

Neural Networks for Natural Language Processing

Lili Mou

doublepower.mou@gmail.com

<http://sei.pku.edu.cn/~moull12>

Neural Networks for Natural Language Processing

Q: Why not “deep learning”?

A: Neural networks are not necessarily deep.

Lili Mou

doublepower.mou@gmail.com

<http://sei.pku.edu.cn/~moull12>

Neural Networks for Natural Language Processing

Q: What is a neural network?

A: A composite function, or just simply, a function.

Lili Mou

doublepower.mou@gmail.com

<http://sei.pku.edu.cn/~moull12>

Outline

- **Unsupervised Learning: Word Embeddings**
- Discriminative Sentence Models
- Natural Language Generation
- Conclusion and Discussion

Language Modeling

- One of the most fundamental problems in NLP.
- Given a corpus $\mathbf{w}=w_1w_2\dots w_t$, the goal is to maximize $p(\mathbf{w})$

Language Modeling

- One of the most fundamental problems in NLP.
- Given a corpus $\mathbf{w}=w_1w_2\dots w_t$, the goal is to maximize $p(\mathbf{w})$
- **Philosophical discussion:** Does “probability of a corpus/sentence” make sense?
 - Recognize speech
 - Wreck a nice beach
- All in all, NLP (especially publishing in NLP) is pragmatic.

Decomposition of the Joint probability

- $p(\mathbf{w}) = p(w_1)p(w_2|w_1)p(w_3|w_1w_2) \dots p(w_t|w_1w_2\dots w_{t-1})$

Decomposition of the Joint probability

- $p(\mathbf{w}) = p(w_1)p(w_2|w_1)p(w_3|w_1w_2) \dots p(w_t|w_1w_2\dots w_{t-1})$

Minor question:

- Can we decompose any probabilistic distribution into this form? Yes.
- Is it necessary to decompose a probabilistic distribution into this form? No.

Markov Assumption

- $p(\mathbf{w}) = p(w_1)p(w_2|w_1)p(w_3|w_1w_2) \dots p(w_t|w_1w_2\dots w_{t-1})$
 $\approx p(w_1)p(w_2|w_1)p(w_3|w_2) \dots p(w_t|w_{t-1})$

- A word is dependent only on its previous $n-1$ words and independent of its position,

I.e., provided with the previous $n-1$ words, the current word is independent of other random variables.

$$p(\mathbf{w}) \approx \prod_{t=1}^m p(w_t | \mathbf{w}_{t-n+1}^{t-1})$$

Multinomial Estimate

- Maximum likelihood estimation for a multinomial distribution is merely counting.

$$p(w_n | \mathbf{w}_1^{n-1}) = \frac{\#w_1^n}{\#w_1^{n-1}}$$

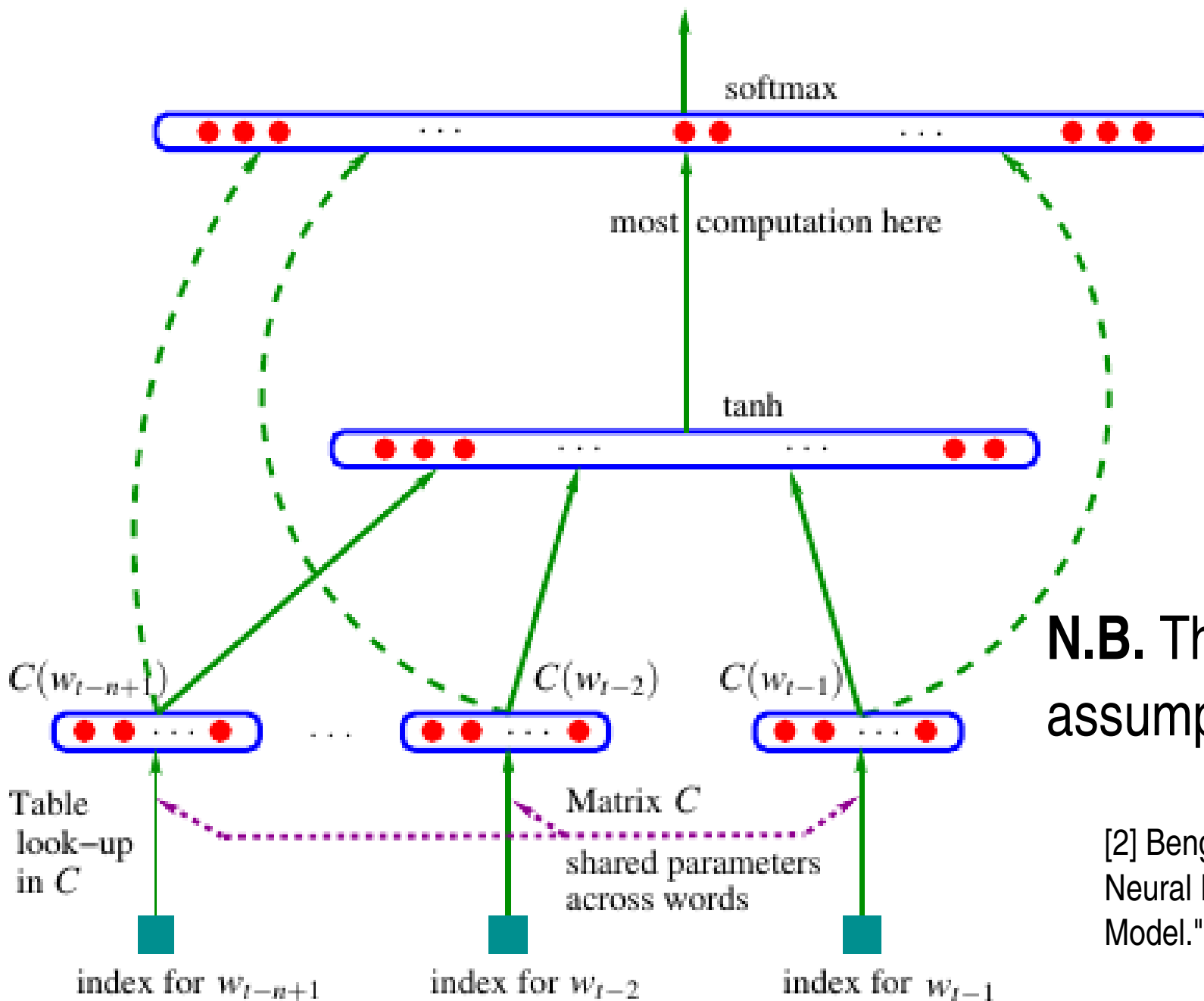
- Problems
 - #para grows exp. w.r.t. n
 - Even for very small n (e.g., 2 or 3), we come across severe data sparsity because of the Zipf distribution

Parameterizing LMs with Neural Networks

- Each word is mapped to a real-valued vector, called *embeddings*.
- Neural layers capture context information (typically previous words).
- The probability $p(w| \cdot)$ is predicted by a softmax layer.

Feed-Forward Language Model

$$i\text{-th output} = P(w_t = i \mid \text{context})$$

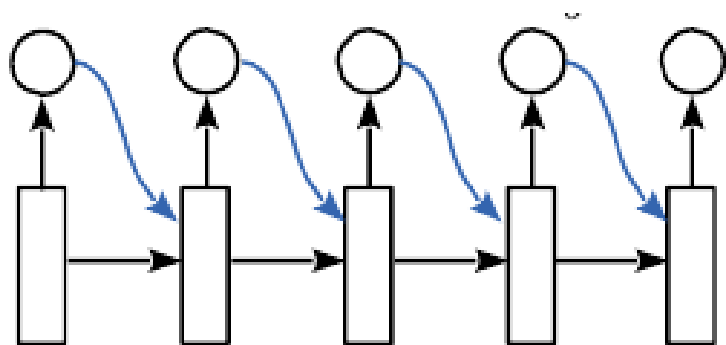


N.B. The Markov assumption also holds.

[2] Bengio, Yoshua, et al. "A Neural Probabilistic Language Model." JMLR. 2003.

Recurrent Neural Language Model

- RNN keeps one or a few hidden states
- The hidden states change at each time step according to the input



$$\begin{aligned} \mathbf{h}_t &= \text{RNN}(\mathbf{x}_t, \mathbf{h}_{t-1}) \\ &= f(W_{\text{in}}\mathbf{x}_t + W_{\text{hid}}\mathbf{h}_{t-1}) \end{aligned}$$

$$p(w_t | \mathbf{w}_0^{t-1}) \approx \text{softmax}(W_{\text{out}}\mathbf{h}_t)$$

- RNN directly parametrizes $p(\mathbf{w}) = \prod_{t=1}^m p(w_t | \mathbf{w}_1^{t-1})$
rather than $p(\mathbf{w}) \approx \prod_{t=1}^m p(w_t | \mathbf{w}_{t-n+1}^{t-1})$

Complexity Concerns

- Time complexity
 - Hinge loss [4]
 - Hierarchical softmax [5]
 - Noisy contrastive estimation [6]
- Model complexity
 - Shallow neural networks are still too “deep.”
 - CBOW, SkipGram [6]
 - Model compression [under review]

[4] Collobert R, Weston J, Bottou L, Karlen M, Kavukcuoglu K, Kuksa P. Natural language processing (almost) from scratch. JMLR, 2011.

[5] Mnih A, Hinton GE. A scalable hierarchical distributed language model. NIPS, 2009.

[6] Mikolov T, Chen K, Corrado G, Dean J. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781. 2013

The Role of Word Embeddings?

- Word embeddings are essentially a connectional weight matrix, whose input is a one-hot vector.
- Implementing by a look-up table is much faster than matrix multiplication.
- Each column of the matrix corresponds to a word.

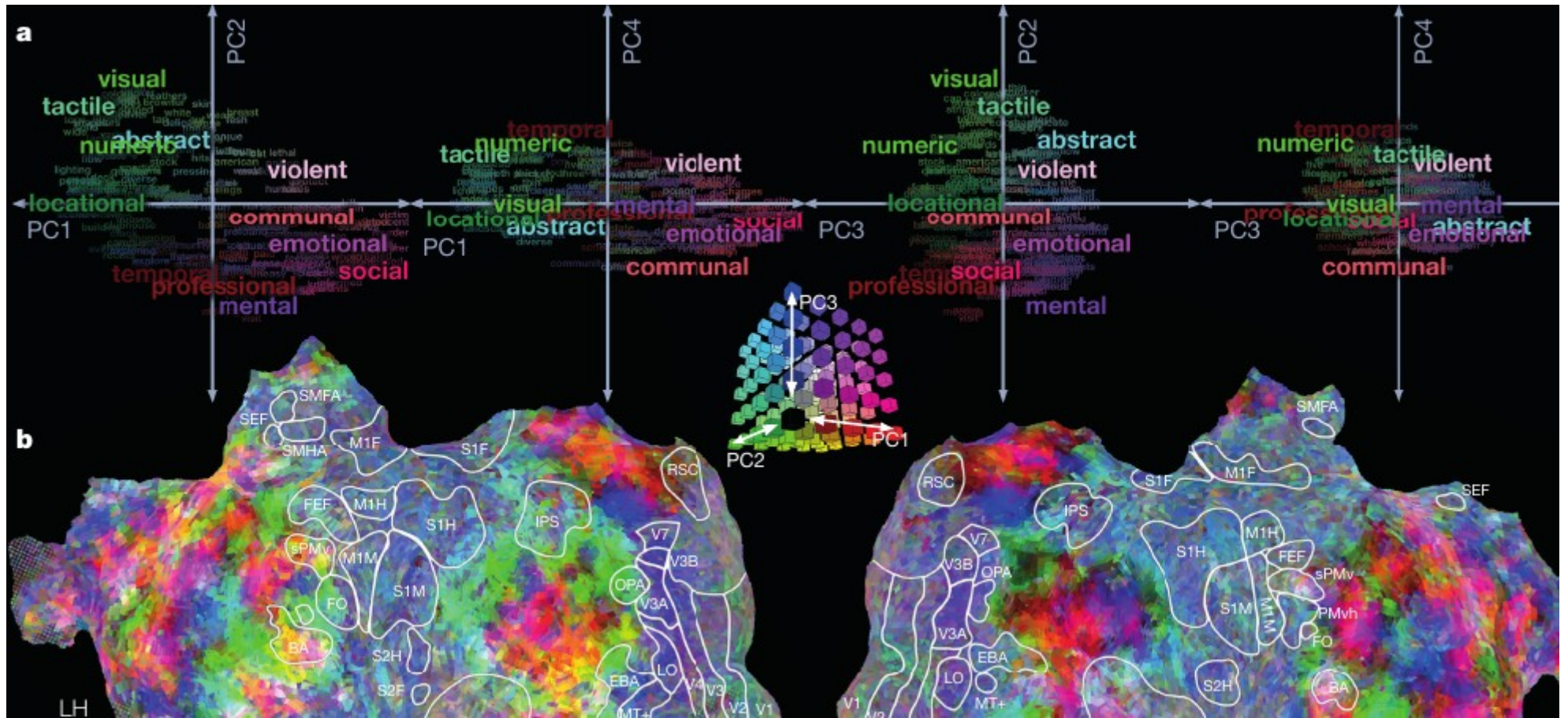
The diagram illustrates the process of retrieving a word embedding. On the left, a large left-facing square bracket contains a vertical blue bar representing a column in a matrix. Below this is the text "Embedding of word i retrieved by matrix-vector multiplication". To the right of the blue bar is a dot, followed by a large right-facing square bracket containing a vertical list of values: 0, 1, a dot, a dot, and 0. This represents a one-hot vector.

One-hot representation of word i (sparse)

How can we use word embeddings?

- Embeddings demonstrate the internal structures of words
 - Relation represented by vector offset
 - “man” – “woman” = “king” – “queen”
 - Word similarity
- Embeddings serve as the initialization of almost every supervised task
 - A way of pretraining
 - **N.B.:** may not be useful when the training set is large enough

Word Embeddings in our Brain



[7] Huth, Alexander G., et al. "Natural speech reveals the semantic maps that tile human cerebral cortex." *Nature* 532.7600 (2016): 453-458.

“Somatotopic Embeddings” in our Brain

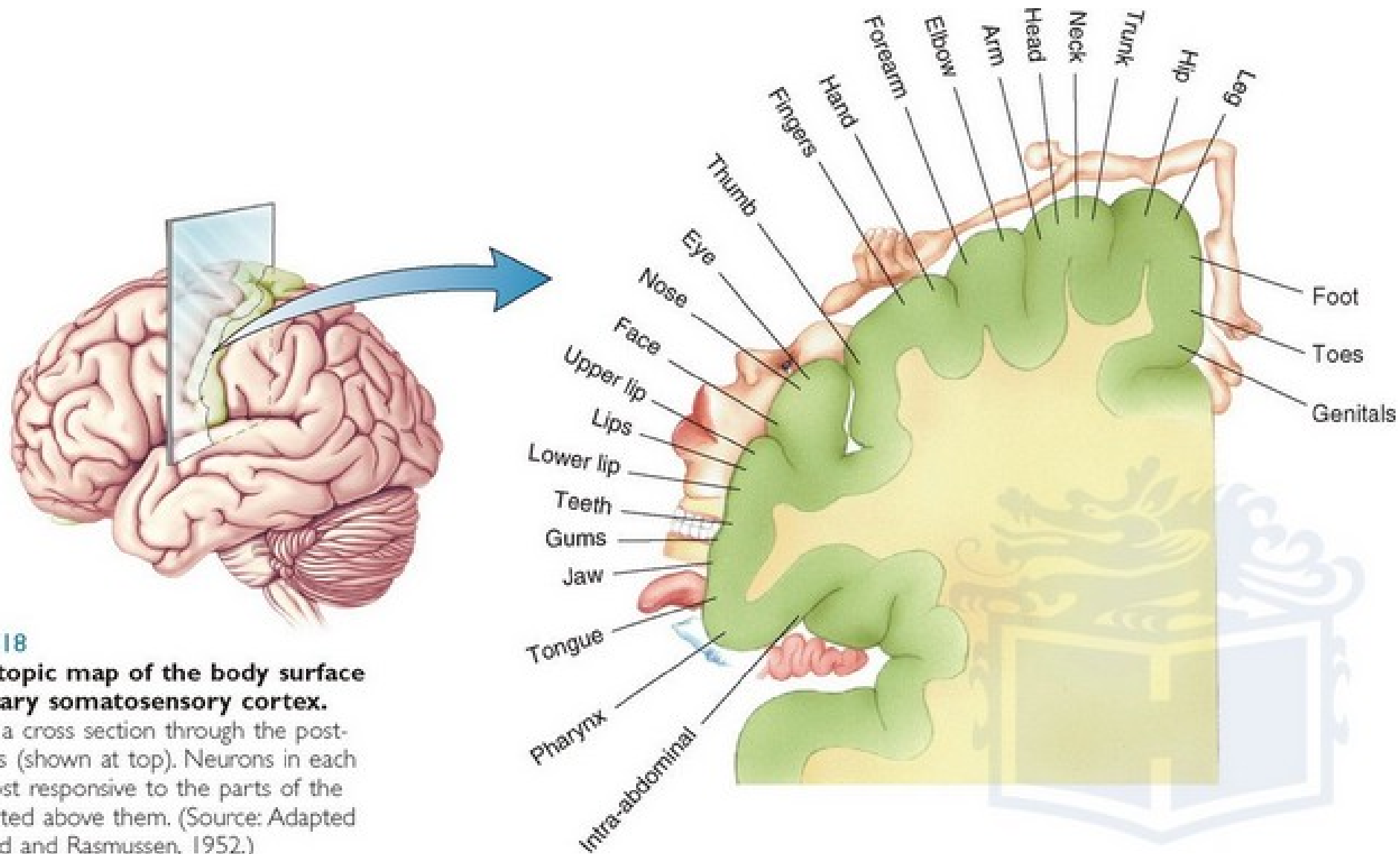


FIGURE 12.18

A somatotopic map of the body surface onto primary somatosensory cortex.

This map is a cross section through the post-central gyrus (shown at top). Neurons in each area are most responsive to the parts of the body illustrated above them. (Source: Adapted from Penfield and Rasmussen, 1952.)

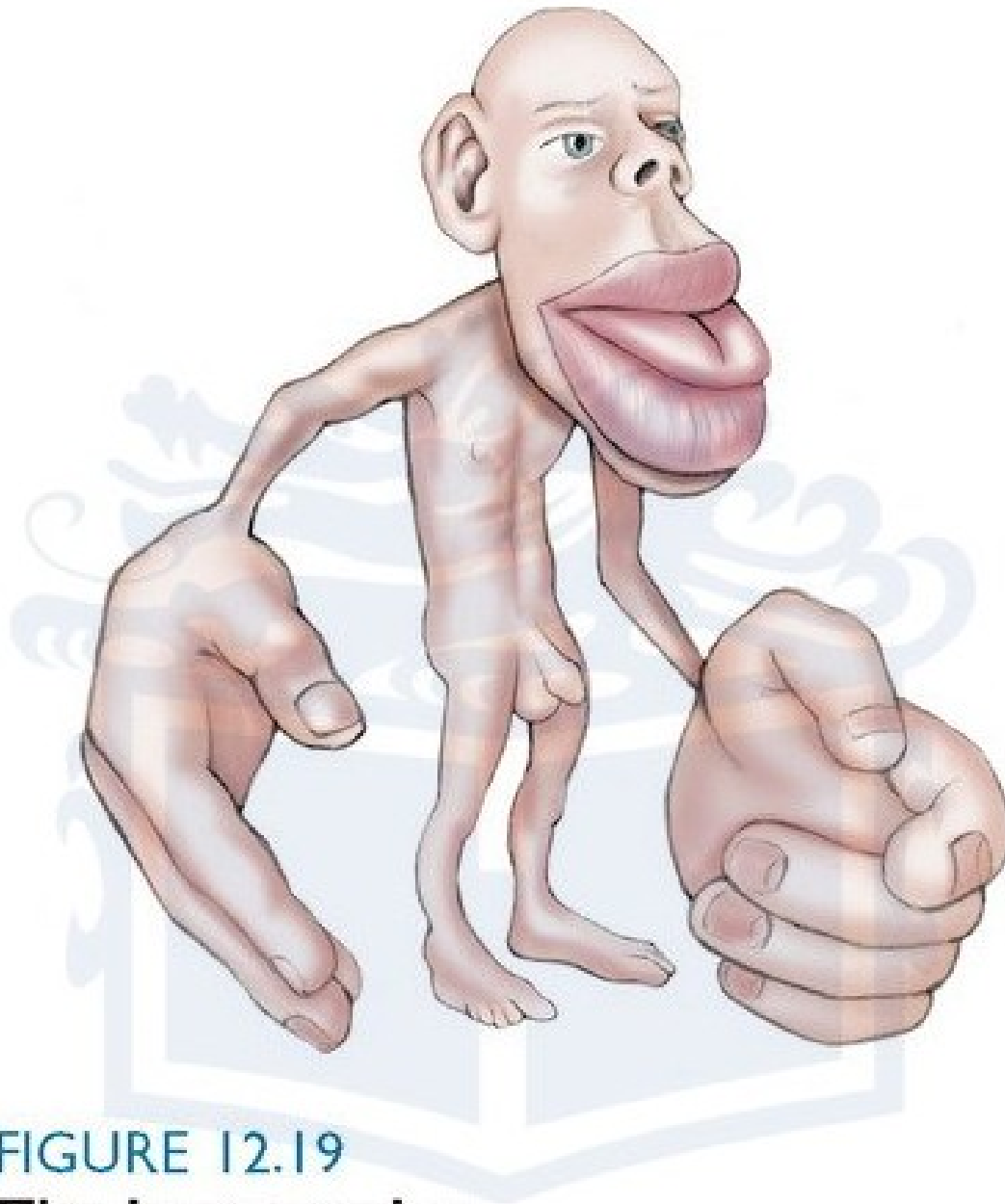
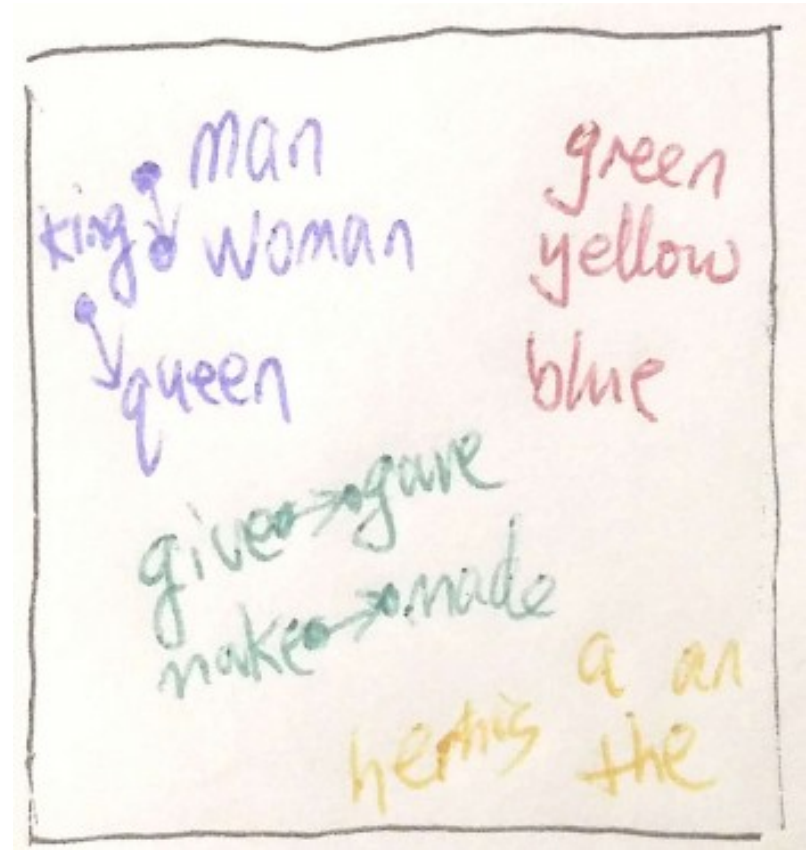
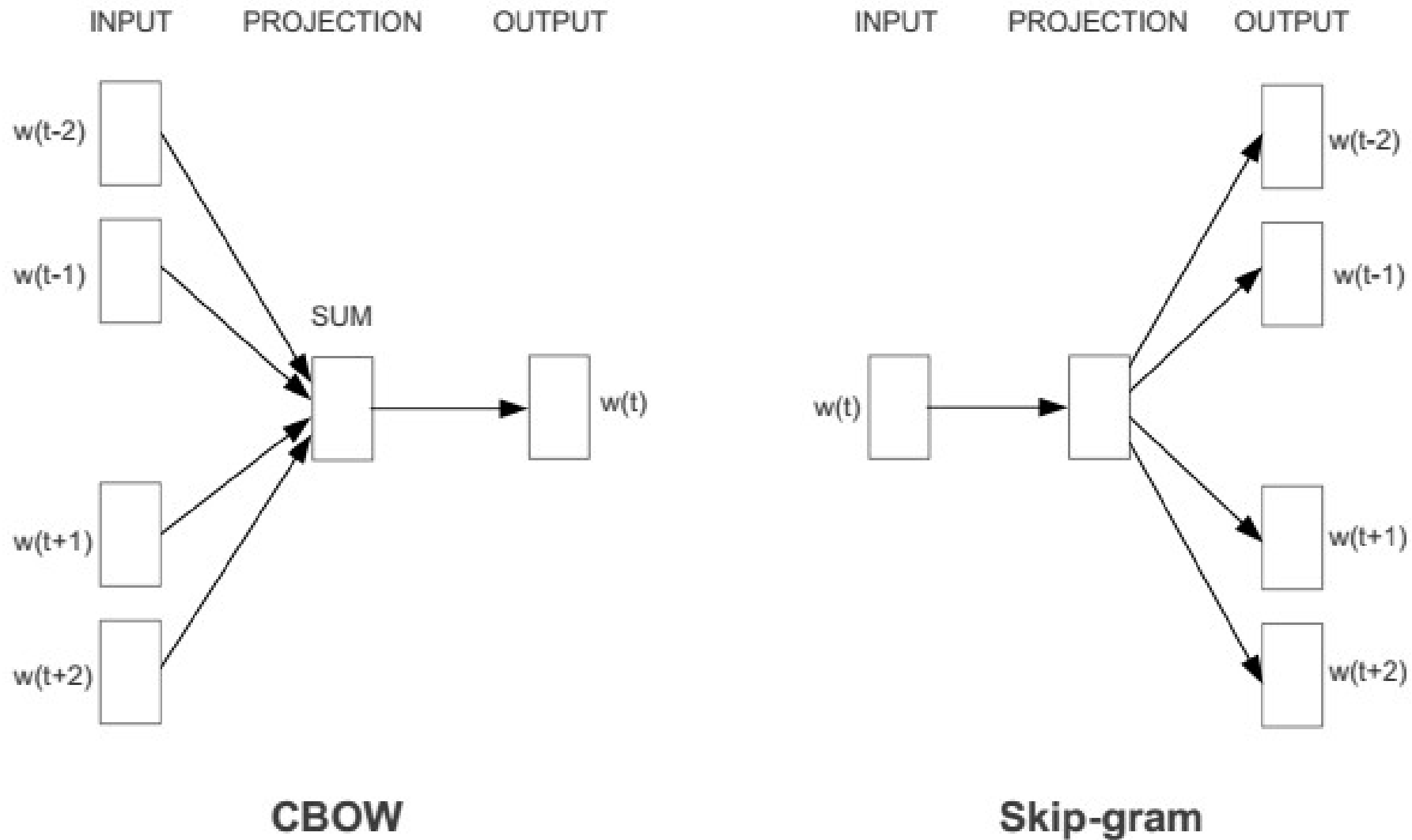


FIGURE 12.19
The homunculus.

Deep neural networks:
To be, or not to be? That is the question.



CBOW, SkipGram (word2vec)



[6] Mikolov T, Chen K, Corrado G, Dean J. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781. 2013

Hierarchical Softmax and Negative Contrastive Estimation

- HS

$$p(w|w_I) = \prod_{j=1}^{L(w)-1} \sigma \left(\mathbb{I}[n(w, j+1) = \text{ch}(n(w, j))] \cdot v'_{n(w, j)}{}^\top v_{w_I} \right)$$

- NCE

$$\log \sigma(v'_{w_O}{}^\top v_{w_I}) + \sum_{i=1}^k \mathbb{E}_{w_i \sim P_n(w)} \left[\log \sigma(-v'_{w_i}{}^\top v_{w_I}) \right]$$

Tricks in Training Word Embeddings

- The # of negative samples?
 - The more, the better.
- The distribution from which negative samples are generated? Should negative samples be close to positive samples?
 - The closer, the better.
- Full softmax vs. NCE vs. HS vs. hinge loss?

Outline

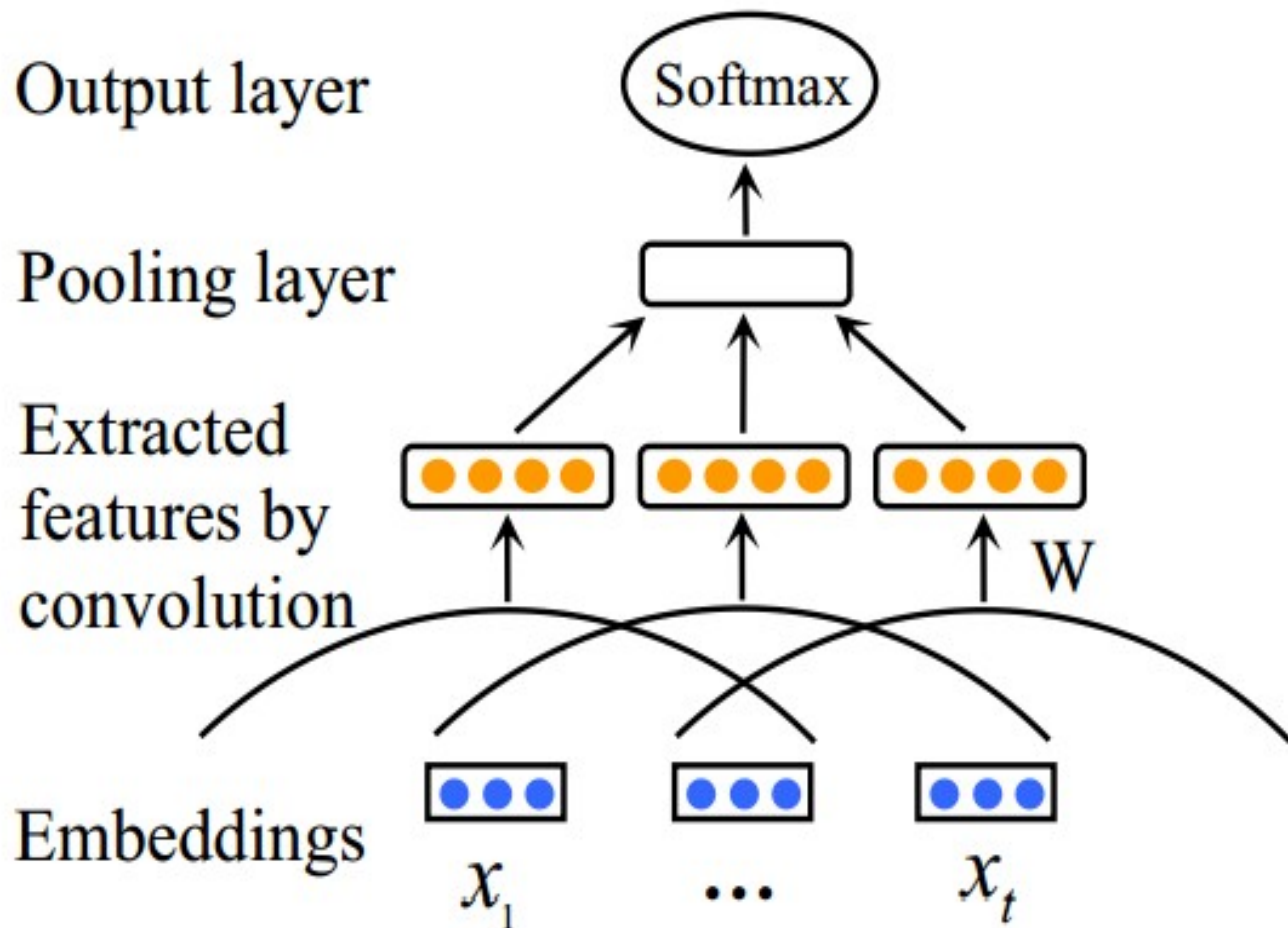
- Unsupervised Learning: Word Embeddings
- **Discriminative Sentence Models**
- Natural Language Generation
- Conclusion and Discussion

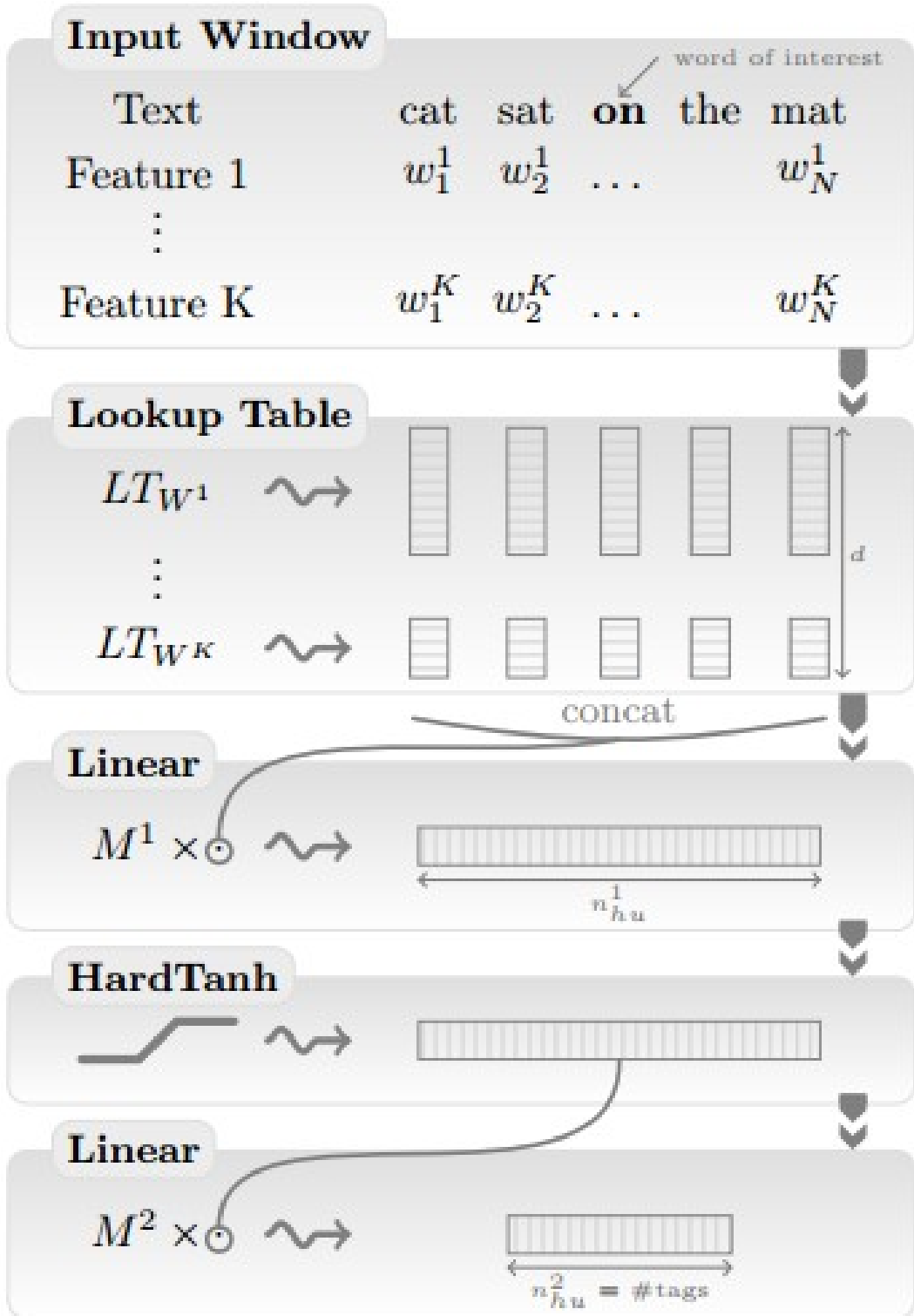
(Discriminative) Sentence Modeling

- To encode a sentence as a vector, capturing some semantics/meanings of the sentence
- Sentence classification (e.g., sentiment analysis)
- A whole bunch of downstream applications
 - Sentence matching,
 - Discourse analysis,
 - Extractive summarization, and even
 - Parsing

Convolutional Neural Networks (CNNs)

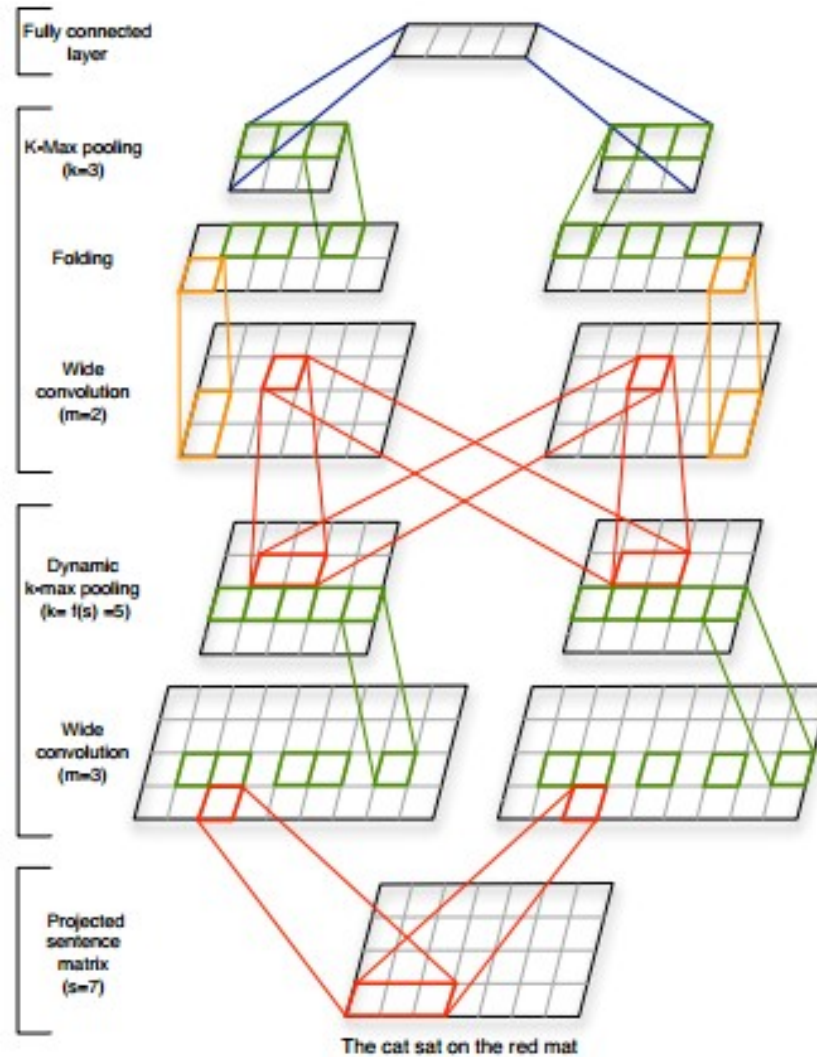
- **Convolution** in signal processing: Linear time-invariant system
 - Flip, inner-product, and slide
- **Convolution** in the neural network regime
 - Sliding window





[4] Collobert R, Weston J, Bottou L, Karlen M, Kavukcuoglu K, Kuksa P. Natural language processing (almost) from scratch. JMLR, 2011.

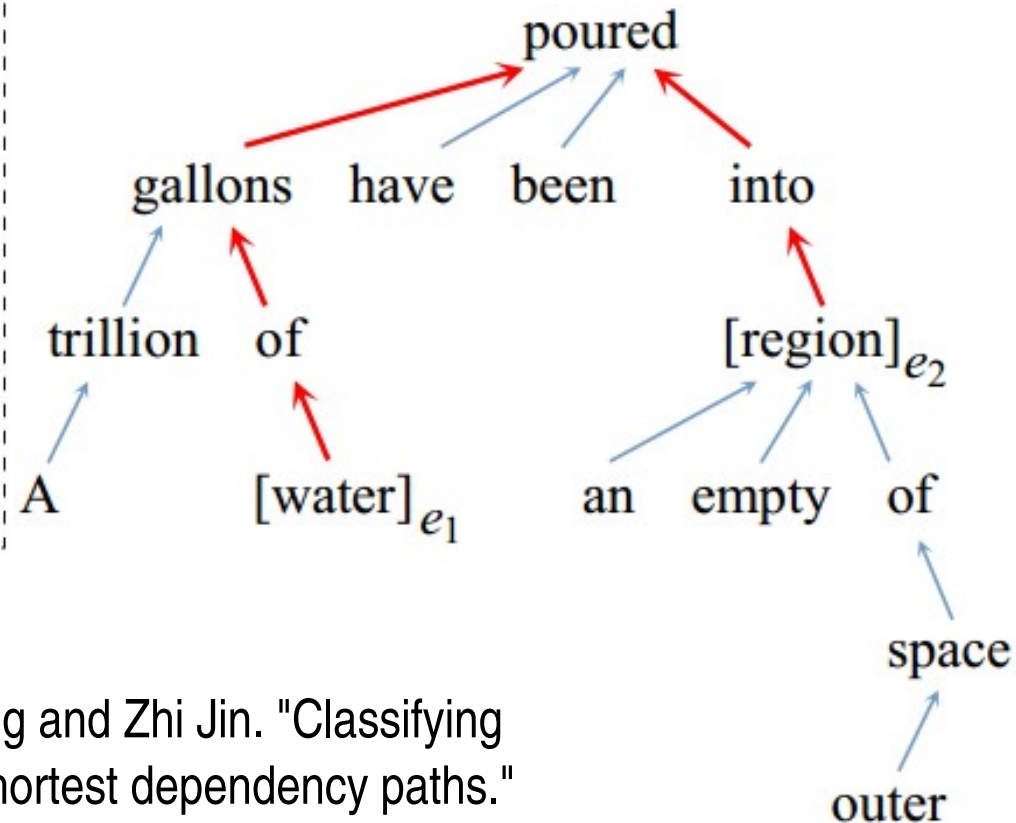
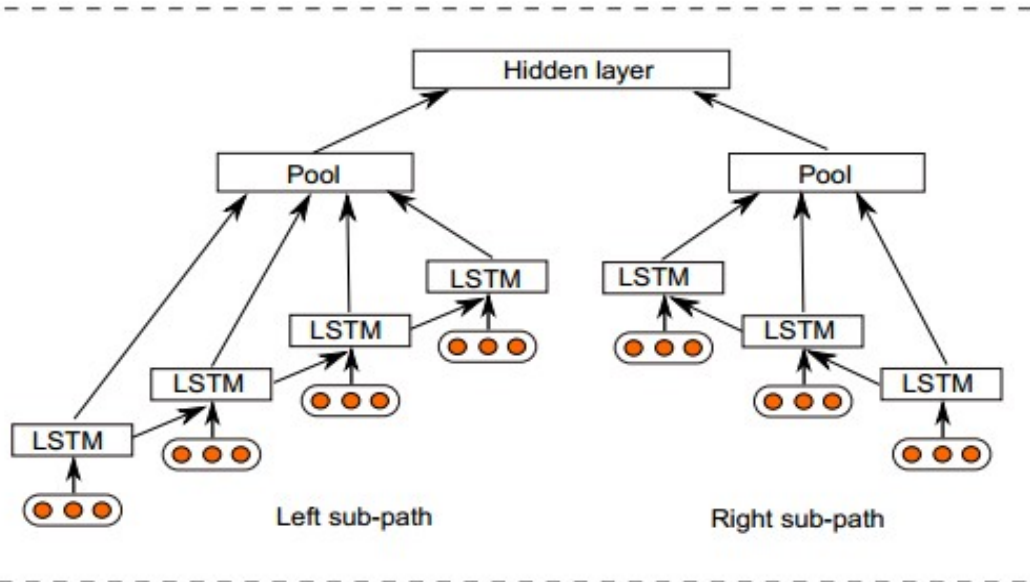
Convolutional Neural Networks (CNNs)



[9] Blunsom, Phil, Edward Grefenstette, and Nal Kalchbrenner. "A Convolutional Neural Network for Modelling Sentences." ACL, 2014.

Recurrent Neural Networks (RNNs)

- Pretty much similar to RNN LM



[10] Yan Xu, Lili Mou, Ge Li, Yunchuan Chen, Hao Peng and Zhi Jin. "Classifying relations via long short term memory networks along shortest dependency paths." In EMNLP, pages 1785--1794, 2015.

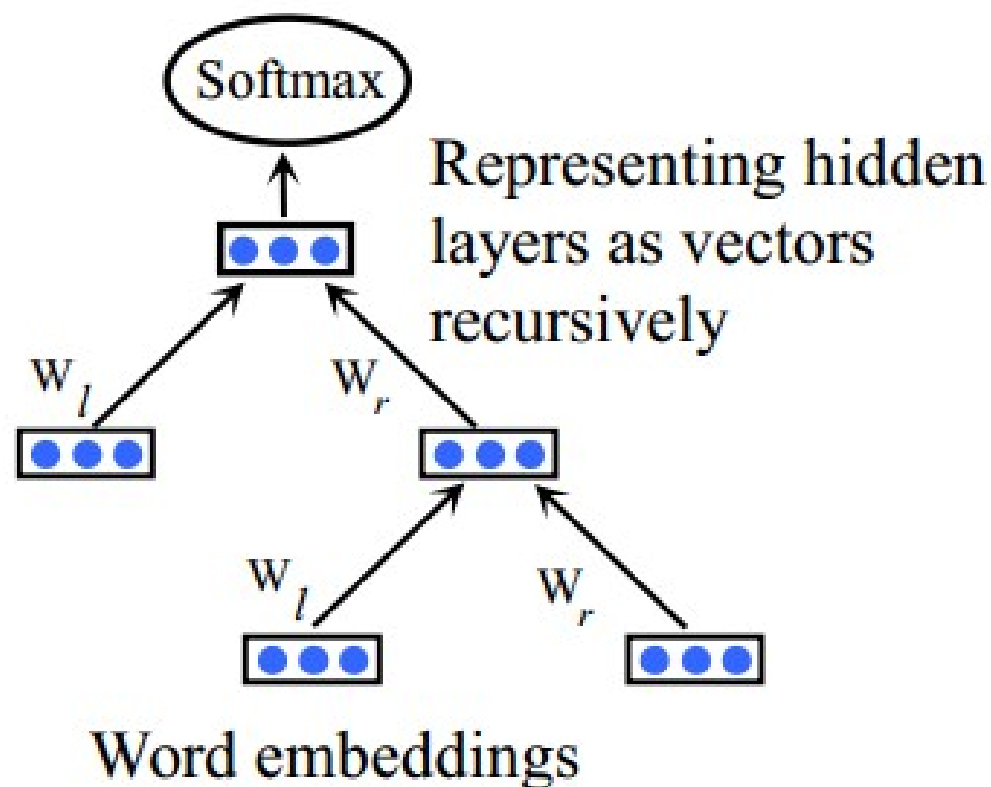
[11] Yan Xu, Ran Jia, Lili Mou, Ge Li, Yunchuan Chen, Yangyang Lu, Zhi Jin. "Improved relation classification by deep recurrent neural networks with data augmentation." arXiv preprint arXiv:1601.03651, 2016.

Recursive Neural Networks (RNNs again)

- Where does the tree come from?
 - Dynamically constructing a tree structure similar to constituency
 - Parsed by external parsers

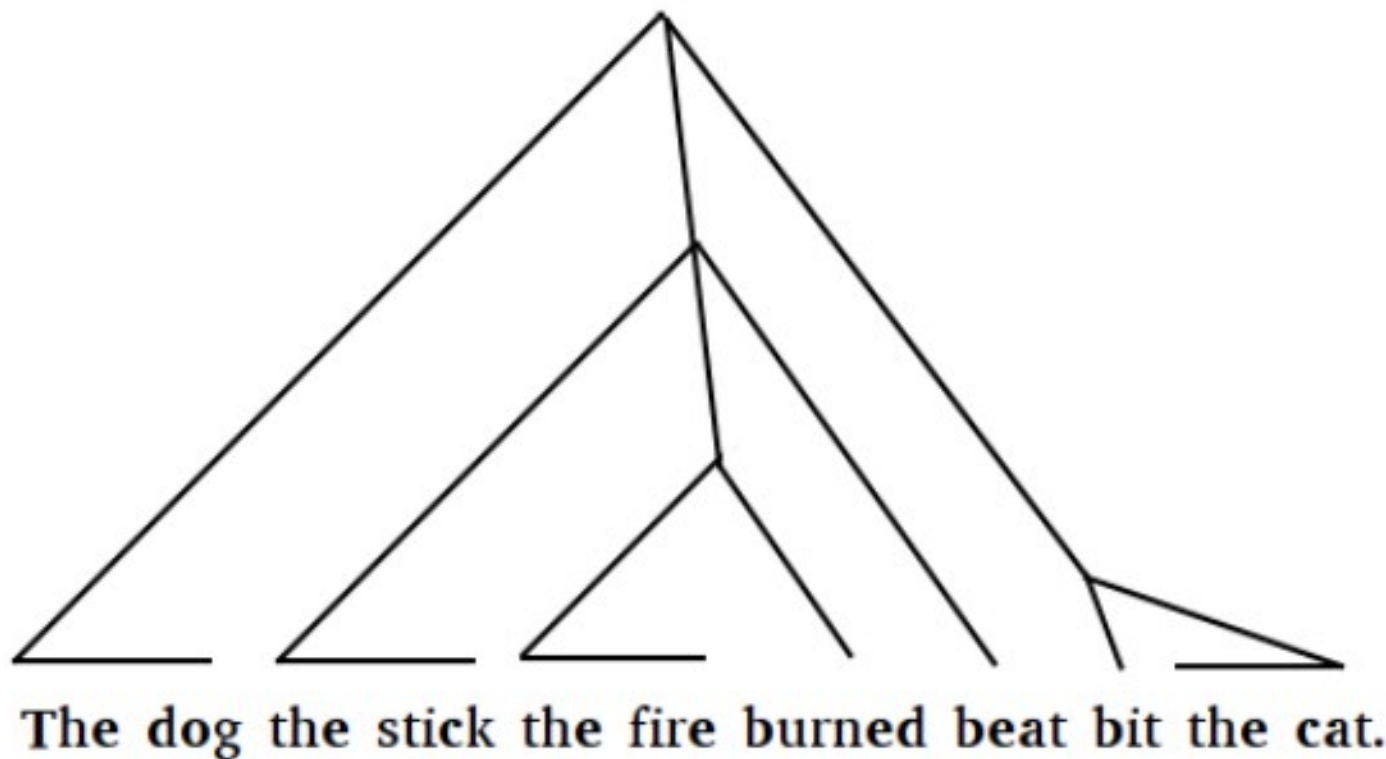
Constituency tree

- Leaf nodes = words
- Interior nodes = abstract components of a sentence (e.g., noun phrase)
- Root nodes = the whole sentence



Why parse trees may be important?

Tree structure



Convolution



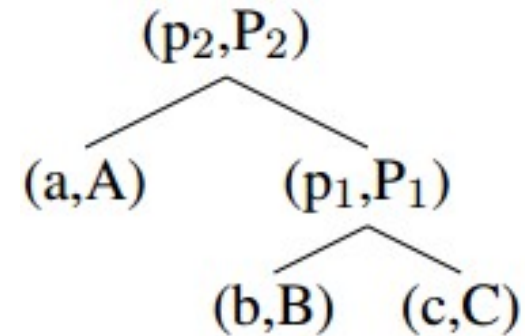
Recursive Propagation

- Perception-like interaction [13]

$$p = f(W[c_1 \ c_2]^T)$$

- Matrix-vector interaction [14]

$$p_1 = f\left(W \begin{bmatrix} Cb \\ Bc \end{bmatrix}\right), P_1 = f\left(W_M \begin{bmatrix} B \\ C \end{bmatrix}\right)$$



- Tensor interaction [15]

$$p_1 = f\left(\begin{bmatrix} b \\ c \end{bmatrix}^T V^{[1:d]} \begin{bmatrix} b \\ c \end{bmatrix} + W \begin{bmatrix} b \\ c \end{bmatrix}\right)$$

[13] Socher R, et al. Semi-supervised recursive autoencoders for predicting sentiment distributions. EMNLP, 2011

[14] Socher, R, et al. "Semantic compositionality through recursive matrix-vector spaces." EMNLP-CoNLL, 2012.

[15] Socher, R, et al. "Recursive deep models for semantic compositionality over a sentiment treebank." EMNLP, 2013.

Recurrent Propagation

- Perception-like interaction [13]

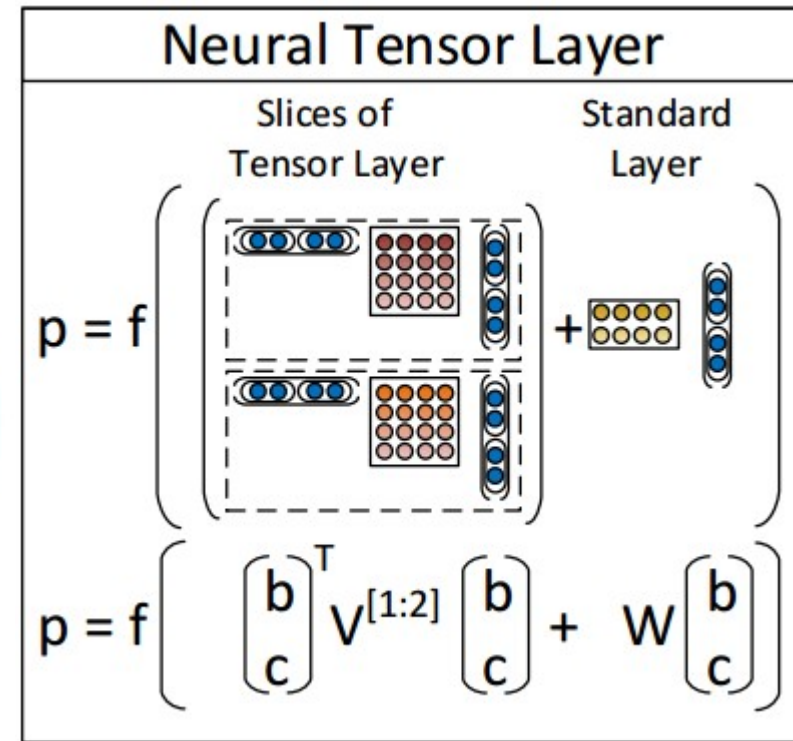
$$p = f(W[c_1 \ c_2]^T)$$

- Matrix-vector interaction [14]

$$p_1 = f\left(W \begin{bmatrix} Cb \\ Bc \end{bmatrix}\right), P_1 = f\left(W_M \begin{bmatrix} B \\ C \end{bmatrix}\right)$$

- Tensor interaction [15]

$$p_1 = f\left(\begin{bmatrix} b \\ c \end{bmatrix}^T V^{[1:d]} \begin{bmatrix} b \\ c \end{bmatrix} + W \begin{bmatrix} b \\ c \end{bmatrix}\right)$$



[13] Socher R, et al. Semi-supervised recursive autoencoders for predicting sentiment distributions. EMNLP, 2011

[14] Socher, R, et al. "Semantic compositionality through recursive matrix-vector spaces." EMNLP-CoNLL, 2012.

[15] Socher, R, et al. "Recursive deep models for semantic compositionality over a sentiment treebank." EMNLP, 2013.

Even More Interaction

- LSTM interaction [16, 17, 18]

$$\tilde{h}_j = \sum_{k \in C(j)} h_k,$$

$$i_j = \sigma \left(W^{(i)} x_j + U^{(i)} \tilde{h}_j + b^{(i)} \right),$$

$$f_{jk} = \sigma \left(W^{(f)} x_j + U^{(f)} h_k + b^{(f)} \right),$$

$$o_j = \sigma \left(W^{(o)} x_j + U^{(o)} \tilde{h}_j + b^{(o)} \right),$$

$$u_j = \tanh \left(W^{(u)} x_j + U^{(u)} \tilde{h}_j + b^{(u)} \right),$$

$$c_j = i_j \odot u_j + \sum_{k \in C(j)} f_{jk} \odot c_k,$$

$$h_j = o_j \odot \tanh(c_j),$$

[16] Tai, Kai Sheng, Richard Socher, and Christopher D. Manning. "Improved semantic representations from tree-structured long short-term memory networks." ACL, 2015

[17] Zhu, Xiaodan, Parinaz Sobihani, and Hongyu Guo. "Long short-term memory over recursive structures." ICML, 2015.

[18] Le, Phong, and Willem Zuidema. "Compositional distributional semantics with long short term memory." arXiv:1503.02510 (2015).

Even More Interaction

- LSTM interaction [16, 17, 18]

To be a good scientist:

– Challenge authority.

To be a good academia:

– Cite authority. Make friends.

$$\tilde{h}_j = \sum_{k \in C(j)} h_k,$$

$$i_j = \sigma \left(W^{(i)} x_j + U^{(i)} \tilde{h}_j + b^{(i)} \right),$$

$$f_{jk} = \sigma \left(W^{(f)} x_j + U^{(f)} h_k + b^{(f)} \right),$$

$$o_j = \sigma \left(W^{(o)} x_j + U^{(o)} \tilde{h}_j + b^{(o)} \right),$$

$$u_j = \tanh \left(W^{(u)} x_j + U^{(u)} \tilde{h}_j + b^{(u)} \right),$$

$$c_j = i_j \odot u_j + \sum_{k \in C(j)} f_{jk} \odot c_k,$$

$$h_j = o_j \odot \tanh(c_j),$$

[16] Tai, Kai Sheng, Richard Socher, and Christopher D. Manning. "Improved semantic representations from tree-structured long short-term memory networks." ACL, 2015

[17] Zhu, Xiaodan, Parinaz Sobihani, and Hongyu Guo. "Long short-term memory over recursive structures." ICML, 2015.

[18] Le, Phong, and Willem Zuidema. "Compositional distributional semantics with long short term memory." arXiv:1503.02510, 2015.

Tree-Based Convolutional Neural Network (TBCNN)

- CNNs

- ☺ Efficient feature learning and extraction

- The propagation path is irrelevant to the length of a sentence

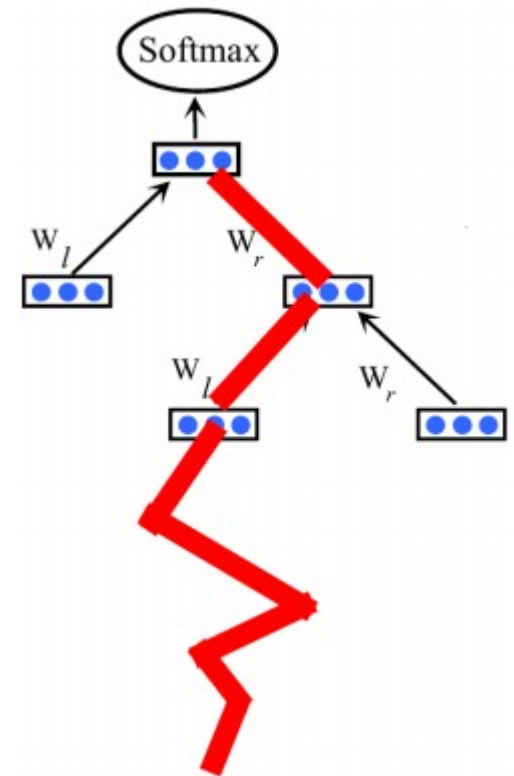
- ☹ Structure insensitive

- Recursive networks

- ☺ Structure sensitive

- ☹ Long propagation path

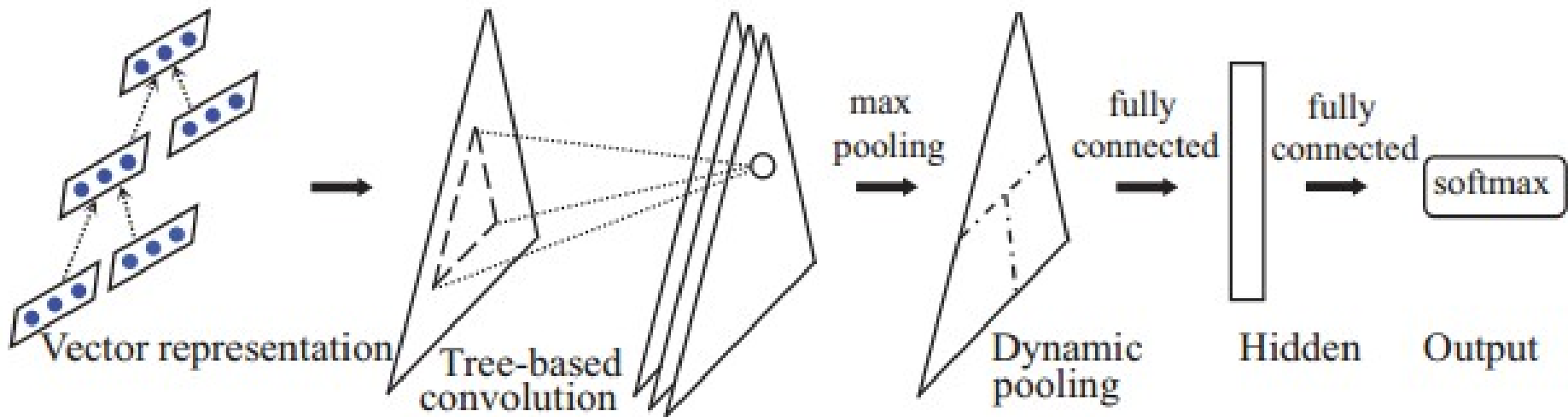
- The problem of “gradient vanishing or explosion”



Our intuition

- Can we combine?
 - Structure sensitive as recursive neural networks
 - Short propagation path as convolutional neural networks
- Solution
 - The tree-based convolutional neural network (TBCNN)
 - Recall convolution = sliding window in the NN regime
 - Tree-based convolution = sliding window of a subtree

Tree-Based Convolution

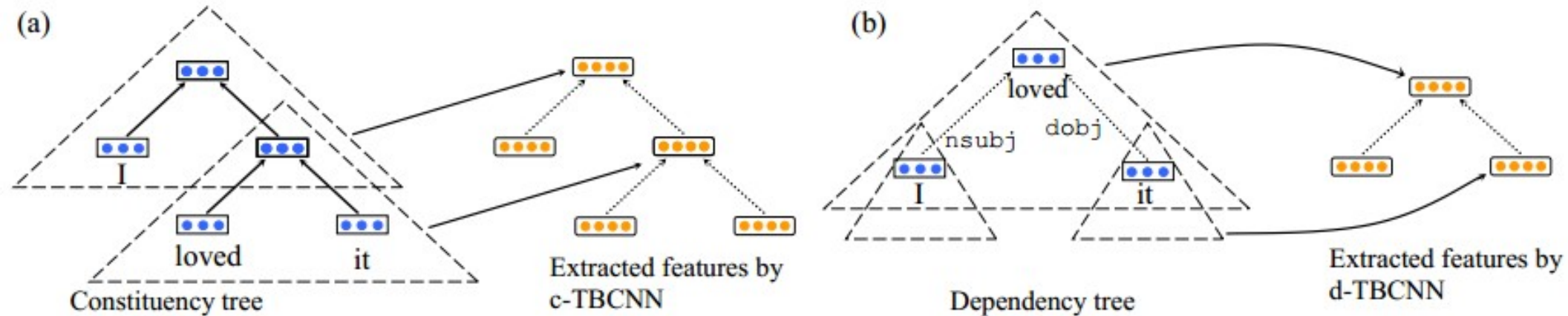


$$\mathbf{y} = f \left(\sum_{i=1}^t W_i \cdot \mathbf{x}_i + \mathbf{b} \right)$$

[19] Lili Mou, Ge Li, Lu Zhang, Tao Wang, Zhi Jin. "Convolutional neural networks over tree structures for programming language processing." In AACL, pages 1287--1293, 2016.

[20] Lili Mou, Hao Peng, Ge Li, Yan Xu, Lu Zhang, Zhi Jin. "Discriminative neural sentence modeling by tree-based convolution." In EMNLP, pages 2315--2325, 2015.

A Few Variants

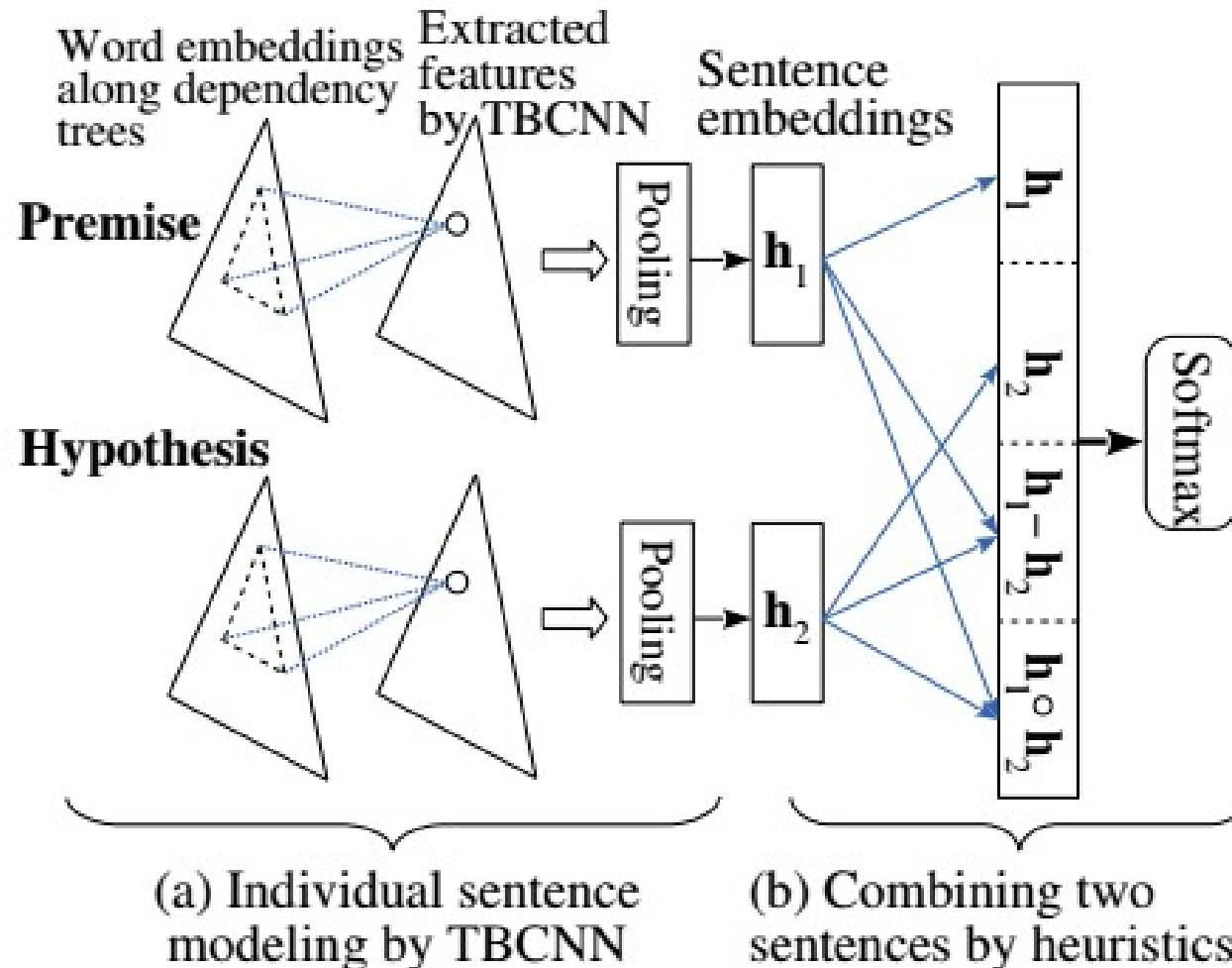


$$\mathbf{y} = f \left(W_p^{(c)} \cdot \mathbf{p} + W_l^{(c)} \cdot \mathbf{c}_l + W_r^{(c)} \cdot \mathbf{c}_r + \mathbf{b}^{(c)} \right) \quad \mathbf{y} = f \left(W_p^{(d)} \cdot \mathbf{p} + \sum_{i=1}^n W_{r[c_i]}^{(d)} \cdot \mathbf{c}_i + \mathbf{b}^{(d)} \right)$$

Wrap Up

		Way of information propagation	
		Iterative	Sliding
Structure	Flat	Recurrent	Convolution
	Tree	Recursive	Tree-base convolution

A glance at how sentence modeling benefits downstream tasks



[21] Lili Mou, Rui Men, Ge Li, Yan Xu, Lu Zhang, Rui Yan, Zhi Jin. "Natural language inference by tree-based convolution and heuristic matching." ACL(2), 2016.

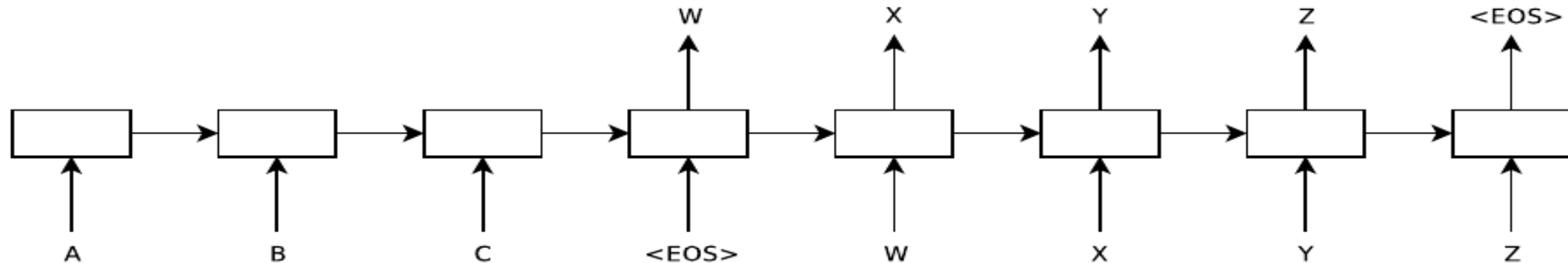
Outline

- Unsupervised Learning: Word Embeddings
- Discriminative Sentence Models
- **Natural Language Generation**
- Conclusion and Discussion

Applications of Natural Language Generation

- Machine translation
- Question answering
- Conversation systems
- Generative summarization

Sequence to Sequence Generation



- Training phrase: X, Y, and Z are the ground truth (words in the corpus)
- Predicting phrase: X, Y, and Z are those generated by RNN
- Seq2seq model is essentially an LM (of XYZ) conditioned on another LM (of ABC)

[22] Sutskever, Ilya, Oriol Vinyals, and Quoc V. Le. "Sequence to sequence learning with neural networks." NIPS. 2014.

The Attention Mechanism

- During sequence generation, the output sequence's hidden state \mathbf{h}_t is related to
 - That of the last time step \mathbf{h}_{t-1} , and
 - A context vector \mathbf{c} , which is a combination of the input sequence's states

$$\mathbf{h}_t = \text{RNN}(\mathbf{h}_{t-1}, \mathbf{c}) = f(W[\mathbf{h}_{t-1}; \mathbf{c}])$$

[23] Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate." arXiv preprint arXiv:1409.0473 (2014).

Context Vector

The context vector \mathbf{c} is a combination of the input sequence's states

$$\mathbf{c} = \sum_i \alpha_i \mathbf{c}_i$$

where the coefficient α_i is related to

- The local context \mathbf{c}_i , and
- The last output state \mathbf{h}_{t-1}
- α_i is normalized

$$\alpha_i = \frac{\exp\{\tilde{\alpha}_i\}}{\sum_j \exp\{\tilde{\alpha}_j\}} \quad \tilde{\alpha}_i = W[\mathbf{h}_{t-1}; \mathbf{c}_i]$$

Sequence-Level Training

- Motivation: We don't have the ground truth
 - In a dialogue system, “The nature of of open-domain conversations shows that a variety of replies are plausible, but some are more meaningful, and others are not.” [21]
- Optimize the sequence generator as a whole in terms of external metrics

[24] Xiang Li, Lili Mou, Rui Yan, Ming Zhang. "StalemateBreaker: A proactive content-introducing approach to automatic human-computer conversation." IJCAI, 2016.

REINFORCE

[25] Ranzato, Marc'Aurelio, et al. "Sequence Level Training with Recurrent Neural Networks." ICLR, 2016.

- Define an external cost function on a generated sequence
- Generate words by sampling
- Take the derivative of generated samples

$$L_{\theta} = - \sum_{w_1^g, \dots, w_T^g} p_{\theta}(w_1^g, \dots, w_T^g) r(w_1^g, \dots, w_T^g) = -\mathbb{E}_{[w_1^g, \dots, w_T^g] \sim p_{\theta}} r(w_1^g, \dots, w_T^g)$$

$\partial p(\mathbf{w}) = p(\mathbf{w}) \partial \log p(\mathbf{w})$ because $p(\mathbf{w}) = \exp\{\log p(\mathbf{w})\}$

- $$\partial J = \sum_{\mathbf{w}} [\partial p(\mathbf{w} | \dots)] r(\mathbf{w}) = \sum_{\mathbf{w}} p(\mathbf{w}) [\partial \log p(\mathbf{w})] r(\mathbf{w})$$
$$= \sum_{\mathbf{w}} (p_{\theta}(w_{t+1} | w_t^g, \mathbf{h}_{t+1}, \mathbf{c}_t) - \mathbf{1}(w_{t+1}^g)) r(\mathbf{w})$$

where \mathbf{o}_t is the input to the softmax.

Outline
















- Unsupervised Learning: Word Embeddings
- Discriminative Sentence Models
- Natural Language Generation
- **Conclusion and Discussion**

Outline

- Unsupervised Learning: Word Embeddings
- Discriminative Sentence Models
- Natural Language Generation
- **Conclusion** and Discussion

Discussion

Challenge of end-to-end learning:

	avg	sum	max	attention	argmax
Differentiability					
Supervision					
Scalability					

Intuition

- Using external information to guide an NN instead of designing end-to-end machines
 - Better performance in short term
 - May or may not conform to the goal of AI, depending on how strict the external information is

	Hard mechanism
Differentiability	☺
Supervision	☺
Scalability	☺

Thank you for listening!

Questions?

References

- [1] Lili Mou, Rui Yan, Ge Li, Lu Zhang, Zhi Jin. "Backward and forward language modeling for constrained sentence generation." arXiv preprint arXiv:1512.06612, 2015.
- [2] Bengio, Yoshua, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. "A Neural Probabilistic Language Model." JMLR, 2003.
- [3] Mikolov T, Karafiát M, Burget L, Cernocký J, Khudanpur S. Recurrent neural network based language model. In INTERSPEECH, 2010.
- [4] Collobert R, Weston J, Bottou L, Karlen M, Kavukcuoglu K, Kuksa P. Natural language processing (almost) from scratch. JMLR, 2011.
- [5] Mnih A, Hinton GE. A scalable hierarchical distributed language model. NIPS, 2009.
- [6] Mikolov T, Chen K, Corrado G, Dean J. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781, 2013.
- [7] Huth, Alexander G., et al. "Natural speech reveals the semantic maps that tile human cerebral cortex." Nature 532.7600 (2016): 453-458.
- [8] Bear MF, Connors BW, Michael A. Paradiso. Neuroscience: Exploring the Brain. 2007

[9] Blunsom, Phil, Edward Grefenstette, and Nal Kalchbrenner. "A Convolutional Neural Network for Modelling Sentences." ACL, 2014.

[10] Yan Xu, Lili Mou, Ge Li, Yunchuan Chen, Hao Peng and Zhi Jin. "Classifying relations via long short term memory networks along shortest dependency paths." In EMNLP, pages 1785--1794, 2015.

[11] Yan Xu, Ran Jia, Lili Mou, Ge Li, Yunchuan Chen, Yangyang Lu, Zhi Jin. "Improved relation classification by deep recurrent neural networks with data augmentation." arXiv preprint arXiv:1601.03651, 2016.

[12] Pinker, Steven. The Language Instinct: The New Science of Language and Mind. Penguin UK, 1995.

[13] Socher R, et al. Semi-supervised recursive autoencoders for predicting sentiment distributions. EMNLP, 2011

[14] Socher, R, et al. "Semantic compositionality through recursive matrix-vector spaces." EMNLP-CoNLL, 2012.

[15] Socher, R, et al. "Recursive deep models for semantic compositionality over a sentiment treebank." EMNLP, 2013.

[16] Tai, Kai Sheng, Richard Socher, and Christopher D. Manning. "Improved semantic representations from tree-structured long short-term memory networks." ACL, 2015

- [17] Zhu, Xiaodan, Parinaz Sobihani, and Hongyu Guo. "Long short-term memory over recursive structures." ICML, 2015.
- [18] Le, Phong, and Willem Zuidema. "Compositional distributional semantics with long short term memory." arXiv:1503.02510 (2015).
- [19] Lili Mou, Ge Li, Lu Zhang, Tao Wang, Zhi Jin. "Convolutional neural networks over tree structures for programming language processing." In AAI, pages 1287--1293, 2016.
- [20] Lili Mou, Hao Peng, Ge Li, Yan Xu, Lu Zhang, Zhi Jin. "Discriminative neural sentence modeling by tree-based convolution." In EMNLP, pages 2315--2325, 2015.
- [21] Lili Mou, Rui Men, Ge Li, Yan Xu, Lu Zhang, Rui Yan, Zhi Jin. "Natural language inference by tree-based convolution and heuristic matching." ACL(2), 2016.
- [22] Sutskever, Ilya, Oriol Vinyals, and Quoc V. Le. "Sequence to sequence learning with neural networks." NIPS. 2014.
- [23] Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate." ICLR, 2015.
- [24] Xiang Li, Lili Mou, Rui Yan, Ming Zhang. "StalemateBreaker: A proactive content-introducing approach to automatic human-computer conversation." IJCAI, 2016.
- [25] Ranzato, Marc'Aurelio, et al. "Sequence Level Training with Recurrent Neural Networks." ICLR, 2016.