell

- Produces a distribution over characters conditioned on all characters seen previously

$$
\begin{aligned}
c_i &= \text{AttentionContext}(s_i, \mathbf{h}) \\
s_i &= \text{RNN}(s_{i-1}, y_{i-1}, c_{i-1}) \\
P(y_i|\mathbf{x}, y_{<i}) &= \text{CharacterDistribution}(s_i, c_i)
\end{aligned}
$$

- At each decoder time-step *i*, AttentionContext computes a score for each encoder step *u*, which is then converted into softmax probabilities that are linearly combined to compute c_i

$$
\begin{aligned}
e_{i,u} &= \langle \phi(s_i), \psi(h_u) \rangle \\
\alpha_{i,u} &= \frac{\exp(e_{i,u})}{\sum_{u'} \exp(e_{i,u'})} \\
c_i &= \sum_u \alpha_{i,u} h_u
\end{aligned}
$$

Training and Decoding

- Training
  - Train the parameters of the model to maximize the log probability of the training instances

$$\tilde{\theta} = \max_{\theta} \sum_{i} \log P(y_i | \mathbf{x}, \tilde{y}_{<i}; \theta)$$

- Decoding
  - Simple left-to-right beam search
  - Beams can be rescored with a language model

Experiments

Table 1: WER comparison on the clean and noisy Google voice search task. The CLDNN-HMM system is the state-of-the-art, the Listen, Attend and Spell (LAS) models are decoded with a beam size of 32. Language Model (LM) rescoring can be beneficial.

Model	**Clean WER**	**Noisy WER**
CLDNN-HMM [22]	8.0	8.9
LAS	14.1	16.5
LAS + LM Rescoring	10.3	12.0

- Listen function used 3 layers of BLSTM (512 nodes); AttendAndSpell used a 2-layer LSTM (256 nodes)

- Constraining the beam search with a dictionary had no impact on WER

Analysis

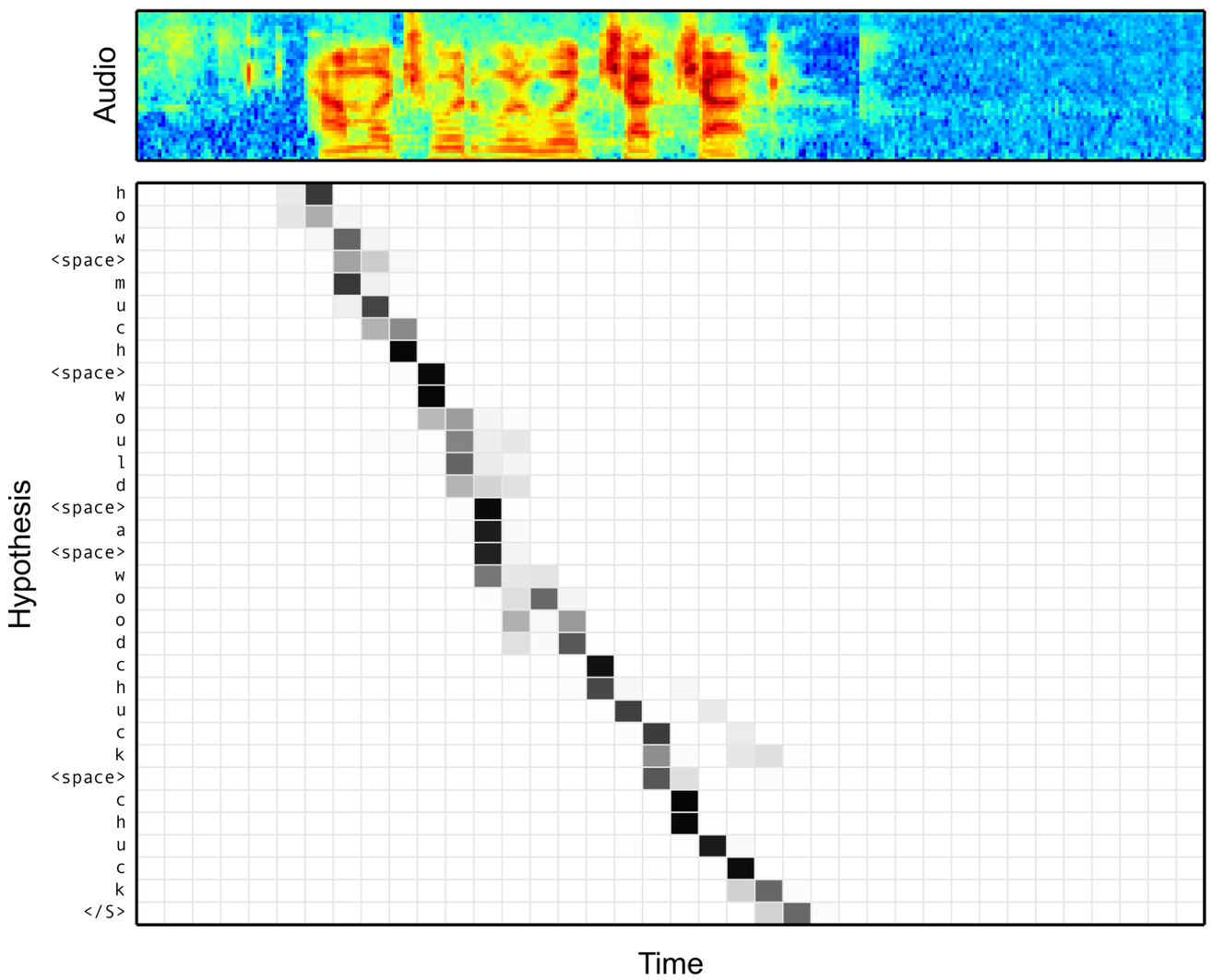


Fig. 2: Alignments between character outputs and audio signal produced by the Listen, Attend and Spell (LAS) model for the utterance "how much would a woodchuck chuck". The content based atten-

Beam	**Text**	$\log P$	**WER**
Truth	call aaa roadside assistance	-	-
1	call aaa roadside assistance	-0.57	0.00
2	call triple a roadside assistance	-1.54	50.00
3	call trip way roadside assistance	-3.50	50.00
4	call xxx roadside assistance	-4.44	25.00

Image from: Chan et al., Listen, Attend and Spell: A NN for LVCSR, ICASSP 2016

Summary

- We saw three ASR systems progressing from:

  A. BiRNN-based models that directly transcribe audio data into text (without any intermediate phonetic representation)

     - However, decoding is still at the word level (integrating a dictionary and language model)

  B. BiRNN-based models operating entirely at the character level

     - Still needs a char-based language model to perform competitively with a baseline GMM-HMM system

  C. BiRNN-based end-to-end model consisting of encoder-decoder RNNs: Entire model, including the LM, is trained jointly.

None of these systems match the performance of an HMM-DNN system yet.