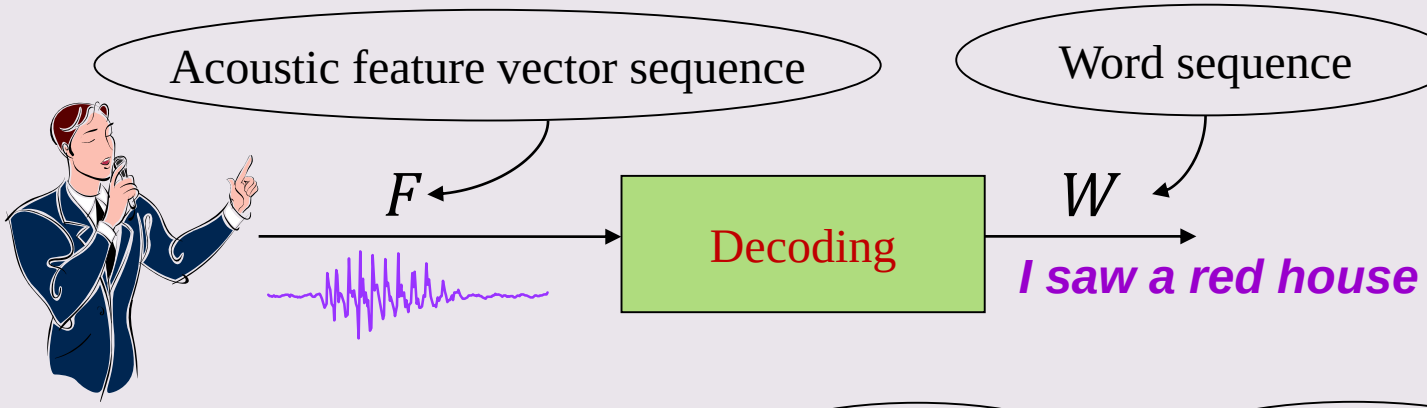
Learning Trans-dimensional Random Fields with Applications to Language Modeling

王斌 （导师：欧智坚）
Speech Processing and Machine Intelligence (SPMI) Lab
清华大学电子工程系

2017/10/08

Language models in Speech Recognition

Speech recognition is formulated as an optimization



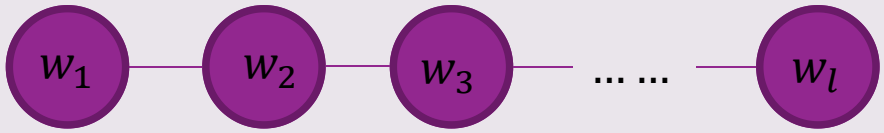
Acoustic model — **Language model**

$$\underset{W}{\operatorname{argmax}}\, P(W|F) = \underset{W}{\operatorname{argmax}} \frac{P(F|W)P(W)}{P(F)}$$

Bayes' theorem

Statistical language models

- a probability distribution over sequences of words



$$p(w_1, w_2, w_3, \ldots, w_l)$$

- Intuitively, we should assign the grammatical or common sentences a higher probability, and assign the ungrammatical or uncommon sentences a lower probability.

$$p(I\ saw\ a\ red\ house) \gg p(house\ red\ a\ saw\ I)$$

- Language modeling is essentially **sequence modeling**.

Directed graphical models

$$P(w_1, w_2, \cdots, w_l) = \prod_{i=1}^{l} P(w_i | w_1, \cdots, w_{i-1})$$

Current word

All previous words

$$\approx \prod_{i=1}^{l} P(w_i | \phi(w_1, \cdots, w_{i-1}))$$

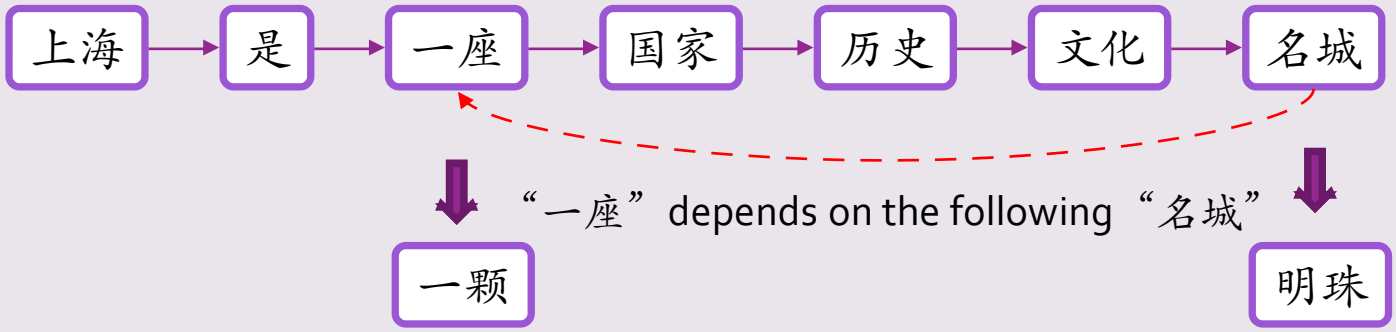
mapping function

n-gram	$\phi(w_1, \ldots, w_{i-1}) = w_{i-n+1}, \ldots, w_{i-1}$ Assuming the current words only depending on the previous $n-1$ words
RNN/LSTM	$\phi(w_1, \ldots, w_{i-1})$ is defined by a recurrent neural networks

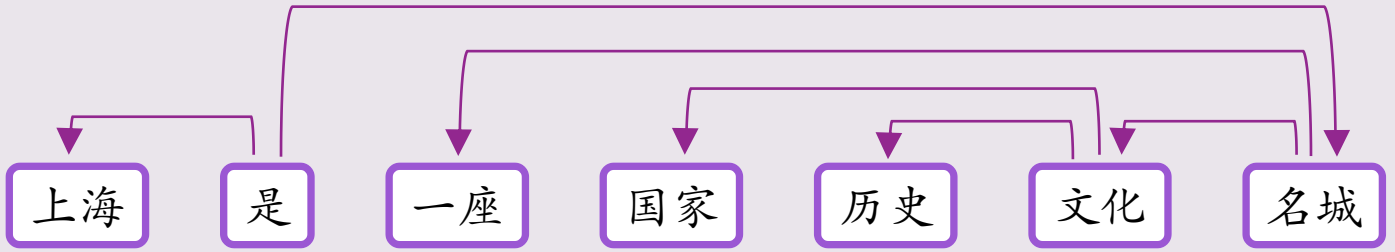
- The language models are commonly optimized to maximize the likelihood on the training set

Limitation of directed graphical models

- In human language, a word may depend on the following words.



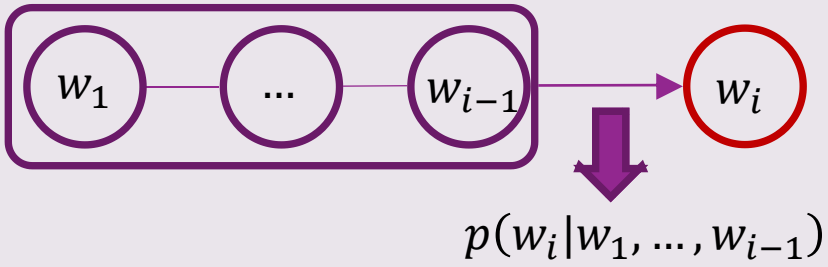
- Word dependence in a sentence is not a chain structure, but a tree structure. The following is a dependence tree.



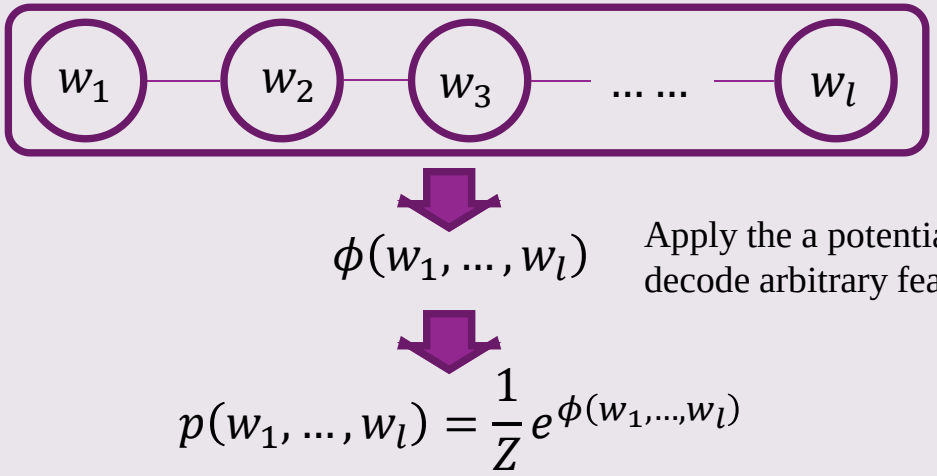
Undirected graphical modeling approaches

$$P(w_1, w_2, \cdots, w_l) = ?$$

- Dominant: Directed graphical approaches / Conditional models



- Alternative: Undirected graphical approaches / Random field models



Trans-dimensional random fields (TRFs)

- Assume the sentences of length $\boldsymbol{l}$ are distributed from an exponential family model:

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

x^l is a word sequence with length l;

$\lambda = (\lambda_1, \dots, \lambda_d)^T$ is the parameter vector; → need to be estimated

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

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

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

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

Feature definition

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

- $f(x^l)$ return the count of specific phrase observed in the input sentence x^l

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

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

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

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

  … …

- Only the phrases (including n-gram and skip n-gram of order ranging from 1 to 10) observed in the training set are added to the features.

Model Estimation (1) : parameter λ

Recall $$p_l(x^l;\lambda) = \frac{1}{Z_l}\exp\left\{\sum_{i=1}^{F}\lambda_i f_i(x^l)\right\}$$

- Objective 1: maximize the log-likelihood on the training set

$$\max_{\lambda,\pi_l} L = \frac{1}{n}\sum_{l=1}^{l_{max}}\sum_{x^l\in D_l}\log \pi_l p_l(x^l;\lambda)$$

D_l is the subset of the training set, containing all the sentence of length l, and $n_l = |D_l|, n = \sum_l n_l$

$$\pi_l = \frac{n_l}{n},$$

$$\frac{1}{n}\sum_{l=1}^{l_{max}}\sum_{x^l\in D_l} f(x^l) - \sum_{l=1}^{l_{max}}\frac{n_l}{n}E_{p_l(x^l;\lambda)}[f(x^l)] = 0$$

Expectation on training set

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

Model Estimation (2): normalization constants

Define $\zeta_l^* = \log \frac{Z_l}{Z_1}$ $l = 1, \ldots, l_{max}$

- Define ζ_l as hypothesized values of the true ζ_l^* and $\zeta = (\zeta_1, \ldots, \zeta_{l_{max}})$.

Rewrite $p_l(x^l; \lambda, \zeta) = \frac{1}{Z_1 e^{\zeta_l}} \exp\left\{\sum_{i=1}^{F} \lambda_i f_i(x^l)\right\}$ $l = 1, \ldots, l_{max}$

- The marginal probability of length l is :

$$p(l; \lambda, \zeta) = \sum_{x^l} p(l, x^l; \lambda, \zeta) = \frac{\pi_l e^{-\zeta_l + \zeta_l^*}}{\sum_{j=1}^{m} \pi_l e^{-\zeta_j + \zeta_j^*}} \xrightarrow{\text{if } \zeta_l = \zeta_l^*} \pi_l$$

- Objective 2:

$$\sum_{x^l} p(l, x^l; \lambda, \zeta) = \boldsymbol{\pi_l}, \quad \boldsymbol{l = 1, \ldots, l_{max}}$$

For more details, see
Z. Tan, "Optimally adjusted mixture sampling and locally weighted histogram analysis," *Journal of Computational and Graphical Statistics*, 2017.

Stochastic approximation (SA)

$$\sum_{l=1}^{l_{max}} \frac{n_l}{n} E_{p_l(x^l;\lambda)}[f(x^l)] = \frac{1}{n}\sum_{l=1}^{l_{max}} \sum_{x^l \in D_l} f(x^l),$$

estimate the parameter λ

Intractable!!

$$\sum_{x^l} p(l, x^l; \lambda, \zeta) = \pi_l = \frac{n_l}{n}, \quad l = 1, \ldots, l_{max}$$

estimate the normalization constants ζ

■ Introduction to Stochastic Approximation (SA)

Problem: The objective is to find a solution θ to $E_{Y\sim f(\cdot;\,\theta)}[H(Y;\theta)] = \alpha$, where $\theta \in R^d$, noisy observation $H(Y;\theta) \in R^d$

Method:

(1) Generate $Y_t \sim K(Y_{t-1}, \cdot\,; \theta_{t-1})$, a Markov transition kernel that admits $f(\cdot;\,\theta_{t-1})$ as the invariant distribution.

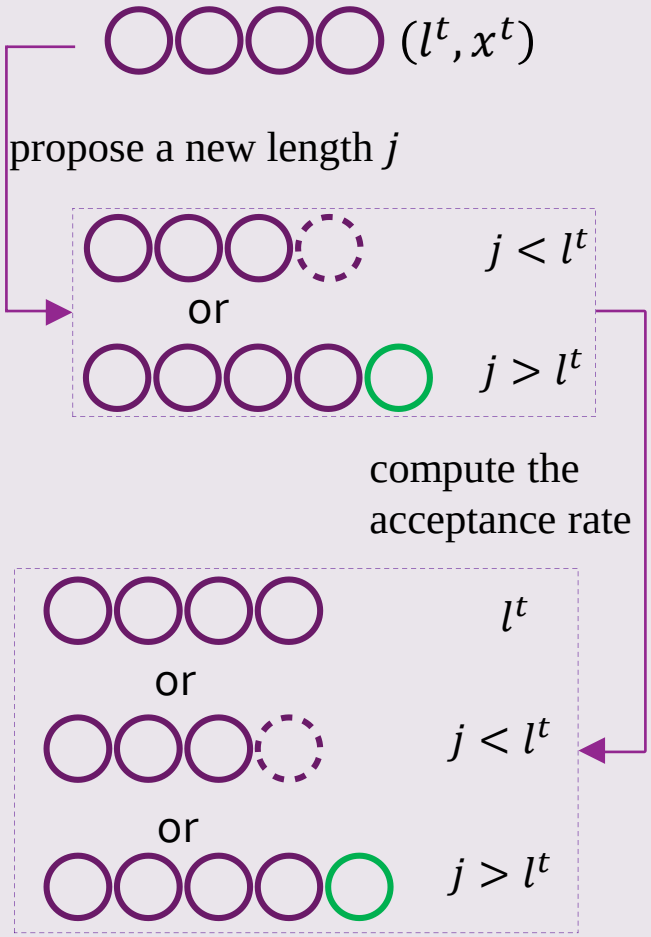
(2) Set $\theta_t = \theta_{t-1} + \gamma_t\{\alpha - H(Y_t; \theta_{t-1})\}$

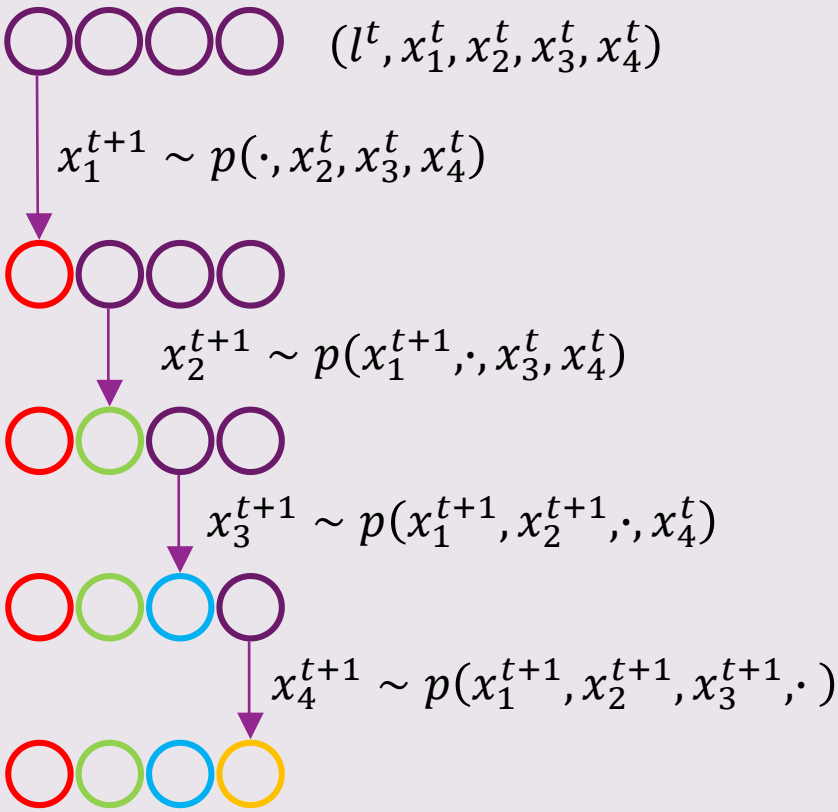
(l, x^l): sentences of varying lengths

Trans-dimensional mixture sampling

Step 1: local jump
- change the sequence length
- Metropolis-Hastings method

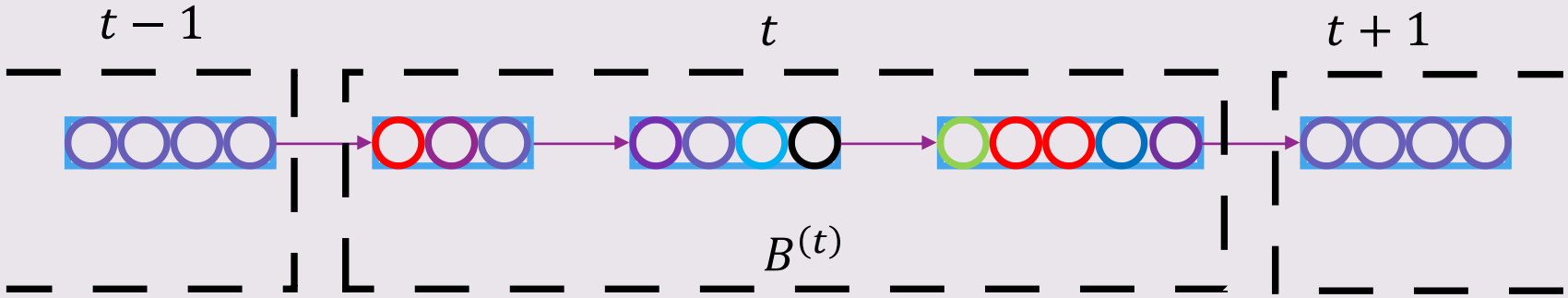
Step 2: Markov move
- change the values at each position
- Gibbs method





Augmented Stochastic Approximation

- Perform the ***trans-dimensional mixture sampling*** leaving $p(l, x^l; \lambda)$ as the stationary distribution, and get the sample set $B^{(t)}$. $(K = |B^{(t)}|)$



- Calculate the feature expectation on $B^{(t)}$ to update λ:

$$\lambda^{(t)} = \lambda^{(t-1)} + \gamma_{\lambda,t}\sigma^{-1}\left\{\frac{1}{n}\sum_{l=1}^{l_{max}}\sum_{x^l \in D_l} f(x^l) - \frac{1}{K}\sum_{(l,x^l)\in B^{(t)}} f(x^l)\right\}$$

Feature expectation on sample set $B^{(t)}$

the empirical variance of features

- Calculate the length expectation on $B^{(t)}$ to update ζ:

$$\zeta^{\left(t-\frac{1}{2}\right)} = \zeta^{(t-1)} + \gamma_{\zeta,t}\left\{\frac{\delta_1(B^{(t)})}{\pi_1}, \dots, \frac{\delta_m(B^{(t)})}{\pi_m}\right\}$$

$$\zeta^{(t)} = \zeta^{\left(t-\frac{1}{2}\right)} - \zeta_1^{\left(t-\frac{1}{2}\right)}$$

$\delta_l(B^{(t)})$ denotes the frequency of length l in sample set $B^{(t)}$

Experiment 1:
Speech recognition on PTB-WSJ dataset

- LM dataset – Penn Treebank part of WSJ dataset
  - *Vocabulary : 10K words*
  - *Training data : 887K words, 42K sentences*
  - *Development data : 70K words*
- Test speech — WSJ'92 test set (330 utterances)
  - *by rescoring the 1000-best list（oracle WER=0.93）*
- Language models
  - *5-gram LMs with modified Kneser-Ney smoothing (KN5)*
  - *Recurrent neural network (RNN) with 250 hidden units*
  - *Long-Short Term Memory (LSTM) with 2 hidden layers and 250 units for each layer*
  - *Trans-dimensional random field (TRF) language models*

Experiment 1: Speech recognition on PTB-WSJ dataset

Tsinghua University

Models	WER(± std. dev.)	PPL(± std. dev.)	#feat
KN5	8.78	284.4	2.3M
RNN	7.91	257.6	5.1M
LSTM(2layer*250)	7.87	306.3	6.0M
TRF	7.898±0.07	253.95±8.84	6.9M
RNN+KN5	7.996		
LSTM+KN5	8.089		
LSTM+TRF	7.585±0.06		

*TRF results are means over 10 independent AugSA training runs, ± standard deviation.

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	$(c_{-9:-6}, c_0)\ (w_{-9:-6}, w_0)$

- TRF outperforms KN5 with 10% relative reduction.
- Results of TRF is close to RNN and LSTM. But TRF improves the rescoring efficiency, as it avoids the computation of Softmax. The rescoring times for the 1000-best list of one utterance are:
  - TRF: 0.16s (CPU used)
  - RNN: 40s (CPU used)
  - LSTM: 10s (GPU used)
- Interpolated TRF and LSTM achieves the lowest WER 7.585. The relative reduction is 13.6% over KN5 and 3.6% over LSTM, and 6.2% over LSTM+KN4

Experiment 2:
Speech Recognition on Google 1-billion dataset

- Dataset: Google 1-billion dataset
  - *Training set contains 99 files and each file contains about 8 million words.*
  - *The held-out set contains 50 files and each files contains about 160K words.*
- Configuration
  - *We increase the training set gradually from 8M to 32M words.*
  - *Both the developing set and the test set contain about 160K words.*
- Test speech — WSJ'92 test set (330 utterances)
  - *by rescoring the 1000-best list （oracle WER=0.93）*
- LMs
  - *KNn: n-gram LMs with Kneser-Ney smoothing*
  - *LSTM: with 2 hidden layers and each layer contains 250 units*

Experiment 2: Speech Recognition on Google 1-billion dataset

We increase the training set gradually from 8M to 32M words and the WERs are shown in the following figures.

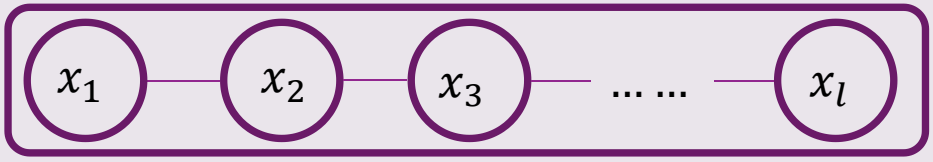
LMs	8M	16M	32M
KN5	8.24	8.26	7.66
RNN	7.47	7.73	7.73
LSTM	7.19	6.88	6.69
TRF	**7.08**	**7.17**	**7.10**
RNN+KN5	7.60	7.66	7.43
LSTM+KN5	7.14	6.82	6.59
RNN+TRF	7.08	7.27	6.93
LSTM+TRF	**6.76**	**6.72**	**6.33**

LSTM+TRF trained on 32M corpus achieves 6.33 WER, which represents relative reduction:

- 17.4% over KN5
- 5.4% over LSTM
- 4.0% over LSTM+KN5

Neural trans-dimensional random fields

- Incorporate the neural network to automatic extract features

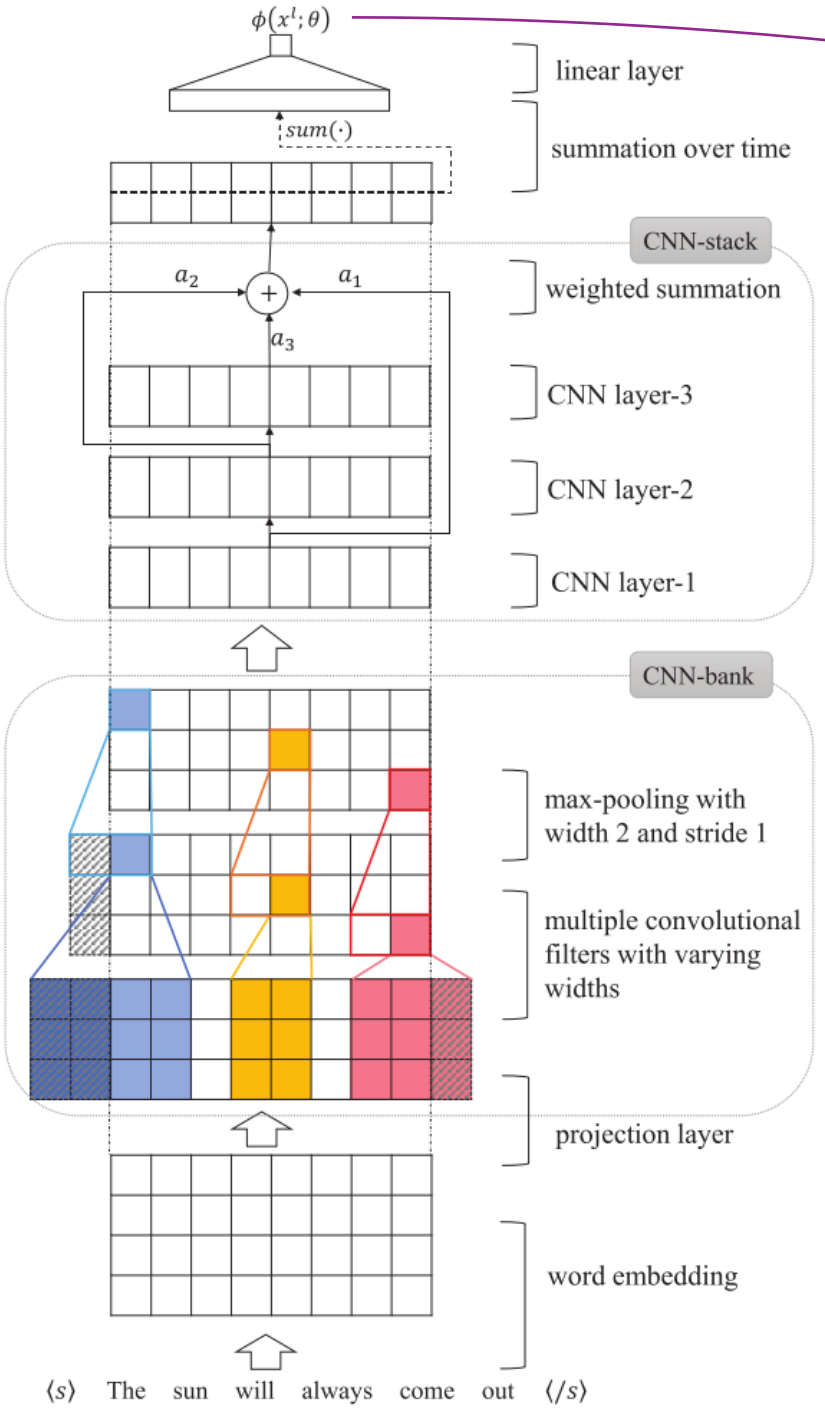


A neural network (with parameter θ), whose output is a read number.

$$\phi(x_1, \dots, x_l; \theta)$$

$$p_l(x_1, \dots, x_l; \theta) = \frac{1}{Z_l} e^{\phi(x_1, \dots, x_l; \theta)}$$

- The log-likelihood is not a convex function with respect to the parameter θ



$$p_l(x^l;\theta) = \frac{1}{Z_l} e^{\phi(x^l;\theta)}$$

Model	WER(%)	#param (M)	inference time (seconds)
KN5	8.78	2.3	0.06
LSTM-2x200	7.96	4.6	6.36
LSTM-2x650	***7.66***	***19.8***	***6.36***
LSTM-2x1500	7.36	66.0	9.09
discrete TRF	7.92	6.4	0.16
neural TRF	***7.60***	***4.0***	***0.40***
LSTM-2x1500 + neural TRF	**7.17**		

Our neural TRF perform slightly better than LSTM LMs with only ***1/5 parameters*** and 16x faster inference efficiency.

Bin Wang, and Zhijian Ou. "Language modeling with Neural trans-dimensional random fields." *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2017.

Summary and future works

- Contributions
  - Propose the trans-dimensional random field (TRF) models and develop the AugSA training algorithm.
  - Evaluate the TRF and AugSA on speech recognition task. TRFs outperform the n-gram LMs and perform competitive with NN-based LMs but being much faster in calculating sentence probabilities.

- Future works
  - Accelerate the mixture sampling.
  - Incorporate other features, such as tree-structure features.
  - Investigate unsupervised training

清華大學 Tsinghua University

Thank You

- Bin Wang, Zhijian Ou, and Zhiqiang Tan. "Learning trans-dimensional random fields with applications to language modeling." *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2017.
- Bin Wang, and Zhijian Ou. "Language modeling with Neural trans-dimensional random fields." *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2017.