

联合神经网络与无向图模型的高效语音识别

欧智坚

清华大学电子工程系

语音处理与机器智能实验室

Speech Processing and Machine Intelligence (SPMI) Lab

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

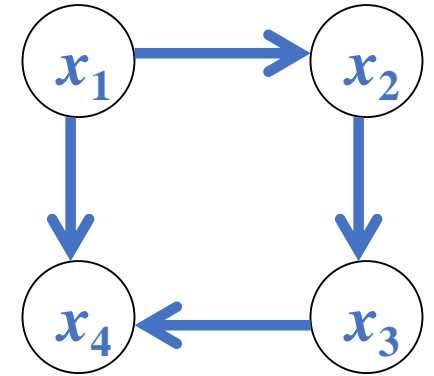
ASRU中英混杂语音识别挑战赛线下技术交流会 2019/11/23

We need probabilistic models, besides neural nets.

• Directed Graphical Models / Bayesian Networks (BNs)

- Self-normalized
- e.g. Hidden Markov Model (HMM), Neural network (NN) based classifier, Variational AutoEncoder (VAE), Generative Adversarial Network (GAN), auto-regressive model (e.g. RNN/LSTM)

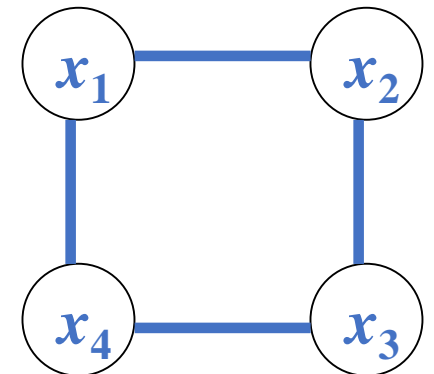
$$P(x_1, x_2, x_3, x_4) = P(x_1)P(x_2|x_1)P(x_3|x_2)P(x_4|x_1, x_3)$$



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

- Involves the normalizing constant (the partition function) Z
- e.g. Conditional Random Field (CRF), Restricted Boltzmann Machine (RBM)

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



Content

- CRF-based end-to-end speech recognition

基于条件随机场的高效端到端语音识别

- ✓ ICASSP 2019
- ✓ Data efficient

- Random field approach to language modeling

随机场语言模型

- ✓ ACL Long Paper 2015, ASRU 2017, T-PAMI 2018, ICASSP 2018, SLT 2018
- ✓ Representational efficient, computational efficient

CTC-CRF

基于条件随机场的高效端到端语音识别

Hongyu Xiang, Zhijian Ou.

CRF-based Single-stage Acoustic Modeling with CTC Topology.

ICASSP 2019. [Oral]

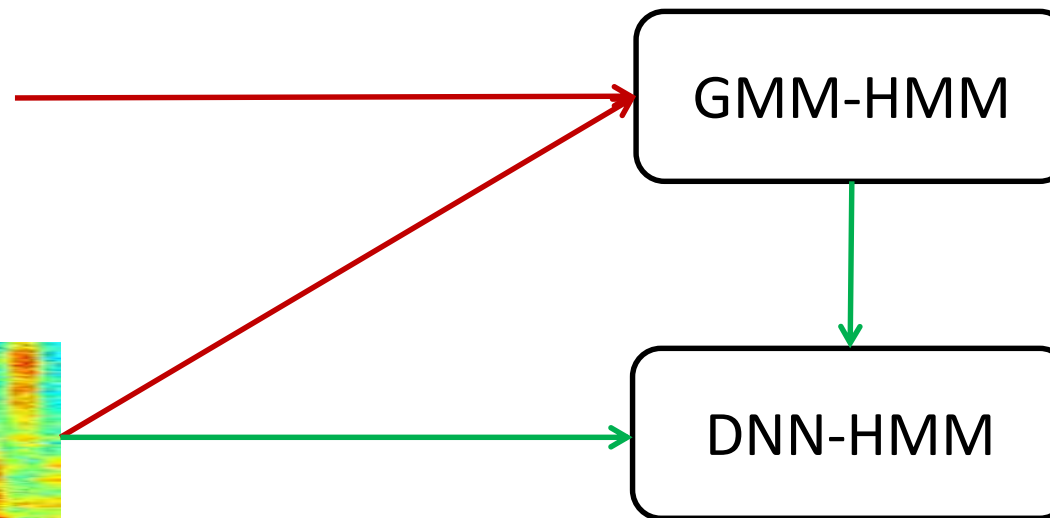
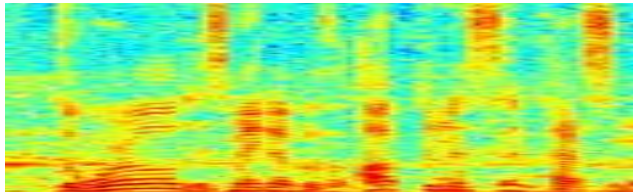


Introduction

- ASR is a discriminative problem
 - For acoustic observations $\mathbf{x} \triangleq x_1, \dots, x_T$, find the most likely labels $\mathbf{y} \triangleq y_1, \dots, y_L$
- ASR state-of-the-art: DNNs of various network architectures (Hinton NIPSw2009, Microsoft IS2011)
- Conventionally, multi-stage
 - Monophone \rightarrow alignment & triphone tree building \rightarrow triphone \rightarrow alignment \rightarrow DNN-HMM

Labels \mathbf{y} :
Nice to meet you.

Acoustic features \mathbf{x} :



Motivation

- End-to-end system:

- Eliminate GMM-HMM pre-training and tree building, and can be trained from scratch (flat-start or single-stage).

- In a more strict sense:

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

We advocate 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

Related work

ASR is a 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 $L \neq T$

- Explicitly by state sequence $\boldsymbol{\pi} \triangleq \pi_1, \dots, \pi_T$ in HMM, CTC, RNN-T, or implicitly in Seq2Seq

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

Related work How to handle alignment, since $L \neq T$

- Explicitly by state sequence $\boldsymbol{\pi} \triangleq \pi_1, \dots, \pi_T$ in HMM, CTC, RNN-T, or implicitly in Seq2Seq
- State topology : determines a mapping \mathcal{B} , which map $\boldsymbol{\pi}$ to a unique l

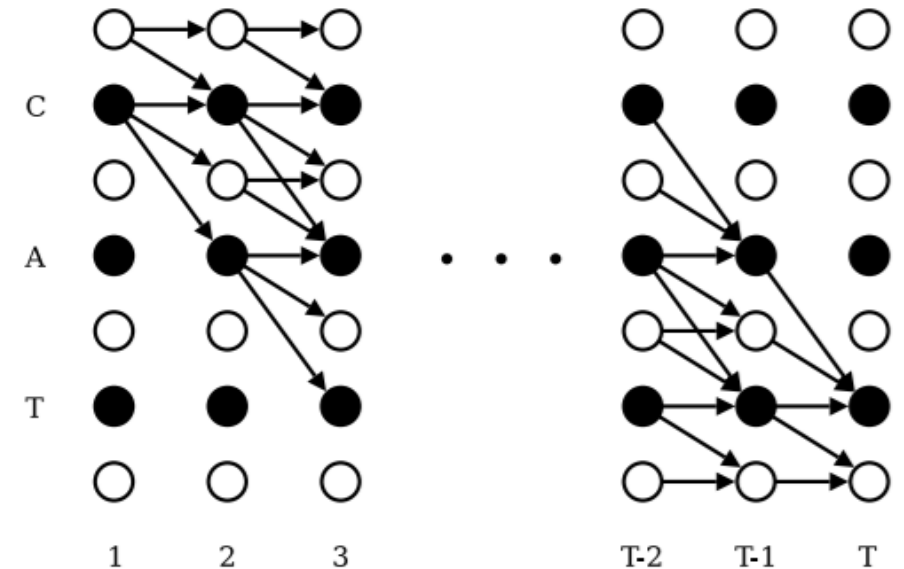
$$p(\mathbf{y}|\mathbf{x}) = \sum_{\boldsymbol{\pi} \in \mathcal{B}^{-1}(l)} p(\boldsymbol{\pi}|\mathbf{x})$$

CTC topology : a mapping \mathcal{B} maps $\boldsymbol{\pi}$ to l by

1. removing all repetitive symbols between the blank symbols.
2. removing all blank symbols.

$$\mathcal{B}(-CC - -AA - T -) = CAT$$

- ☺ Admit the smallest number of units in state inventory, by adding only one `<blk>` to label inventory.
- ☺ Avoid ad-hoc silence insertions in estimating denominator LM of labels.



Related work How to obtain $p(\mathbf{y} | \mathbf{x})$

- Directed Graphical Model/Locally normalized

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

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

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

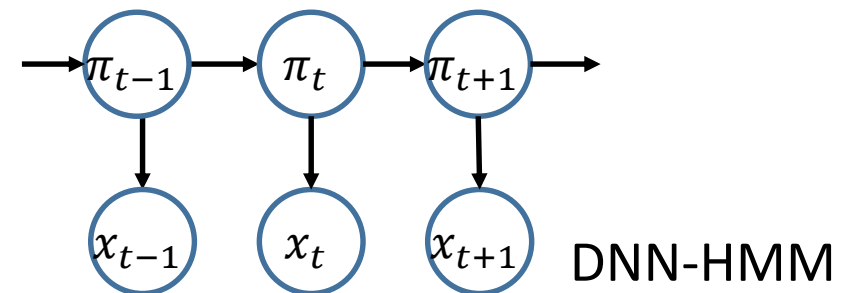
- Undirected Graphical Model/Globally normalized

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

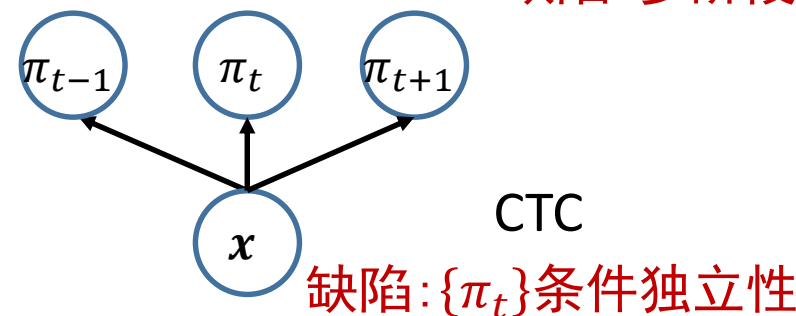
MMI training of GMM-HMMs is equiv. to

CML training of CRFs (using 0/1/2-order features in potential definition).

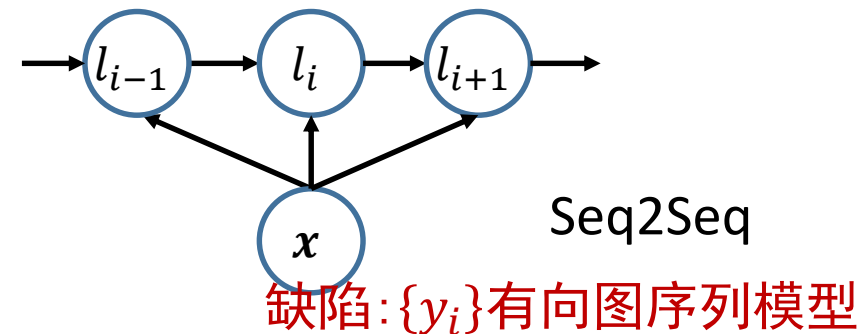
Heigold, et al., "Equivalence of generative and log-linear models", T-ASLP 2011.



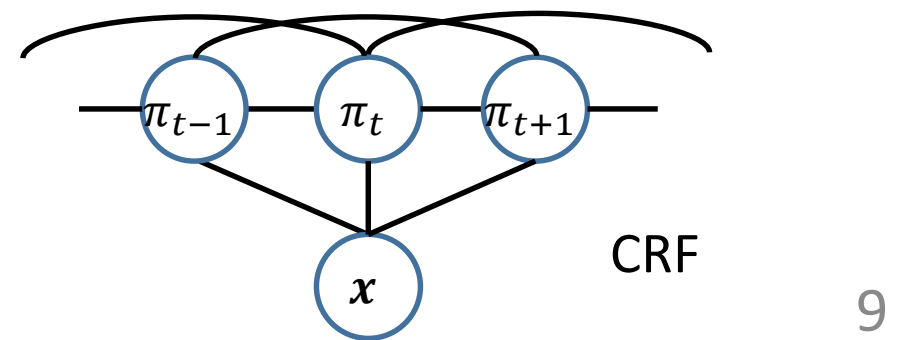
DNN-HMM
缺陷: 多阶段



CTC
缺陷: $\{\pi_t\}$ 条件独立性

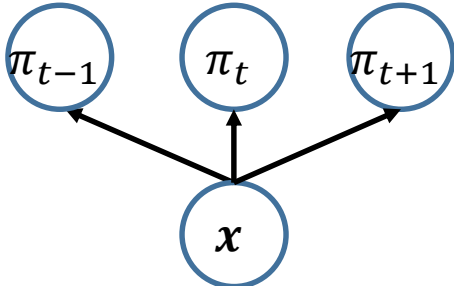
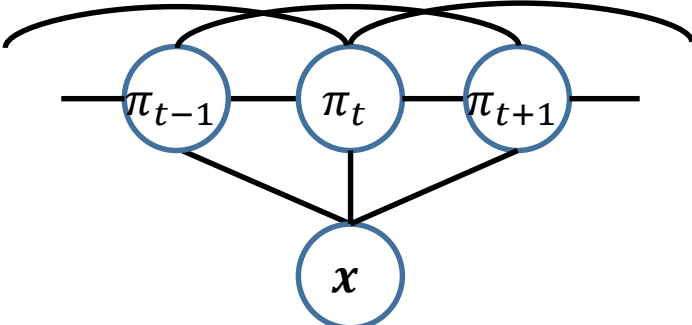


Seq2Seq
缺陷: $\{y_i\}$ 有向图序列模型



CRF

CTC vs CTC-CRF

CTC	CTC-CRF
$p(\mathbf{l} \mathbf{x}) = \sum_{\pi \in \mathcal{B}^{-1}(\mathbf{l})} p(\pi \mathbf{x}), \text{ using CTC topology } \mathcal{B}$	
<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>
$\frac{\partial \log p(\mathbf{l} \mathbf{x}; \theta)}{\partial \theta} = \mathbb{E}_{p(\pi \mathbf{l}, \mathbf{x}; \theta)} \left[\frac{\partial \log p(\pi \mathbf{x}; \theta)}{\partial \theta} \right]$	$\frac{\partial \log p(\mathbf{l} \mathbf{x}; \theta)}{\partial \theta} = \mathbb{E}_{p(\pi \mathbf{l}, \mathbf{x}; \theta)} \left[\frac{\partial \phi(\pi, \mathbf{x}; \theta)}{\partial \theta} \right] - \mathbb{E}_{p(\pi' \mathbf{x}; \theta)} \left[\frac{\partial \phi(\pi', \mathbf{x}; \theta)}{\partial \theta} \right]$
	

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(6,320), 13M
CTC-CRF monophone	9.8	18.7	14.3	2019, VGG BLSTM(6,320), 16M
DNN-HMM triphone	9.8	19.0	14.4	RWTH IS 2018
DNN-HMM triphone	9.6	19.3	14.5	IBM IS 2019
Seq2Seq subword	11.8	25.7	18.8	RWTH IS 2018, LSTM-LM
Seq2Seq subword	10.7	20.7	15.7	Espnet ASRU19

10%

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
DNN-HMM triphone	4.4	10.0	RWTH IS 2019
Seq2Seq subword	4.8	15.3	RWTH IS 2018
Seq2Seq subword	4.0	12.0	Espnet ASRU19
Seq2Seq subword	4.1	12.5	Google IS19 (w/o SpecAugment)

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

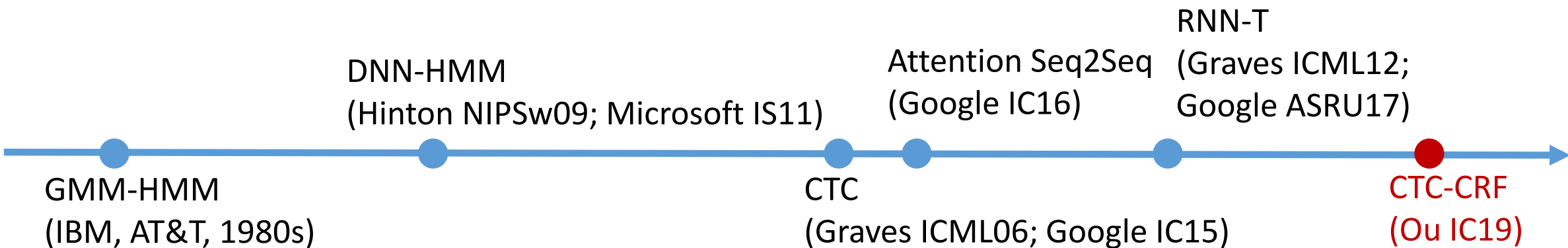
RWTH IS 2019, “RWTH ASR Systems for LibriSpeech Hybrid vs Attention -- Data Augmentation”.

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

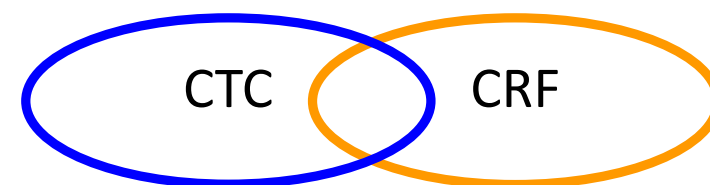
Espnet ASRU19, “Espresso: A Fast End-to-end Neural Speech Recognition Toolkit”.

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

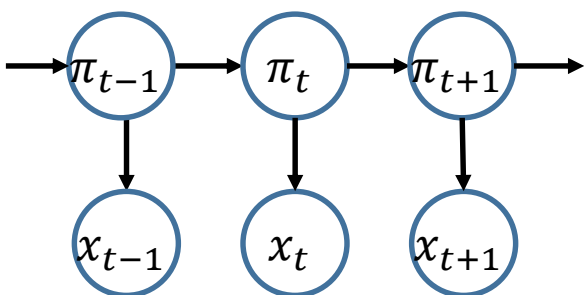
基于条件随机场的高效端到端语音识别 – 总结



历史上的各类模型具有不同的图结构，ctc-crf占有独特位置！

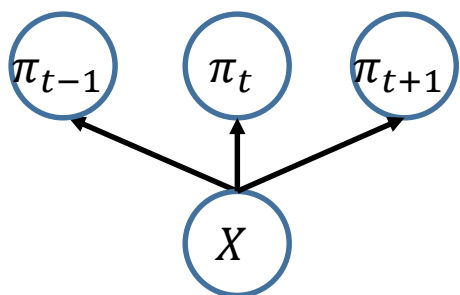


联合神经网络与概率图模型



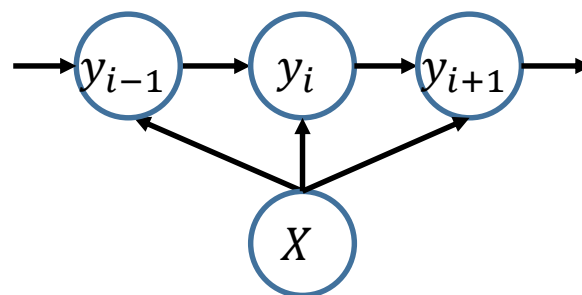
DNN-HMM

缺陷: 多阶段



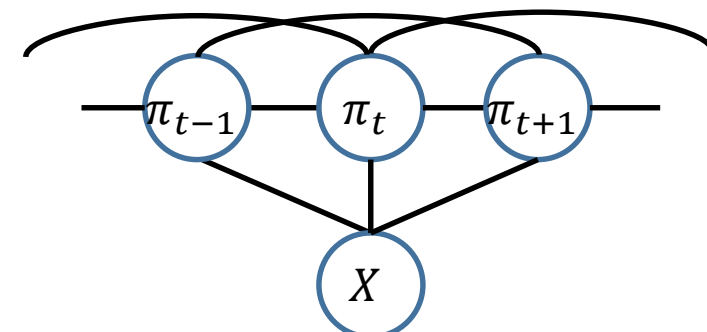
CTC

缺陷: $\{\pi_t\}$ 条件独立性



Attention

缺陷: $\{y_i\}$ 有向图序列模型



CTC-CRF

基于条件随机场的高效端到端语音识别 – 总结

- 在WSJ、Switchboard、Librispeech, 性能表现均超过了
 - CTC、Attention Seq2Seq (15%相对改进),
 - 现在广为流行的Kaldi工具包中的端对端模型e2e Chain-model (10%相对改进),
 - 与多阶段Chain-model持平,
- 同时具有训练流程简洁、能充分利用词典及语言模型从而数据利用效率高等优势。

开源 Crf-based Asr Toolkit (CAT)



1. CAT contains a full-fledged implementation of CTC-CRF

- Fast parallel calculation of gradients using CUDA C/C++ interface

2. CAT adopts PyTorch to build DNNs

3. CAT provides a complete workflow with example recipes

4. Flexibility and future work

- CRFs with different topologies
- Streaming ASR
- Speaker-adapted recognition
- Multi-lingual, Code-mix
- ...



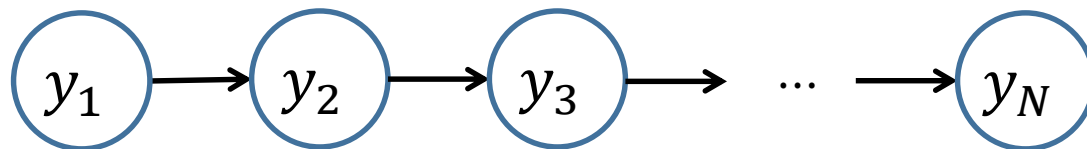
TRF

随机场语言模型

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

Motivation

有向图序列模型



$$P(y_1, y_2, \dots, y_N) = \prod_t P(y_{t+1} | y_t)$$

两个缺陷:

如: N-Grams、神经网络语言模型, 用于语音识别、机器翻译等

1. 从左到右单向边不能有效对上下文建模

The cat is **on** the table.

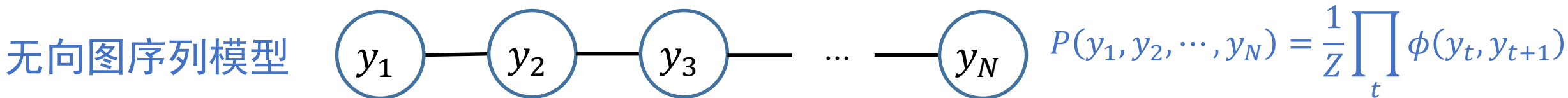
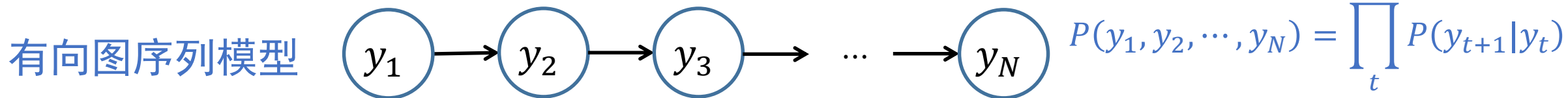
The cat is **in** the house.

2. 每个位置: 局部归一化计算 $P(y_{t+1} | y_t)$, 计算开销大, 正比于 $O(\text{词汇表大小} \times \text{特征维数})$

$$P(y_{t+1} = k | y_t) = \frac{h_t^T e_k}{\sum_{j=1}^V h_t^T e_j}$$

例如, 词汇表 $V = 10^4 \sim 10^6$, $h_t, e_k \in \mathbb{R}^d$, 特征维数 $d = 250 \sim 1024$

提出随机场语言模型 (TRF)



- **首次**将无向图模型的应用从定维情形扩展到序列情形，为语言模型乃至一般的序列建模打开一条**新思路**
- 重要意义在于，新模型
 - 有效地建模**上下文**
 - 通过**全局归一化**，带来似然计算效率的显著提升 (**38x**)
- ACL2015 (计算语言学), ASRU2017 (语音识别与理解), ICASSP2018 (信号处理), SLT2018 (口语处理), **T-PAMI2018** (人工智能顶级期刊, 影响因子9.455), **清华大学优秀博士学位论文2018**。

Google one-billion-word benchmark

模型	错误率 (%)	参数个数 (兆)	似然计算时间
N-Gram	6.13	133	0.491 s
LSTM-2x1024	5.55	191	0.909 s
mix-TRF	5.28 ↓	216	0.024 s ↓

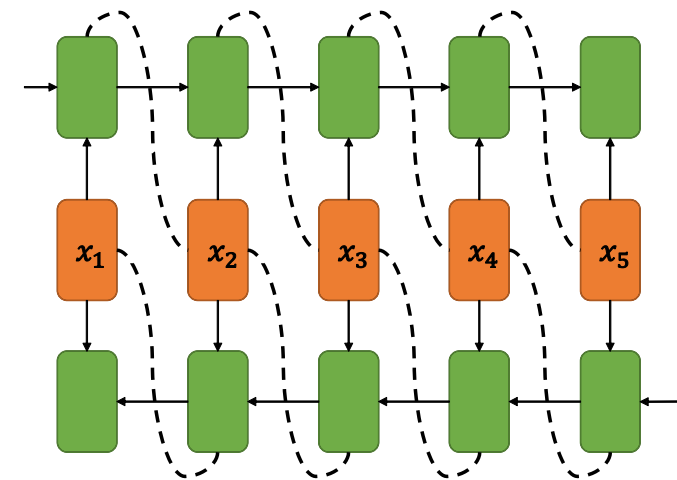
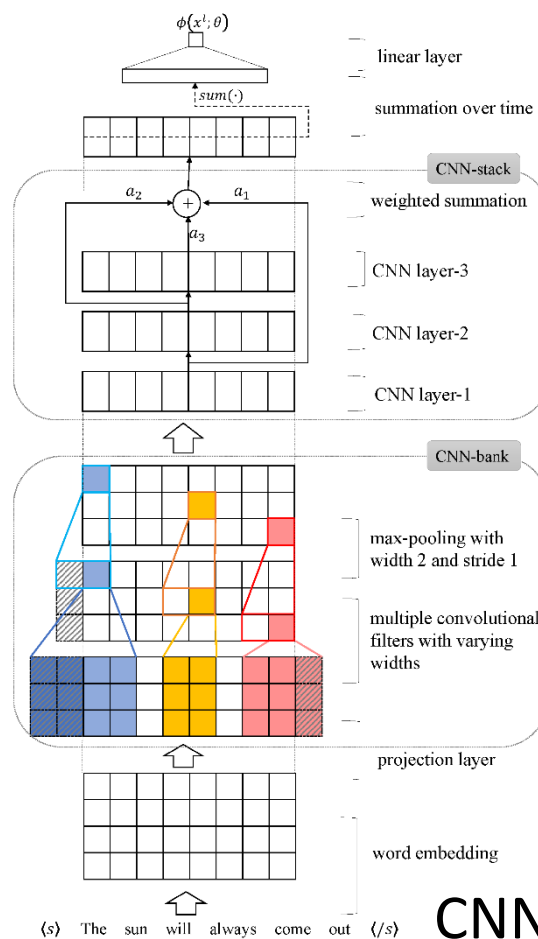
5% 38x

Trans-dimensional Random Field (TRF) - model definition

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

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



BLSTM features

CNN features

Review the development of TRF LMs

- Maximum-likelihood training $\nabla_{\lambda} = E_{\tilde{p}(l, x^l)} [f(x^l)] - E_{p(l, x^l; \lambda)} [f(x^l)]$

Expectation under
empirical distribution $\tilde{p}(l, x^l)$

Expectation under
model distribution $p(l, x^l; \lambda)$

ACL-2015

- Discrete features

TPAMI-2018

- Augmented stochastic approximation (**AugSA**) for model training

ASRU-2017

- Potential function as a deep CNN.
- Model training by **AugSA plus JSA** (joint stochastic approximation)

ICASSP-2018

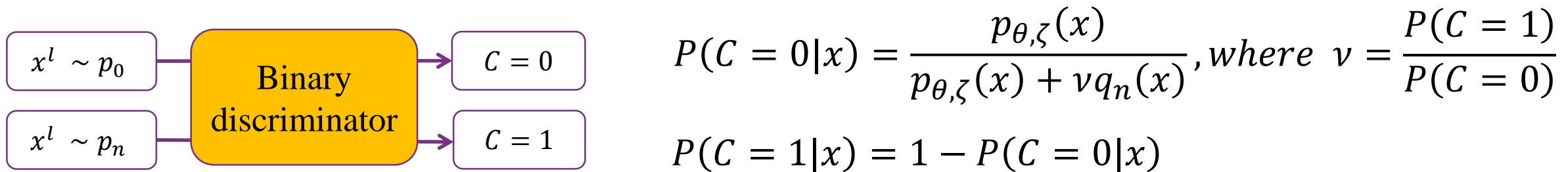
- Use LSTM on top of CNN
- **NCE** is introduced to train TRF LMs

SLT-2018

- Simplify the potential definition by using only Bidirectional LSTM
- Propose **Dynamic NCE** for improved model training

TRF - 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).

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		

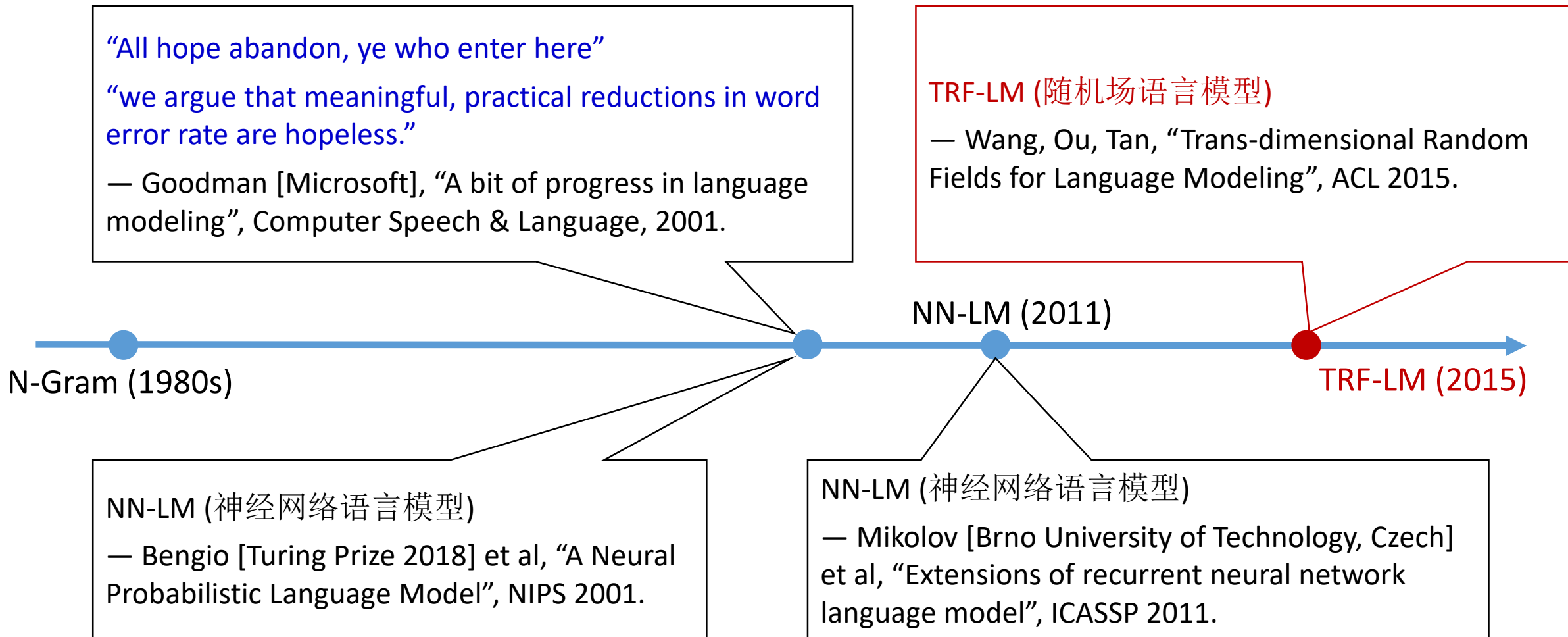
Annotations: A red arrow points from 6.13 to 5.55 with "5%" next to it. A red arrow points from 191 to 324 with "33%" next to it. A red arrow points from 0.909 to 0.024 with "38x" next to it.

开源SPMILM toolkit

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



随机场语言模型 (TRF) - 总结



Summary

- We need probabilistic models, besides neural nets.
- CRF-based end-to-end speech recognition

基于条件随机场的高效端到端语音识别

- ✓ ICASSP 2019
- ✓ **Data efficient**

- Random field approach to language modeling

随机场语言模型

- ✓ ACL Long Paper 2015, ASRU 2017, T-PAMI 2018, ICASSP 2018, SLT 2018
- ✓ **Representational efficient, computational efficient**

非常感谢聆听！

