



The 6th Asian Conference on Pattern Recognition (ACPR2021) Tutorial

State-of-the-Art of End-to-End Speech Recognition

Zhijian Ou

Speech Processing and Machine Intelligence (SPMI) Lab, Tsinghua University

<http://oa.ee.tsinghua.edu.cn/ouzhijian/>

November 9, 2021

ASR brief history

1970 – 2010: 1st Generation

HMM	<ul style="list-style-type: none">• F. Jelinek, “Continuous speech recognition by statistical methods”, Proc. of the IEEE, 1976.• J. Baker, “The DRAGON system--An overview”, T-ASSP, 1975.
GMM	<ul style="list-style-type: none">• B.H. Juang, “Maximum-likelihood estimation for mixture multivariate stochastic observations of Markov chains”, AT&T Technical Journal, 1985.
N-gram, Smoothing	<ul style="list-style-type: none">• F. Jelinek & R.L. Mercer, “Interpolated estimation of Markov source parameters from sparse data”, Proc. Workshop on Pattern Recognition in Practice, 1980.• F. Jelinek, “The development of an Experimental Discrete Dictation Recognizer”, Proc. of the IEEE, 1985.
Tree based state tying	<ul style="list-style-type: none">• S. Young, J.J. Odell, P.C. Woodland, “Tree-based state tying for high accuracy acoustic modeling”, HLT workshop, 1994.
MAP, MLLR	<ul style="list-style-type: none">• C.H. Lee, C.H. Lin, B.H. Juang, “A study on speaker adaptation of the parameters of continuous density hidden Markov models”, T-IP, 1991.• C.J. Leggetter & P.C. Woodland, “Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models”, Computer Speech and Language, 1995.
fMLLR, Speaker adaptive training	<ul style="list-style-type: none">• M.J.F. Gales, “Maximum likelihood linear transformations for HMM-based speech recognition”, Computer Speech and Language, 1998.
WFST	<ul style="list-style-type: none">• M. Mohri. Finite-State Transducers in Language and Speech Processing. Computational Linguistics, 1997.• M. Mohri, F. Pereira, and M. Riley, “Speech Recognition with Weighted Finite-State Transducers”, 2008.
Discriminative Training, MMI, MPE	<ul style="list-style-type: none">• D. Povey, “Discriminative training for large vocabulary speech recognition”, Ph.D. dissertation, 2003.

ASR brief history

2011 – now: 2nd Generation

DNN-HMM	<ul style="list-style-type: none">• A. Mohamed, G. Dahl, and G. Hinton, “Deep belief networks for phone recognition”, NIPS Workshop Deep Learning for Speech Recognition and Related Applications, 2009.• G. Dahl, et al, “Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition”, T-ASLP, 2012.• F. Seide, G. Li, and D. Yu, “Conversational speech transcription using context-dependent deep neural networks”, Interspeech, 2011.• D. Povey, et al, "Purely sequence-trained neural networks for ASR based on lattice-free MMI", Interspeech 2016.
NN-LM	<ul style="list-style-type: none">• Bengio, et al, “A Neural Probabilistic Language Model”, NIPS, 2001.• Mikolov, et al, "Recurrent neural network based language model", Interspeech, 2010.
CTC	<ul style="list-style-type: none">• A. Graves, et al, “Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks”, ICML, 2006.• H. Sak, et al, “Learning acoustic frame labeling for speech recognition with recurrent networks”, ICASSP, 2015.• Y. Miao, et al, “EESSEN: End-to-end speech recognition using deep RNN models and WFST-based decoding”, ASRU, 2015.
Attention seq2seq	<ul style="list-style-type: none">• D. Bahdanau, et al, “Neural machine translation by jointly learning to align and translate”, ICLR 2015.• J. K. Chorowski, et al, “Attention-based models for speech recognition,” NIPS, 2015.• W. Chan, et al @ google, “Listen, attend and spell: A neural network for large vocabulary conversational speech recognition”, ICASSP, 2016.
RNN Transducer	<ul style="list-style-type: none">• A. Graves, “Sequence transduction with recurrent neural networks,” ICML 2012 Workshop on Representation Learning.• E. Battenberg, et al @ Baidu, “Exploring neural transducers for end-to-end speech recognition”, ASRU 2017.• K. Rao, et al @ Google, “Exploring architectures, data and units for streaming end-to-end speech recognition with RNN-transducer”, ASRU 2017
Transformer	<ul style="list-style-type: none">• A. Vaswani, et al @ google, "Attention Is All You Need", NIPS, 2017.
CRF	<ul style="list-style-type: none">• H. Xiang, Z. Ou. "CRF-based Single-stage Acoustic Modeling with CTC Topology", ICASSP, 2019.

Content

I. Basics for end-to-end speech recognition (15*6=90 min)

1. Probabilistic graphical modeling (PGM) framework
 2. Classic hybrid DNN-HMM models
 3. Connectionist Temporal Classification (CTC)
 4. Attention based Encoder-Decoder (AED)
 5. RNN Transducer (RNNT)
 6. Conditional random fields and sequence discriminative training
-

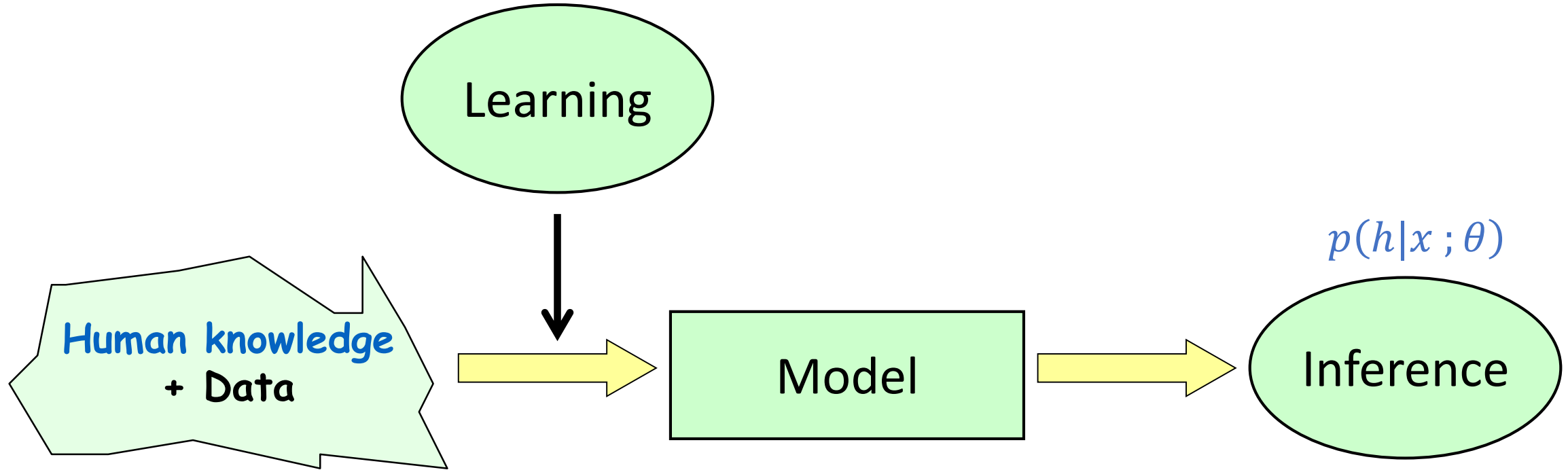
II. Improving end-to-end speech recognition (20*4=80 min)

15-minute break

1. Data-efficiency
2. Neural architecture search
3. Multilingual and crosslingual ASR
4. Language modeling

III. Open questions and future directions (10 min)

Probabilistic Framework



$p(x, h; \theta)$: Generative model, e.g., Hidden Markov Model (HMM)

$p(h|x; \theta)$: Discriminative model, e.g., Conditional Random Field (CRF)

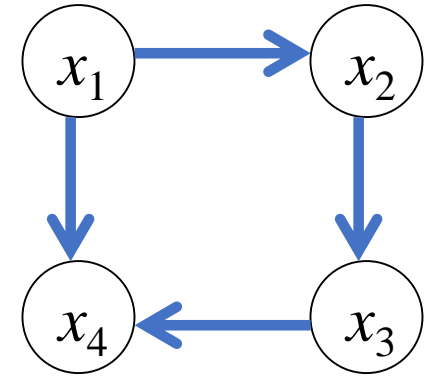
We need probabilistic models, besides neural nets.

Probabilistic Graphical Modeling (PGM) Framework

• Directed Graphical Models / Bayesian Networks (BNs)

- Self-normalized/Local-normalized
- e.g. Hidden Markov Models (HMMs), Neural network (NN) based classifiers, Variational AutoEncoders (VAEs), Generative Adversarial Networks (GANs), auto-regressive models (e.g. RNNs/LSTMs)

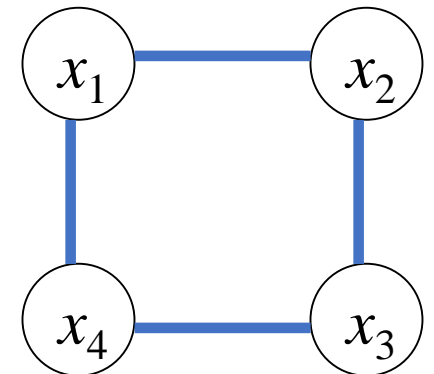
$$p(x_1, x_2, x_3, x_4) = p(x_1)p(x_2|x_1)p(x_3|x_2)p(x_4|x_1, x_3)$$



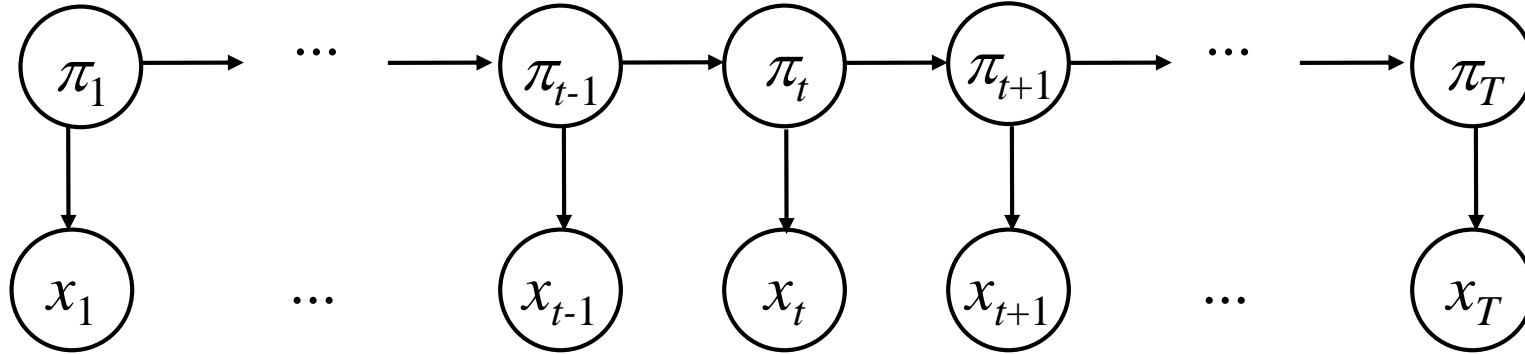
• Undirected Graphical Models / Random Fields (RFs) / Energy-based models

- Involves the normalizing constant Z / Globally-normalized
- e.g. Ising model, Conditional Random Fields (CRFs)

$$p(x_1, x_2, x_3, x_4) = \frac{1}{Z} \Phi(x_1, x_2) \Phi(x_2, x_3) \Phi(x_3, x_4) \Phi(x_1, x_4)$$



HMM Viewed as Directed Graphical Model



The joint probability distribution of a hidden Markov model (HMM) :

$$p(\pi_{1:T}, x_{1:T}) = p(\pi_1) \prod_{t=1}^{T-1} p(\pi_{t+1} | \pi_t) \prod_{t=1}^T p(x_t | \pi_t)$$

State Initial
Distr.

State Transition
Distr.

State Observation
Distr.

I. Basics for end-to-end speech recognition

1. Probabilistic graphical modeling (PGM) framework
 - 2. **Classic hybrid DNN-HMM models**
 3. Connectionist Temporal Classification (CTC)
 4. Attention based encoder-decoder (AED)
 5. RNN transducer (RNNT)
 6. Conditional random fields and sequence discriminative training
-

II. Improving end-to-end speech recognition

15-minute break

1. Data-efficiency
2. Neural architecture search
3. Multilingual and crosslingual ASR
4. Language modeling

III. Open questions and future directions

ASR: Basics

ASR (Automatic Speech Recognition) is a seq. discriminative problem

- For acoustic observations $\mathbf{x} \triangleq x_1, \dots, x_T$, find the most likely labels $\mathbf{y} \triangleq y_1, \dots, y_L$

1. How to obtain $p(\mathbf{y} | \mathbf{x})$

2. How to handle alignment, since $L \neq T$

Separate
neural network architectures
and probabilistic model definitions !

Labels

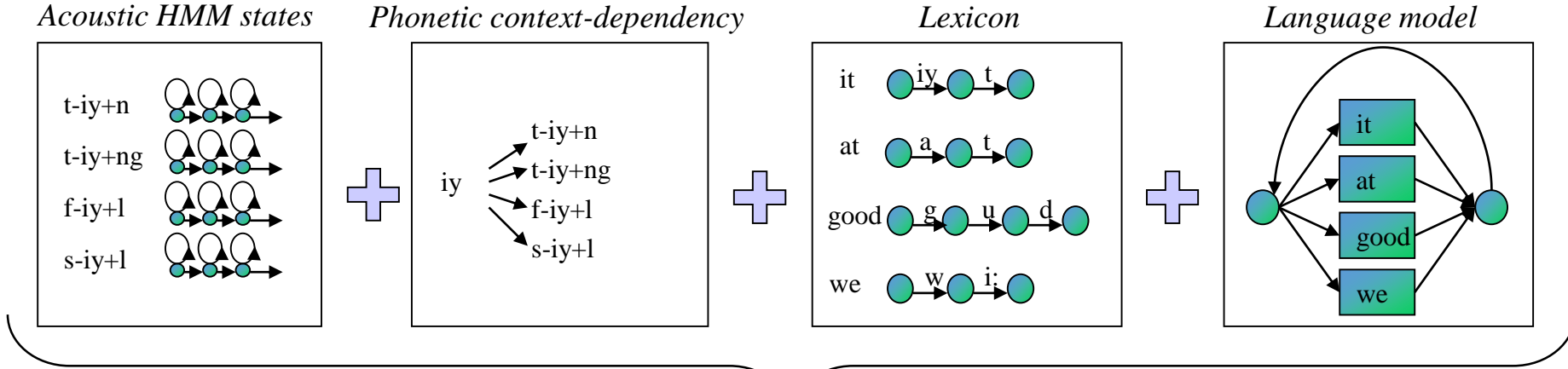
\mathbf{y} $L \neq T$

\parallel							π_7	π_8
y_1						π_6		
\vdots			π_3	π_4	π_5			
y_L	π_1	π_2						

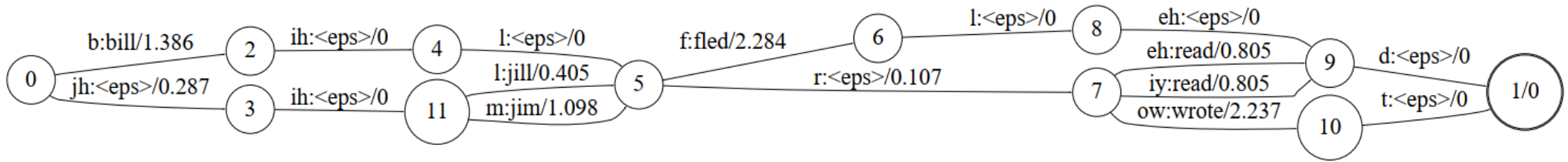
Observations $\mathbf{x} = x_1 \dots x_T$

Example of alignment

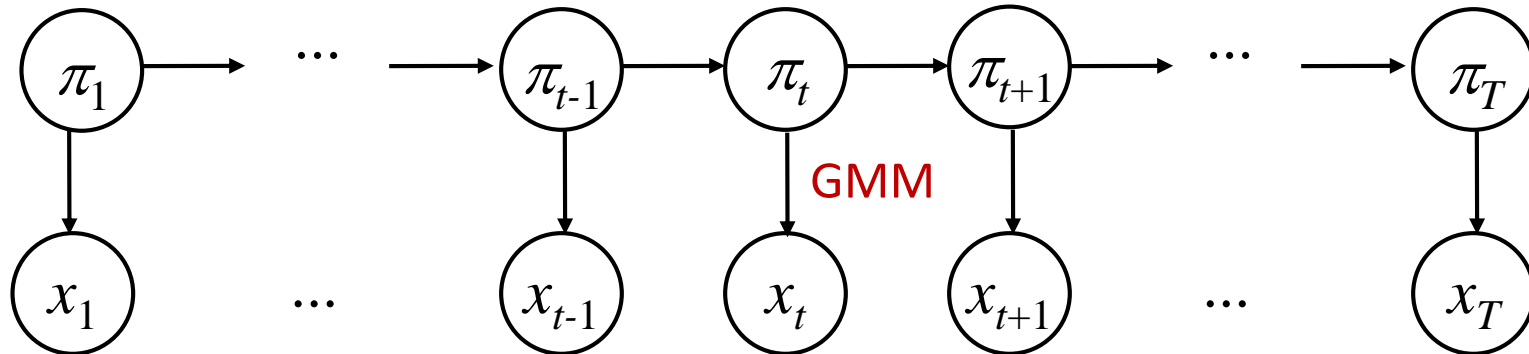
GMM-HMM: state transitions



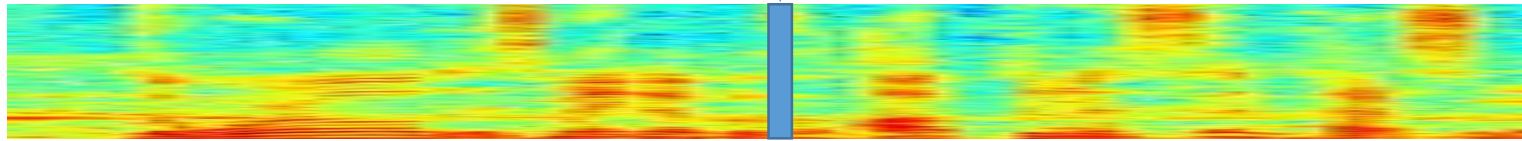
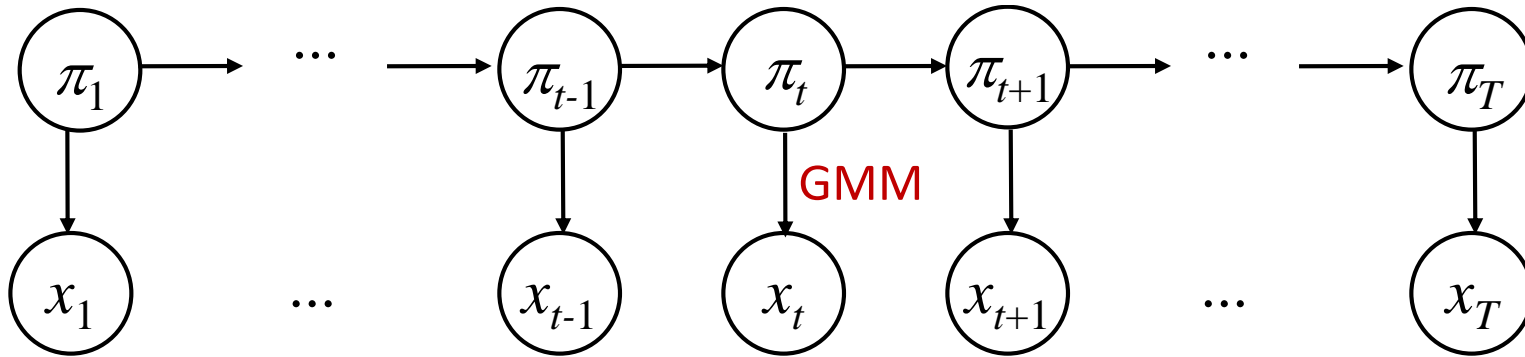
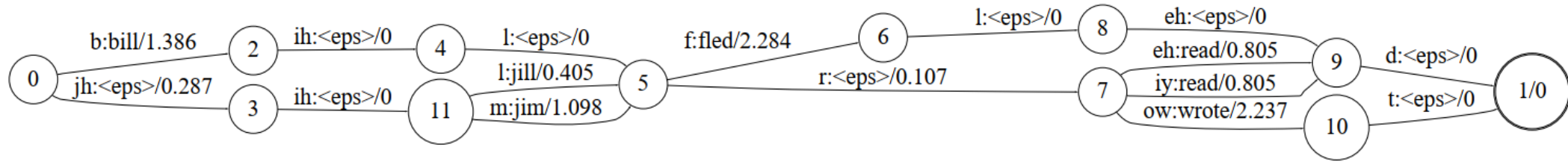
State transitions in π are determined by a state transition graph (WFST), constrained by \uparrow



A path $\pi \triangleq \pi_1, \dots, \pi_T$ uniquely determines a label sequence y , but not vice versa.



GMM-HMM



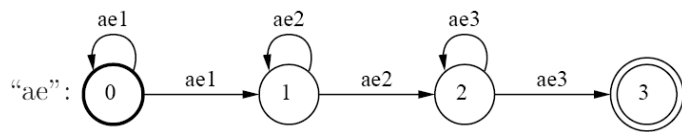
A path π uniquely determines \mathbf{y} via mapping \mathcal{B}_{HMM}

- ▶ Training: Maximum likelihood $p(\mathbf{y}, \mathbf{x}) = \sum_{\pi: \mathcal{B}_{HMM}(\pi)=\mathbf{y}} p(\pi, \mathbf{x})$ via the forward-backward algo.
- ▶ Inference: Viterbi Decoding via $\max_{\pi} p(\pi, \mathbf{x})$

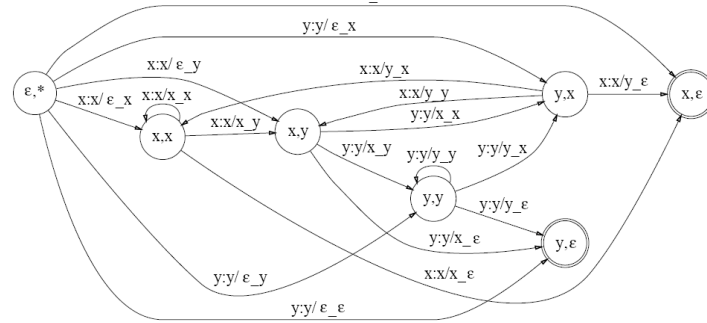
WFST

- WFSTs (weighted finite-state transducers) for Viterbi decoding
 - Pioneered by AT&T in late 1990's [Mohri et al., 2008]

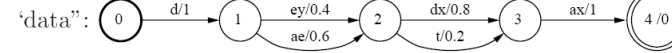
Acoustic HMMs: H



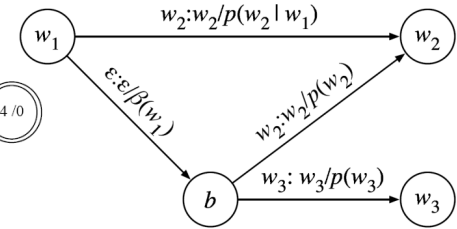
Phonetic context-dependency: C



Lexicon: L



Language model: G



Composed and optimized into a single WFST

$$N = \min \left(\det \left(H \circ \det \left(C \circ \det \left(L \circ G \right) \right) \right) \right)$$

which represents $p(\pi_{t+1} | \pi_t)$ and is used in Viterbi decoder.

Well implemented in Kaldi toolkit <https://github.com/kaldi-asr/kaldi>

DNN-HMM

- ASR state-of-the-art: DNNs of various network architectures (MLP, LSTM, CNN, Transformer, etc.), initially DNN-HMM

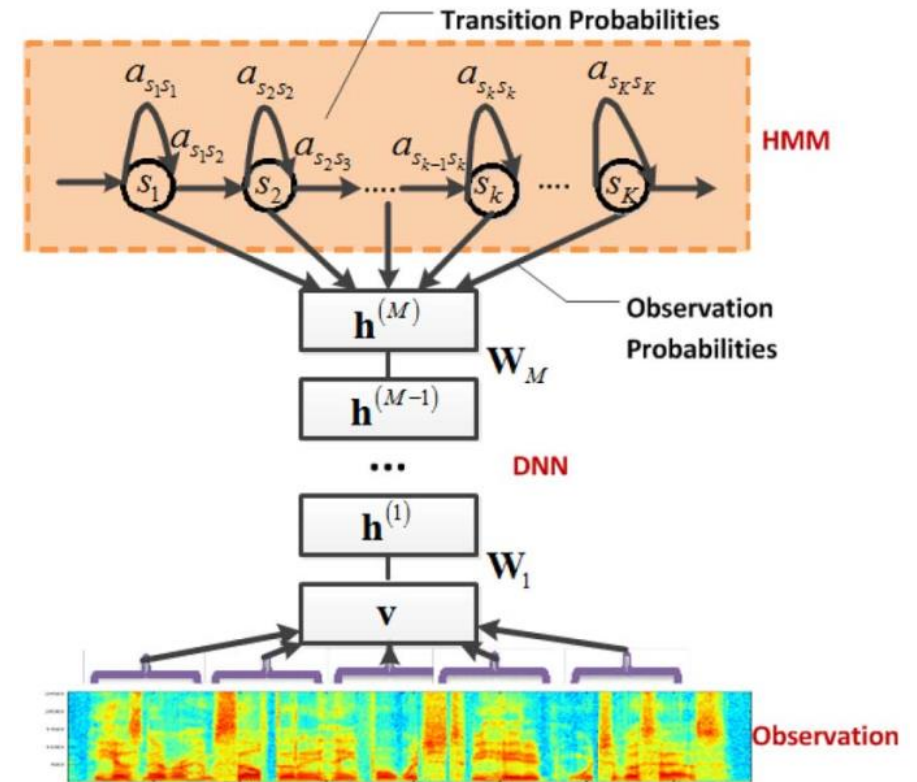
State posterior prob.
estimated from the DNN, which needs
frame-level alignments

Can be ignored.

$$p(x_t|\pi_t) = \frac{p(\pi_t|x_t)p(x_t)}{p(\pi_t)}$$

State prior prob.
estimated from the training data

- Conventionally, multi-stage
 - monophone GMM-HMM
 - alignment & triphone tree building
 - triphone GMM-HMM
 - alignment
 - triphone DNN-HMM



[Dahl, et al., TASLP 2012]

G. Dahl, et al., "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition", TASLP, 2012.

Advancing to end-to-end ASR: motivation

- End-to-end in the sense that:

- Eliminate the construction of GMM-HMMs and phonetic decision-trees, and can be trained from scratch (**flat-start** or **single-stage**)

- In a more strict/ambitious sense:

- Remove the need for a pronunciation lexicon and, even further, train the acoustic and language models jointly rather than separately
- Trained to optimize criteria that are related to the final evaluation metric that we are interested in (typically, word error rate)

- Motivation

- Simplify system pipeline, reduce expert knowledge and labor (such as compiling the ProLex, building phonetic decision trees)

Advancing to end-to-end ASR: techniques

ASR is a *sequence discriminative* problem

- For acoustic observations $\mathbf{x} \triangleq x_1, \dots, x_T$, find the most likely labels $\mathbf{y} \triangleq y_1, \dots, y_L$

- How to obtain $p(\mathbf{y} | \mathbf{x})$
- How to handle alignment, since $L \neq T$

- Need a differentiable sequence-level loss of mapping acoustic sequence \mathbf{y} to label sequence \mathbf{x}

- Explicitly:** introduce hidden state sequence $\boldsymbol{\pi}$, as in Connectionist Temporal Classification (CTC), RNN Transducer (RNNT), CRF
- Implicitly:** as in Attention based Encoder-Decoder (AED)

Labels

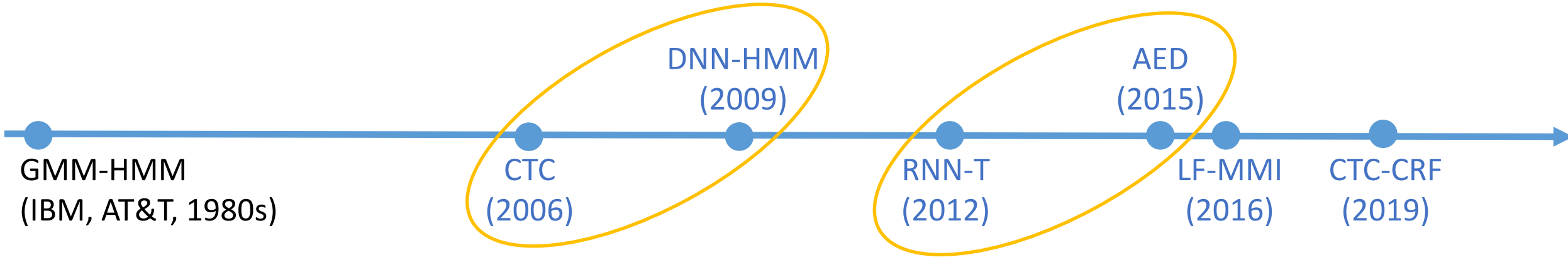
$L \neq T$

\mathbf{y}									
\parallel							π_7	π_8	
y_1							π_6		
\vdots									
y_L									
	π_1	π_2							

Observations $\mathbf{x} = x_1 \dots x_T$

Example of explicit alignment

History



- [CTC] Graves, et al., “Connectionist Temporal Classification: Labelling unsegmented sequence data with RNNs”, ICML 2006.
- [DNN-HMM] A. Mohamed, et al., “Deep belief networks for phone recognition”, NIPS Workshop Deep Learning for Speech Recognition and Related Applications, 2009.
- [RNNT] A. Graves, “Sequence transduction with recurrent neural networks”, ICML 2012 Workshop on Representation Learning.
- [AED] D. Bahdanau, et al., “Neural machine translation by jointly learning to align and translate”, ICLR 2015.
- [LF-MMI] D. Povey, et al., "Purely sequence-trained neural networks for ASR based on lattice-free MMI", INTERSPEECH 2016.
- [CTC-CRF] Xiang&Ou. "CRF-based Single-stage Acoustic Modeling with CTC Topology", ICASSP, 2019.

Content

I. Basics for end-to-end speech recognition

1. Probabilistic graphical modeling (PGM) framework
 2. Classic hybrid DNN-HMM models
 - 3. **Connectionist Temporal Classification (CTC)**
 4. Attention based encoder-decoder (AED)
 5. RNN transducer (RNNT)
 6. Conditional random fields and sequence discriminative training
-

II. Improving end-to-end speech recognition

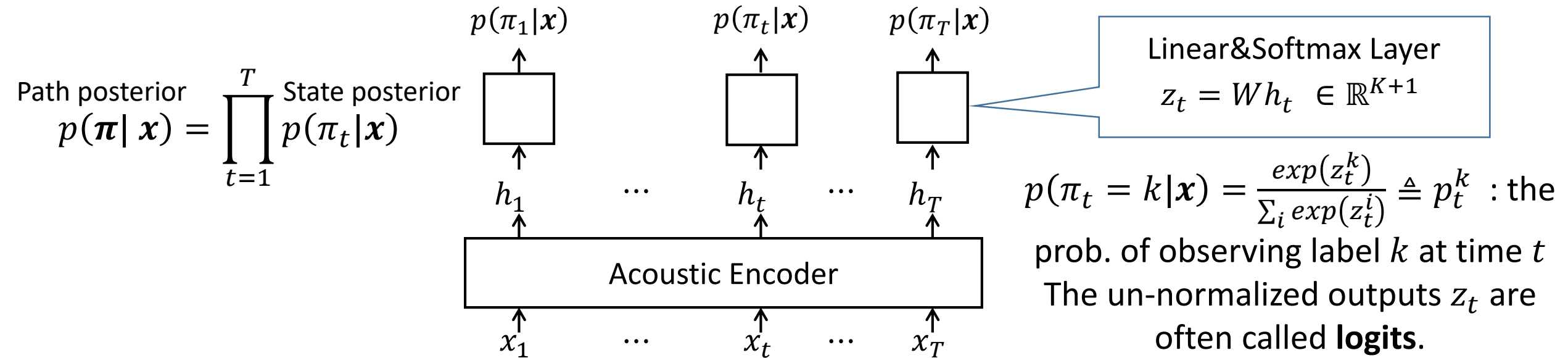
15-minute break

1. Data-efficiency
2. Neural architecture search
3. Multilingual and crosslingual ASR
4. Language modeling

III. Open questions and future directions

CTC: introducing **blank** symbol

- Motivation: training $p(\mathbf{y} | \mathbf{x})$ without the need for frame-level alignments between the acoustics \mathbf{x} and the transcripts \mathbf{y}
 - Introduce a state sequence $\boldsymbol{\pi} \triangleq \pi_1, \dots, \pi_T$, where $\pi_t \in \text{the-alphabet-of-labels} \cup \langle \mathbf{b} \rangle$



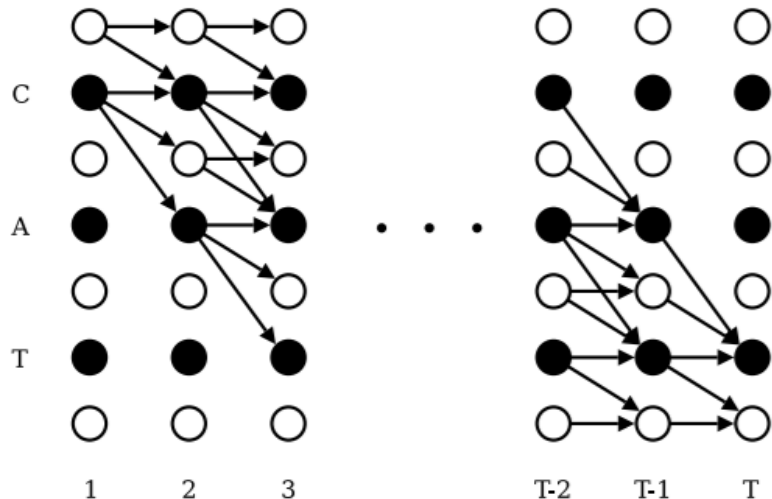
CTC topology

- State topology** refers to the state transition structure in π , which basically determines the mapping \mathcal{B}_{CTC} from π to \mathbf{y}

CTC topology : a mapping \mathcal{B}_{CTC} maps π to \mathbf{y} by

- reducing repetitive symbols to a single symbol;
- removing all blank symbols.

$$B(-CC - -AA - T -) = CAT$$



Path posterior

$$p(\boldsymbol{\pi}|\mathbf{x}) = \prod_{t=1}^T p(\pi_t|\mathbf{x})$$

Label-seq posterior

$$p(\mathbf{y}|\mathbf{x}) = \sum_{\boldsymbol{\pi}: \mathcal{B}_{CTC}(\boldsymbol{\pi})=\mathbf{y}} p(\boldsymbol{\pi}|\mathbf{x})$$

Summing over all possible paths, which map to \mathbf{y}

B	B	c	B	B	a	a	B	B	t
B	c	c	B	a	B	B	B	B	t
...									
B	c	B	B	a	B	B	t	t	B

CTC: the gradient & the forward-backward algorithm I.3 CTC

For logit $z_t^k, 1 \leq t \leq T$

$$\frac{\partial \log p(\mathbf{y}|\mathbf{x})}{\partial z_t^k} = E_{p(\boldsymbol{\pi}|\mathbf{x},\mathbf{y})} \left[\frac{\partial \log p(\boldsymbol{\pi}|\mathbf{x})}{\partial z_t^k} \right] \quad \because \text{Fisher Equality [Ou, arxiv 2018]}$$

$$= E_{p(\boldsymbol{\pi}|\mathbf{x},\mathbf{y})} \left[\frac{\partial \log p_t^{\pi_t}}{\partial z_t^k} \right] \quad \because p(\boldsymbol{\pi}|\mathbf{x}) = \prod_{t=1}^T p_t^{\pi_t}$$

$$= E_{p(\boldsymbol{\pi}|\mathbf{x},\mathbf{y})} [\delta(\pi_t = k) - p_t^k]$$

$$= p(\pi_t = k|\mathbf{x}, \mathbf{y}) - p_t^k$$

i.e., the **error signal** received by the acoustic encoder NN during training

i.e., γ_t^k , the posterior **state occupation probability**, calculated using the alpha-beta variables from the forward-backward algorithm [Rabiner, 1989]

Providing easy derivation and giving insight, not appeared in [Graves, et al., 2006] and elsewhere

CTC: LM integration with WFSTs

- Best-path-decoding or Prefix-search-decoding

$$\max_{\pi} p(\pi | \mathbf{x})$$

$$\max_{\mathbf{y}} p(\mathbf{y} | \mathbf{x})$$

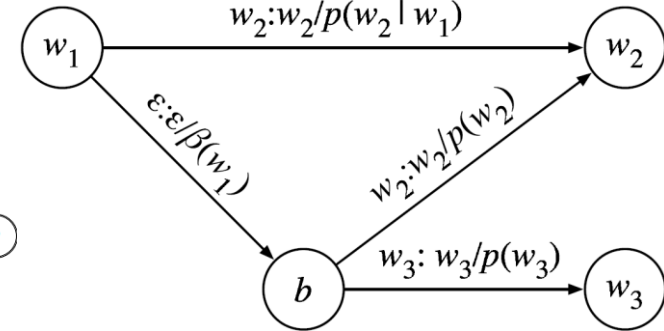
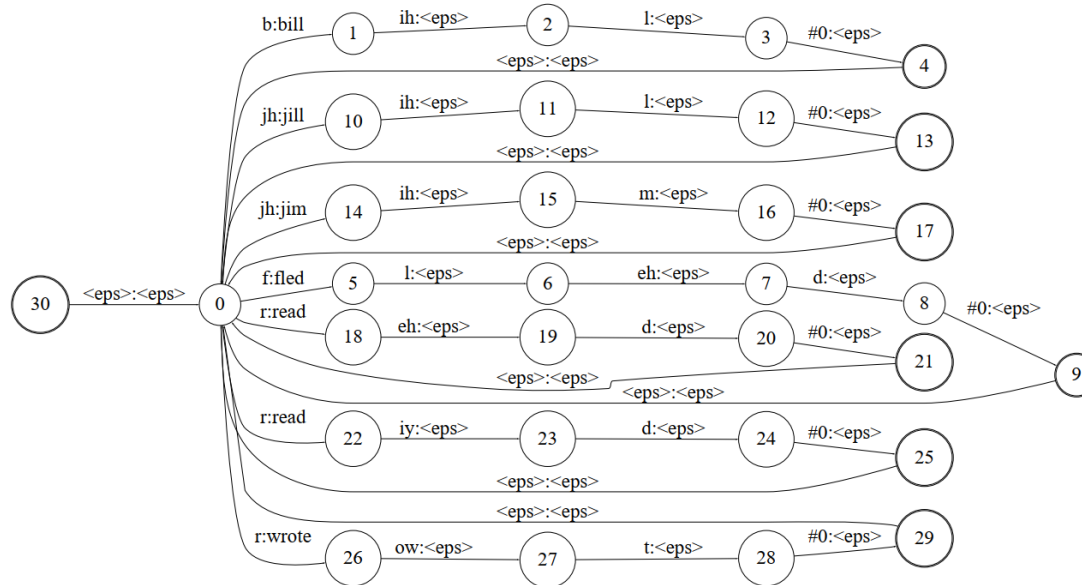
- Incorporate lexicon and LM to improve best-path-decoding

$$\max_{\pi} p(\pi | \mathbf{x}) LM(\mathcal{B}_{CTC}(\pi))$$

WFST representing CTC topology: T

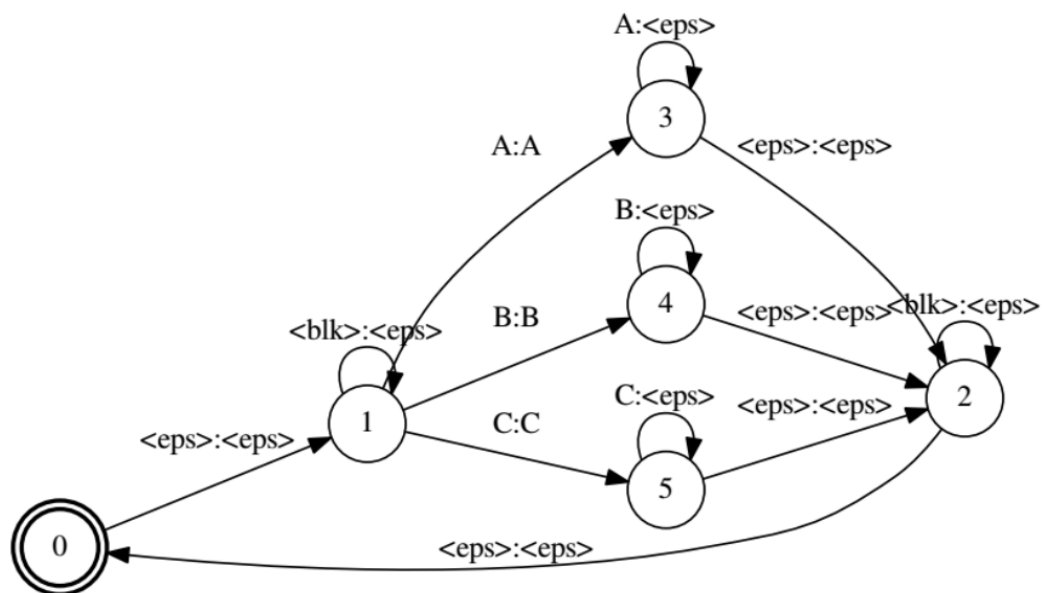
Lexicon: L

Language model: G

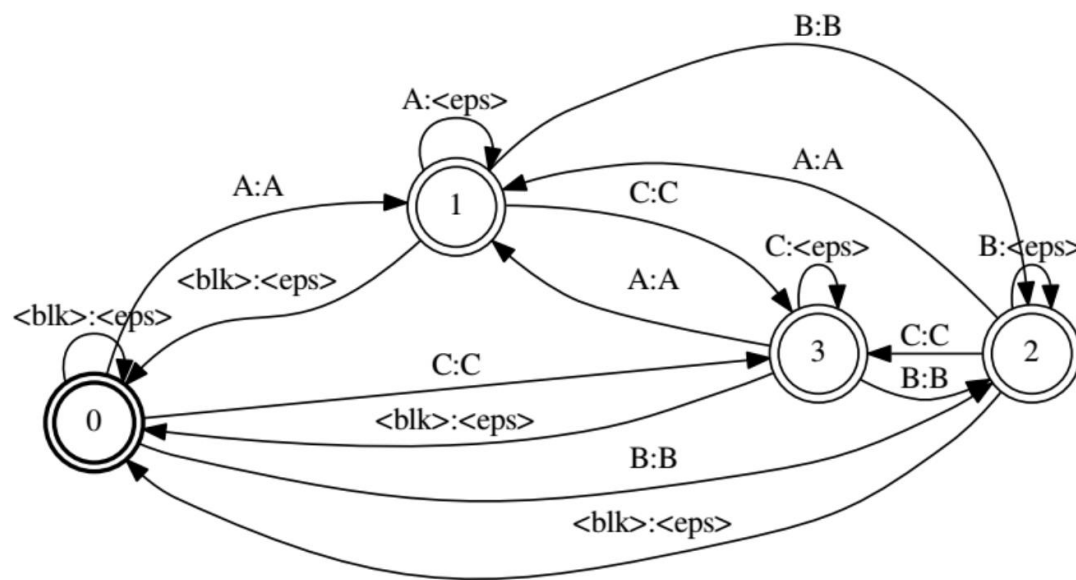


Composed and optimized into a single WFST

WFST representation of CTC topology [Xiang&Ou, 2019] ^{L3 CTC}



EESEN T.fst ✘



Corrected T.fst

WFST	dev		test	
	clean	other	clean	other
Eesen T.fst	3.90%	10.32%	4.11%	10.68%
Corrected T.fst	3.87%	10.28%	4.09%	10.65%

WFST	TLG size	decoding time
Eesen T.fst	208M	700s
Corrected T.fst	181M	672s

Using corrected T.fst performs slightly better; The decoding graph size smaller, and the decoding speed faster.

- Miao, et al., "EESEN: End-to-end speech recognition using deep RNN models and WFST-based decoding", ASRU 2015.
- Xiang&Ou. "CRF-based Single-stage Acoustic Modeling with CTC Topology", ICASSP, 2019.

CTC: Label units

- For WERs in apple-to-apple comparisons
 - mono-phone **clearly better** mono-char, over WSJ-80h, Switchboard-300h, Librispeech-960h [Xiang&Ou, 2019]
 - For low degree of grapheme-phoneme correspondence (e.g., English), wordpiece **slightly worse** than mono-phone; For high degree (e.g., German), **equally strong** [Zheng, et al., 2021]
- **Longer span, more training data needed**
 - Word-level CTC targets, trained on 3,400 hours of speech [Li et al., 2018]

Basic Units of Labels	Label Sequence
phoneme	DH AE1 T N IY1 DH ER0 AH1 V DH EH1 M HH AE1 D K R AO1 S T DH AH0 TH R EH1 SH OW2 L D S IH1 N S DH AH0 D AA1 R K D EY1
character /grapheme	that_neither_of_them_had_crossed_the_threshold_since_the_dark_day_
subword /wordpiece	that_neither_of_them_had_crossed_the_threshold_since_the_dark_day_
word	that neither of them had crossed the threshold since the dark day

- Zheng, et al., "Advancing CTC-CRF Based End-to-End Speech Recognition with Wordpieces and Conformers", 2021.
- J. Li, et al., "Advancing acoustic-to-word CTC model", ICASSP 2018.

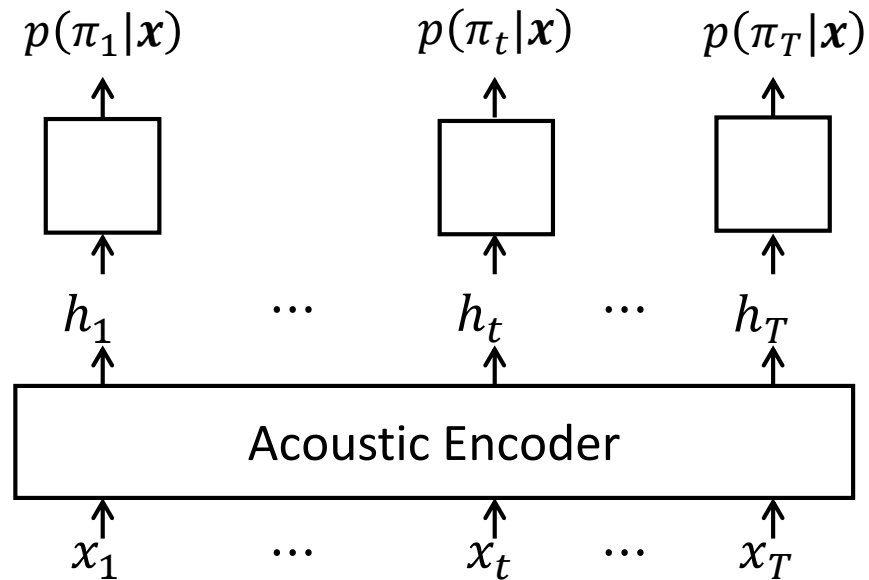
CTC: shortcoming

- Conditional independence assumption

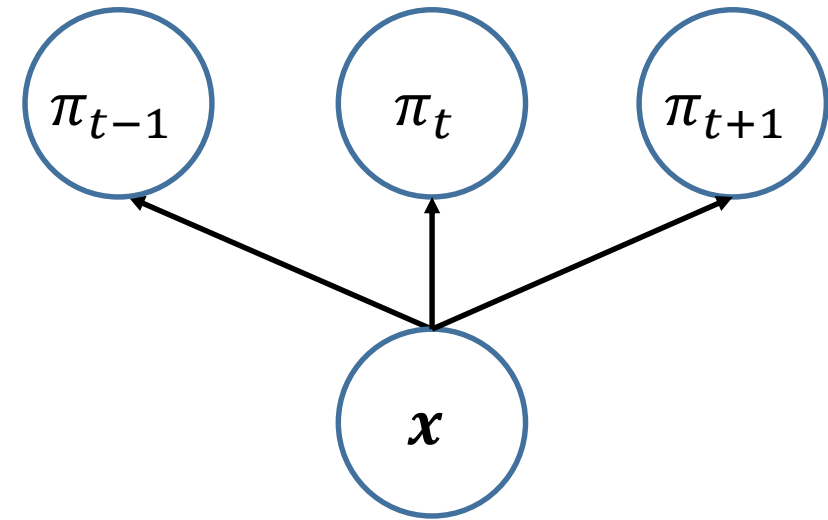
$$p(\boldsymbol{\pi} | \boldsymbol{x}) = \prod_{t=1}^T p(\pi_t | \boldsymbol{x})$$

Overcome

RNN-T
CTC-CRF



Computational flow



Graphical Model Representation

Content

I. Basics for end-to-end speech recognition

1. Probabilistic graphical modeling (PGM) framework
 2. Classic hybrid DNN-HMM models
 3. Connectionist Temporal Classification (CTC)
 - 4. Attention based encoder-decoder (AED)
 5. RNN transducer (RNNT)
 6. Conditional random fields and sequence discriminative training
-

II. Improving end-to-end speech recognition

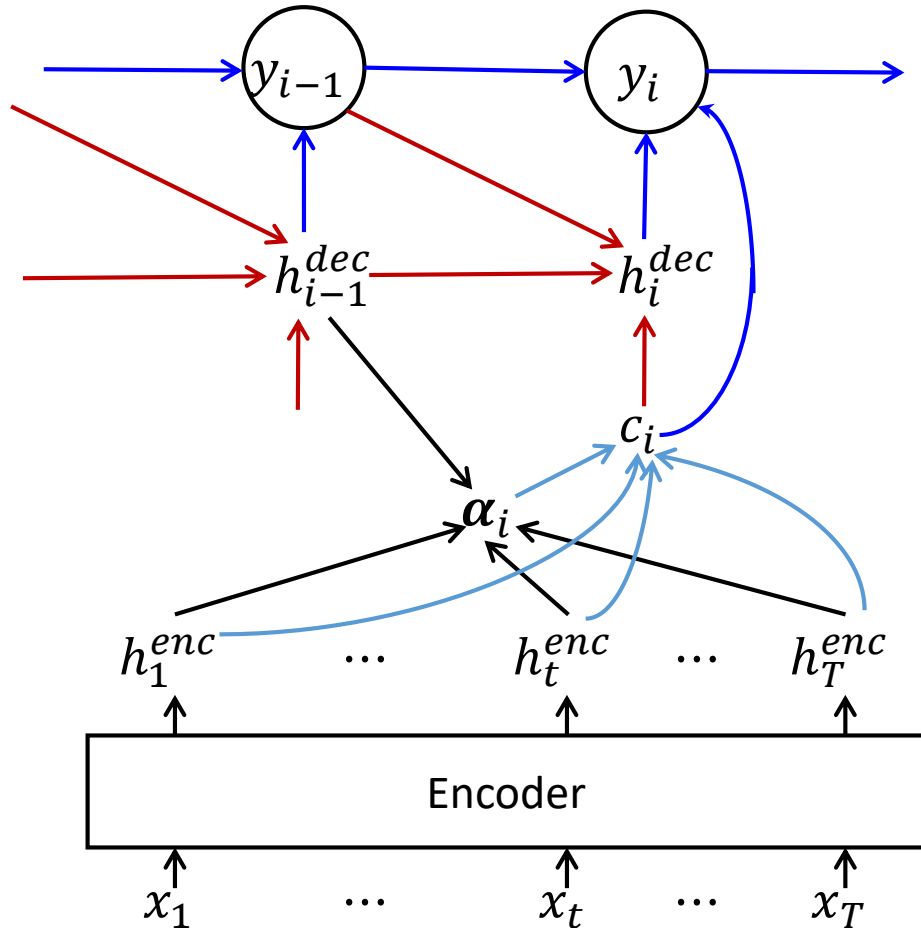
15-minute break

1. Data-efficiency
2. Neural architecture search
3. Multilingual and crosslingual ASR
4. Language modeling

III. Open questions and future directions

AED: basics

$$P(y_{1:L}|x_{1:T}) = \prod_{i=1}^L P(y_i|x_{1:T}, y_{1:i-1})$$



$$y_i = \text{Generate}(y_{i-1}, h_i^{dec}, c_i), i = 1, \dots, L$$

$$h_i^{dec} = \text{Decoder}(h_{i-1}^{dec}, y_{i-1}, c_i)$$

$$c_i = \sum_{t=1}^T \alpha_{i,t} h_t^{enc}, \text{ or simply as, } c_i = \text{Attend}(h_{i-1}^{dec}, h_{1:T}^{enc})$$

$$\alpha_{i,t} = \text{AttentionWeight}(h_{i-1}^{dec}, h_t^{enc}), t = 1:T$$

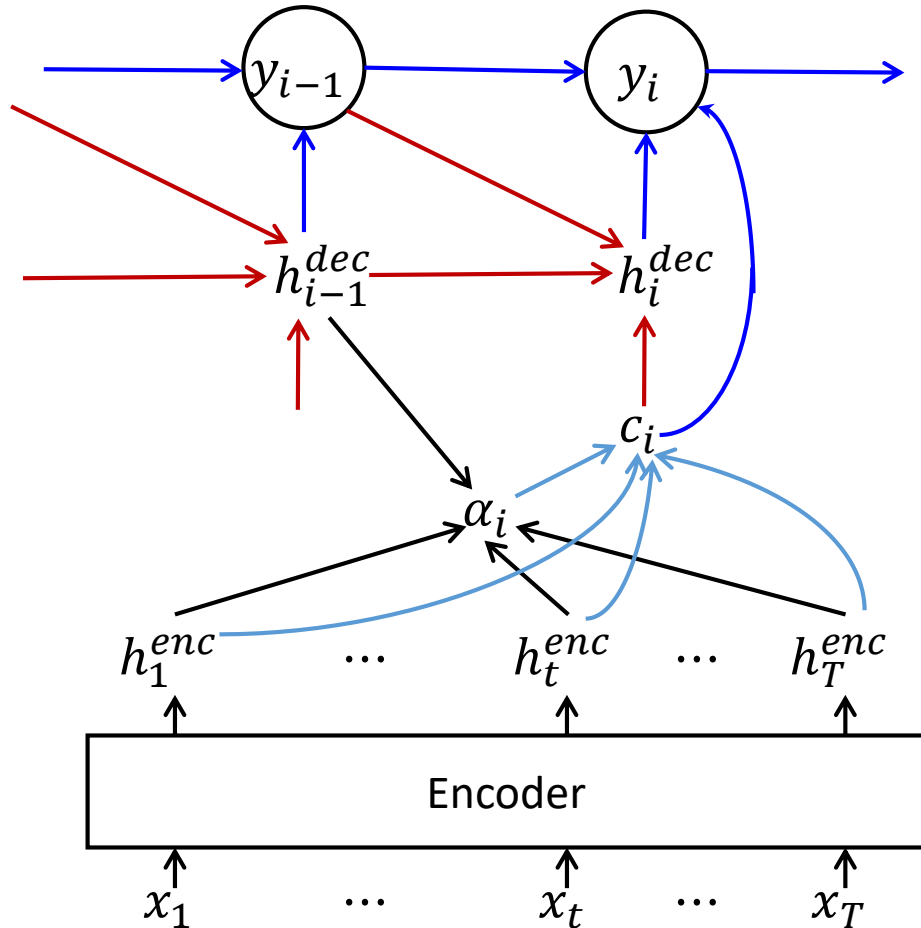
$$h_{1:T}^{enc} = \text{Encode}(x_{1:T})$$

Emerged first in the context of NMT, then applied to ASR

- D. Bahdanau, et al, "Neural machine translation by jointly learning to align and translate", ICLR 2015.
- W. Chan, et al., "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition", ICASSP 2016.
- J. Chorowski, et al., "Attention-based models for speech recognition", NIPS 2015.

AED: intuition

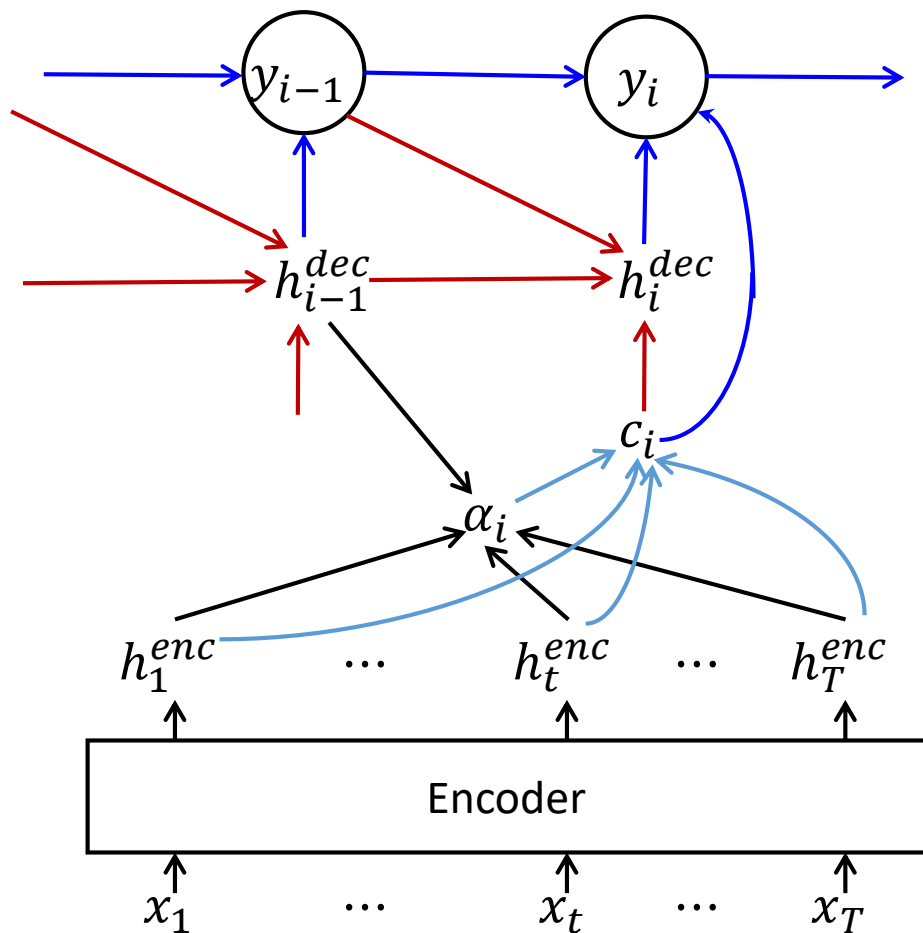
$$P(y_{1:L}|x_{1:T}) = \prod_{i=1}^L P(y_i|x_{1:T}, y_{1:i-1})$$



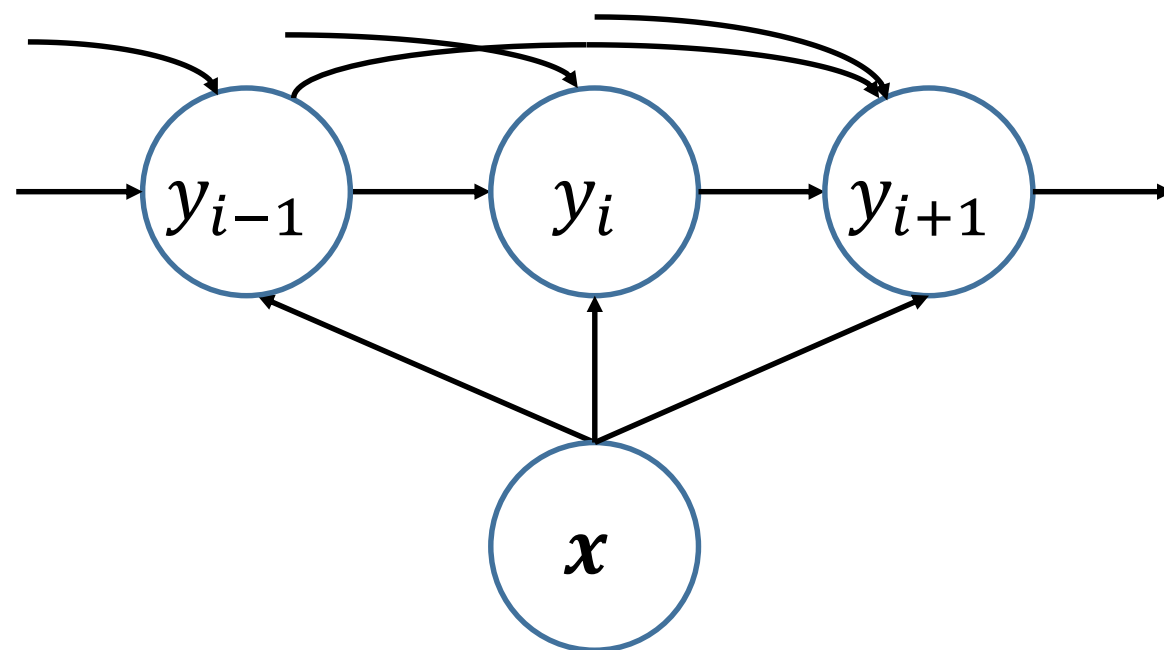
- ▶ **Encoder (analogous to AM):**
 - Transforms input speech into higher-level representation
- ▶ **Attention (alignment model):**
 - Identifies encoded frames that are relevant to producing current output
- ▶ **Decoder (analogous to LM):**
 - Operates autoregressively by predicting each output token, as a function of the previous predictions

AED: shortcoming

$$P(y_{1:L}|x_{1:T}) = \prod_{i=1}^L P(y_i|x_{1:T}, y_{1:i-1})$$



Computational flow



Graphical Model Representation

- As directed sequential model /Auto-regressive model, AED potentially suffers from Label Bias and Exposure Bias
- AED is not streaming; there are efforts...

Content

I. Basics for end-to-end speech recognition

1. Probabilistic graphical modeling (PGM) framework
2. Classic hybrid DNN-HMM models
3. Connectionist Temporal Classification (CTC)
4. Attention based encoder-decoder (AED)

→ 5. RNN transducer (RNNT)

6. Conditional random fields and sequence discriminative training
-

II. Improving end-to-end speech recognition

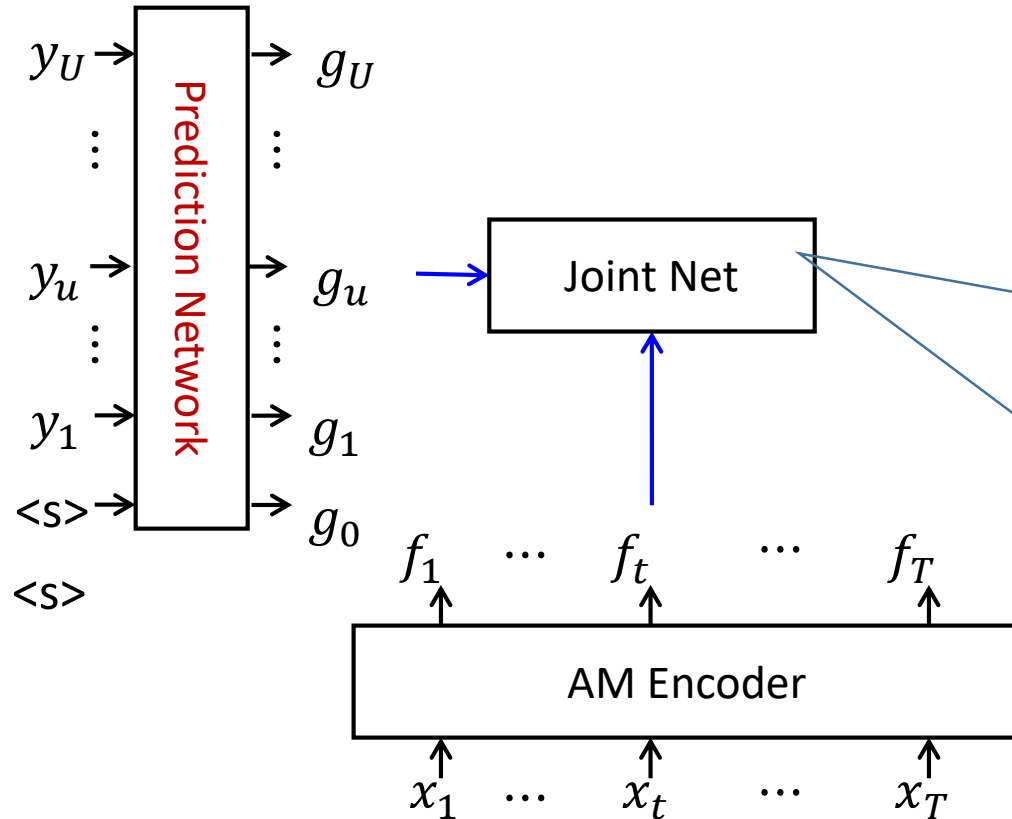
15-minute break

1. Data-efficiency
2. Neural architecture search
3. Multilingual and crosslingual ASR
4. Language modeling

III. Open questions and future directions

RNNT: introducing prediction network for labels I.5 RNNT

- Motivation: extending CTC by considering output-output dependencies
- Introduce the **prediction network**, which attempts to model each output in $y_{1:U}$ given the previous ones



y_0 is the special token $\langle s \rangle$

Defines the conditional output distribution at (t, u) :
 $JointNet(\cdot | t, u)$
which is a softmax over $K + 1$ units, including a blank ϕ , and used to determine the state transition probabilities in a lattice.

In the original paper, $f_t, g_u \in \mathbb{R}^{K+1}$ and $f_t + g_u$ directly defines the logits for the softmax layer.

RNNT: introducing a new definition of path

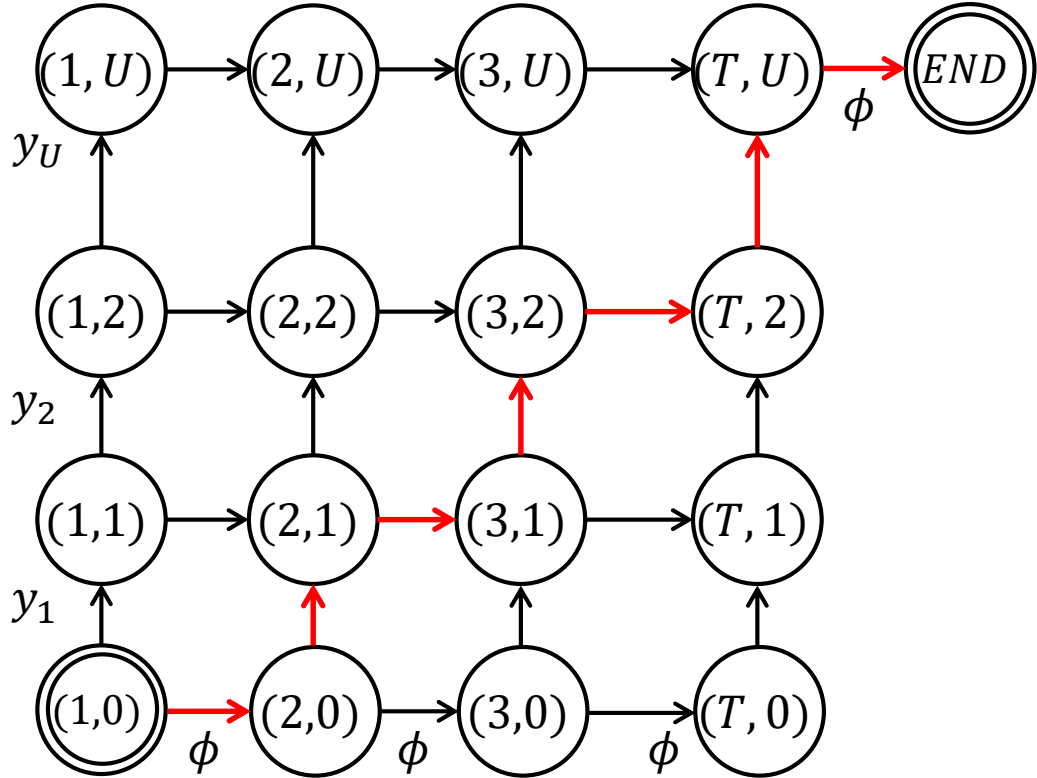
- Introduce the state sequence (a path) $\pi \triangleq \pi_1, \dots, \pi_{T+U}$ for input $x_{1:T}$ and output $y_{1:U}$ over a lattice
- Each state π_j is a tuple (t_j, u_j, o_j) , namely an arc starting from (t_j, u_j) and associated with an output label o_j , either being ϕ or from $y_{1:U}$
- A path π consists of T horizontal and U vertical arcs.

Path posterior

$$P(\pi_{1:T+U} | x_{1:T}) = \prod_{j=1}^{T+U} P(\pi_j | \pi_{1:j-1})$$

$$P(\pi_j | \pi_{1:j-1}) = \text{JointNet}(o_j | t_j, u_j) \triangleq p_{(t_j, u_j)}^{o_j}$$

$$o_j = \begin{cases} y_{u_j+1} & \pi_j \text{ is vertical} \\ \phi & \pi_j \text{ is horizontal} \end{cases}$$



RNNT: label-seq posterior

Path posterior

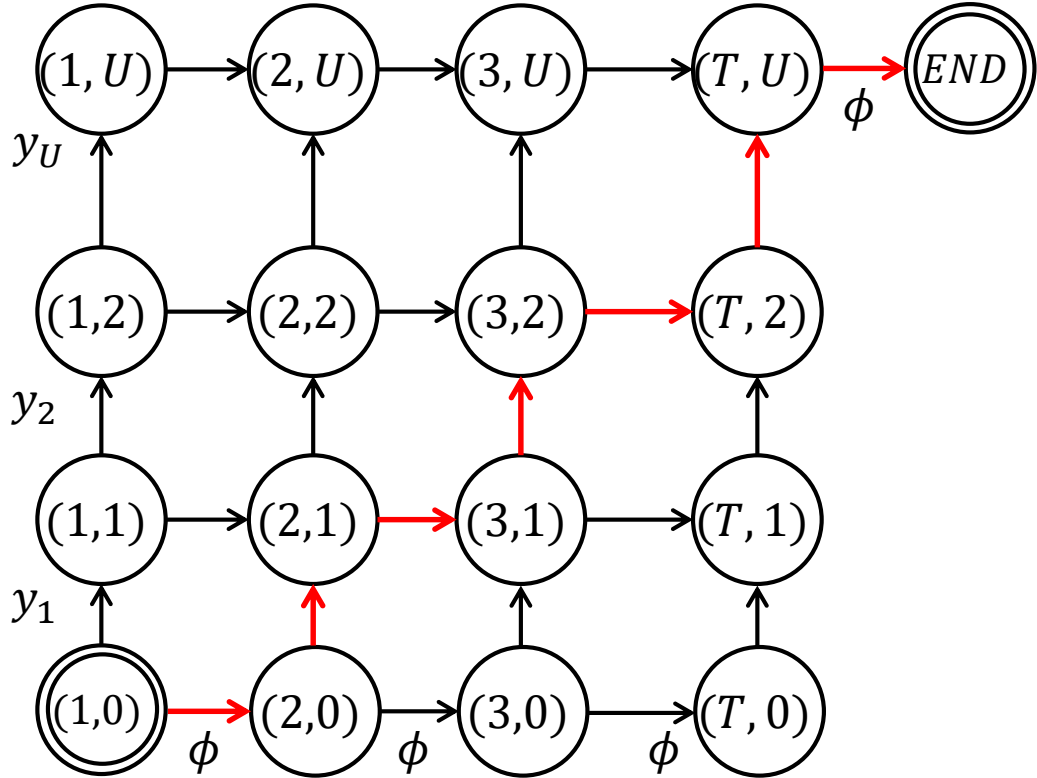
$$P(\pi_{1:T+U} | x_{1:T}) = \prod_{j=1}^{T+U} P(\pi_j | \pi_{1:j-1})$$

Label-seq posterior

$$P(y_{1:U} | x_{1:T}) = \sum_{\pi_{1:T+U} : \mathcal{B}_{RNNT}(\pi_{1:T+U}) = y_{1:U}} P(\pi_{1:T+U} | x_{1:T})$$

Summing over all possible paths, which map to $y_{1:U}$

RNNT topology : a mapping \mathcal{B}_{RNNT} maps π to y by removing outputs from all horizontal transitions in π



RNNT: the gradient & the forward-backward algorithm

For logits $z_{(t,u)} \in \mathbb{R}^{K+1}$ from $JointNet(\cdot | t, u)$, $1 \leq t \leq T, 0 \leq u \leq U$

$$\frac{\partial \log p(\mathbf{y} | \mathbf{x})}{\partial z_{(t,u)}^k} = E_{p(\boldsymbol{\pi} | \mathbf{x}, \mathbf{y})} \left[\frac{\partial \log p(\boldsymbol{\pi} | \mathbf{x})}{\partial z_{(t,u)}^k} \right] \quad \because \text{Fisher Equality [Ou, arxiv 2018]}$$

$$= E_{p(\boldsymbol{\pi} | \mathbf{x}, \mathbf{y})} \left[\frac{\partial \log p^{o_j}_{(t_j, u_j)}}{\partial z_{(t,u)}^k} \right] \quad \because p(\boldsymbol{\pi} | \mathbf{x}) = \prod_{j=1}^{T+U} p^{o_j}_{(t_j, u_j)}$$

$$= \sum_{j=1}^{T+U} E_{p(\boldsymbol{\pi} | \mathbf{x}, \mathbf{y})} [\delta(t_j = t, u_j = u, o_j = k)] - p_{(t,u)}^k$$

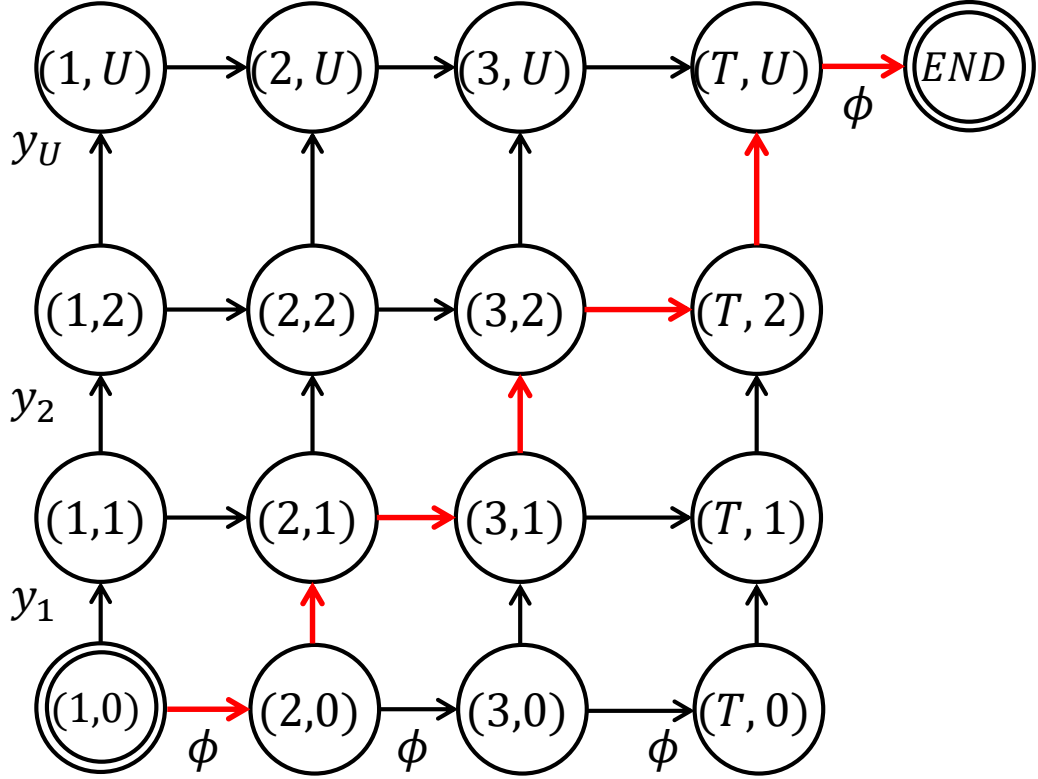
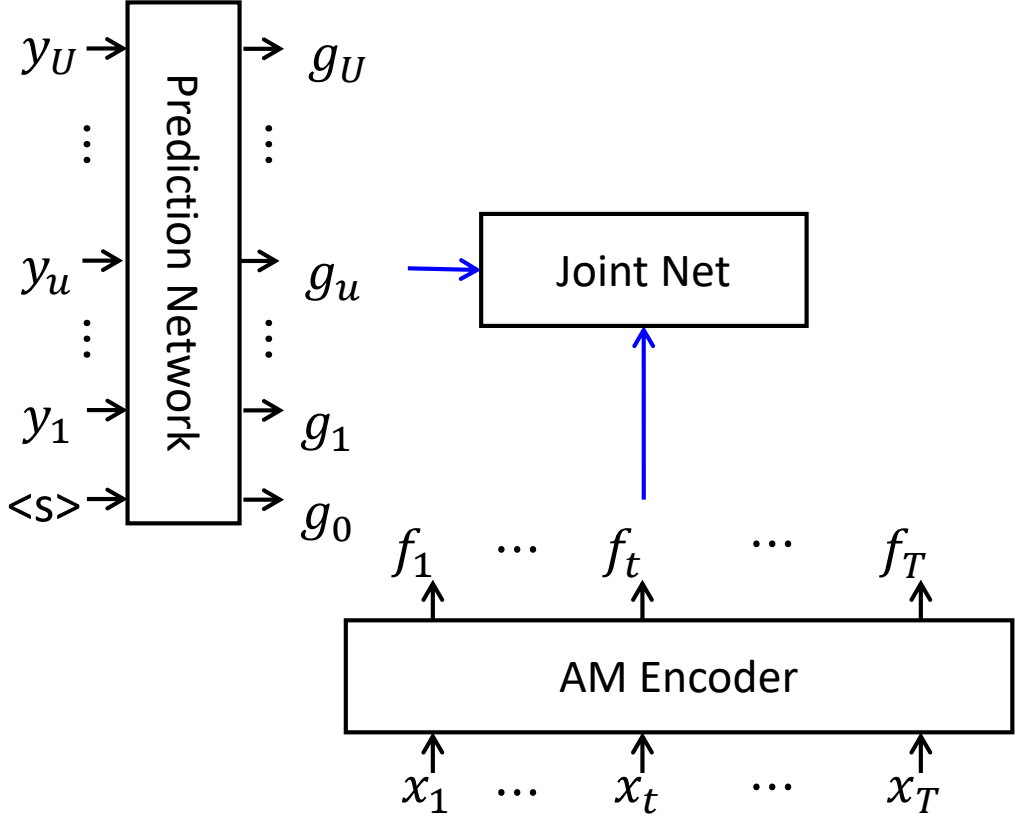
$$= \sum_{j=1}^{T+U} p(t_j = t, u_j = u, o_j = k | \mathbf{x}, \mathbf{y}) - p_{(t,u)}^k$$

i.e., the **error signal** received by the acoustic encoder NN during training

i.e., $\gamma_{(t,u)}^k$, the posterior **state occupation probability**, calculated using the alpha-beta variables from the forward-backward algorithm [Rabiner, 1989]

Providing easy derivation and giving insight, not appeared in [Graves, et al., 2006] and elsewhere

RNNT: intuition



► Encoder (analogous to AM):
 Transforms input speech into higher-level representation

► Prediction Net (analogous to LM?):
 Operates over output tokens

► Joint Net (Alignment model?):
 Determines when to emit output tokens

RNNT: discussion

- AM encoder initialized from CTC-trained acoustic model: generally improves performance.
- PN initialized from recurrent LM: mixed results. Reported to be helpful in [Rao et al., 2017], but not ↓

[E. Variani, et al., 2020]

“PN decoder network deviates from being a language model.”

Feeding limited context in PN performs as good as infinite context.

Context	0	1	2	4	∞
1st-pass WER	8.5	7.4	6.6	6.6	6.6
posterior cost	34.6	5.6	5.2	4.7	4.6

Table 2: Effect of limited context history.

[M. Ghodsi, et al., 2020]

Our results suggest that the RNNT prediction network does not function as the LM in classical ASR. Instead, it merely helps the model align to the input audio, while the RNNT encoder and joint networks capture both the acoustic and the linguistic information.

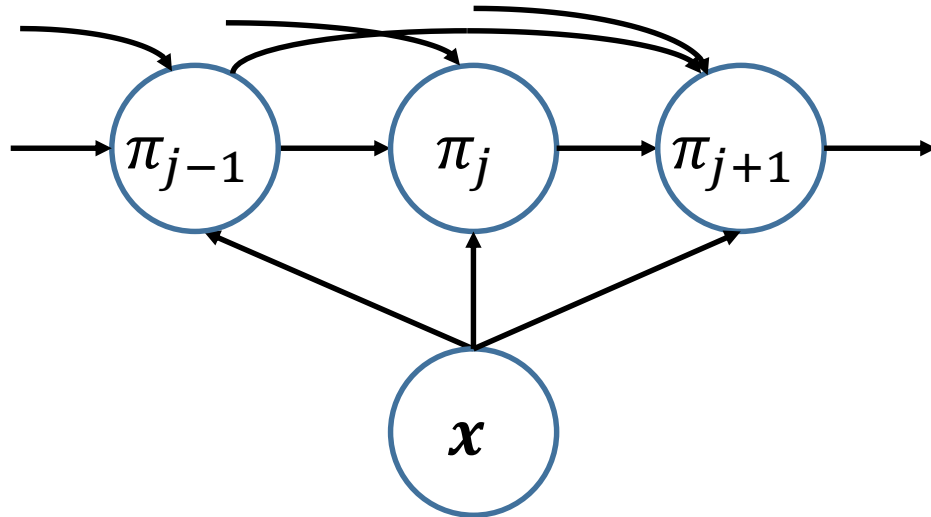
- K. Rao, H. Sak, R. Prabhavalkar, "Exploring Architectures, Data and Units For Streaming End-to-End Speech Recognition with RNN-Transducer", ASRU, 2017.
- E. Variani, et al., "Hybrid Autoregressive Transducer (HAT)", ICASSP, 2020.
- M. Ghodsi, et al., "RNN-transducer with stateless prediction network", ICASSP 2020.

RNNT: shortcoming

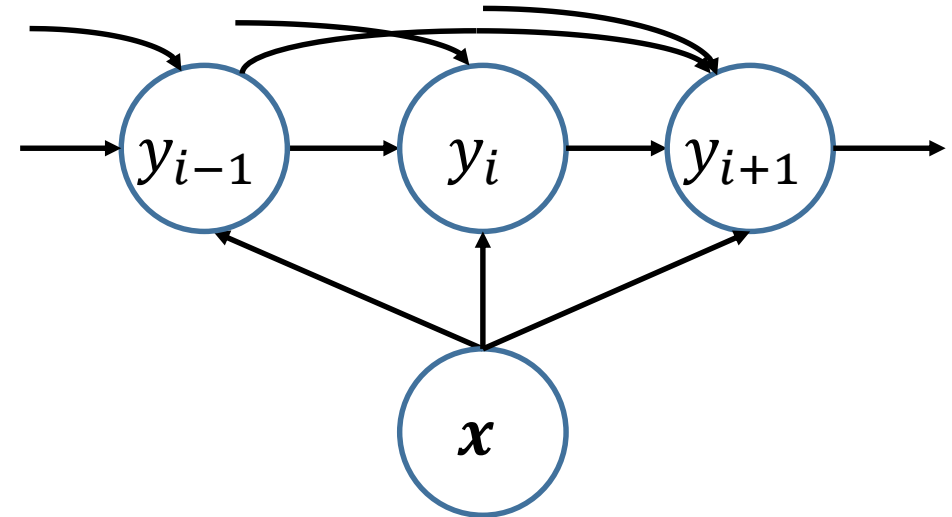
$$P(\pi_{1:T+U} | x_{1:T}) = \prod_{j=1}^{T+U} P(\pi_j | \pi_{1:j-1})$$

Marginalize

$$P(y_{1:U} | x_{1:T}) = \prod_{i=1}^U P(y_i | x_{1:T}, y_{1:i-1})$$



Graphical Model Representation



Graphical Model Representation

- RNNT is more suitable for streaming recognition, but as directed sequential model /Auto-regressive model, RNNT potentially suffers from Exposure Bias and Label Bias. A recent effort in [Cui, et al., 2021].

Content

I. Basics for end-to-end speech recognition

1. Probabilistic graphical modeling (PGM) framework
2. Classic hybrid DNN-HMM models
3. Connectionist Temporal Classification (CTC)
4. Attention based encoder-decoder (AED)
5. RNN transducer (RNNT)

→ 6. Conditional random fields and sequence discriminative training

II. Improving end-to-end speech recognition

15-minute break

1. Data-efficiency
2. Neural architecture search
3. Multilingual and crosslingual ASR
4. Language modeling

III. Open questions and future directions

Sequence discriminative training

- **Historically**

- GMM-HMMs are generative models

- DNN-HMMs are interpreted as generative models (interpreting $p(x_t|\pi_t) = \frac{p(\pi_t|x_t)p(x_t)}{p(\pi_t)}$ as pseudo-likelihood), though strictly not

- **A large body of works to improve GMM-HMMs and DNN-HMMs, by using sequence-discriminative criteria, like**

- Maximum Mutual Information (MMI), boosted MMI (BMMI), Minimum Phone Error (MPE), Minimum Bayes Risk (MBR) [Karel, et al., 2013]
- Minimum Word Error Rate (MWER) [Stolcke, et al., 1997]

- V. Karel, et al., "Sequence-discriminative training of deep neural networks", INTERSPEECH 2013.
- A. Stolcke, et al., "Explicit word error minimization in N-best list rescoring", Eurospeech, 1997.

MMI and CML

- MMI training of a GMM-HMM, for acoustic input \mathbf{x} and transcript \mathbf{y} , is equivalent to CML (conditional maximum likelihood) training of a CRF (using 0/1/2-order features in potential definition) [Heigold, et al., 2011].

$$J_{MMI} = \log \frac{p(\mathbf{x} | \mathbf{y})}{p(\mathbf{x})} = \log \frac{p(\mathbf{y} | \mathbf{x})}{p(\mathbf{y})}$$

$$J_{CML} = \log p(\mathbf{y} | \mathbf{x})$$

- **LF-MMI**: no division by the prior, uniform transition probabilities, using log-softmax prob. of states as the log of a pseudo-likelihood [Povey, et al., 2016]
- For the two manners - indirectly formulated as MMI training of a pseudo HMM [Povey, et al., 2016] or directly formulated as CML training of a CRF, it would be **conceptually simpler** to adopt the later manner.

- G. Heigold, et al., "Equivalence of generative and log-linear models", TASLP, 2011.
- D. Povey, et al., "Purely sequence-trained neural networks for ASR based on lattice-free MMI", INTERSPEECH 2016.

Conditional random field (CRF)

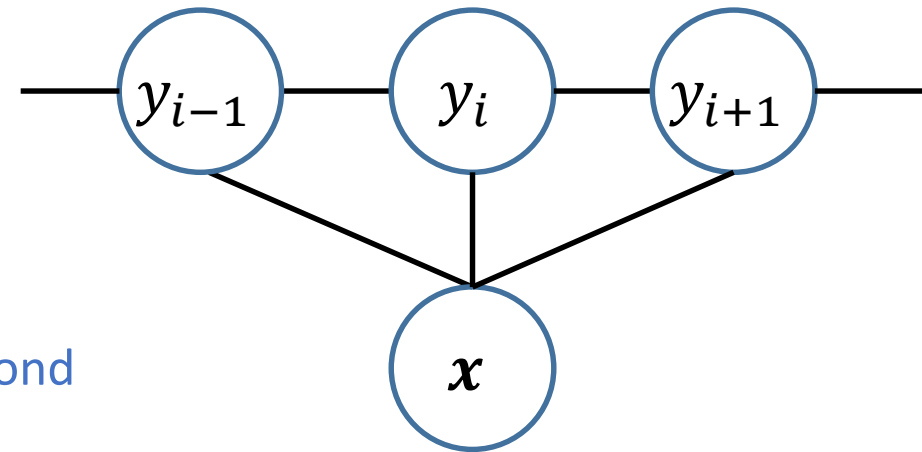
A CRF define a **conditional** distribution over output sequence y^l given input sequence x^l of length l :

$$p_{\theta}(y^l|x^l) = \frac{1}{Z_{\theta}(x^l)} \exp(u_{\theta}(x^l, y^l)) \quad Z_{\theta}(x^l) = \sum_{y^l} \exp(u_{\theta}(x^l, y^l))$$

Potential function:

$$u_{\theta}(x^l, y^l) = \sum_{i=1}^l \phi_i(y_i, x^l) + \sum_{i=1}^l \psi_i(y_{i-1}, y_i, x^l)$$

Node potential Edge potential
↙ ↙



Example of a linear-chain CRF

- ▶ CRFs was explored for phone classification, using zero, first and second order features [Gunawardana, et al., 2005].
- ▶ CRFs can overcome “label bias” and “exposure bias”, but are hard to be trained.
- ▶ CTC-CRF: the **first** CRF successfully developed for end-to-end ASR

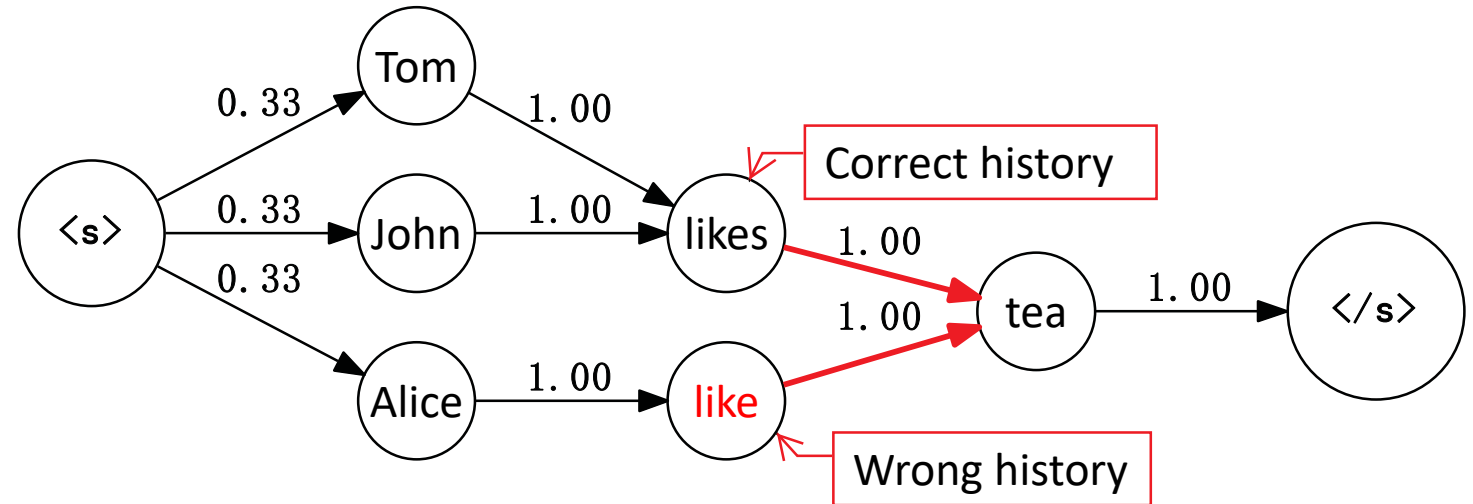
- Lafferty, et al., "Conditional random fields: Probabilistic models for segmenting and labeling sequence data", ICML 2001.
- A. Gunawardana, et al., "Hidden conditional random fields for phone classification", Eurospeech, 2005.

Label bias [Lafferty, et al., 2001]

► Word probabilities at each time-step are locally normalized, so successors of incorrect histories receive the same mass as do the successors of the true history. [Wiseman, et al., 2016]

Training data

Tom likes tea
John likes tea
Alice like tea



► [Andor, et al., 2016]

- “Intuitively, we would like the model to **be able to revise an earlier decision** made during search, when later evidence becomes available that rules out the earlier decision as incorrect.”
- “the label bias problem means that locally normalized models often have a very weak ability to **revise earlier decisions**.”
- A **proof** that globally normalized models are strictly more expressive than locally normalized models.
- Wiseman, et al., "Sequence-to-sequence Learning as Beam-Search Optimization", EMNLP, 2016.
- Andor, et al., "Globally Normalized Transition-Based Neural Networks", ACL, 2016.

Exposure bias

- ▶ **Mismatch** between **training** (teacher forcing) and **testing** (prediction) of locally-normalized sequence models:
 - **Training**: maximize the likelihood of each successive target word, conditioned on the gold history of the target word.
 - **Testing**: the model predict the next step, using its own predicted samples in testing.
- ▶ The model is **never exposed to its own errors during training**, and so the inferred histories at test-time do not resemble the gold training histories. [Wiseman, et al., 2016]
- ▶ **Exposure bias** results from training in a certain way, **Label bias** results from properties of the model itself.

I. Basics for end-to-end speech recognition

1. Probabilistic graphical modeling (PGM) framework
 2. Classic hybrid DNN-HMM models
 3. Connectionist Temporal Classification (CTC)
 4. Attention based encoder-decoder (AED)
 5. RNN transducer (RNNT)
 6. Conditional random fields and sequence discriminative training
-

II. Improving end-to-end speech recognition

15-minute break

- 1. **Data-efficiency**
- 2. Neural architecture search
- 3. Multilingual and crosslingual ASR
- 4. Language modeling

III. Open questions and future directions

Section Content

1. Motivation

2. Related work

3. Method: **CTC-CRF**

4. Experiments

5. Conclusion

- H. Xiang, Z. Ou. "CRF-based Single-stage Acoustic Modeling with CTC Topology", **ICASSP**, 2019.
- K. An, H. Xiang, Z. Ou. "CAT: A CTC-CRF based ASR Toolkit Bridging the Hybrid and the End-to-end Approaches towards Data Efficiency and Low Latency", **INTERSPEECH**, 2020.
- Fan, et al., "The SLT 2021 children speech recognition challenge: Open datasets, rules and baselines", **SLT**, 2021.
- H. Zheng, W. Peng, Z. Ou, J. Zhang. "Advancing CTC-CRF Based End-to-End Speech Recognition with Wordpieces and Conformers", arXiv:2107.03007, 2021.

Data-efficient $Efficiency = \frac{Performance}{Labeling\ Cost}$

- Current ASR: heavy reliance on supervised learning and large amounts of manually-labeled data
- Different from: computation-efficient (MIPS, million instructions per second), power-efficient (MIPS/Watt)
 - — Efficiency of learning by machines
- A spectrum of data-efficient modeling and learning methods
 - ✓ Model architecture
 - ✓ unsupervised, semi-supervised, self-supervised learning
 - ✓ Pre-training
 - ✓ Transfer learning
 - ✓ Active learning
 - ✓ Meta-learning

Motivation: data-efficient end2end

- End-to-end system:

- Eliminate the construction of GMM-HMMs and phonetic decision-trees, and can be trained from scratch (flat-start or single-stage)

- In a more strict/ambitious sense:

- Remove the need for a pronunciation lexicon and, even further, train the acoustic and language models jointly rather than separately
- Data-hungry

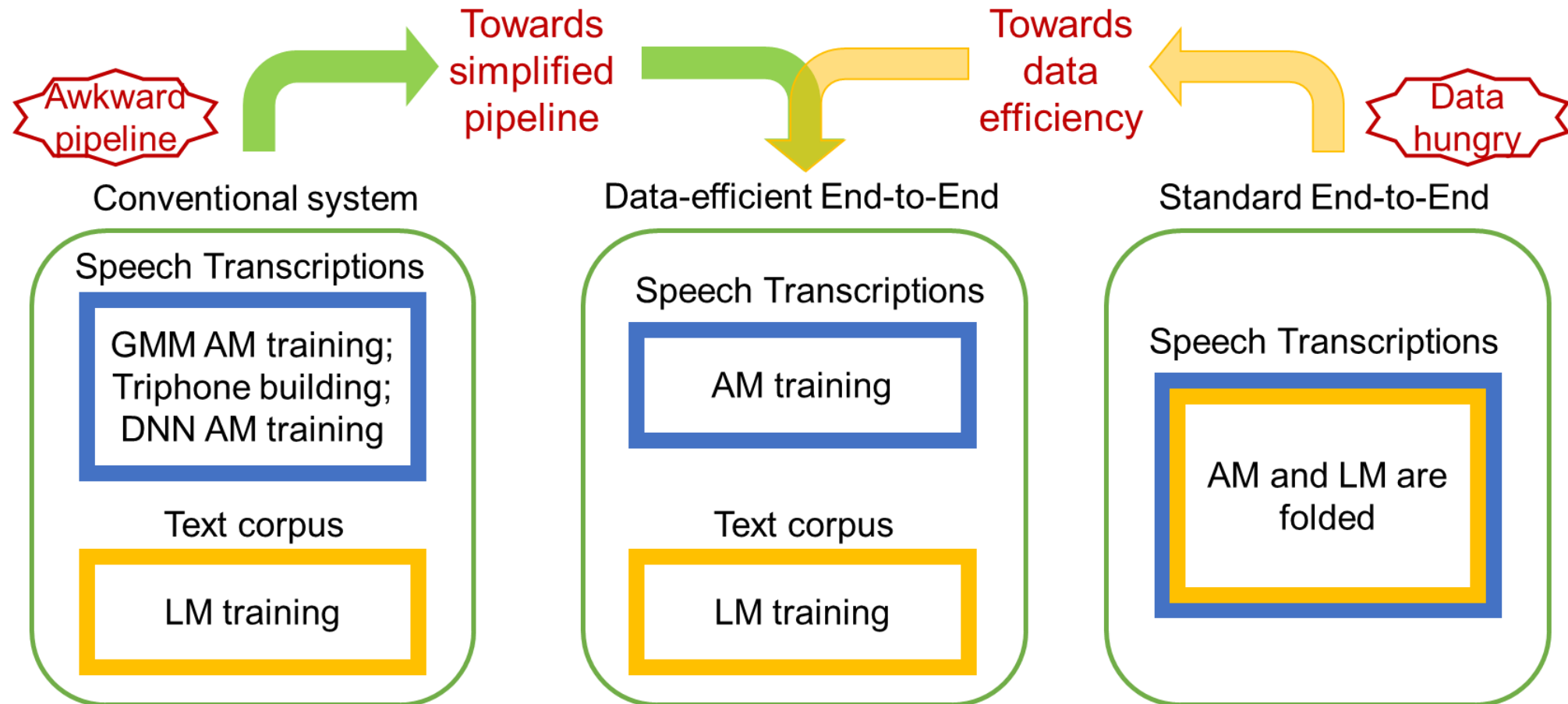
We need data-efficient end2end speech recognition, which uses a separate language model (LM) with or without a pronunciation lexicon.

- Text corpus for language modeling are cheaply available.
- Data-efficient

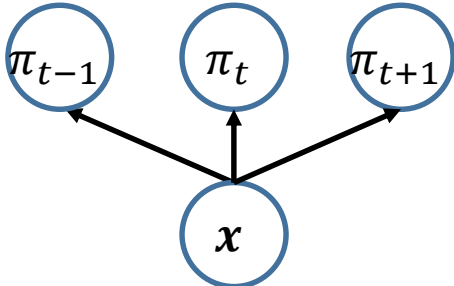
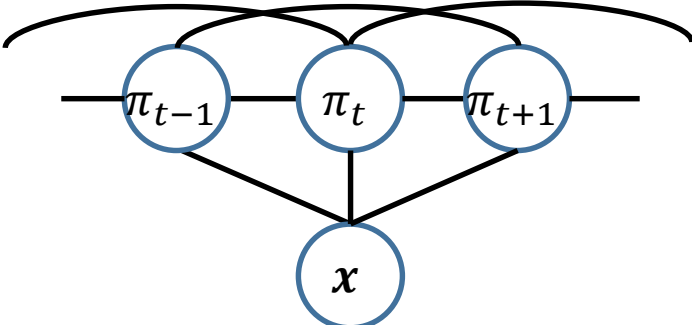
Motivation: bridging

Modularization promote Data-efficiency

- ✓ Keep necessary factorization of AM and LM



CTC vs CTC-CRF

CTC	CTC-CRF
$p(\mathbf{y} \mathbf{x}) = \sum_{\pi: \mathcal{B}(\pi)=\mathbf{y}} p(\pi \mathbf{x}), \text{ using CTC topology } \mathcal{B}$	
<p>State Independence</p> $p(\pi \mathbf{x}; \theta) = \prod_{t=1}^T p(\pi_t \mathbf{x})$	$p(\pi \mathbf{x}; \theta) = \frac{e^{\phi(\pi, \mathbf{x}; \theta)}}{\sum_{\pi'} e^{\phi(\pi', \mathbf{x}; \theta)}}$ <p style="text-align: right; color: red;">Node potential, by NN</p> $\phi(\pi, \mathbf{x}; \theta) = \sum_{t=1}^T \left(\log p(\pi_t \mathbf{x}) + \log p_{LM}(\mathcal{B}(\pi)) \right)$ <p style="text-align: right; color: red;">Edge potential, by n-gram denominator LM of labels, like in LF-MMI</p>
$\frac{\partial \log p(\mathbf{y} \mathbf{x}; \theta)}{\partial \theta} = \mathbb{E}_{p(\pi \mathbf{y}, \mathbf{x}; \theta)} \left[\frac{\partial \log p(\pi \mathbf{x}; \theta)}{\partial \theta} \right]$	$\frac{\partial \log p(\mathbf{y} \mathbf{x}; \theta)}{\partial \theta} = \mathbb{E}_{p(\pi \mathbf{x}, \mathbf{y}; \theta)} \left[\frac{\partial \phi(\pi, \mathbf{x}; \theta)}{\partial \theta} \right] - \mathbb{E}_{p(\pi' \mathbf{x}; \theta)} \left[\frac{\partial \phi(\pi', \mathbf{x}; \theta)}{\partial \theta} \right]$
	

Related work

■ Directed Graphical Model/Locally normalized

➤ DNN-HMM : Model $p(\boldsymbol{\pi}, \boldsymbol{x})$ as an HMM, could be discriminatively trained, e.g. by $\max_{\boldsymbol{\theta}} p_{\boldsymbol{\theta}}(\boldsymbol{y} | \boldsymbol{x})$

➤ CTC : $p(\boldsymbol{\pi} | \boldsymbol{x}) = \prod_{t=1}^T p(\pi_t | \boldsymbol{x})$

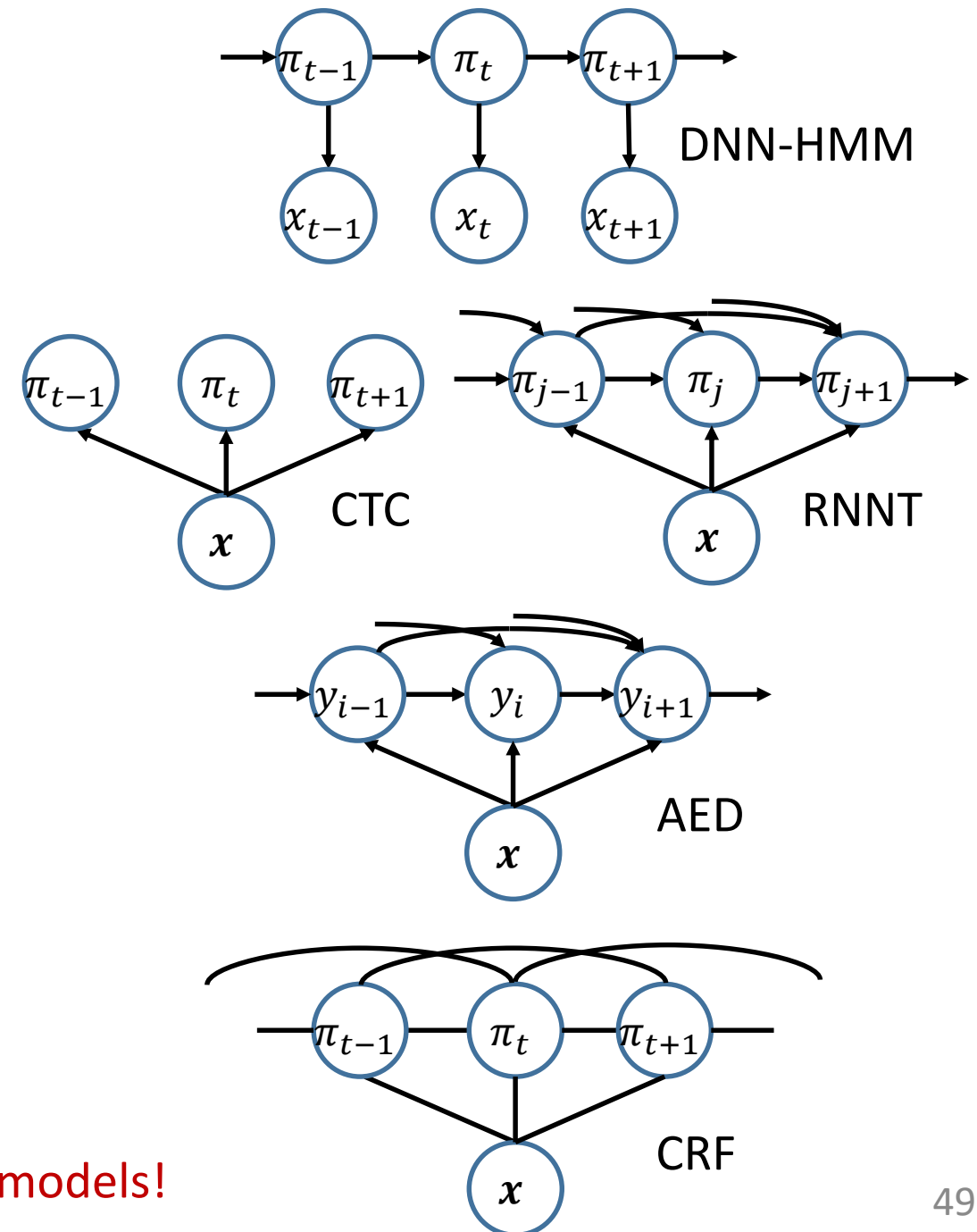
➤ RNNT : $p(\pi_{1:T+U} | \boldsymbol{x}_{1:T}) = \prod_{j=1}^{T+U} p(\pi_j | \pi_{1:j-1})$

➤ Seq2Seq : $p(\boldsymbol{y} | \boldsymbol{x}) = \prod_{i=1}^L p(y_i | y_1, \dots, y_{i-1}, \boldsymbol{x})$

■ Undirected Graphical Model/Globally normalized

➤ CRF : $p(\boldsymbol{\pi} | \boldsymbol{x}) \propto \exp[\phi(\boldsymbol{\pi}, \boldsymbol{x})]$

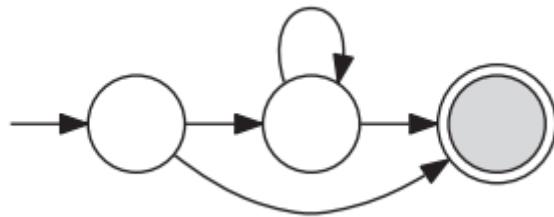
CTC-CRF is fundamentally different from all history models!



Related work (SS-LF-MMI/EE-LF-MMI)

- Single-Stage (SS) Lattice-Free Maximum-Mutual-Information (LF-MMI)

- 10 - 25% relative WER reduction on 80-h WSJ, 300-h Switchboard and 2000-h Fisher+Switchboard datasets, compared to **CTC**, **Seq2Seq**, **RNN-T**.
- Cast as MMI-based discriminative training of an HMM (generative model) with *Pseudo state-likelihoods calculated by the bottom DNN*, *Fixed state-transition probabilities*.
- 2-state HMM topology
- Including a silence label



CTC-CRF

- Cast as a CRF;
- CTC topology;
- No silence label.

SS-LF-MMI vs CTC-CRF

	SS-LF-MMI	CTC-CRF
State topology	HMM topology with two states	CTC topology
Silence label	Using silence labels. Silence labels are randomly inserted when estimating denominator LM.	No silence labels. Use <blk> to absorb silence. 😊 No need to insert silence labels to transcripts.
Decoding	No spikes.	The posterior is dominated by <blk> and non-blank symbols occur in spikes. 😊 Speedup decoding by skipping blanks.
Implementation	Modify the utterance length to one of 30 lengths; use leaky HMM.	😊 No length modification; no leaky HMM.

Experiments

- We conduct our experiments on three benchmark datasets:
 - WSJ 80 hours
 - Switchboard 300 hours
 - Librispeech 1000 hours
- **Acoustic model:** 6 layer BLSTM with **320** hidden dim, 13M parameters
- **Adam optimizer** with an initial learning rate of 0.001, decreased to 0.0001 when cv loss does not decrease
- **Implemented with Pytorch.**
- **Objective function** (use the CTC objective function to help convergences):

$$\mathcal{J}_{CTC-CRF} + \alpha \mathcal{J}_{CTC}$$

- **Decoding score function** (use word-based language models, WFST based decoding):

$$\log p(\mathbf{l}|\mathbf{x}) + \beta \log p_{LM}(\mathbf{l})$$

Experiments (Comparison with CTC, phone based)

WSJ 80h

Model	Unit	LM	SP	dev93	eval92
CTC	Mono-phone	4-gram	N	10.81%	7.02%
CTC-CRF	Mono-phone	4-gram	N	6.24%	3.90%

44.4% reduction in eval92 error rate for CTC-CRF compared to CTC.

Switchboard 300h

Model	Unit	LM	SP	SW	CH
CTC	Mono-phone	4-gram	N	12.9%	23.6%
CTC-CRF	Mono-phone	4-gram	N	11.0%	21.0%

14.7% reduction in SW error rate and 11% reduction in CH error rate for CTC-CRF compared to CTC.

Librispeech 1000h

Model	Unit	LM	SP	Dev Clean	Dev Other	Test Clean	Test Other
CTC	Mono-phone	4-gram	N	4.64%	13.23%	5.06%	13.68%
CTC-CRF	Mono-phone	4-gram	N	3.87%	10.28%	4.09%	10.65%

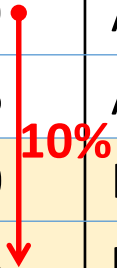
19.1% reduction in Test Clean error rate and 22.1% reduction in Test Other error rate for CTC-CRF compared to CTC.

SP: speed perturbation for 3-fold data augmentation.

Experiments (Comparison with STOA)

Switchboard 300h

Model	SW	CH	Average	Source
Kaldi chain triphone	9.6	19.3	14.5	IS 2016
Kaldi e2e chain monophone	11.0	20.7	15.9	ASLP 2018, 26M
Kaldi e2e chain biphone	9.8	19.3	14.6	ASLP 2018, 26M
CTC-CRF monophone	10.3	19.7	15.0	ICASSP 2019, BLSTM, 13M
CTC-CRF monophone	9.8	18.8	14.3	IS 2020, VGG BLSTM, 16M



RWTH IS 2018, “Improved training of end-to-end attention models for speech recognition”.

RWTH IS 2019, “RWTH ASR Systems for LibriSpeech Hybrid vs Attention -- Data Augmentation”.

IBM IS19, “Forget a Bit to Learn Better Soft Forgetting for CTC-based Automatic Speech Recognition”.

Espnet ASRU19, “Espresso: A Fast End-to-end Neural Speech Recognition Toolkit”.

Google IS19, “SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition”.

Experiments (Comparison with STOA)

Librispeech 1000h

Model	Test Clean	Test Other	Source
Kaldi chain triphone	4.28	-	IS 2016
CTC-CRF monophone	4.0	10.6	ICASSP 2019, BLSTM (6,320), 13M

RWTH IS 2018, “Improved training of end-to-end attention models for speech recognition”.

RWTH IS 2019, “RWTH ASR Systems for LibriSpeech Hybrid vs Attention -- Data Augmentation”.

IBM IS19, “Forget a Bit to Learn Better Soft Forgetting for CTC-based Automatic Speech Recognition”.

Espnet ASRU19, “Espresso: A Fast End-to-end Neural Speech Recognition Toolkit”.

Google IS19, “SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition”.

Mandarin Aishell-1 results

- 170 hours mandarin speech corpus
- 400 speakers from different accent areas
- 15% CER reduction compared with LF-MMI
- 5% CER reduction compared with end-to-end transformer

Model	%CER
LF-MMI with i-vector [1]	7.43
Transformer [2]	6.7
CTC-CRF [3]	6.34

[1] D. Povey, A. Ghoshal, and et al, “The Kaldi speech recognition toolkit,” ASRU 2011.

[2] S. Karita, N. Chen, and et al, “A comparative study on transformer vs RNN in speech applications,” ASRU 2019.

[3] Keyu An, Hongyu Xiang, and **Zhijian Ou**, “CAT: A CTC-CRF based ASR toolkit bridging the hybrid and the end-to-end approaches towards data efficiency and low latency,” INTERSPEECH 2020.

2021 SLT CHILDREN SPEECH RECOGNITION CHALLENGE (CSRC)

ORGANIZER :  西北工业大学  清华大学  厦門大學  标贝科技 

- 400 hours of data, targeting to boost children speech recognition research.
- Evaluated on 10 hours of children's reading and conversational speech.
- 3 baselines (Chain model, Transformer and CTC-CRF) are provided.

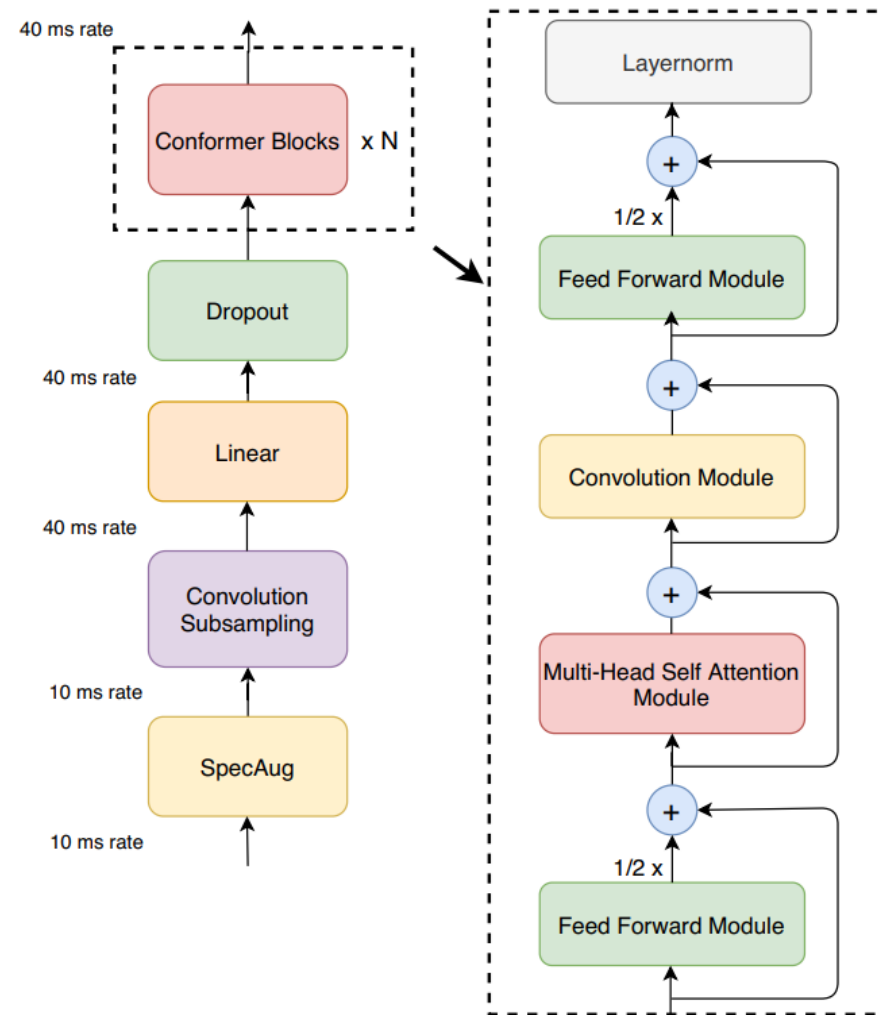
model	Chain model	Transformer	CTC-CRF
CER%	28.75	27.28	25.34

Fan Yu, Zhuoyuan Yao, Xiong Wang, Keyu An, Lei Xie, **Zhijian Ou**, Bo Liu, Xiulin Li, Guanqiong Miao. The SLT 2021 children speech recognition challenge: Open datasets, rules and baselines. SLT 2021.

Advancing CTC-CRF Based End-to-End Speech Recognition with Wordpieces and Conformers

Huahuan Zheng, Wenjie Peng, **Zhijian Ou** and Jinsong Zhang, arXiv:2107.03007

Basic Units of Labels	Label Sequence
phoneme	DH AE1 T N IY1 DH ER0 AH1 V DH EH1 M HH AE1 D K R AO1 S T DH AH0 TH R EH1 SH OW2 L D S IH1 N S DH AH0 D AA1 R K D EY1
character /grapheme	that_neither_of_them_had_crossed_the_threshold_since_the_dark_day_
subword /wordpiece	that_neither_of_them_had_crossed_the_ threshold_since_the_dark_day_
word	that neither of them had crossed the threshold since the dark day



Experiments (Comparison between different units, WER%)

Switchboard 300h

Model	Unit	LM	Augmentation	Eval2000	SW	CH
Conformer (this work)	monophone	4-gram	SP, SA	12.1	7.9	16.1
	monophone	Trans.*	SP, SA	10.7	6.9	14.5
	wordpiece	4-gram	SP, SA	12.7	8.7	16.5
	wordpiece	Trans.*	SP, SA	11.1	7.2	14.8

Librispeech 1000h

Model	Unit	LM	Augmentation	Test Clean	Test Other
Conformer (this work)	monophone	4-gram	SA	3.61	8.10
	monophone	Trans.**	SA	2.51	5.95
	wordpiece	4-gram	SA	3.59	8.37
	wordpiece	Trans.**	SA	2.54	6.33

SP: speed perturbation for 3-fold data augmentation.

SA: our implementation of SpecAug with ratio

* Latest **Kaldi Transformer LM rescoring**

** RWTH 42-layer Transformer

English: a low degree of grapheme-phoneme correspondence

Experiments (Comparison between different units, WER%)

CommonVoice German 700h

Model	#params	unit	LM	Augmentation	Test
Conformer (This work)	25.03	char	4-gram	SP, SA	12.7
	25.03	char	Trans.	SP, SA	11.6
	25.03	monophone	4-gram	SP, SA	10.7
	25.03	monophone	Trans.	SP, SA	10.0
	25.06	wordpiece	4-gram	SP, SA	10.5
	25.06	wordpiece	Trans.	SP, SA	9.8

German: a high degree of grapheme-phoneme correspondence

Experiments (Comparison with STOA)

Switchboard 300h

Model	#params	LM	unit	SW	CH	Eval2000
RNN-T, 2021 [10]	57	RNN LM	char	6.4	13.4	9.9
Conformer [9]	44.6	Trans.	bpe	6.8	14.0	10.4
TDNN-F [11]	-	Trans.*	triphone	7.2	14.4	10.8
TDNN-F [11]	-	Trans.**	triphone	6.5	13.9	10.2
VGGBLSTM [2]	39.15	RNN LM	monophone	8.8	17.4	[13.0]
Conformer (This work)	51.82	Trans.	monophone	6.9	14.5	10.7
	51.85	Trans.	wordpiece	7.2	14.8	11.1

* N-best rescoring, ** Iterative lattice rescoring

[2] “CAT: A CTC-CRF based ASR toolkit bridging the hybrid and the end-to-end approaches towards data efficiency and low latency,” INTERSPEECH 2020.

[9] “Conformer: Convolution-augmented Transformer for Speech Recognition”, Interspeech 2020.

[10] “Advancing RNN transducer technology for speech recognition,” ICASSP 2021.

[11] “A parallelizable lattice rescoring strategy with neural language models,” ICASSP, 2021

Section Conclusion

- The CTC-CRF framework inherits the **data-efficiency** of the hybrid approach and the **simplicity** of the end-to-end approach.
- CTC-CRF significantly **outperforms** regular CTC on a wide range of benchmarks, and is **on par with** other state-of-the-art end-to-end models.
 - English WSJ-80h, Switchboard-300h, Librispeech-1000h; Mandarin Aishell-170h; ...
- **Flexibility**
 - Streaming ASR <- INTRESPEECH 2020
 - Neural Architecture Search <- SLT 2021
 - Children Speech Recognition <- SLT 2021
 - Wordpieces, Conformer architectures
 - Multilingual and Crosslingual <- ASRU2021
 - ...



<https://github.com/thu-spmi/cat>

Content

I. Basics for end-to-end speech recognition

1. Probabilistic graphical modeling (PGM) framework
 2. Classic hybrid DNN-HMM models
 3. Connectionist Temporal Classification (CTC)
 4. Attention based encoder-decoder (AED)
 5. RNN transducer (RNNT)
 6. Conditional random fields and sequence discriminative training
-

II. Improving end-to-end speech recognition

15-minute break

1. Data-efficiency
- 2. **Neural architecture search**
3. Multilingual and crosslingual ASR
4. Language modeling

III. Open questions and future directions

Section Content

1. Motivation

2. Related work

3. Method: **ST-NAS**

4. Experiments

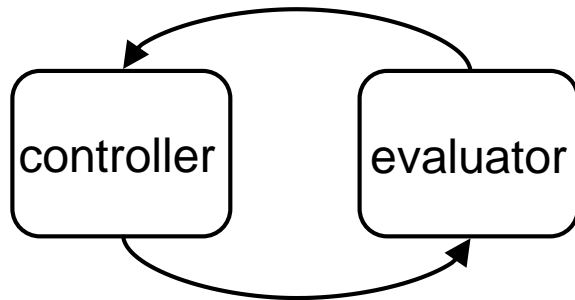
5. Conclusion

- H. Zhen, K. An, Z. Ou. “Efficient Neural Architecture Search for End-to-end Speech Recognition via Straight-Through Gradients”, SLT 2021.

Motivation

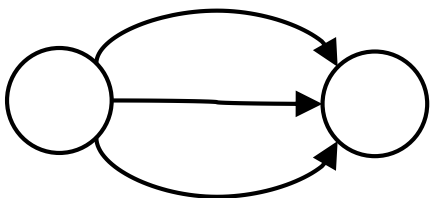
- End-to-end ASR reduces expert efforts by automating *feature engineering*, but raises a demand for *architecture engineering*.
- Neural architecture search (NAS), the process of automating architecture engineering, is an appealing next step to advancing end-to-end ASR.

1. Early NAS methods



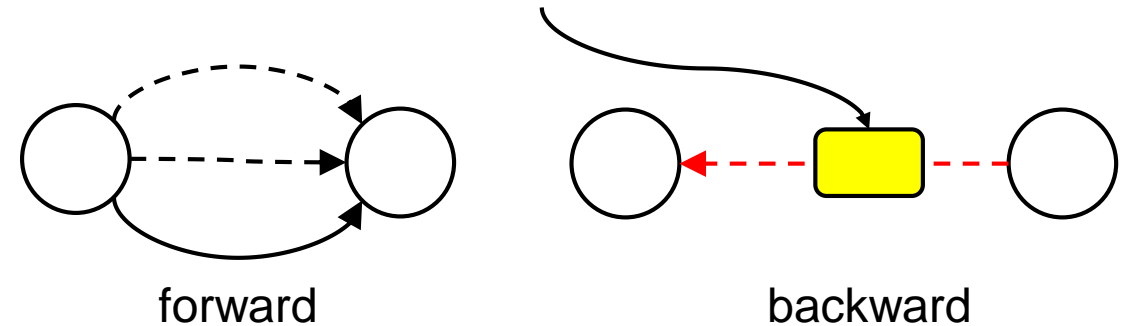
computation expensive,
1000+ GPU days 😞

2. Recent gradient-based NAS methods (DARTS, SNAS, ProxylessNAS)



Improved,
but still memory expensive (DARTS, SNAS),
or using ad-hoc trick (ProxylessNAS) 😞

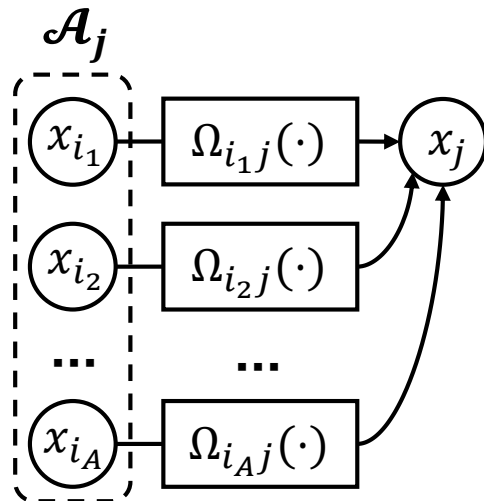
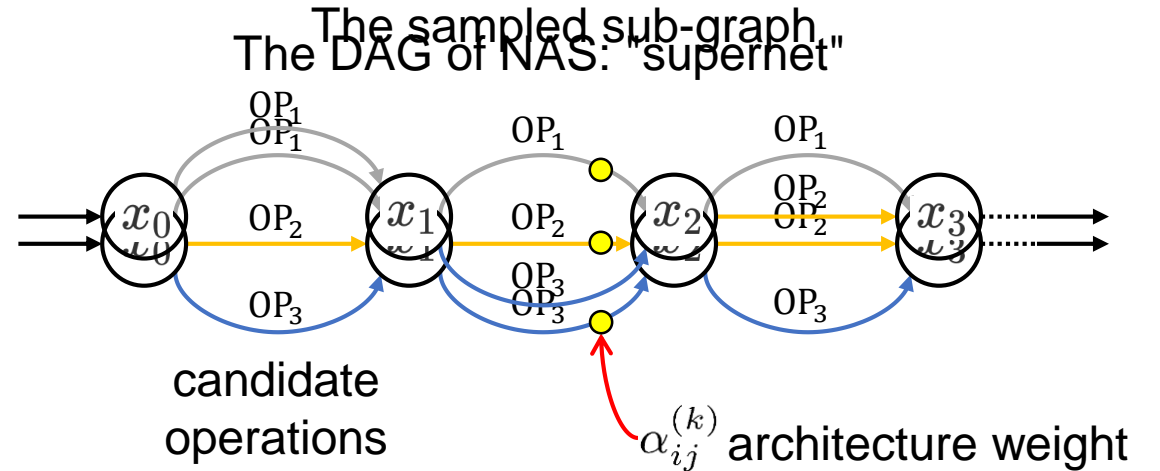
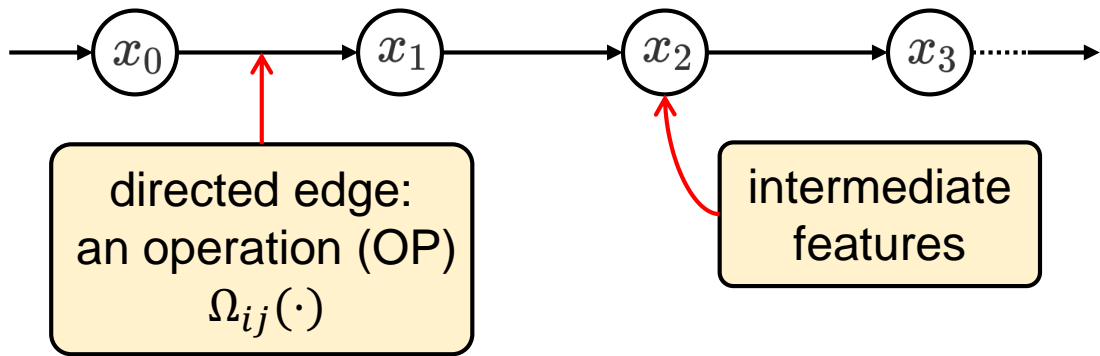
3. (ours) Straight-Through gradient NAS (ST-NAS)



Back-Prop ST gradients through the sampled edge,
Efficient in both memory and computation,
Less than 3-fold computation time 😊

Gradient-based NAS: representing the search space as a weighted directed acyclic graph (DAG)

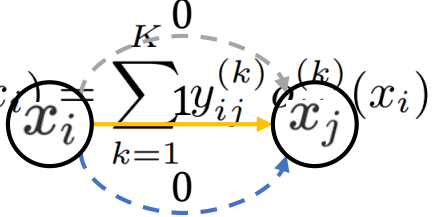
NN architecture as graph



Forward computation in general

$$\Rightarrow x_j = \sum_{i \in \mathcal{A}_j} \Omega_{ij}(x_i)$$

Related work: DARTS, SNAS and ProxylessNAS

	DARTS	SNAS	Ideally	ProxylessNAS
Definition	$\Omega_{ij}^{\text{DARTS}}(x_i) = \sum_{k=1}^K \pi_{ij}^{(k)} o_{ij}^{(k)}(x_i)$ $\pi_{ij}^{(k)} = \frac{\exp(\alpha_{ij}^{(k)})}{\sum_{k'=1}^K \exp(\alpha_{ij}^{(k')})}$	$\tilde{\mathcal{L}}(\alpha, \theta) = \mathbb{E}_{z \sim p_\alpha(z)} [\mathcal{L}_\theta(z)]$ $\Omega_{ij}^{\text{SNAS}}(x_i) = \sum_{k=1}^K y_{ij}^{(k)} o_{ij}^{(k)}(x_i)$  $y_{ij}^{(k)} = G_i z_{ij}^{(k)} = \begin{cases} 1, & \text{OP}_k \text{ is sampled} \\ 0, & \text{others} \end{cases}$	$\Omega_{ij}(x_i) = \sum_{k=1}^K z_{ij}^{(k)} o_{ij}^{(k)}(x_i)$	$z_{ij} \sim \text{Multi}(\pi_{ij})$
Limitation	<div style="border: 2px solid red; padding: 5px; display: inline-block;"> Continuous relaxation; Require $K \times$ memory and time </div>		$S_i z_{ij} \sim \text{Multi}(\pi_{ij})$ is one-hot vector	The loss is not explicitly shown in the original paper

- [1] H Liu, K Simonyan, and Y Yang, “**DARTS**: Differentiable architecture search,” in ICLR, 2019.
- [2] S Xie, H Zheng, et al, “**SNAS**: stochastic neural architecture search,” in ICLR, 2019.
- [3] H Cai, L Zhu, and S Han, “**ProxylessNAS**: Direct neural architecture search on target task and hardware,” in ICLR, 2019.
- [4] E Jang, S Gu, and B Poole, “Categorical reparameterization with Gumbel-Softmax,” in ICLR, 2017.
- [5] M Courbariaux, Y Bengio, and J David, “BinaryConnect: training deep neural networks with binary weights during propagations,” NIPS, 2015.

ST (Straight-Through) NAS

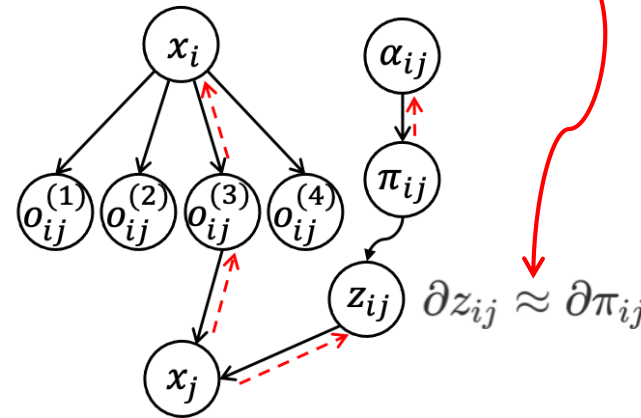
$$\mathcal{L}(\alpha, \theta) = \mathbb{E}_{z \sim p_{\alpha}(z)} [\mathcal{L}_{\theta}(z)]$$

$$\Omega_{ij}(x_i) = \sum_{k=1}^K z_{ij}^{(k)} o_{ij}^{(k)}(x_i)$$

$$z_{ij} \sim \text{Multi}(\pi_{ij})$$

$$\pi_{ij}^{(k)} = \frac{\exp(\alpha_{ij}^{(k)})}{\sum_{k'=1}^K \exp(\alpha_{ij}^{(k')})}$$

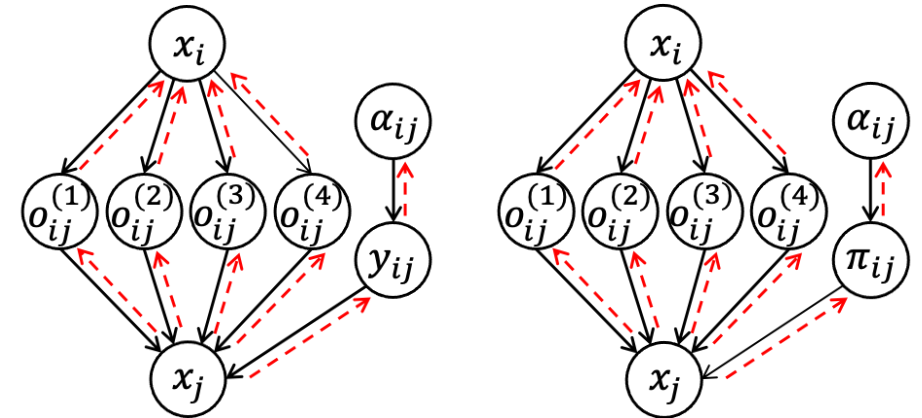
straight-through gradient



continuous relaxation

SNAS

DARTS



Using ST gradients to support sub-graph sampling is key to achieve efficient NAS beyond DARTS and SNAS.

Comparison of different gradient-based NAS methods.

Methods	Loss	α gradient	Memory	Backward computation
DARTS	$\mathcal{L}^{\text{DARTS}}(\alpha, \theta)$	continuous	KC_1	$O(K)$
SNAS	$\tilde{\mathcal{L}}(\alpha, \theta)$	continuous	KC_1	$O(K)$
Proxyless	$\mathcal{L}_{\theta}(z)$	BinaryConnect	$C_1 + (K - 1)C_2$	$O(1)$
ST-NAS	$\mathcal{L}(\alpha, \theta)$	ST	$C_1 + (K - 1)C_2$	$O(1)$

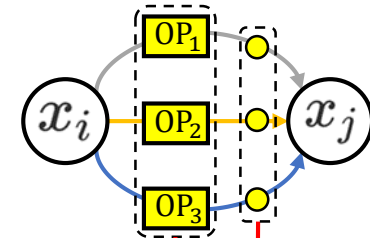
- Computation costs are estimated relative to training a single model.
- C_1 : the memory size for training a single model.
- C_2 : the average memory size for storing the output features for all connected pairs of nodes in a sub-graph. Usually we have $C_2 \ll C_1$.

ST-NAS: overview

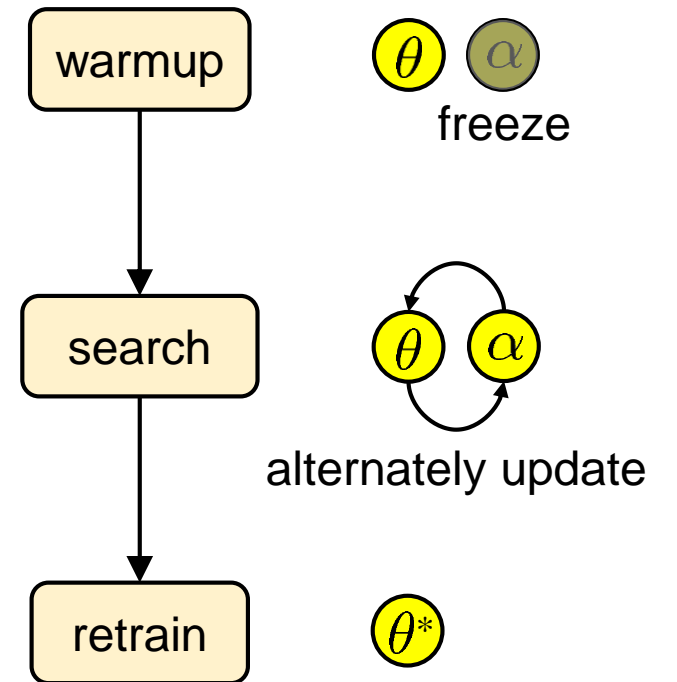
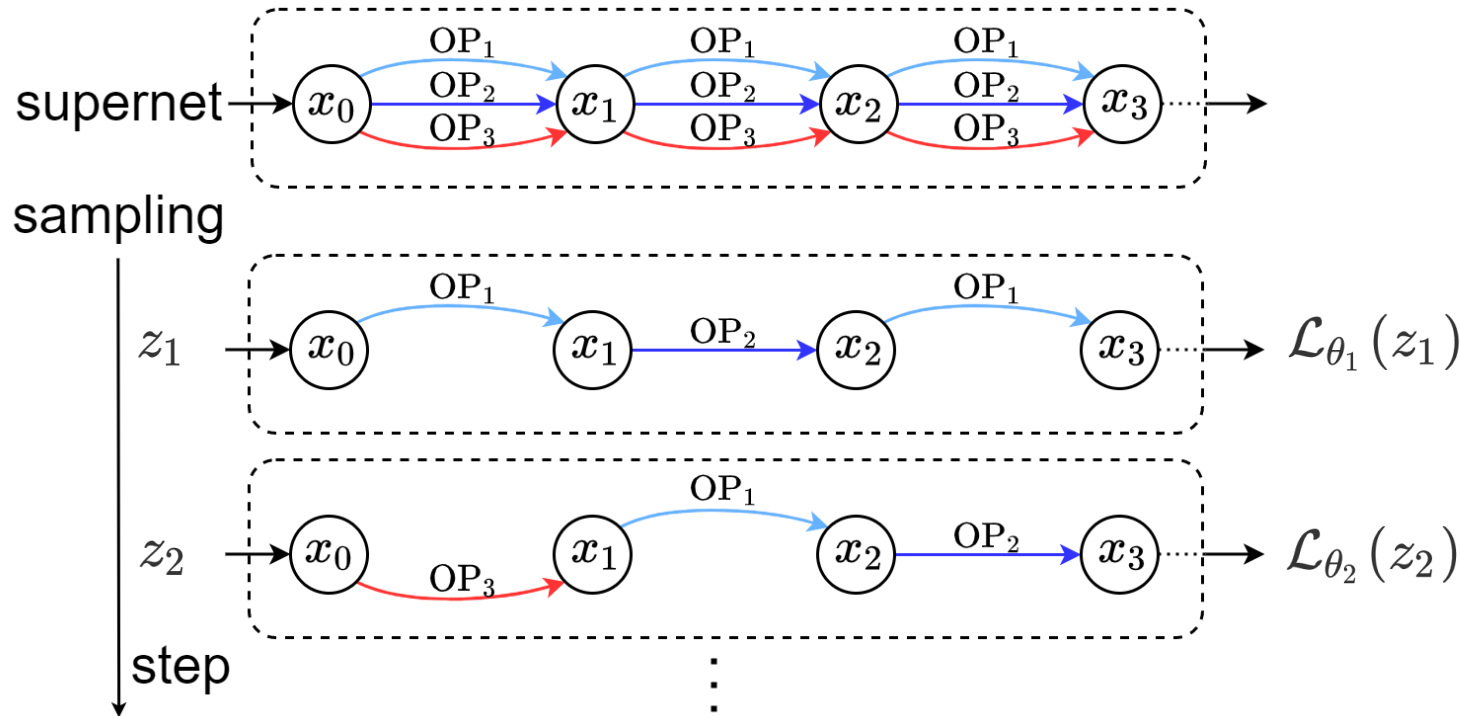
Objective: $\mathcal{L}(\alpha, \theta) = \mathbb{E}_{z \sim p_\alpha(z)} [\mathcal{L}_\theta(z)]$

Monte Carlo estimation

sampled sub-graph



NN parameters θ α architecture weights

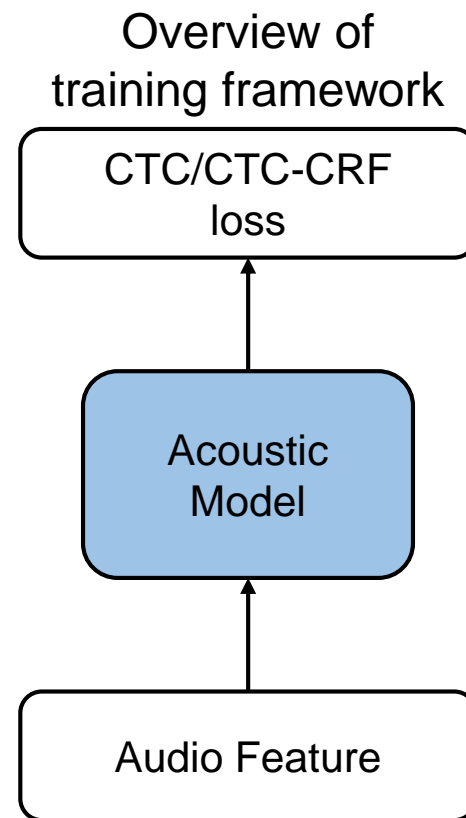
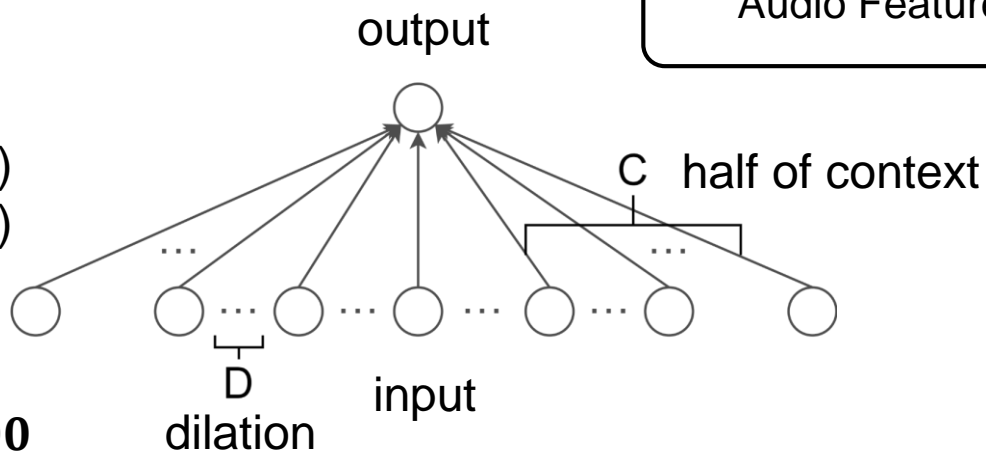


ST-NAS procedure

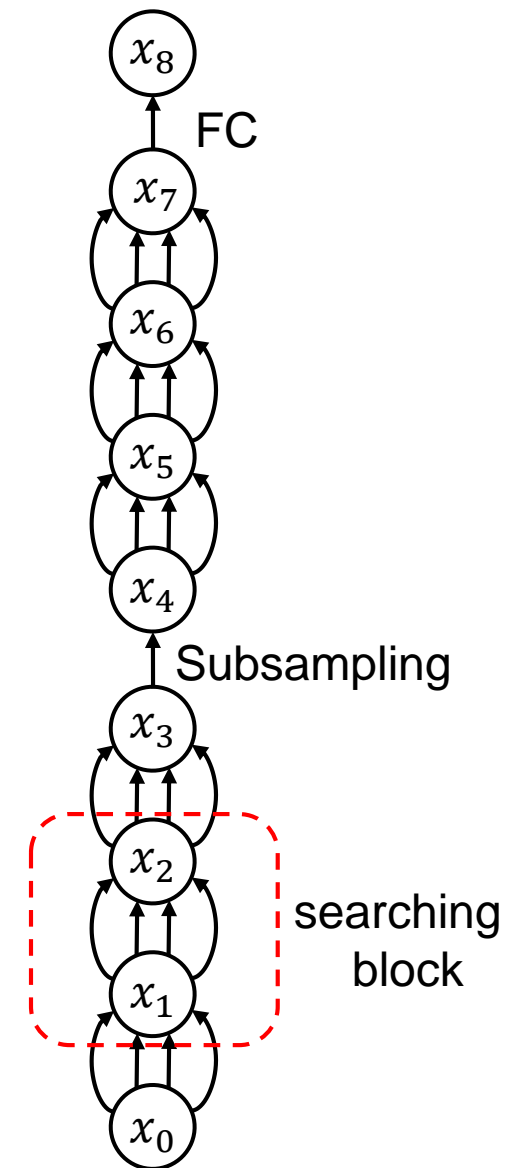
Experiments: ASR system

Settings:

1. Datasets: 80-hour WSJ and 300-hour Switchboard.
2. LM: n-gram model.
3. Loss: CTC/CTC-CRF based on [CAT](#) [1].
4. Candidate operations:
 - TDNN-1-1 (-{C}-{D})
 - TDNN-1-2
 - TDNN-2-1
 - TDNN-2-2
 - TDNN-3-1 (Switchboard)
 - TDNN-3-2 (Switchboard)
5. Search space:
 - WSJ: $4^6 = 4096$
 - Switchboard: $6^6 \approx 47000$

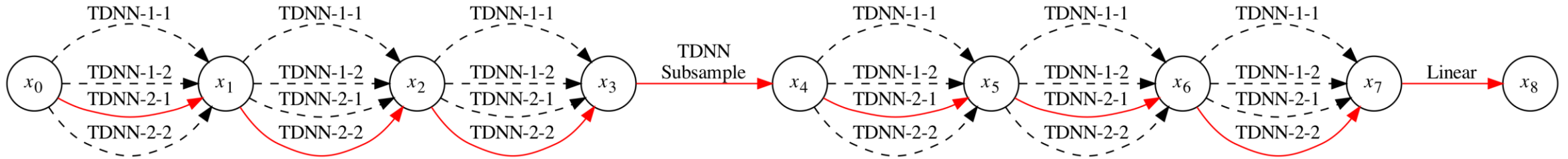


Backbone of supernet

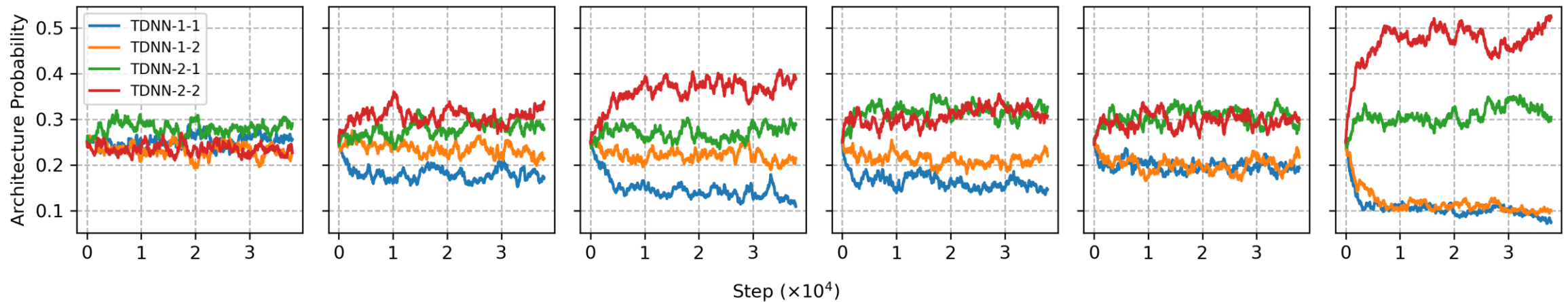


[1] K An, H Xiang, and Z Ou, "CAT: A CTC-CRF based ASR toolkit bridging the hybrid and the end-to-end approaches towards data efficiency and low latency," INTERSPEECH, 2020.

Experiments: WSJ



The red lines indicate one of the derived single model from the 5 runs of NAS on WSJ.



The evolution of architecture probabilities for the searching blocks in the NAS run that yields the derived single model above.

Experiments: results on WSJ

WERs on the WSJ.

Methods		eval92	dev93
EE-Policy-CTC [1]		5.53	9.21
SS-LF-MMI [2]		3.0	6.0
EE-LF-MMI [3]		3.0	-
FC-SR [4]		3.5	6.8
ESPRESSO [5]		3.4	5.9
CTC	BLSTM	4.93	8.57
	ST-NAS	4.72±0.03	8.82±0.07
CTC-CRF	BLSTM [6]	3.79	6.23
	VGG-BLSTM [7]	3.2	5.7
	TDNN-D* [8]	2.91	6.24
	Random search	2.82±0.01	5.71±0.03
	ST-NAS	2.77±0.00	5.68±0.01
ST-NAS with fully CTC-CRF		2.81±0.01	5.74±0.02

* Obtained based on our implementation of the “TDNN-D” in [8].

outperforming all other end-to-end ASR models

[1] Y Zhou, C Xiong, et al, “Improving end-to-end speech recognition with policy learning,” ICASSP, 2018.

[2] H Hadian, H Sameti, et al, “Flat-start single-stage discriminatively trained HMM-Based models for ASR,” TASLP, 2018.

[3] H Hadian, H Sameti, et al, “End-to-end speech recognition using lattice-free MMI,” INTERSPEECH, 2018.

[4] N Zeghidour, Q Xu, et al, “Fully convolutional speech recognition,” arXiv preprint arXiv:1812.06864, 2018.

[5] Y Wang, T Chen, et al, “Espresso: A fast endto-end neural speech recognition toolkit,” ASRU, 2019.

[6] H Xiang and Z Ou, “CRF-based single-stage acoustic modeling with CTC topology,” ICASSP, 2019.

[7] K An, H Xiang, et al, “CAT: A CTC-CRF based ASR toolkit bridging the hybrid and the end-to-end approaches towards data efficiency and low latency,” INTERSPEECH, 2020.

[8] V Peddinti, Y Wang, et al, “Low latency acoustic modeling using temporal convolution and LSTMs,” IEEE SPL, 2018.

Experiments: results on WSJ

WERs on the WSJ.

Methods		eval92	dev93
EE-Policy-CTC [1]		5.53	9.21
SS-LF-MMI [2]		3.0	6.0
EE-LF-MMI [3]		3.0	-
FC-SR [4]		3.5	6.8
ESPRESSO [5]		3.4	5.9
CTC	BLSTM	4.93	8.57
	ST-NAS	4.72±0.03	8.82±0.07
CTC-CRF	BLSTM [6]	3.79	6.23
	VGG-BLSTM [7]	3.2	5.7
	TDNN-D* [8]	2.91	6.24
	Random search	2.82±0.01	5.71±0.03
	ST-NAS	2.77±0.00	5.68±0.01
ST-NAS with fully CTC-CRF		2.81±0.01	5.74±0.02

* Obtained based on our implementation of the “TDNN-D” in [8].

Better performance with lighter model,
under the same CTC-CRF loss

[1] Y Zhou, C Xiong, et al, “Improving end-to-end speech recognition with policy learning,” ICASSP, 2018.

[2] H Hadian, H Sameti, et al, “Flat-start single-stage discriminatively trained HMM-Based models for ASR,” TASLP, 2018.

[3] H Hadian, H Sameti, et al, “End-to-end speech recognition using lattice-free MMI,” INTERSPEECH, 2018.

[4] N Zeghidour, Q Xu, et al, “Fully convolutional speech recognition,” arXiv preprint arXiv:1812.06864, 2018.

[5] Y Wang, T Chen, et al, “Espresso: A fast endto-end neural speech recognition toolkit,” ASRU, 2019.

[6] H Xiang and Z Ou, “CRF-based single-stage acoustic modeling with CTC topology,” ICASSP, 2019.

[7] K An, H Xiang, et al, “CAT: A CTC-CRF based ASR toolkit bridging the hybrid and the end-to-end approaches towards data efficiency and low latency,” INTERSPEECH, 2020.

[8] V Peddinti, Y Wang, et al, “Low latency acoustic modeling using temporal convolution and LSTMs,” IEEE SPL, 2018.

Experiments: results on WSJ

WERs on the WSJ.

Methods		eval92	dev93
EE-Policy-CTC [1]		5.53	9.21
SS-LF-MMI [2]		3.0	6.0
EE-LF-MMI [3]		3.0	-
FC-SR [4]		3.5	6.8
ESPRESSO [5]		3.4	5.9
CTC	BLSTM	4.93	8.57
	ST-NAS	4.72±0.03	8.82±0.07
BLSTM [6]		3.79	6.23
VGG-BLSTM [7]		3.2	5.7
CTC-CRF	TDNN-D* [8]	2.91	6.24
	Random search	2.82±0.01	5.71±0.03
	ST-NAS	2.77±0.00	5.68±0.01
ST-NAS with fully CTC-CRF		2.81±0.01	5.74±0.02

* Obtained based on our implementation of the “TDNN-D” in [8].

Better than strong baseline.

[1] Y Zhou, C Xiong, et al, “Improving end-to-end speech recognition with policy learning,” ICASSP, 2018.

[2] H Hadian, H Sameti, et al, “Flat-start single-stage discriminatively trained HMM-Based models for ASR,” TASLP, 2018.

[3] H Hadian, H Sameti, et al, “End-to-end speech recognition using lattice-free MMI,” INTERSPEECH, 2018.

[4] N Zeghidour, Q Xu, et al, “Fully convolutional speech recognition,” arXiv preprint arXiv:1812.06864, 2018.

[5] Y Wang, T Chen, et al, “Espresso: A fast endto-end neural speech recognition toolkit,” ASRU, 2019.

[6] H Xiang and Z Ou, “CRF-based single-stage acoustic modeling with CTC topology,” ICASSP, 2019.

[7] K An, H Xiang, et al, “CAT: A CTC-CRF based ASR toolkit bridging the hybrid and the end-to-end approaches towards data efficiency and low latency,” INTERSPEECH, 2020.

[8] V Peddinti, Y Wang, et al, “Low latency acoustic modeling using temporal convolution and LSTMs,” IEEE SPL, 2018.

Experiments: results on WSJ

WERs on the WSJ.

Methods		eval92	dev93
EE-Policy-CTC [1]		5.53	9.21
SS-LF-MMI [2]		3.0	6.0
EE-LF-MMI [3]		3.0	-
FC-SR [4]		3.5	6.8
ESPRESSO [5]		3.4	5.9
CTC	BLSTM	4.93	8.57
	ST-NAS	4.72±0.03	8.82±0.07
BLSTM [6]		3.79	6.23
VGG-BLSTM [7]		3.2	5.7
CTC-CRF	TDNN-D* [8]	2.91	6.24
	Random search	2.82±0.01	5.71±0.03
	ST-NAS	2.77±0.00	5.68±0.01
ST-NAS with fully CTC-CRF		2.81±0.01	5.74±0.02

* Obtained based on our implementation of the “TDNN-D” in [8].

Architectures searched with CTC are transferable to be retrained with CTC-CRF.

[1] Y Zhou, C Xiong, et al, “Improving end-to-end speech recognition with policy learning,” ICASSP, 2018.

[2] H Hadian, H Sameti, et al, “Flat-start single-stage discriminatively trained HMM-Based models for ASR,” TASLP, 2018.

[3] H Hadian, H Sameti, et al, “End-to-end speech recognition using lattice-free MMI,” INTERSPEECH, 2018.

[4] N Zeghidour, Q Xu, et al, “Fully convolutional speech recognition,” arXiv preprint arXiv:1812.06864, 2018.

[5] Y Wang, T Chen, et al, “Espresso: A fast endto-end neural speech recognition toolkit,” ASRU, 2019.

[6] H Xiang and Z Ou, “CRF-based single-stage acoustic modeling with CTC topology,” ICASSP, 2019.

[7] K An, H Xiang, et al, “CAT: A CTC-CRF based ASR toolkit bridging the hybrid and the end-to-end approaches towards data efficiency and low latency,” INTERSPEECH, 2020.

[8] V Peddinti, Y Wang, et al, “Low latency acoustic modeling using temporal convolution and LSTMs,” IEEE SPL, 2018.

Experiments: results on Switchboard

WERs on the Switchboard.

Methods		SW	CH	Params
	TDNN-D-Small	15.2	26.8	7.64M
	TDNN-D-Large	14.6	25.5	11.85M
ST-NAS	Transferred from WSJ	12.5	23.2	11.89M
	Searched on Switchboard	12.6	23.2	15.98M

The architecture searched in WSJ is transferable to Switchboard.

1. All experiments are trained with CTC-CRF. TDNN-D-Small is with the hidden size of 640, which is the same as that of our searched models. TDNN-D-Large is with the hidden size of 800.
2. The transferred model is randomly taken from one of the 5 runs of NAS with CTC over WSJ, and retrained on Switchboard.

Section Conclusion

NAS is an appealing next step to advancing end-to-end ASR.

1. We review existing gradient-based NAS methods and develop an **efficient** NAS method via Straight-Through gradients (ST-NAS).
2. We **successfully** apply ST-NAS to end-to-end ASR. Our ST-NAS induced architectures significantly outperform the human-designed architecture across the WSJ and Switchboard datasets.
3. The ST-NAS method is **flexible** and can be further explored with various backbones of the supernet and candidate operations.

Content

I. Basics for end-to-end speech recognition

1. Probabilistic graphical modeling (PGM) framework
 2. Classic hybrid DNN-HMM models
 3. Connectionist Temporal Classification (CTC)
 4. Attention based encoder-decoder (AED)
 5. RNN transducer (RNNT)
 6. Conditional random fields and sequence discriminative training
-

II. Improving end-to-end speech recognition

15-minute break

1. Data-efficiency
2. Neural architecture search
- 3. Multilingual and crosslingual ASR
4. Language modeling

III. Open questions and future directions

Section Content

1. Motivation

2. Related work

3. Method: **JoinAP**

4. Experiments

5. Conclusion

- Chengrui Zhu, Keyu An, Huahuan Zheng, Zhijian Ou. “Multilingual and Crosslingual Speech Recognition using Phonological-Vector based Phone Embeddings”, ASRU 2021.

Motivation

- There are more than 7100 languages in the world, and most of them are low-resourced languages.
- Multilingual speech recognition
 - Training data from a number of languages (seen languages) are merged to train a multilingual AM.
- Crosslingual speech recognition
 - The target language is unseen in training the multilingual AM.
 - In **few-shot** setting , the AM can be finetuned on limited target language data.
 - In **zero-shot** setting , the AM is directly used without finetuning*.

* Suppose that text corpus from the target language are available.

Intuitively, the key to successful multilingual and crosslingual recognition is to promote the information sharing in multilingual training and maximize the knowledge transferring from the well trained multilingual model to the model for recognizing the utterances in the new language.

Phonological features

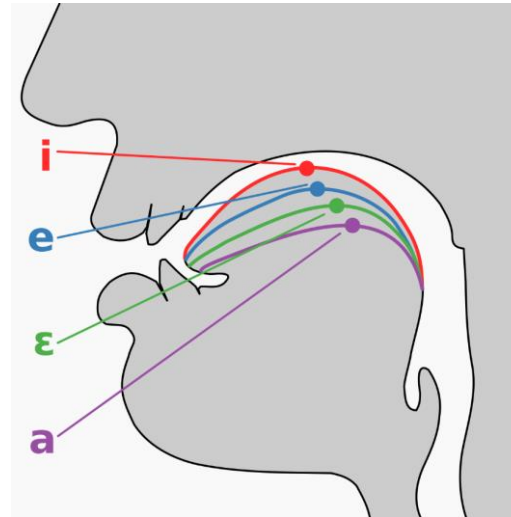
Describe phones by phonological features

■ Vowels

- vowel height
- vowel backness

■ Consonants

- Place of articulation
- Manner of articulation



Phonological feature	d	ε	ð	ə	i	ɖ	kʲ
syllabic	-	+	-	+	+	-	-
sonorant	-	+	-	+	+	-	-
consonantal	+	-	+	-	-	+	+
continuant	-	+	+	+	+	-	-
delayed release	-	-	-	-	-	+	-
lateral	-	-	-	-	-	-	-
nasal	-	-	-	-	-	-	-
strident	0	0	0	0	0	0	0
voice	+	+	+	+	+	+	-
spread glottis	-	-	-	-	-	-	-
constricted glottis	-	-	-	-	-	-	-
anterior	+	0	+	0	0	-	-
coronal	+	-	+	-	-	+	-
distributed labial	-	0	+	0	0	+	0
labial	-	-	-	-	-	-	-
high	-	-	-	-	+	+	+
low	-	-	-	-	-	-	-
back	-	-	-	+	-	-	-
round	-	-	-	-	-	-	-
velaric	-	-	-	-	-	-	-
tense	0	-	0	-	+	0	0
long	-	-	-	-	-	-	-
hitone	0	0	0	0	0	0	0
hireg	0	0	0	0	0	0	0

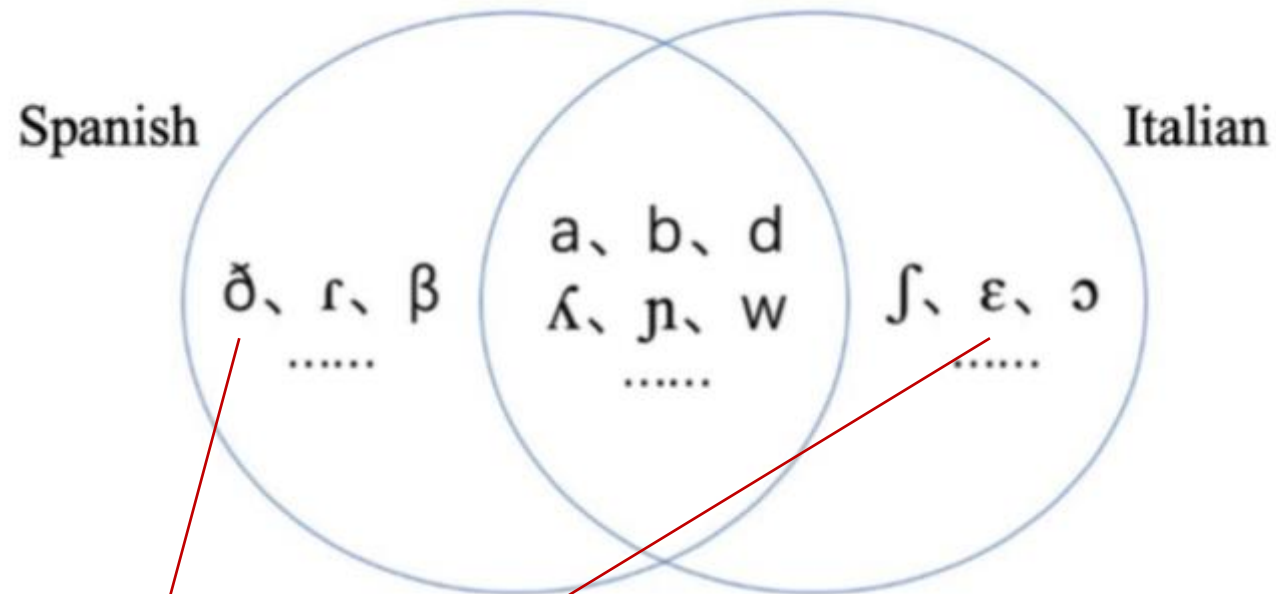
Phonological features: micro-decomposition of phones

- Like atoms could be split into nucleus and electrons, phones can be expressed by phonological features.

Matter	Speech
Atoms	Phones
Periodic table of elements	IPA table
Nucleus, electrons	Phonological features

Phonological features: promote information sharing

- Even language-specific phones are connected by using phonological features.

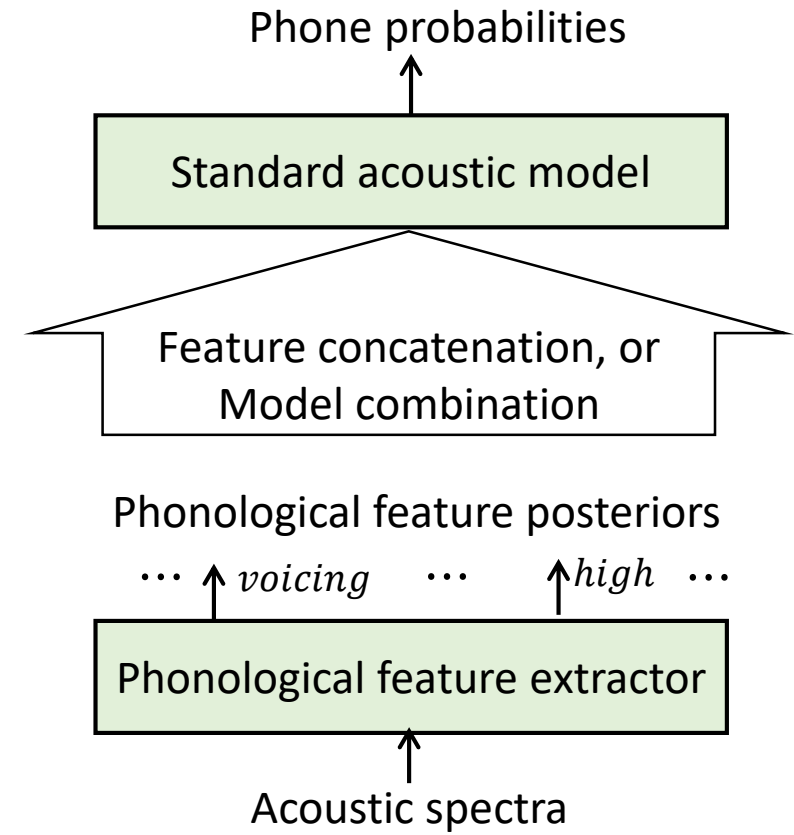


ð : -, +, +, -, -, -, 0, +, -, -, +, +, +, -, -, -, -, -, -, 0, -, -, 0, 0

ε : +, +, -, +, -, -, -, 0, +, -, -, 0, -, 0, -, -, -, +, -, -, +, -, -, 0, 0

Related work

- Phonological features(PFs) have been applied in multilingual and crosslingual ASR
- Previous studies generally take a bottom-up approach, and suffer from:
 - The acoustic-to-PF extraction in a bottom-up way is itself **difficult**.
 - Do not provide a principled model to calculate the phone probabilities **for unseen phones** from the new language towards zero-shot crosslingual recognition.



From phonological features to phonological-vector

- Phonological-vector

- Encode each phonological feature by a 2-bit binary vector. (24PFs -> 48bits)

+	-	0
10	01	00

- Plus 3 bits to indicate <blk>, <spn>, <nsm>
- Phonological-vector: Total 51 bits

Joining of Acoustics and Phonology (JoinAP)

- The JoinAP method

- DNN based acoustic feature extraction (bottom-up) and phonology driven phone embedding (top-down) are joined to calculate the **logits**.

- JoinAP-Linear

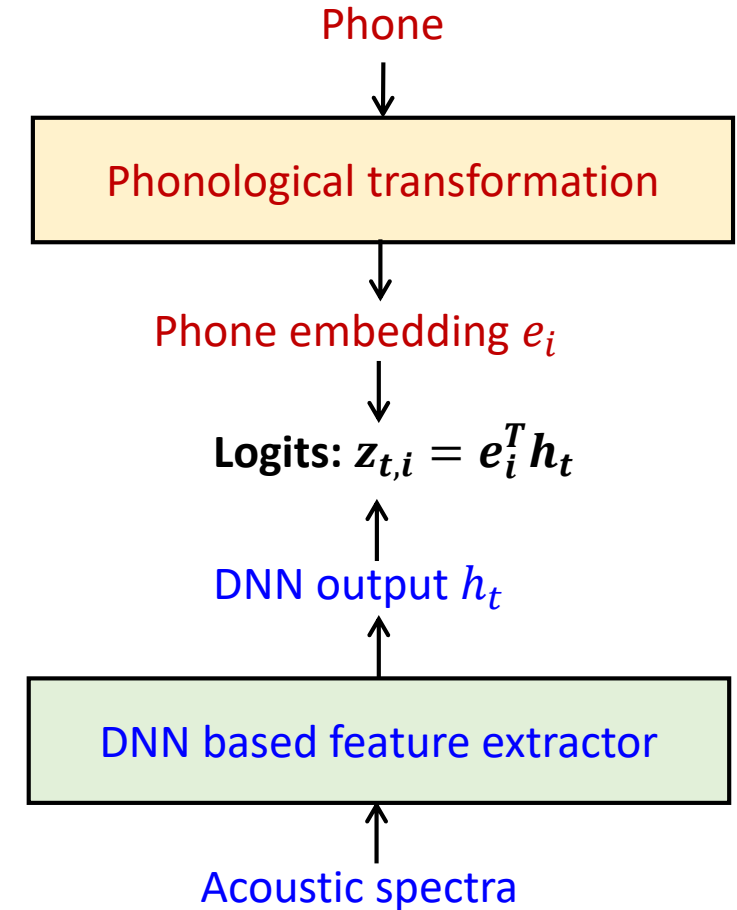
- Linear transformation of phonological-vector p_i to define the embedding vector for phone i :

$$e_i = Ap_i \in \mathbb{R}^H$$

- JoinAP-Nonlinear

- Apply nonlinear transformation, multilayered neural networks:

$$e_i = A_2 \sigma(A_1 p_i) \in \mathbb{R}^H$$



Experiments

- Train multilingual AM on German, French, Spanish and Polish.
- Zero-shot and few-shot crosslingual ASR on Polish and Mandarin.
- Employ Phonetisaurus G2P to generate IPA lexicons
- Use CTC-CRF based ASR toolkit, CAT
 - **Acoustic model**: 3 layer VGGBLSTM with **1024** hidden dim
 - **Adam optimizer**: with an initial learning rate of 0.001, decreased to 1/10 until less than 0.00001
 - **Dropout** 0.5

Language	Corpora	#Phones	Train	Dev	Test
German	CommonVoice	40	639.4	24.7	25.1
French	CommonVoice	57	465.2	21.9	23.0
Spanish	CommonVoice	30	246.4	24.9	25.6
Italian	CommonVoice	33	89.3	19.7	20.8
Polish	CommonVoice	46	93.2	5.2	6.1
Mandarin	AISHELL-1	96	150.9	18.1	10.0

Experiments

- Multilingual experiments

Language	Flat-Phone monolingual	Flat-Phone w/o finetuning	Flat-Phone finetuning	JoinAP-Linear w/o finetuning	JoinAP-Linear finetuning	JoinAP-Nonlinear w/o finetuning	JoinAP-Nonlinear finetuning
German	13.09	14.36	12.42	13.72	12.45	13.97	12.64
French	18.96	22.73	18.91	22.73	19.54	22.88	19.62
Spanish	15.11	13.93	13.06	13.93	13.19	14.10	13.26
Italian	24.57	25.97	21.77	25.85	21.70	24.06	20.29
Average	17.93	19.25	16.54	19.06	16.72	18.75	16.45

- Language-degree of a phone: how many languages a phone appears

		Language-degree			
Language		4	3	2	1
	German	18	6	8	8
	French	18	6	7	26
	Spanish	18	4	1	7
	Italian	18	5	4	6

On average, both JoinAP-Nonlinear and JoinAP-Linear perform better than Flat-Phone, and JoinAP-Nonlinear is the strongest.

Experiments

- Crosslingual experiments

- Polish:

#Finetune	Flat-Phone	JoinAP-Linear	JoinAP-Nonlinear
0	33.15	35.73	31.80
10 minutes	8.70	7.50	8.10

- Mandarin:

#Finetune	Flat-Phone	JoinAP-Linear	JoinAP-Nonlinear
0	97.10	89.51	88.41
1 hour	25.39	25.21	24.86

- Statistics about Polish and Mandarin:

Language	#Phones	#Unseen phones
Polish	46	18
Mandarin	96	79

On average, both JoinAP-Nonlinear and JoinAP-Linear perform better than Flat-Phone, and JoinAP-Nonlinear is the strongest.

Section Conclusion

- In the multilingual and crosslingual experiments, **JoinAP-Nonlinear** generally performs better than **JoinAP-Linear** and the traditional **flat-phone** method on average. The improvements for target language depend on its data amount and language-degree.
- Our JoinAP method provides **a principled, data-efficient approach** to multilingual and crosslingual speech recognition.
- Promising directions: exploring DNN based phonological transformation, and pretraining over increasing number of languages.

Content

I. Basics for end-to-end speech recognition

1. Probabilistic graphical modeling (PGM) framework
 2. Classic hybrid DNN-HMM models
 3. Connectionist Temporal Classification (CTC)
 4. Attention based encoder-decoder (AED)
 5. RNN transducer (RNNT)
 6. Conditional random fields and sequence discriminative training
-

II. Improving end-to-end speech recognition

15-minute break

1. Data-efficiency
2. Neural architecture search
3. Multilingual and crosslingual ASR
- 4. Language modeling

III. Open questions and future directions

Section Content

1. Motivation

2. Related work

3. Method: **RF LMs**

4. Experiments

5. Conclusion

- Bin Wang, Zhijian Ou, Zhiqiang Tan. Trans-dimensional Random Fields for Language Modeling. *ACL Long Paper*, 2015.
- Bin Wang, Zhijian Ou, Zhiqiang Tan. Learning Trans-dimensional Random Fields with Applications to Language Modeling. *TPAMI*, 2018.
- Bin Wang, Zhijian Ou. Language modeling with neural trans-dimensional random fields. *ASRU*, 2017.
- Bin Wang, Zhijian Ou. Learning neural trans-dimensional random field language models with noise-contrastive estimation. *ICASSP*, 2018.
- Bin Wang, Zhijian Ou. Improved training of neural trans-dimensional random field language models with dynamic noise-contrastive estimation. *SLT*, 2018.
- Silin Gao, Zhijian Ou, Wei Yang, Huifang Xu. Integrating discrete and neural features via mixed-feature trans-dimensional random field language models. *ICASSP*, 2020. [Oral]

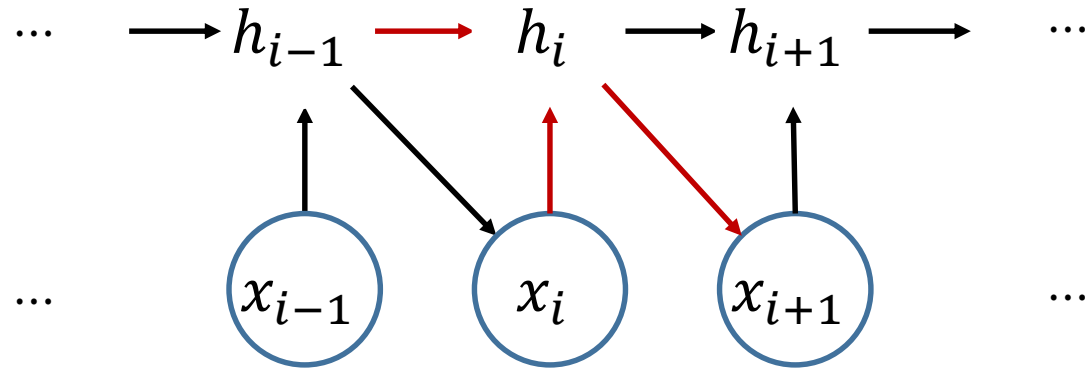
N-gram LMs

- Language modeling (LM) is to determine the joint probability of a sentence, i.e. a word sequence.
- Dominant: Directed modeling approach

$$p(x_1, x_2, \dots, x_l) = \prod_{i=1}^l p(x_i | x_1, \dots, x_{i-1})$$
$$\approx \prod_{i=1}^l p(x_i | \underline{x_{i-n+1}, \dots, x_{i-1}})$$

- Using Markov assumption leads to the N-gram LMs
 - One of the state-of-the-art LMs

Recurrent Neural Nets (RNNs)/LSTM/Transformer LMs



$$p(x_i | x_1, \dots, x_{i-1}) \approx p(x_i | h_{i-1}(x_1, \dots, x_{i-1})) \approx \frac{h_{i-1}^T w_k}{\sum_{k=1}^V h_{i-1}^T w_k}$$

☹️.1 Computational expensive in both training and testing ¹

e.g. $V = 10^4 \sim 10^6$, $w_k \in \mathbb{R}^{250 \sim 1024}$

☹️.2 As directed sequential model /Auto-regressive model, potentially suffers from Exposure Bias and Label Bias

¹ Partly alleviated by using un-normalized models (e.g., through NCE) or a small set of tokens (e.g., BPE).

Trans-dimensional Random Field (TRF) LM: motivation

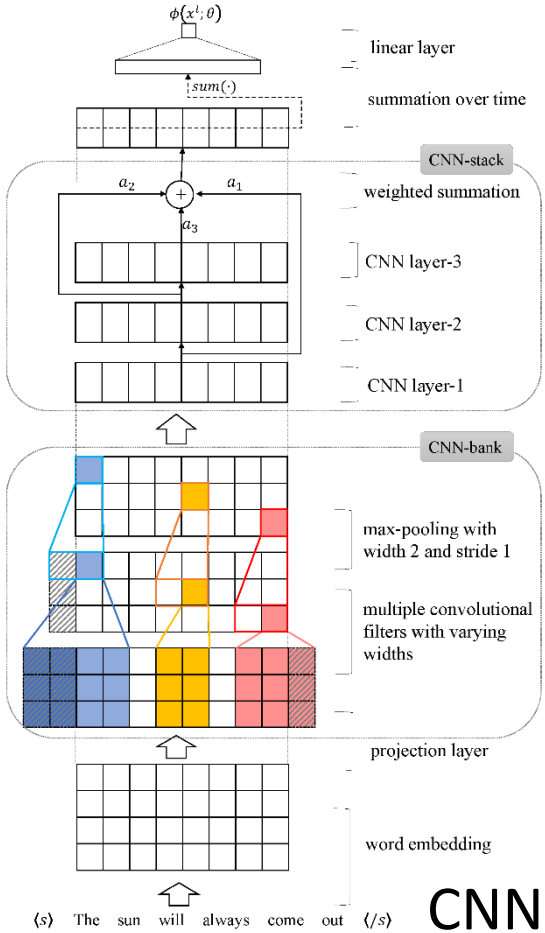
☺.1 Avoid local normalization

$$p_{\theta}(x^l) = \frac{1}{Z_l(\theta)} e^{u_{\theta}(x^l)}, x^l \triangleq x_1, x_2, \dots, x_l$$

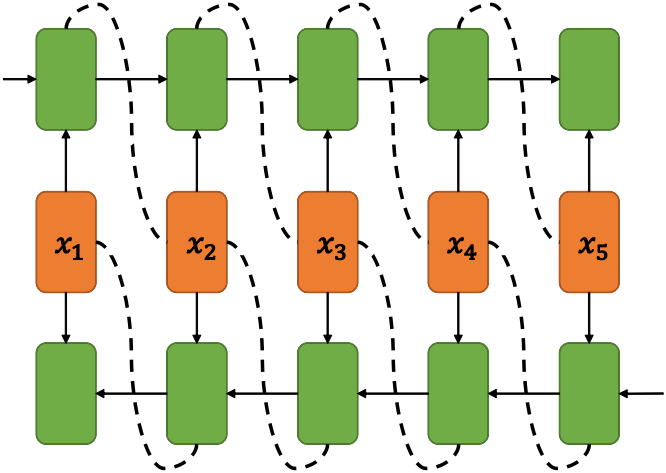
☺.2 Flexible

Type	Features
w	$(w_{-3}w_{-2}w_{-1}w_0)(w_{-2}w_{-1}w_0)(w_{-1}w_0)(w_0)$
c	$(c_{-3}c_{-2}c_{-1}c_0)(c_{-2}c_{-1}c_0)(c_{-1}c_0)(c_0)$
ws	$(w_{-3}w_0)(w_{-3}w_{-2}w_0)(w_{-3}w_{-1}w_0)(w_{-2}w_0)$
cs	$(c_{-3}c_0)(c_{-3}c_{-2}c_0)(c_{-3}c_{-1}c_0)(c_{-2}c_0)$
wsh	$(w_{-4}w_0)(w_{-5}w_0)$
csh	$(c_{-4}c_0)(c_{-5}c_0)$
cpw	$(c_{-3}c_{-2}c_{-1}w_0)(c_{-2}c_{-1}w_0)(c_{-1}w_0)$
tied	$(c_{-9:-6}, c_0)(w_{-9:-6}, w_0)$

Discrete features



CNN features



BLSTM features

Trans-dimensional random fields (TRFs): model

- Assume the sentences of length l are distributed as follows:

$$p_l(x^l; \lambda) = \frac{1}{Z_l(\lambda)} e^{\lambda^T f(x^l)} \quad l = 1, \dots, l_{max}$$

$x^l \triangleq x_1, x_2, \dots, x_l$ is a word sequence with length l ;

$f(x^l) = (f_1(x^l), \dots, f_d(x^l))^T$ is the feature vector;

$\lambda = (\lambda_1, \dots, \lambda_d)^T$ is the parameter vector;

$Z_l(\lambda) = \sum_{x^l} e^{\lambda^T f(x^l)}$ is the normalization constant.

Needed to be estimated

- Assume length l is associated with priori probability π_l . Therefore the pair (l, x^l) is jointly distributed as:

$$p(l, x^l; \lambda) = \pi_l \cdot p_l(x^l; \lambda)$$

Feature definition

$$p_l(x^l; \lambda) = \frac{1}{Z_l(\lambda)} e^{\lambda^T f(x^l)}$$

- $f_i(x^l)$ returns the count of a specific phrase observed in the input sentence x^l

$x^l = he\ is\ a\ teacher\ and\ he\ is\ also\ a\ good\ father.$

$f_{he\ is}(x^l) = \text{count of "he is" observed in } x^l = 2$

$f_{a\ teacher}(x^l) = \text{count of "a teacher" observed in } x^l = 1$

$f_{she\ is}(x^l) = \text{count of "she is" observed in } x^l = 0$

... ..

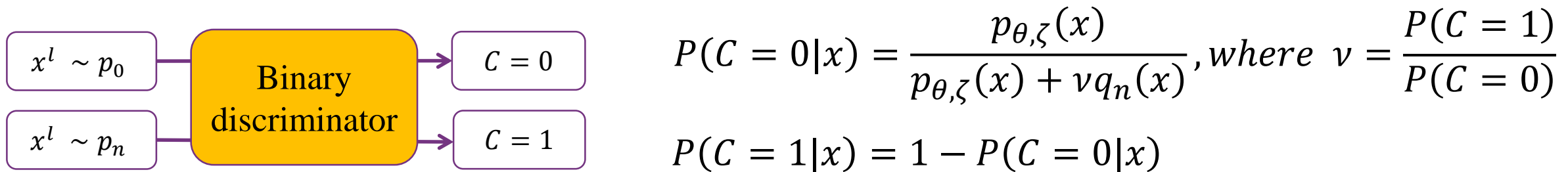
- For example, **n-grams** and **skip n-grams (tied or not)** of orders ranging from 1 to 10, observed in the training set are added to the features.

Review the development of TRF LMs

ACL-2015 TPAMI-2018	<ul style="list-style-type: none">• Discrete features• Augmented stochastic approximation (AugSA) for model training
ASRU-2017	<ul style="list-style-type: none">• Potential function as a deep CNN.• Model training by AugSA plus JSA (joint stochastic approximation)
ICASSP-2018	<ul style="list-style-type: none">• Use LSTM on top of CNN• NCE is introduced to train TRF LMs
SLT-2018	<ul style="list-style-type: none">• Simplify the potential definition by using only Bidirectional LSTM• Propose Dynamic NCE for improved model training

Model training

- The target RF model $p_{\theta}(x) = \frac{1}{Z(\theta)} e^{u_{\theta}(x)}$
- Treat $\log Z(\theta)$ as a parameter ζ and rewrite $p_{\theta, \zeta}(x) \propto e^{u_{\theta}(x) - \zeta}$
- Introduce a **noise distribution** $q_n(x)$, and consider a binary classification



- Noise Contrastive Estimation (NCE):

$$\max_{\theta, \zeta} E_{x \sim p_0(x)} [\log P(C = 0|x)] + E_{x \sim q_n(x)} [\log P(C = 1|x)]$$

☺ $p_{\theta} \rightarrow p_0$ (oracle), under infinite amount of data and infinite capacity of p_{θ} .

☹ Reliable NCE needs a large $\nu \approx 20$; Overfitting. Dynamic-NCE in (Wang&Ou, SLT 2018).

Motivation: Integrating discrete and neural features

- Language models using discrete features (N-gram LMs, Discrete TRF LMs)
 - Mainly capture local lower-order interactions between words
 - Better suited to handling symbolic knowledges
- Language models using neural features (LSTM LMs, Neural TRF LMs)
 - Able to learn higher-order interactions between words
 - Good at learning smoothed regularities due to word embeddings
- Interpolation of LMs^{1, 2}: usually achieves further improvement
 - Discrete and neural features have complementary strength. 😊
 - Two-step model training is sub-optimal. 😞

¹Xie Chen, Xunying Liu, Yu Wang, Anton Ragni, Jeremy HM Wong, and Mark JF Gales, “Exploiting future word contexts in neural network language models for speech recognition,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 9, pp. 1444–1454, 2019.

²Bin Wang, Zhijian Ou, Yong He, and Akinori Kawamura, “Model interpolation with trans-dimensional random field language models for speech recognition,” *arXiv preprint arXiv:1603.09170*, 2016.



Mixed TRF LMs: Definition

- Mixed TRF LMs:

- $p(l, x^l; \eta) = \frac{\pi_l}{Z_l(\eta)} e^{V(x^l, \eta)}$, $V(x^l, \eta) = \lambda^T f(x^l) + \phi(x^l; \theta)$, $\eta = (\lambda, \theta)$

Discrete n-gram features, with parameter λ :

$$f(x^l) = (f_1(x^l), f_2(x^l), \dots, f_N(x^l))$$

N : the total number of types of n-grams

$$f_k(x^l) = c$$

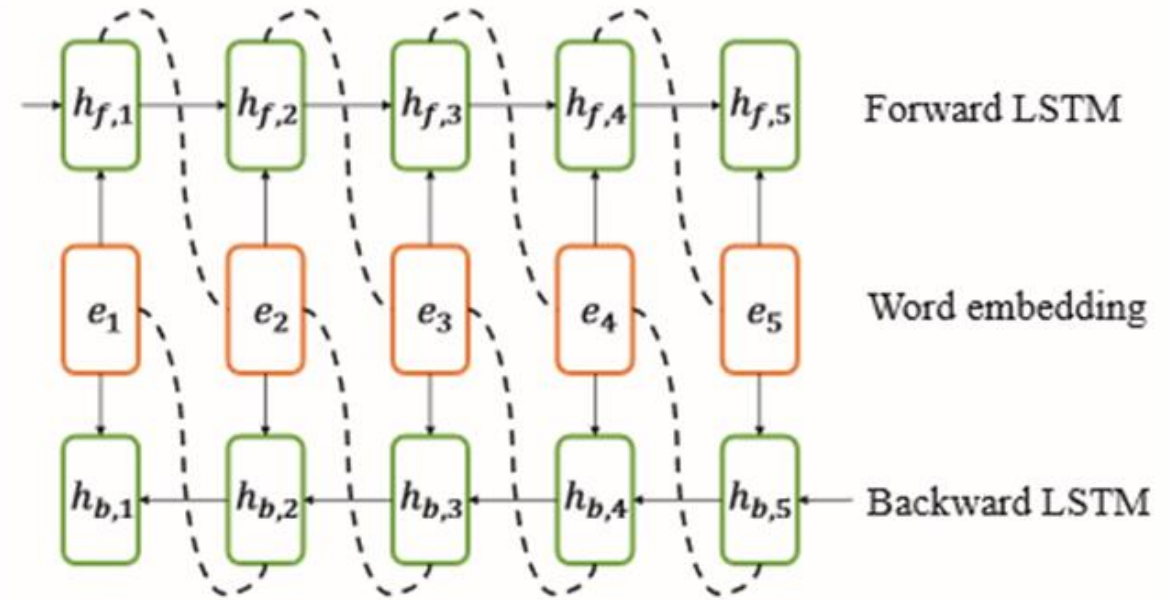
where c is the count of the k th n-gram type in x^l

$x^l = he is a teacher and he is also a good father.$

$f_{he is}(x^l) = \text{count of "he is" in } x^l = 2$

$f_{a teacher}(x^l) = \text{count of "a teacher" in } x^l = 1$

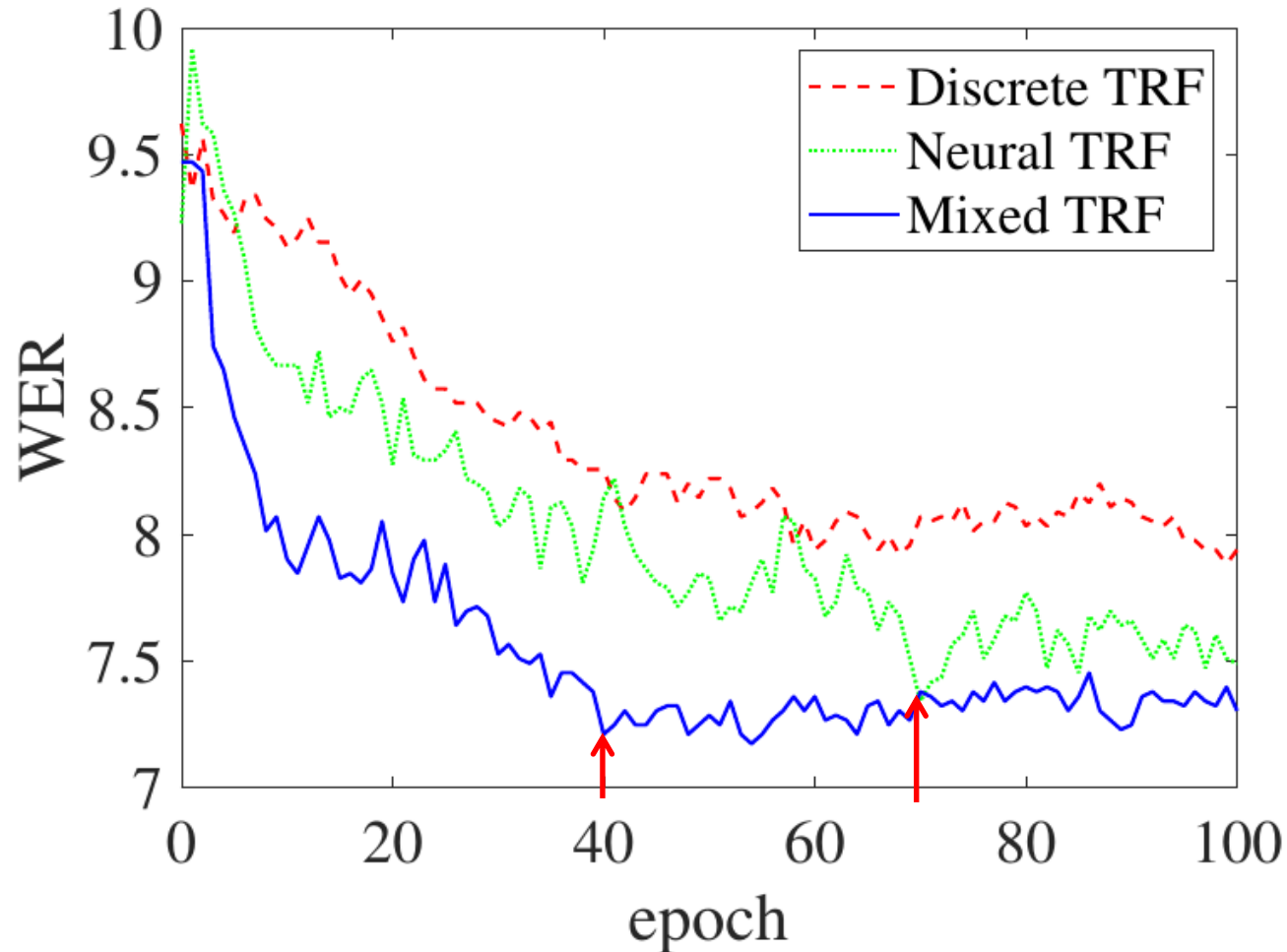
Neural network features, with parameter θ



$$\phi(x^l; \theta) = \sum_{i=1}^{l-1} h_{f,i}^T e_{i+1} + \sum_{i=2}^l h_{b,i}^T e_{i-1}$$

Experiments: PTB dataset

WER curves of the three TRF LMs during the first 100 training epochs:



- Mixed TRF converges faster than the state-of-the-art Neural TRF, using only **58%** training epochs.

😊 The discrete features in Mixed TRF lower the non-convexity of the optimal problem, and reduce the amount of patterns for neural features to capture.

On Google one-billion word benchmark

Training: Google One-Billion word benchmark, 0.8 billion words, 568K vocabulary

Testing: WSJ'92 test data, 330 utterances, rescoreing 1000-best lists

Model	WER (%)	#Param (M)	Training time	Inference Time
KN5	6.13	133	2.5 h (1 CPU)	0.491 s (1 CPU)
LSTM-2x1024	5.55	191	144 h (2 GPUs)	0.909 s (2 GPUs)
discrete-TRF basic	6.04	102	131 h (8 cores and 2 GPUs)	0.022 s (1 CPU)
neural-TRF	5.47	114	336 h (2 GPUs)	0.017 s (2 GPUs)
mix-TRF	5.28	216	297 h (8 cores and 2 GPUs)	0.024 s (1 core and 2 GPUs)
LSTM-2x1024+KN5	5.38	324		

Annotations: Red arrows indicate WER changes. From LSTM-2x1024 to mix-TRF: WER decreases by 5%. From mix-TRF to LSTM-2x1024+KN5: WER decreases by 33%. From LSTM-2x1024 to LSTM-2x1024+KN5: Inference time decreases by 38x.

Open-source LM toolkit

<https://github.com/thu-spmi/SPMILM>



Section Conclusion

- Language models play an important role for ASR!
- Random Field language models
 - Avoid local normalization
 - Being flexible to integrate rich features (both discrete and neural)
 - Overcome “label bias” and “Exposure bias”
- More related work
 - Residual energy-based models for text generation
 - Electric: an energy-based cloze model for representation learning over text
- Yuntian Deng, Anton Bakhtin, Myle Ott, Arthur Szlam, and Marc'Aurelio Ranzato. Residual energy-based models for text generation, ICLR 2020.
- Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. Pre-training transformers as energy-based cloze models, EMNLP 2020.

Content

I. Basics for end-to-end speech recognition

1. Probabilistic graphical modeling (PGM) framework
 2. Classic hybrid DNN-HMM models
 3. Connectionist Temporal Classification (CTC)
 4. Attention based encoder-decoder (AED)
 5. RNN transducer (RNNT)
 6. Conditional random fields and sequence discriminative training
-

II. Improving end-to-end speech recognition

15-minute break

1. Data-efficiency
2. Neural architecture search
3. Multilingual and crosslingual ASR
4. Language modeling

III. Open questions and future directions

“WER we are and WER we think we are”

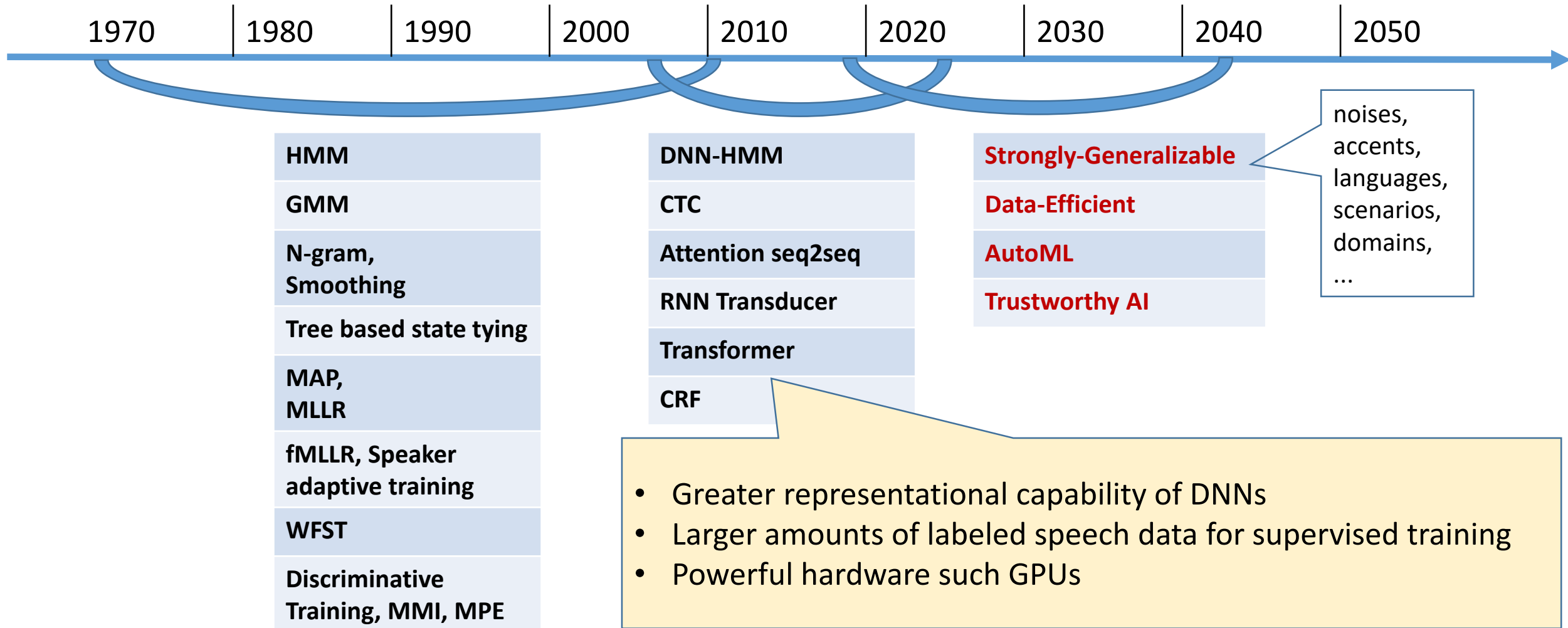
“The conclusions are clear: we are definitely not where we think we are in terms of WERs (Word Error Rates).”

ASR	CCC	SWBD	CallHome
ASR 1	17.9	11.62	17.69
ASR 2	19.2	11.45	18.6
ASR 3	16.5	10.2	15.85

Table 1: WER [%] comparison on benchmarks

- Test: three different state-of-the-art commercial ASR solutions
- Call Center Conversations (CCC)
- The commercial ASR systems in our evaluation achieve **nearly double** the error rates (reported in the literatures) on both HUB’05 evaluation subsets.

New-generation ASR



Thanks for your attention !

Thanks to my collaborators and students :

Hongyu Xiang, Keyu An, Huahuan Zheng, Wenjie Peng, Bin Wang,
Zhiqiang Tan, Silin Gao