Mixed TRF LMs

Integrating Discrete and Neural Features via Mixed-Feature Trans-Dimensional Random Field Language Models

Silin Gao[1], Zhijian Ou[1], Wei Yang[2], Huifang Xu[3]

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

[2]State Grid Customer Service Center

[3]China Electric Power Research Institute

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

Presented at International Conference on Acoustics, Speech and Signal Processing (ICASSP) 2020

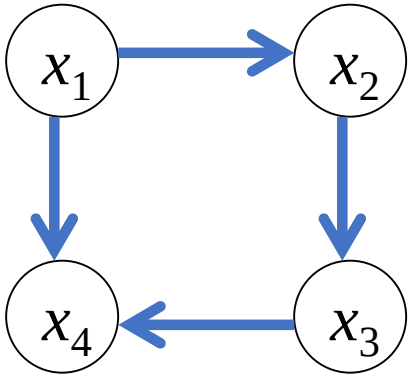
Content

1. Introduction
   - Related Work
   - Motivation
2. Mixed TRF LMs
   - Definition
   - Training
3. Experiments
   - PTB
   - Google one-billion word
4. Conclusions

Introduction

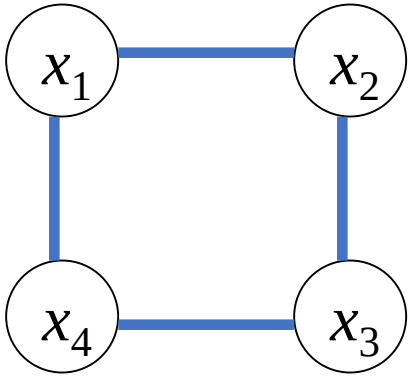
- Language Modeling
  - For the word sequence $\boldsymbol{x} \triangleq x_1 x_2 \cdots x_l$, determine the joint probability $p(\boldsymbol{x})$
- Directed Graphical Language Models
  - Self-normalized, modeling conditional probabilities
  - e.g. N-gram language models, Neural network (NN) based language models (e.g. RNN/LSTM LMs)

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



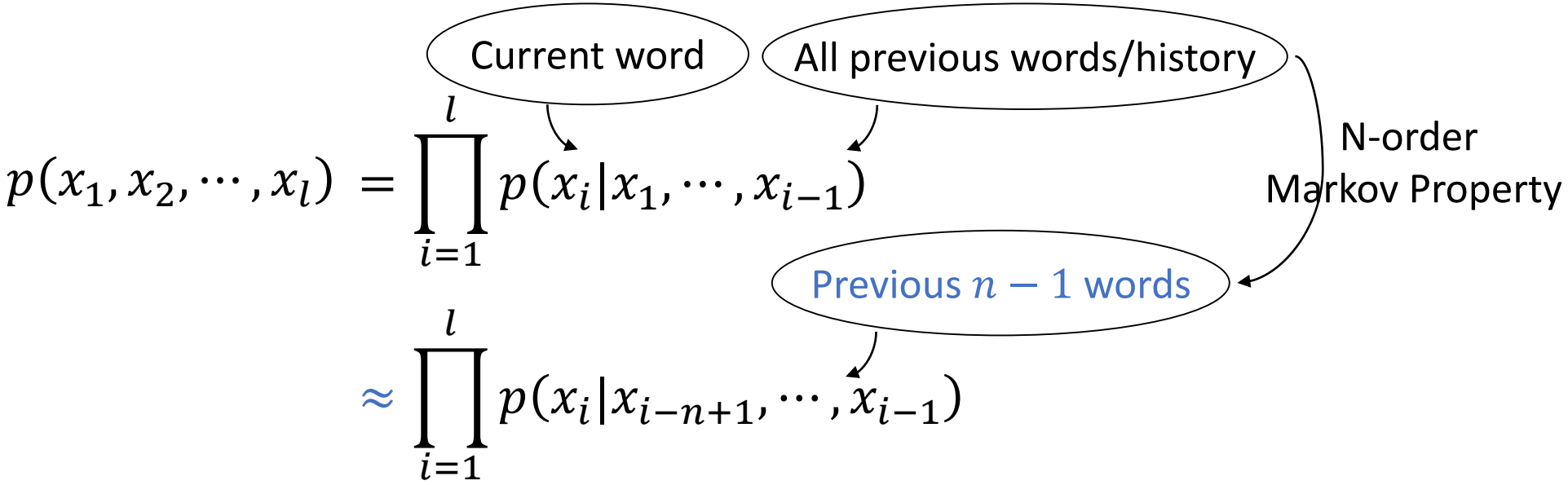
- Undirected Graphical Language Models
  - Involves the normalizing constant Z, potential function Φ
  - e.g. Trans-dimensional random field language models (TRF LMs)

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



Related Work: N-gram LMs

- N-gram Language Models



$$p(x_1, x_2, \cdots, x_l) = \prod_{i=1}^{l} p(x_i | x_1, \cdots, x_{i-1})$$

$$\approx \prod_{i=1}^{l} p(x_i | x_{i-n+1}, \cdots, x_{i-1})$$

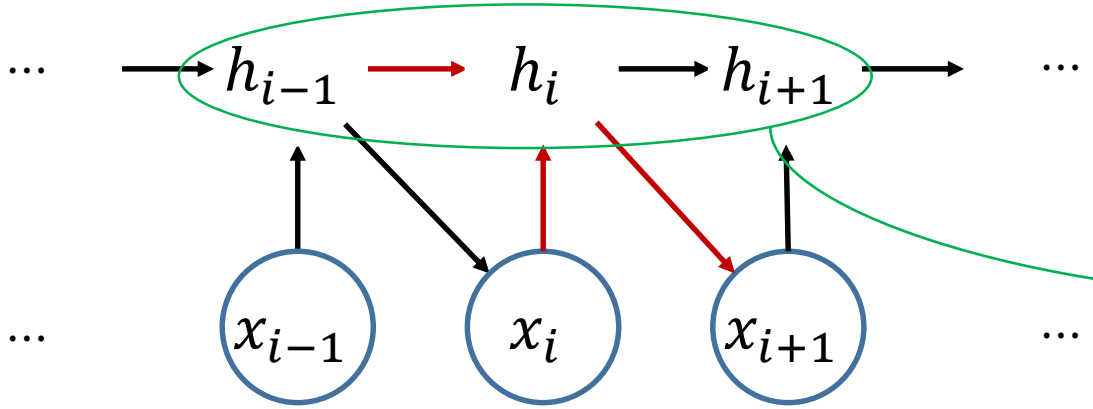
- Back-off N-gram LMs with Kneser-Ney Smoothing[1] (KNn LMs)
  - $p_{KN}(x_i|h) = (1 - \alpha_{KN}(h))\hat{p}(x_i|h) + \alpha_{KN}(h)p_{KN}(x_i|h')$

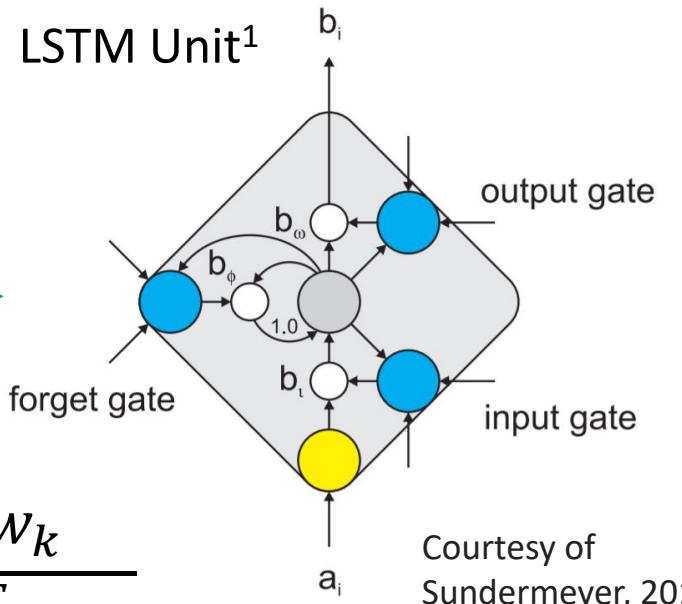
$$h = x_{i-n+1} \cdots x_{i-1} = x_{i-n+1}h'$$

[1]Stanley F Chen and Joshua Goodman, "An empirical study of smoothing techniques for language modeling," *Computer Speech & Language*, vol. 13, no. 4, pp. 359–394, 1999

Related Work: RNNs/LSTM LMs

- Recurrent Neural Nets (RNNs)/Long-Short Time Memory (LSTM) Language Models





Courtesy of Sundermeyer, 2012

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

☹.1 High computational cost of the Softmax output layer

e.g. $V = 10^4 \sim 10^6, w_k \in \mathbb{R}^{250\sim1024}$

☹.2 "Label bias" caused by the teacher-forcing training of the local conditional probabilities

[1]Hochreiter S , Schmidhuber J . "Long Short-Term Memory", *Neural computation*, 1997, 9(8):1735-1780.

Related Work: TRF LMs

- Trans-Dimensional Random Field (TRF) Language Models
  - Assume the sentences of length $\boldsymbol{l}$ are distributed as follows:

$$p_l(x^l;\eta) = \frac{1}{Z_l(\eta)} e^{V(x^l;\eta)},\ x^l \triangleq x_1 x_2 \cdots x_l$$

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

$V(x^l;\eta)$ is the potential function extracting the features of x^l;

η is the parameter of the potential function;

$Z_l(\eta) = \sum_{x^l} e^{V(x^l;\eta)}$ is the normalization constant.

Needed to be estimated

  - Assume length $\boldsymbol{l}$ is associated with prior probability $\boldsymbol{\pi_l}$.
    Therefore the pair $(\boldsymbol{l}, \boldsymbol{x^l})$ is jointly distributed as:

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

Related Work: TRF LMs

$$p(l, x^l; \eta) = \frac{\pi_l}{Z_l(\eta)} e^{V(x^l;\eta)},\ x^l \triangleq x_1 x_2 \cdots x_l$$

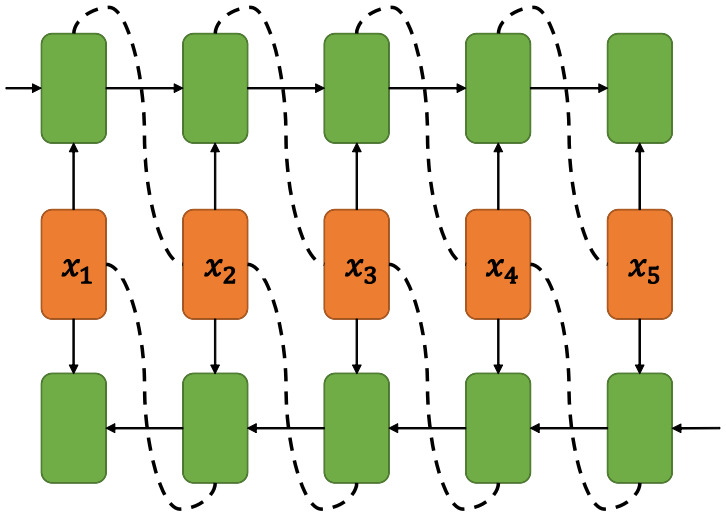
☺.1 Flexible: no acyclic and local normalization constraint

Discrete TRF:

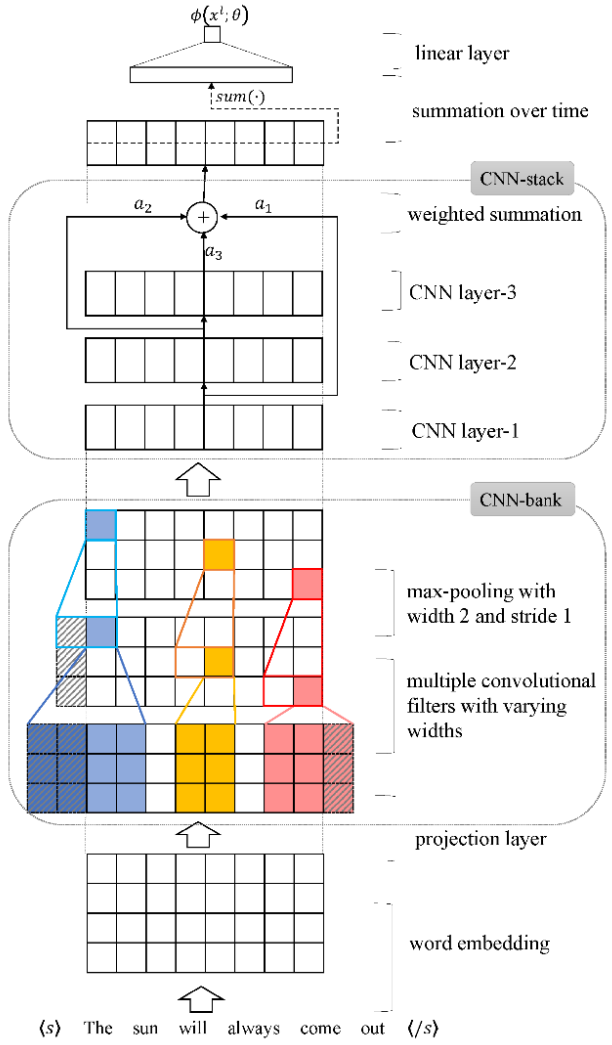
Type	Features
w	$(w_{-3}w_{-2}w_{-1}w_0)(w_{-2}w_{-1}w_0)(w_{-1}w_0)(w_0)$
c	$(c_{-3}c_{-2}c_{-1}c_0)(c_{-2}c_{-1}c_0)(c_{-1}c_0)(c_0)$
ws	$(w_{-3}w_0)(w_{-3}w_{-2}w_0)(w_{-3}w_{-1}w_0)(w_{-2}w_0)$
cs	$(c_{-3}c_0)(c_{-3}c_{-2}c_0)(c_{-3}c_{-1}c_0)(c_{-2}c_0)$
wsh	$(w_{-4}w_0)\ (w_{-5}w_0)$
csh	$(c_{-4}c_0)\ (c_{-5}c_0)$
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

Neural TRF:



Bi-LSTM features



CNN features

☺.2 Avoid high computational cost of the Softmax and "label bias"

- The state-of-the-art Neural TRF LMs perform as good as LSTM LMs, and are computationally more efficient in inference (computing sentence probabilities)

Related Work: TRF LMs

- The development of TRF LMs

ACL-2015 TPAMI-2018	• Discrete features • 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 • Noise Contrastive Estimation (**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

Motivation

- 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. ☹

[1]Xie 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.

[2]Bin 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.

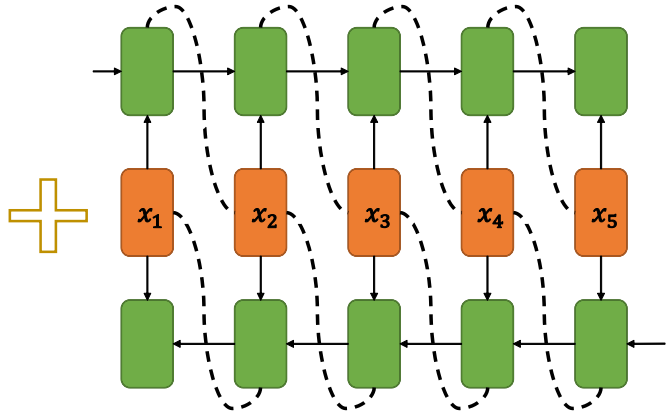
Motivation

TRF LMs: $p(l, x^l; \eta) = \frac{\pi_l}{Z_l(\eta)} e^{V(x^l, \eta)}, \; x^l \triangleq x_1 x_2 \cdots x_l$

☺.1 TRF LMs are flexible to support both discrete and neural features

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



Neural features

Achieve feature integration in an optimal single-step model construction! (Mixed-feature TRF)

☺.2 Lower the non-convexity

- Speed up convergence and reduce training time

☺.3 Complementary strength in language modeling

- Further improve the performance of TRF LMs by using diversified features

Content

1. Introduction
   - Related Work
   - Motivation
2. Mixed TRF LMs
   - Definition
   - Training
3. Experiments
   - PTB
   - Google one-billion word
4. Conclusions

Mixed TRF LMs: Definition

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

Discrete n-gram features, with parameter λ:

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

N: the total number of types of n-grams

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

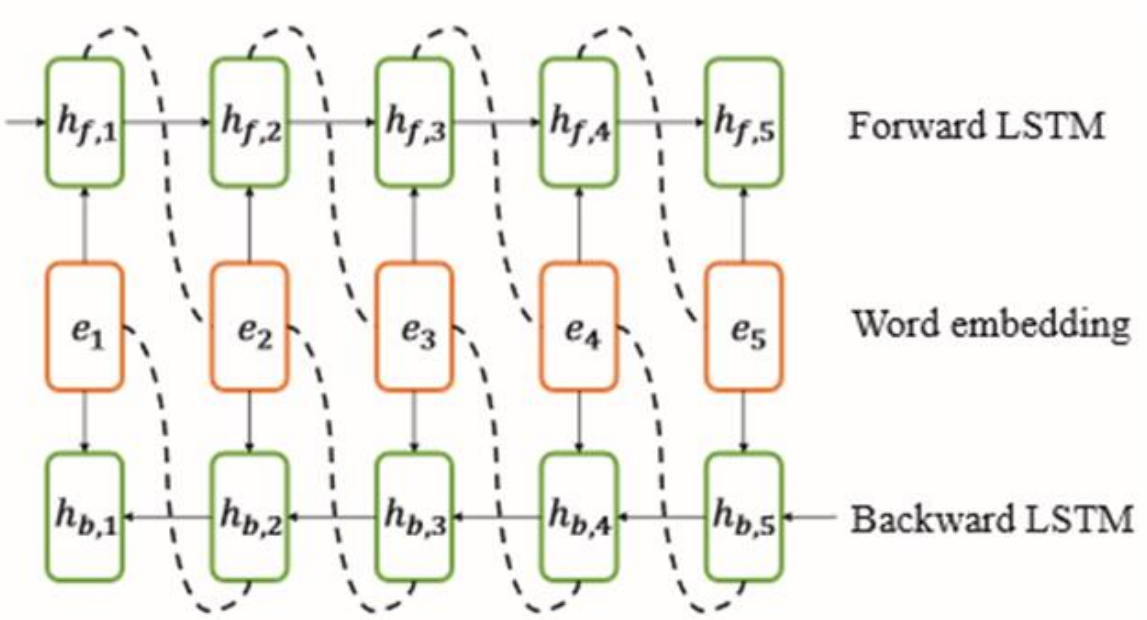
where c is the count of the kth n-gram type in x^l

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

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

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

Neural network features, with parameter θ



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

Mixed TRF LMs: Training, Noise Contrastive Estimation

- Treat $\log Z_l(\eta)$ as a parameter ζ_l and rewrite

$$p(l, x^l; \eta) = \frac{\pi_l}{Z_l(\eta)} e^{V(x^l, \eta)} \longrightarrow p(x; \xi) = \pi_l e^{V(x^l, \eta) - \zeta_l}, x = (l, x^l), \xi = (\eta, \zeta)$$

- Introduce a **noise distribution** $q_n(x)$, and consider a binary classification

$$P(C = 0|x) = \frac{p(x; \xi)}{p(x; \xi) + \nu q_n(x)}, where\ \nu = \frac{P(C = 1)}{P(C = 0)}$$

$$P(C = 1|x) = 1 - P(C = 0|x)$$

- Noise Contrastive Estimation (NCE):

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

☹ Reliable NCE needs a large $\nu \approx 20$; Overfitting.

Dynamic-NCE[1] in Wang & Ou, SLT 2018.

[1]Bin Wang and Zhijian Ou, "Improved training of neural trans-dimensional random field language models with dynamic noise-contrastive estimation," in *2018 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2018, pp. 70–76.

Content

1. Introduction
   - Related Work
   - Motivation
2. Mixed TRF LMs
   - Definition
   - Training
3. Experiments
   - PTB
   - Google one-billion word
4. Conclusions

Experiments: n-best list rescoring

- Two sets of experiments over two training datasets of different scales
  - **Penn Treebank (PTB) dataset**:

    16K sentences, 10K vocabulary (after preprocessing)
  - **Google one-billion-word dataset**:

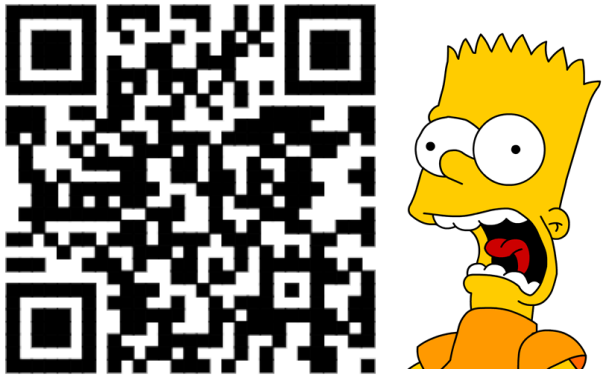
    31M sentences, 568K vocabulary (after cutting off words counting less than 4)

- Test set for LM n-best list rescoring
  - **Wall Street Journal (WSJ) '92 dataset**:

    330 sentences, each corresponds to a 1000-best list

- Implemented with Tensorflow

Open-source: https://github.com/thu-spmi/SPMILM



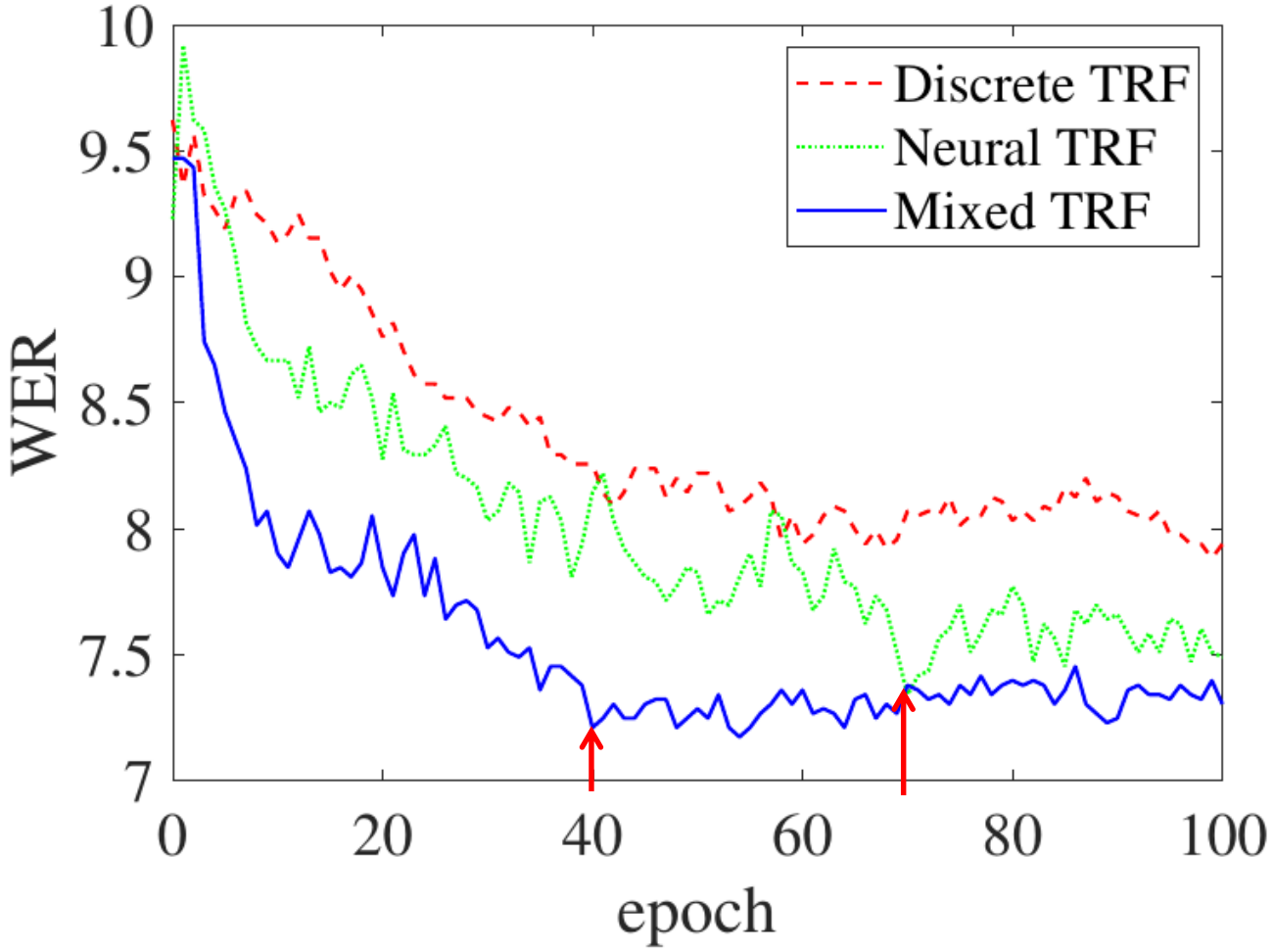
Experiments: PTB dataset

Model	PPL	WER (%)	#param (M)	Training time	Inference time
KN5	141.2	8.78	2.3	22 seconds	0.06 seconds
LSTM-2×1500	78.7	7.36	66.0	23.6 hours	9.09 seconds
Discrete TRF	~128	8.37	2.3	7.28 hours	0.11 seconds
Neural TRF	~75	7.34	2.6	22.1 hours	0.08 seconds
Mixed TRF	~69	7.17	4.9	18.2 hours	0.12 seconds

- Compared to the LSTM-2×1500, Mixed TRF achieves a 2.6% relative reduction on word error rate (WER), with 77.1% training time and only 7.4% parameters.
- Mixed TRF is 76x faster in inference (rescoring sentences) than the LSTM-2×1500.
- Compared to the state-of-the-art Neural TRF, Mixed TRF achieves a 2.3% relative reduction on word error rate (WER), with 82.4% training time, and comparable parameter size and inference speed.

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.

Experiments: PTB dataset

More rescoring results of various interpolated LMs:

Model	WER (%)
Mixed TRF	7.17
LSTM-2×1500 + KN5	7.47
Neural TRF + KN5	7.30
LSTM-2×1500 + Discrete TRF	7.15
Neural TRF + Discrete TRF	7.17
LSTM-2×1500 + Neural TRF	7.01
LSTM-2×1500 + Neural TRF + KN5	6.89
LSTM-2×1500 + Mixed TRF	6.83
LSTM-2×1500 + Mixed TRF + KN5	6.82

"+" denotes the log-linear interpolation with equal weights

- Mixed TRF matches the best interpolated model combining a discrete-feature LM and a neural-feature LM together.
- Updating Neural TRF to Mixed TRF is beneficial in language model interpolations.

Experiments: Google one-billion-word dataset

Model	PPL	WER (%)	#param (M)	Training time	Inference time
KN5	94.5	6.13	133	2.48 hours	0.491 seconds
LSTM-2×1024	72.7	5.55	191	144 hours	0.909 seconds
Discrete TRF	~86	6.04	102	131 hours	0.022 seconds
Neural TRF	~72	5.47	114	336 hours	0.017 seconds
Mixed TRF	~68	5.28	216	297 hours	0.024 seconds

Note: To reduce parameter size and speed up inference, we adopt a small-scale LSTM LM, and apply adaptive softmax strategy[1].

- Compared to the LSTM-2×1024 with adaptive softmax, Mixed TRF achieves a 4.9% relative reduction on word error rate (WER) and a 38x inference speed, though having a bit more parameters and longer training time.
- Compared to the state-of-the-art Neural TRF, Mixed TRF achieves a 3.5% relative reduction on word error rate (WER) with 88.4% training time.
- The LM interpolation results are similar to those on PTB.

Results of various interpolated LMs:

Model	WER (%)
Mixed TRF	5.28
LSTM-2×1024 + KN5	5.38
Neural TRF + KN5	5.51
LSTM-2×1024 + Discrete TRF	5.31
Neural TRF + Discrete TRF	5.27
LSTM-2×1024 + Neural TRF	5.25
LSTM-2×1024 + Neural TRF + KN5	5.06
LSTM-2×1024 + Mixed TRF	5.02
LSTM-2×1024 + Mixed TRF + KN5	4.99

[1]Edouard Grave, Armand Joulin, Moustapha Cissé, Hervé Jégou, et al., "Efficient softmax approximation for gpus," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 1302–1310.

Content

1. Introduction
   - Related Work
   - Motivation
2. Mixed TRF LMs
   - Definition
   - Training
3. Experiments
   - PTB
   - Google one-billion word
4. Conclusions

Conclusions

- We propose a mixed-feature TRF LM and demonstrate its advantage in integrating discrete and neural features.
- The Mixed TRF LMs trained on PTB and Google one-billion datasets achieve strong results in n-best list rescoring experiments for speech recognition.
  - Mixed TRF LMs outperform all the other single LMs, including N-gram LMs, LSTM LMs, Discrete TRF LMs and Neural TRF LMs;
  - The performance of Mixed TRF LMs matches the best interpolated model, and with simplified one-step training process and reduced training time;
  - Interpolating Mixed TRF LMs with LSTM LMs and N-gram LMs can further improve rescoring performance and achieve the lowest word error rate (WER).
- Next: Apply Mixed TRF LMs to one-pass ASR.

Thanks for your attention !

Silin Gao[1], Zhijian Ou[1], Wei Yang[2], Huifang Xu[3]

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

[2]State Grid Customer Service Center

[3]China Electric Power Research Institute

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