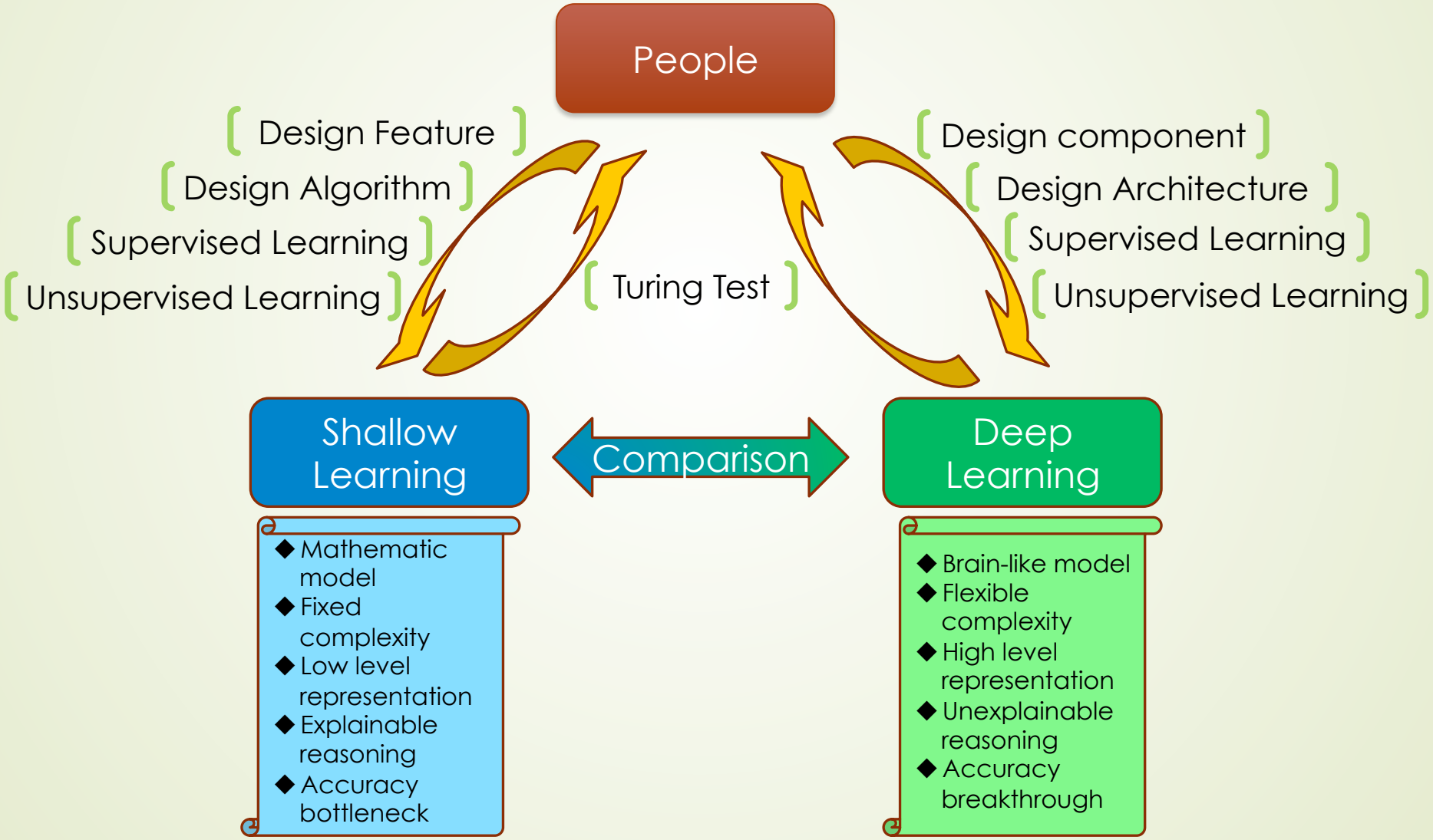
Tutorial: Deep learning-based Chinese Information Processing

Junjie Zhai, Zehua Xie, Chao Liu

About Authors

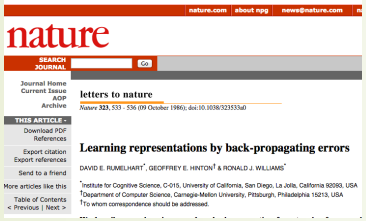
Motivation for Neural networks



Outline

- Background of Deep Learning
- Deep learning for Chinese NLP
- Feed-forward Neural Network
- Recurrent Neural Network

Development of Deep Learning

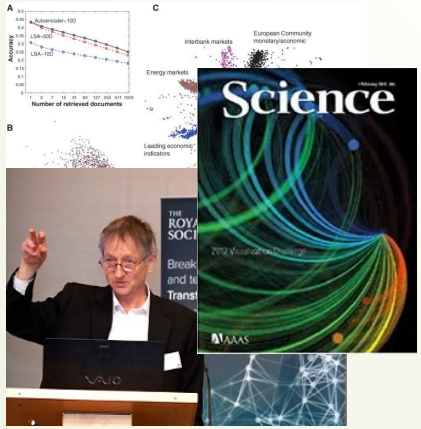


1986

The article about back propagation published on Nature

Overcome the difficulty in training neural network

2006

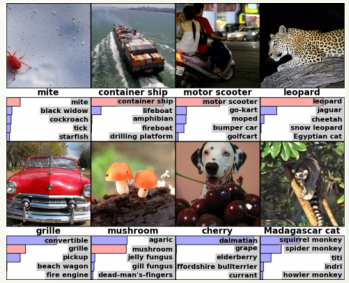


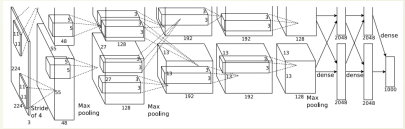
2011

Huge improvement in speech recognition: 33%

Huge improvement in image recognition: 11%

2012





2015

The first time to surpass human-level performance on visual recognition

Applications of Deep Learning

Progress on Voice Field

- The recognition rate has achieved 81% in noisy environment and 94% in quiet environment.

Skype Translator: make it possible for cross-language real-time communication

Voice Assistant: make life more convenient

Applications of Deep Learning

Progress on Image Recognition Field

- The image recognition task has achieved 4.94% top-5 test error on the ImageNet 2012 classification dataset.

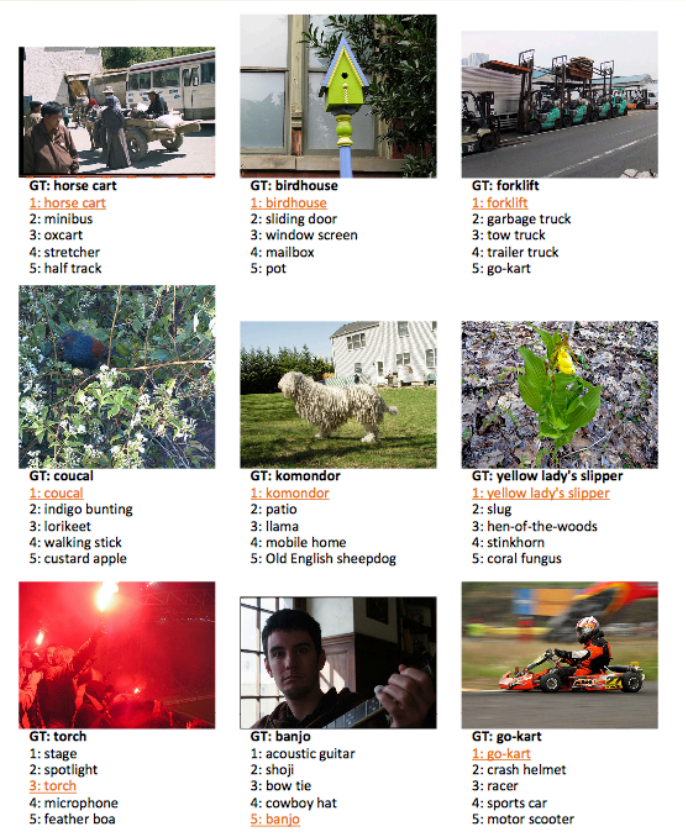
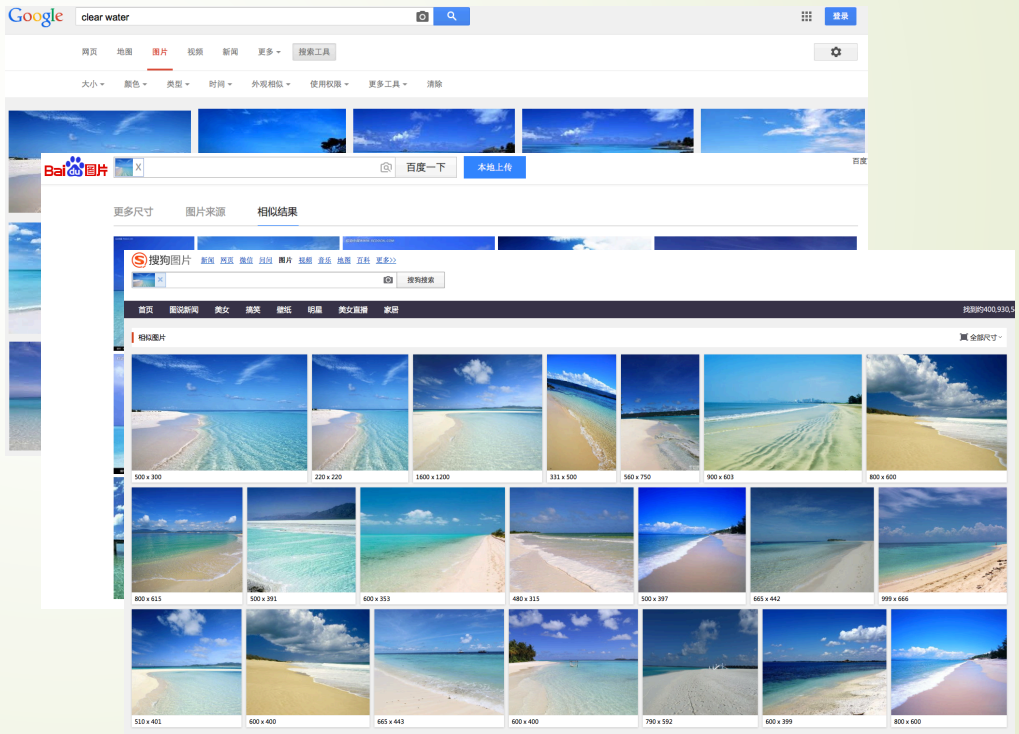
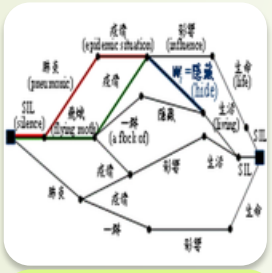


Image Classification Task



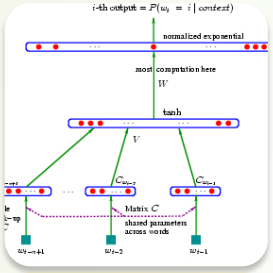
Application on Image Searching

The Roadmap of the Development of Chinese NLP



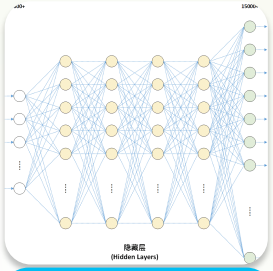
PLM(Probabilistic Language Model)

- Sparse representation
- Sparse feature, imbalanced learning
- Curse of dimension
- Lack of semantic information
- High dimension unfit for neural network



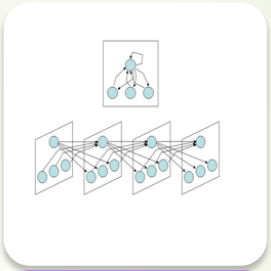
NLM(Neural Language Model)

- Dense representation
- Sparse feature, imbalanced learning
- Curse of dimension
- Semantic information
- Fit for neural network



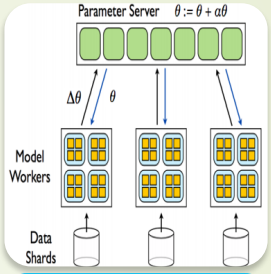
FNN(Feed-forward Neural Network)

- Robust
- Capable for modeling complex problem
- Arbitrary complexity of model
- Avoid the feature engineering
- Unfit for sequence task



RNN(Recurrent Neural network)

- Robust
- Capable for modeling complex problem
- Arbitrary complexity of model
- Avoid the feature engineering
- Born with sequence property



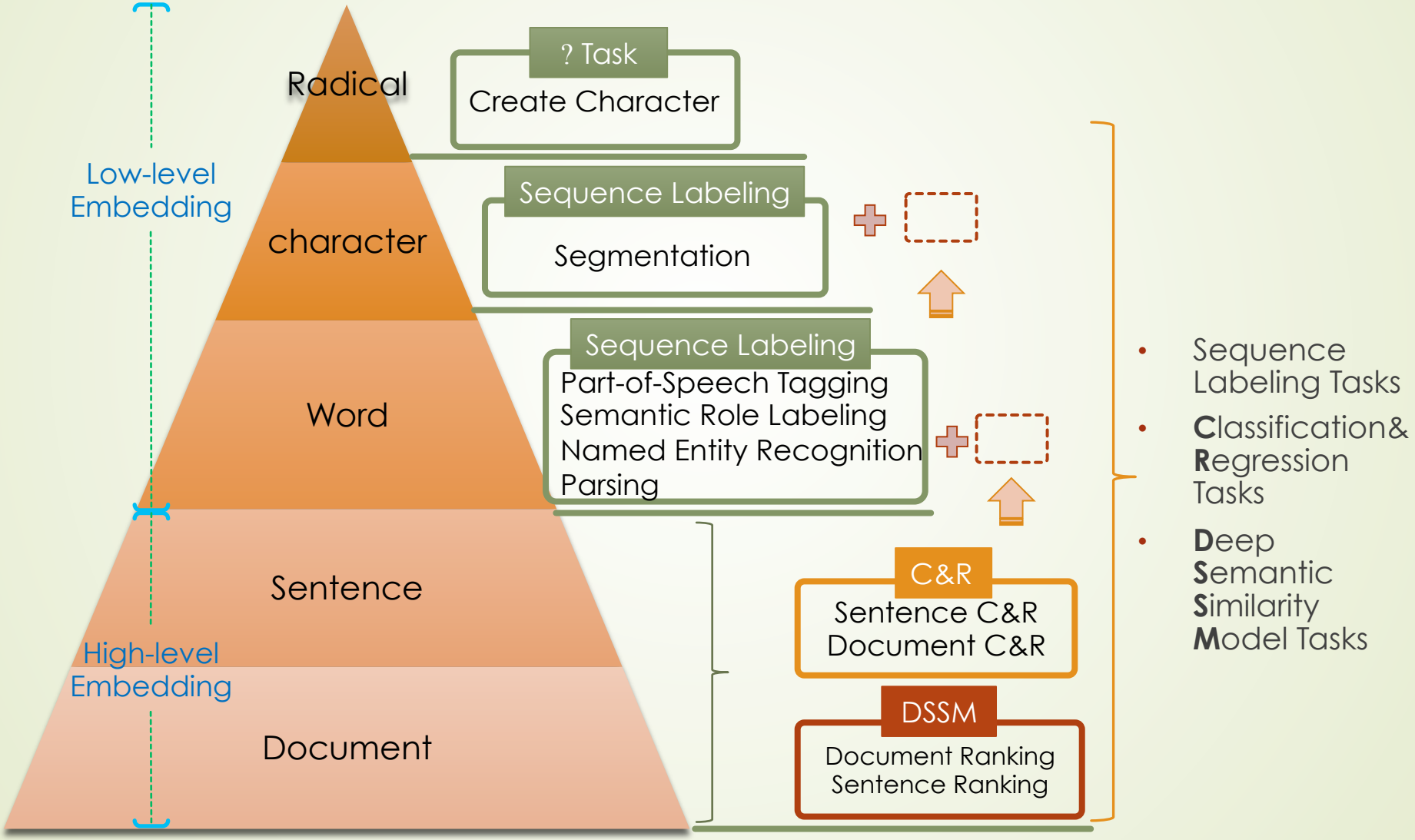
PDP(Parallel Distributed Platform)

- Large data set
- Parallel computing
- Widely algorithm supporting
- Scalable capacity

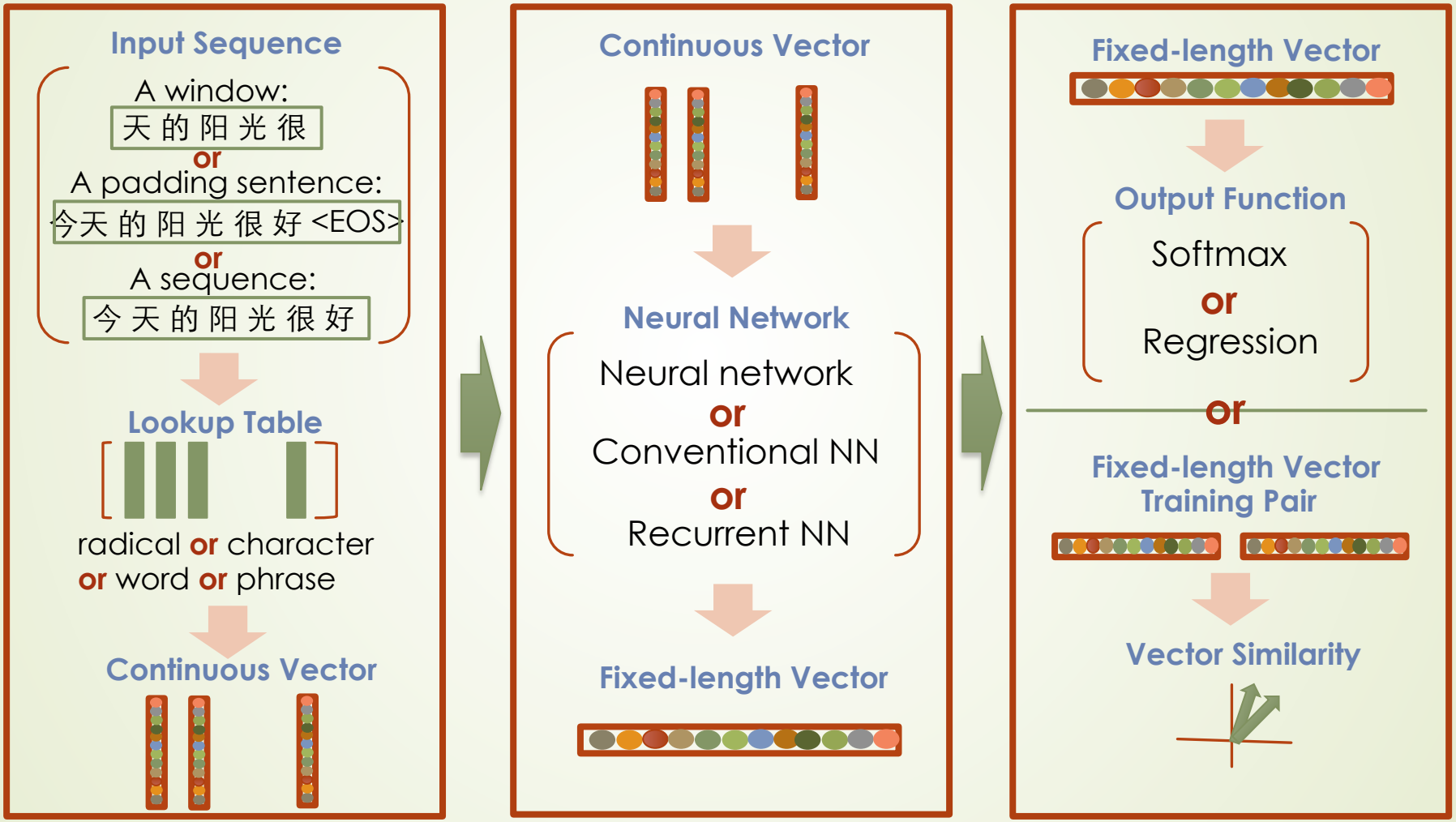
Outline

- Background of Deep Learning
- Deep learning for Chinese NLP
  - Neural Language Model
  - Feed-forward Neural Network
  - Recurrent Neural Network

Pyramid of the Unit of Chinese NLP



Three Functional Blocks of Neural Network



Outline

- Background of Deep Learning
- Deep learning for Chinese NLP
  - Neural Language Model
  - Feed-forward Neural Network
  - Recurrent Neural Network

Deep Learning in Language Model

- Probabilistic Language Model
- Definition:

  Probabilistic distribution over sequences of words.

- Proficiency:
  - Sparsity, Smoothing,
  - Curve of dimension(Heaps rule: M(word size) T(document size) $M=kT\uparrow b$)
  - Lost Semantic similarity
  - Large & sparse input （Neural Network  ）

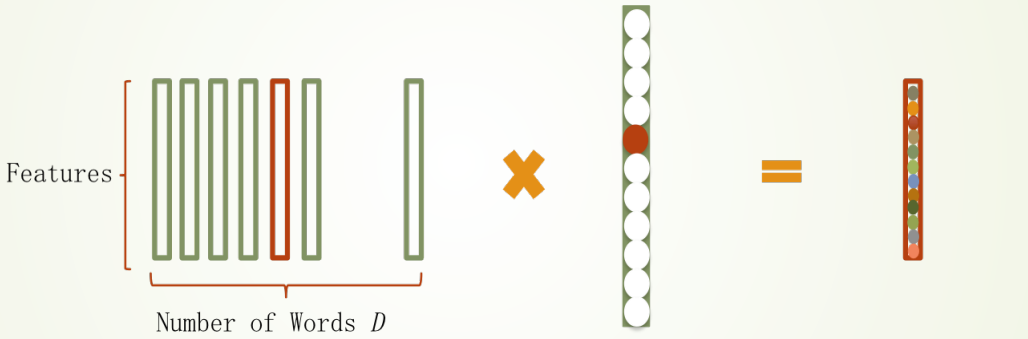
Deep Learning in Language Model

- Neural Language Model（Word Embedding）
- Definition:

  Neural network based dense representation of words
- Advantages:
  - Low-dimensional dense vector(50~1000 dimensions).
  - Conditional word probabilities ➔ word embeddings.
  - Semantic/syntactic similarity exploited.
  - Low-dimension & dense input.（ Neural Network  ）

Deep Learning in Language Model

- Word Embedding
  - A low-dimensional continuous vector representation for each word
  - Captures the word meaning in a semantic space

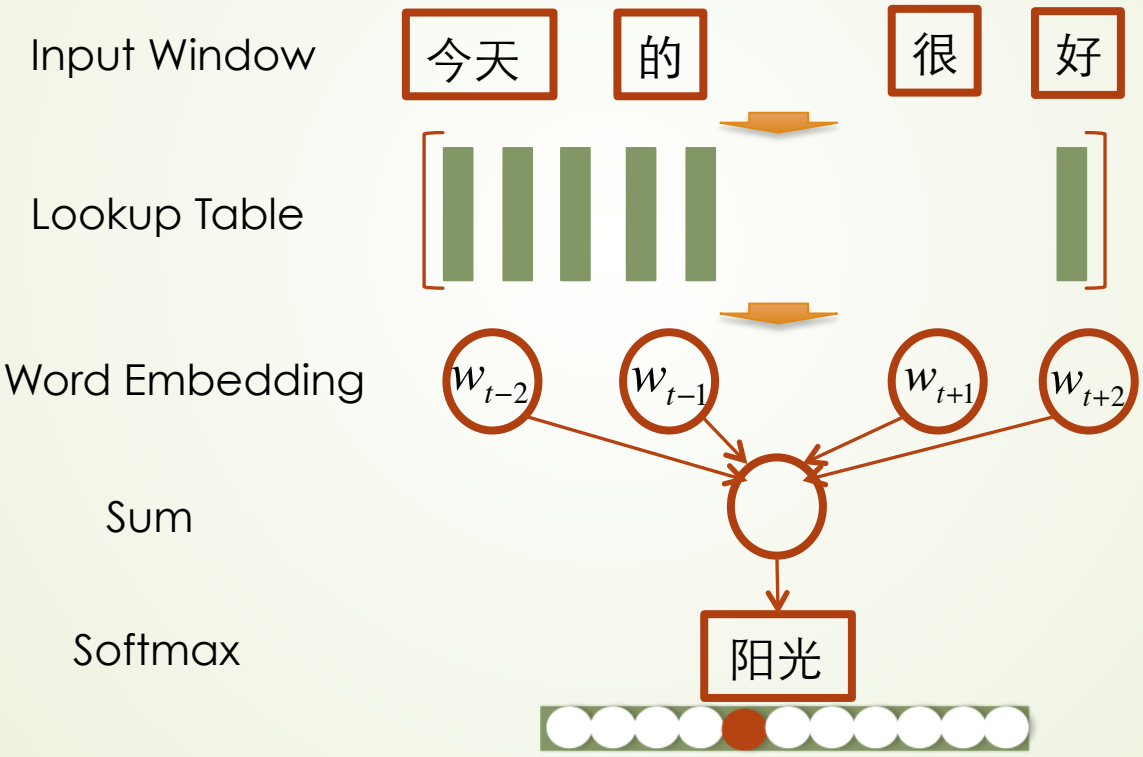


$$L \cdot e_i = w_i \in R^{d \times 1}$$

- Common Neural Network based Word Embedding Approaches
  - CBOW & Skip-gram
  - SENNA embedding
  - RNN language model based embedding

Common Neural Network based Word Embedding Approaches

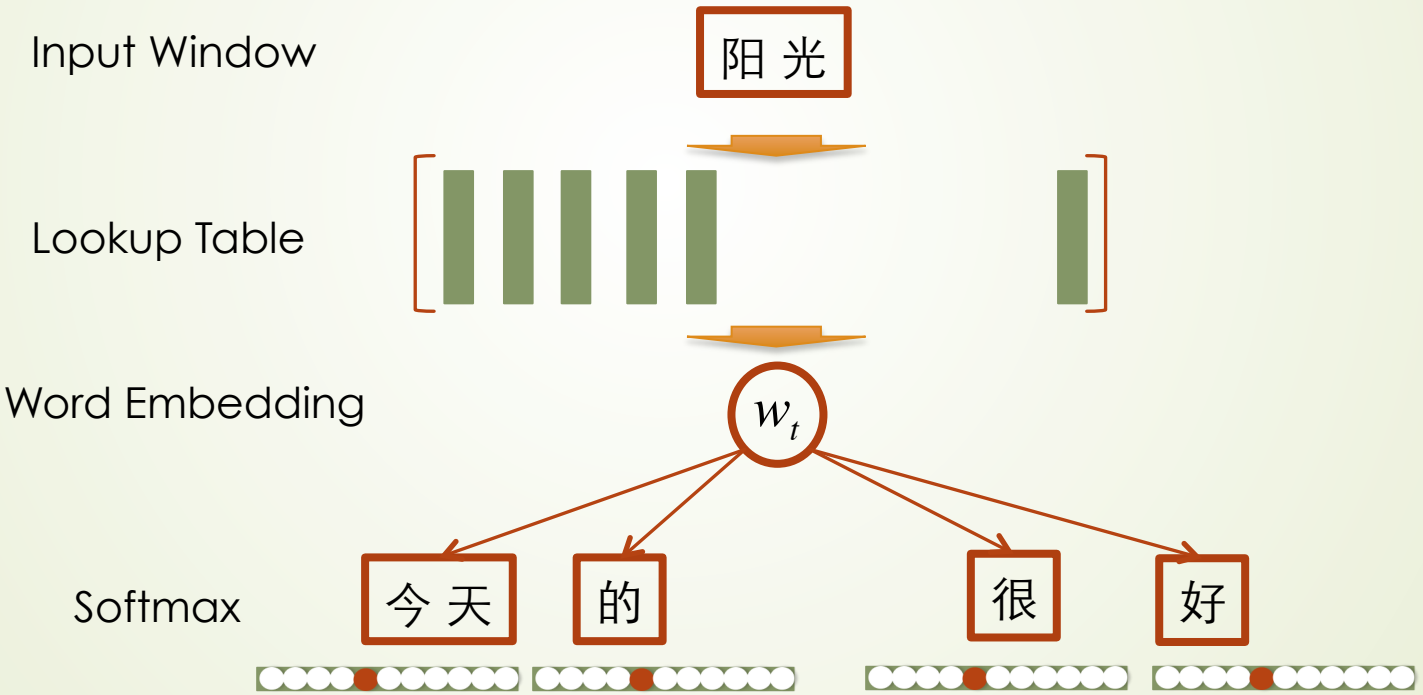
- CBOW



[Mikolov et al., 2013a]

Common Neural Network based Word Embedding Approaches

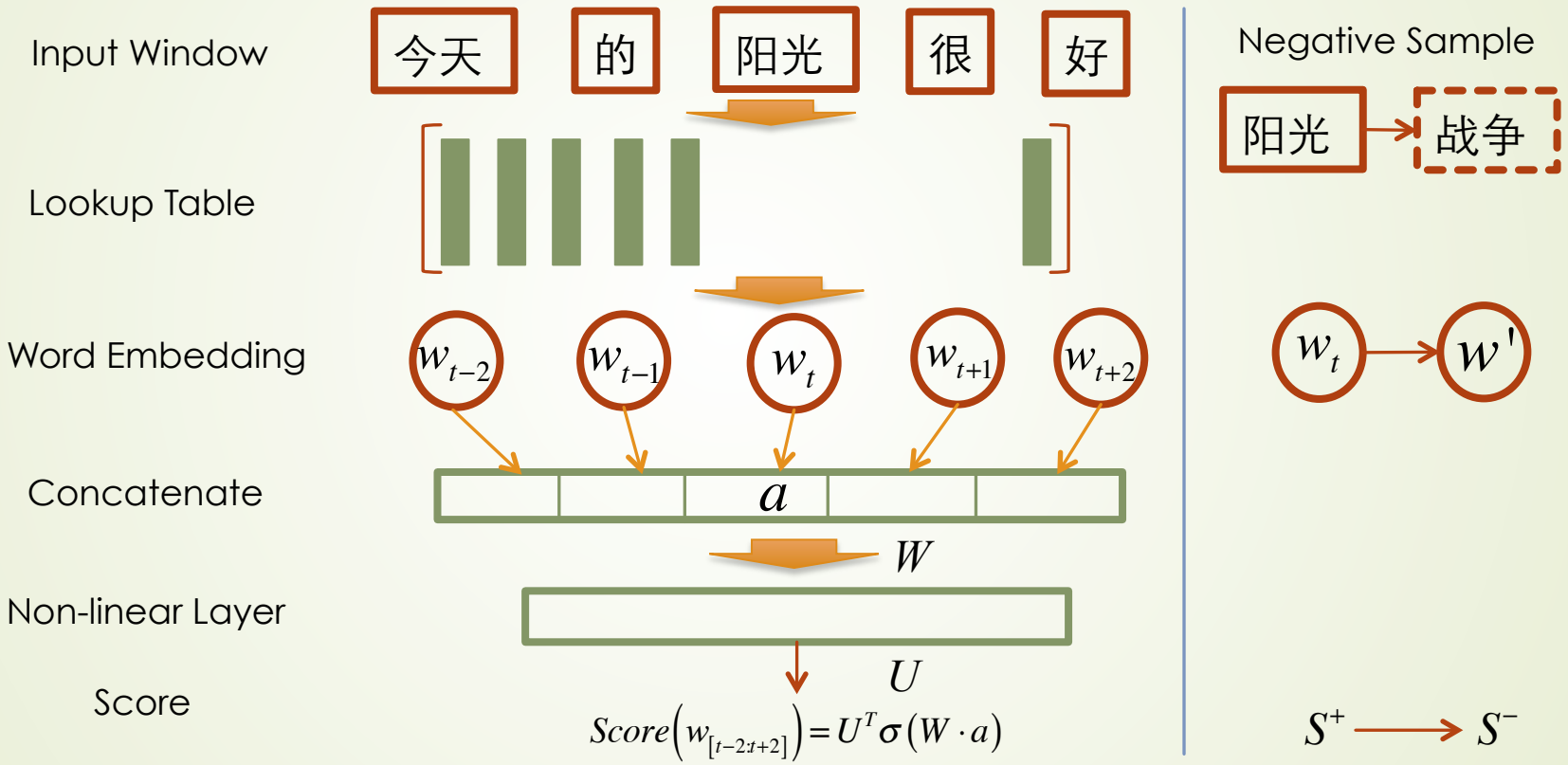
- Skip-gram



[Mikolov et al., 2013a]

Common Neural Network based Word Embedding Approaches

- SENNA



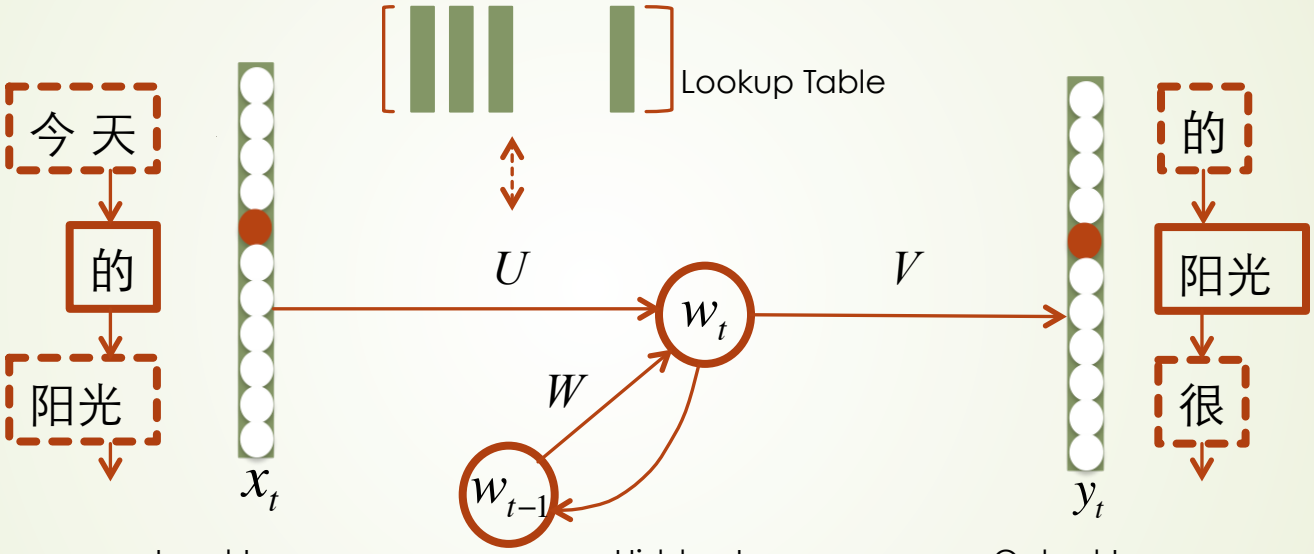
$$S^+ = Score(w_{t-2}, w_{t-1}, w_t, w_{t+1}, w_{t+2}) \qquad S^- = Score(w_{t-2}, w_{t-1}, w', w_{t+1}, w_{t+2})$$

Minimize the objective function: $J = \max(0, 1 - S^+ + S^-)$ Update model until $S^+ > 1 + S^-$

[Ronan Collobert et al. 2011]

Common Neural Network based Word Embedding Approaches

- RNN based Language Model



$x_t \in R^{D\times 1}$ $U \in R^{d\times D}$ $w_t \in R^{d\times 1}$ $y_t \in R^{D\times 1}$

$$w_t = f\left(U \cdot x_t + W \cdot w_{t-1}\right)$$

f: Sigmoid function

$$y_t = g\left(V \cdot w_t\right)$$

g: Softmax function

[Tomas Mikolov et al. 2013]

Common Neural Network based Word Embedding Approaches

- Summary of Three Embedding Approaches
  - Both of CBOW and SENNA adopt Negative Sampling. (Balanced training)
  - Both of CBOW and SENNA based on Contextual Window.

    (Selective dilemma)
  - RNN embedding based on Historical Sequence.

    (Born with sequence processing)

    (Imbalanced training)

    (Bidirectional RNN embedding based on contextual sequence).

Neural Language Models Comparison

- Accuracies on Semantic-Syntactic Word Relationship test set with 640-dim word vector, and the same training data.

Model Architecture	Semantic-Syntactic Word Relationship test set		MSR Word Relatedness Test Set[20]
	Semantic Accuracy [%]	Syntactic Accuracy [%]	
RNNLM	9	36	35
NNLM (Bengio 2003)	23	53	47
CBOW	24	64	61
Skip-gram	55	59	56

- CBOW has the best syntactic and word relationship Information. Skip-gram has the best semantic information.

[T Mikolov et al., 2013]

Neural Language Models Comparison

- Accuracy on subset of the SSWR test, use word vectors from CBOW. Frequent 30k words used.

Dimensionality / Training Words	24M	49M	98M	196M	391M	783M
50	13.4	15.7	18.6	19.1	22.5	23.2
100	19.4	23.1	27.8	28.7	33.4	32.2
300	23.2	29.2	35.3	38.6	43.7	45.9
600	24.0	30.1	36.5	40.8	46.6	50.4

- Dimension: the larger, the better
- Training Words: the more, the better

[T Mikolov et al., 2013]

Neural Language Models Comparison

- Comparison of models trained using the DistBelief distributed framework.

Model	Vector Dimensionality	Training words	Accuracy[%]			Training time [days x CPU cores]
			Semantic	Syntactic	Total	
NNLM	100	6B	34.2	64.5	50.8	14 x 180
CBOW	1000	6B	57.3	68.9	63.7	2 x 140
Skip-gram	1000	6B	66.1	65.1	65.6	2.5 x 125

- It is possible to train high-quality word vectors just using a simple model.
- It is possible to obtain high-dimensional and accurate word vectors from a large dataset.

[T Mikolov et al., 2013]

Neural Language Models Comparison

- Comparison of models trained with the same data but different epochs. Accuracy is reported on the full Semantic-Syntactic data set.

Model	Epoch	Vector Dimensionality	Training Words	Accuracy[%]			Training Time [days]
				Semantic	Syntactic	Total	
CBOW	1	300	783M	13.8	49.9	33.6	0.3
CBOW	1	300	1.6B	16.1	52.6	36.1	0.6
CBOW	1	600	783M	15.4	53.3	36.2	0.7
Skip-gram	1	300	783M	45.6	52.2	49.2	1
Skip-gram	1	300	1.6B	52.2	55.1	53.8	2
Skip-gram	1	600	783M	56.7	54.5	55.5	2.5
CBOW	3	300	783M	15.5	53.1	36.1	1
Skip-gram	3	300	783M	50.0	55.9	53.3	3

- Skip-gram has better representation, but need more training time than CBOW.

[T Mikolov et al., 2013]

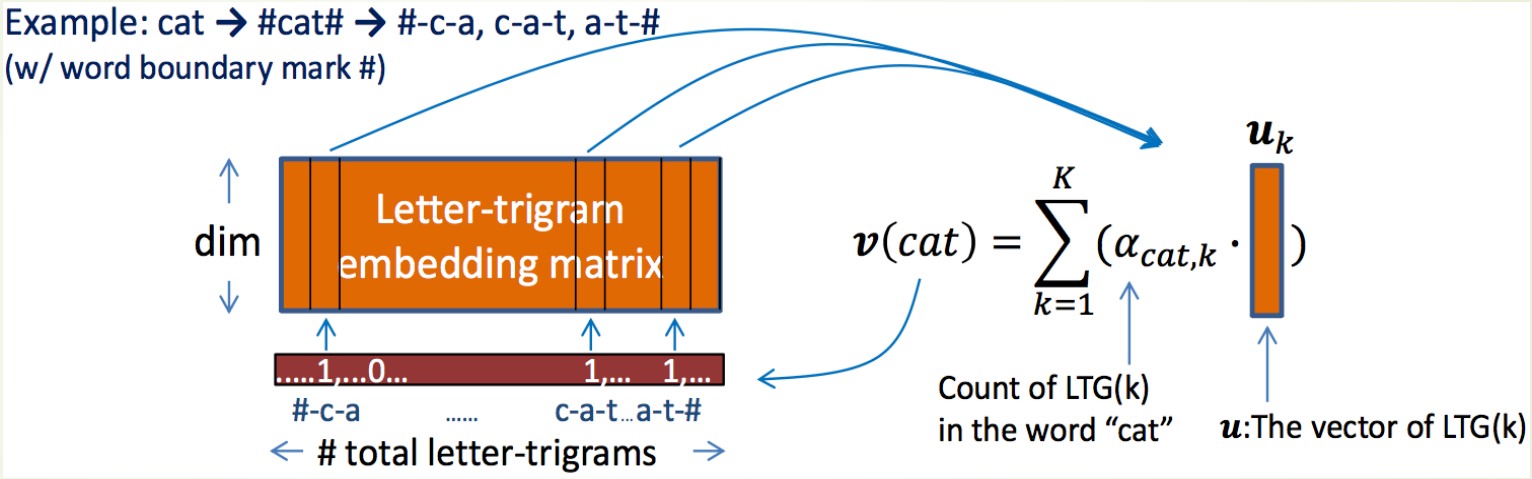
Common Neural Network based Word Embedding Approaches

- Extension of Word Embedding
- Motivation:
  - Vocabulary of real-world big data tasks could be huge (Heaps Rule!!!)

    >100M words in a modern commercial search engine.
  - Common phrases are well represented, rare phrases are terribly represented.
  - New words, misspellings, and word fragments frequently occur.
- Action:
  - Find the proper sub-word embedding. (Based on language itself)
    - Letter trigram
    - Radical ngram

Common Neural Network based Word Embedding Approaches

- Letter TriGram Embedding



- Evaluation Criterion (Collision Rate) $\approx 0.004\%$

Vocabulary Size	Unique LTG Observed	Collision Number
40K	10306	2
500K	30621	22
5M	49292	179

[PS Huang et al. 2013]

Common Neural Network based Word Embedding Approaches

- Experiments on the evaluation data set henceforth(16510 query and each of them are related to 15 urls)

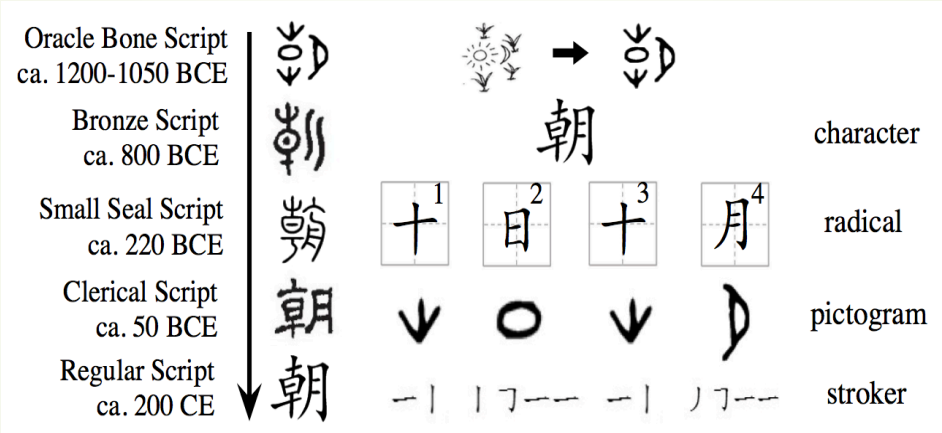
Models	NDCG@1	NDCG@3	NDCG@10
Word-Unigram DNN	0.342	0.410	0.486
Letter-Trigram DNN	0.362	0.425	0.498

- Letter-trigram Embedding has the better performance than Word-Unigram Embedding.

[PS Huang et al. 2013]

Common Neural Network based Word Embedding Approaches

- Radical embedding



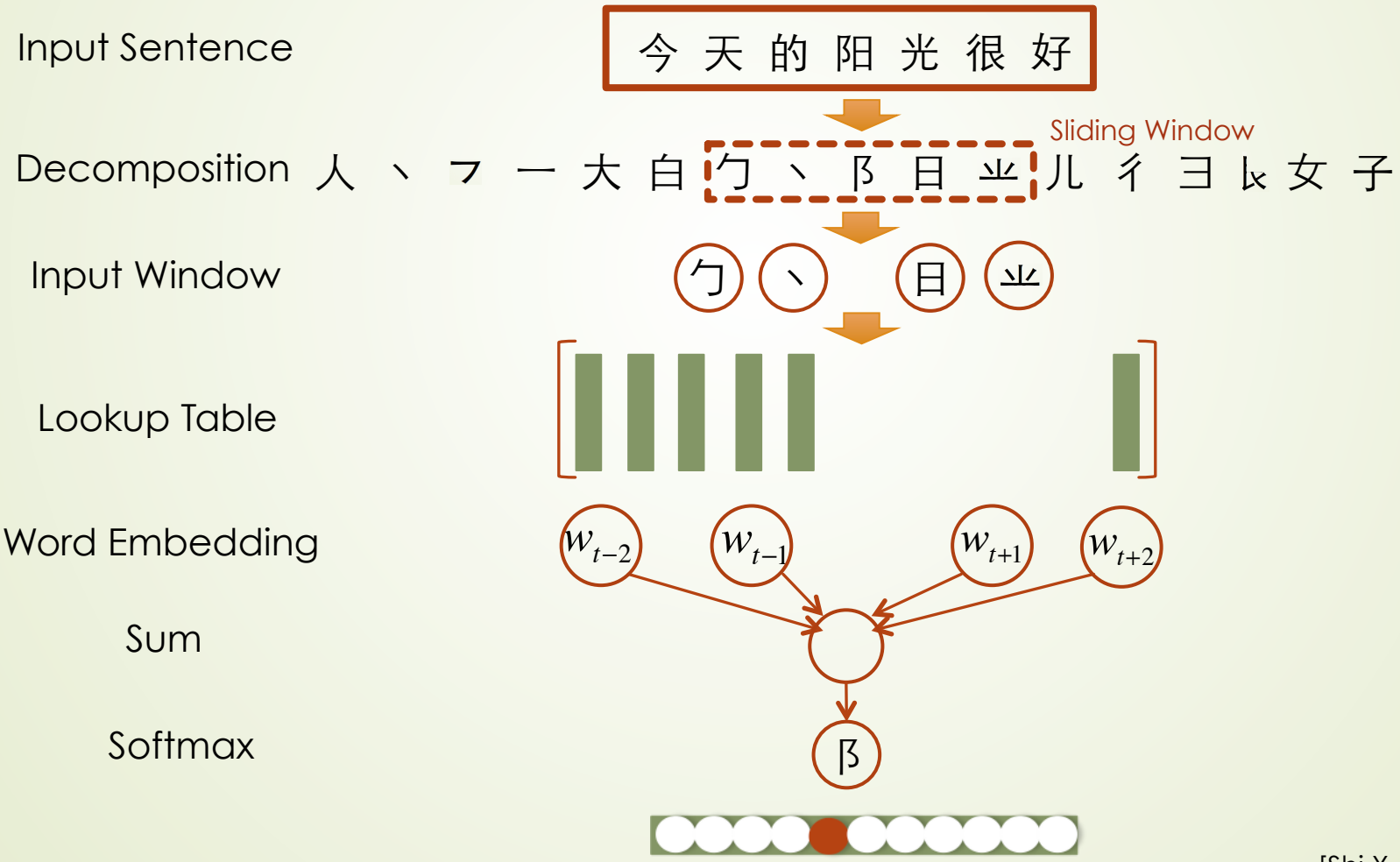
Decomposition of Chinese Characters

- Why is radical
  - Radical is the smallest semantic unit of Chinese
  - Radical is from the earliest pictograph of China
  - Sufficient resources of radical decomposition in Sogou.Inc

[Shi X. et al. 2015]

Common Neural Network based Word Embedding Approaches

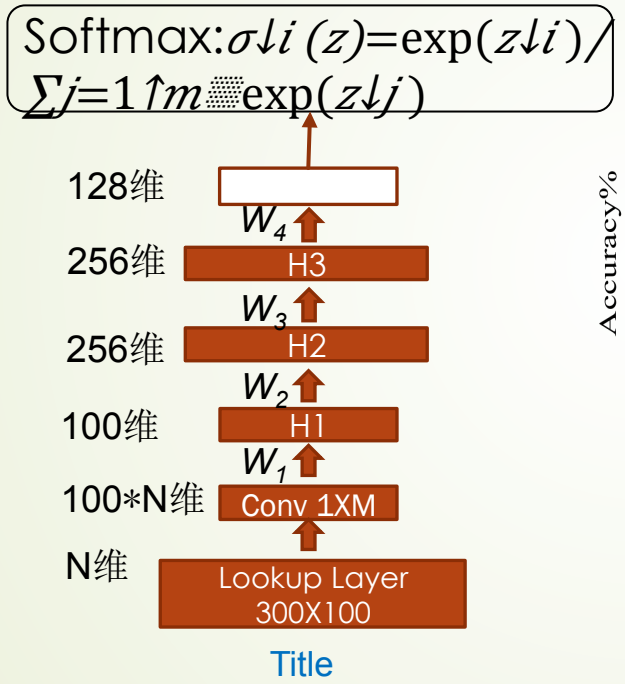
- Radical Embedding

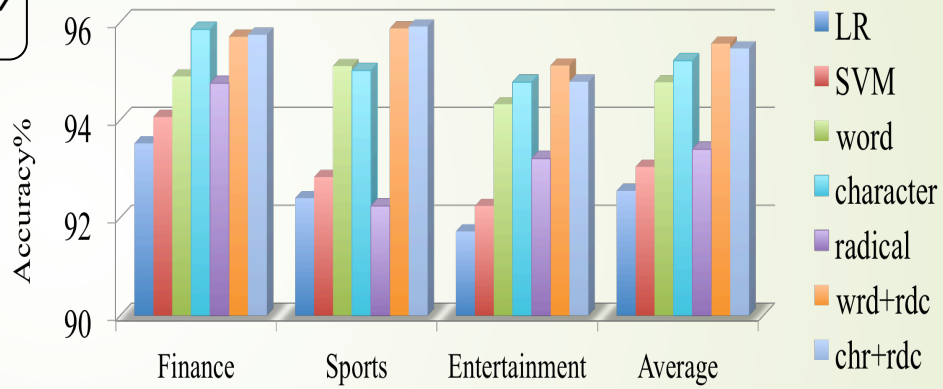


[Shi X. et al. 2015]

Common Neural Network based Word Embedding Approaches

- **Experiment of Radical Embedding on Short-Text Classifier**





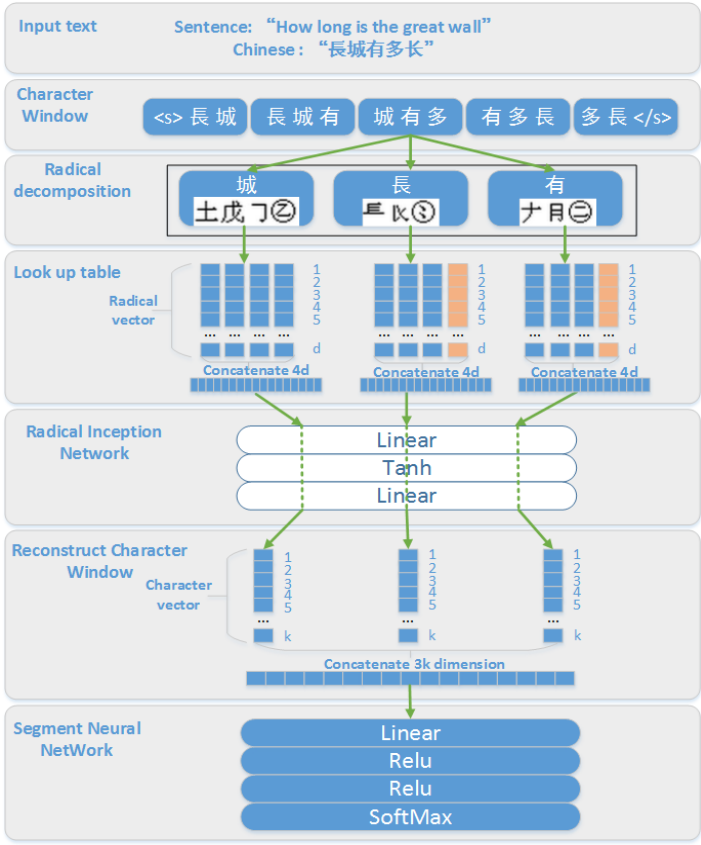
DataSet: train set 400K, test set 40K. Data from SogouCA, SogouCS news corpus.

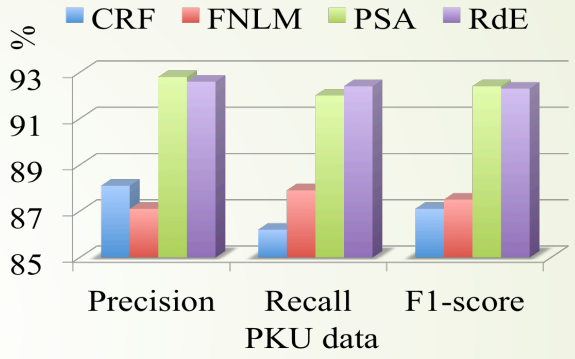
- **Results show radical embedding performs a little bad, but the composition of word+radical performs quite good.**

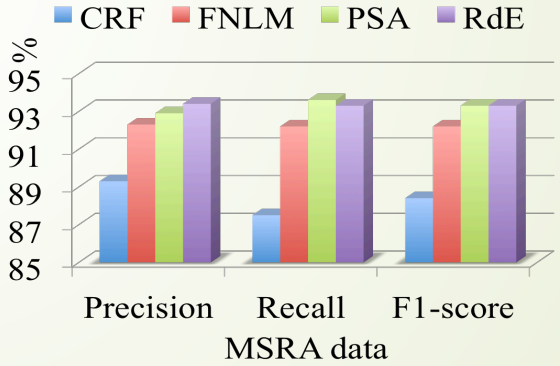
[Shi X. et al. 2015]

Common Neural Network based Word Embedding Approaches

- **Experiment of Radical Embedding on Chinese Word Segmentation**







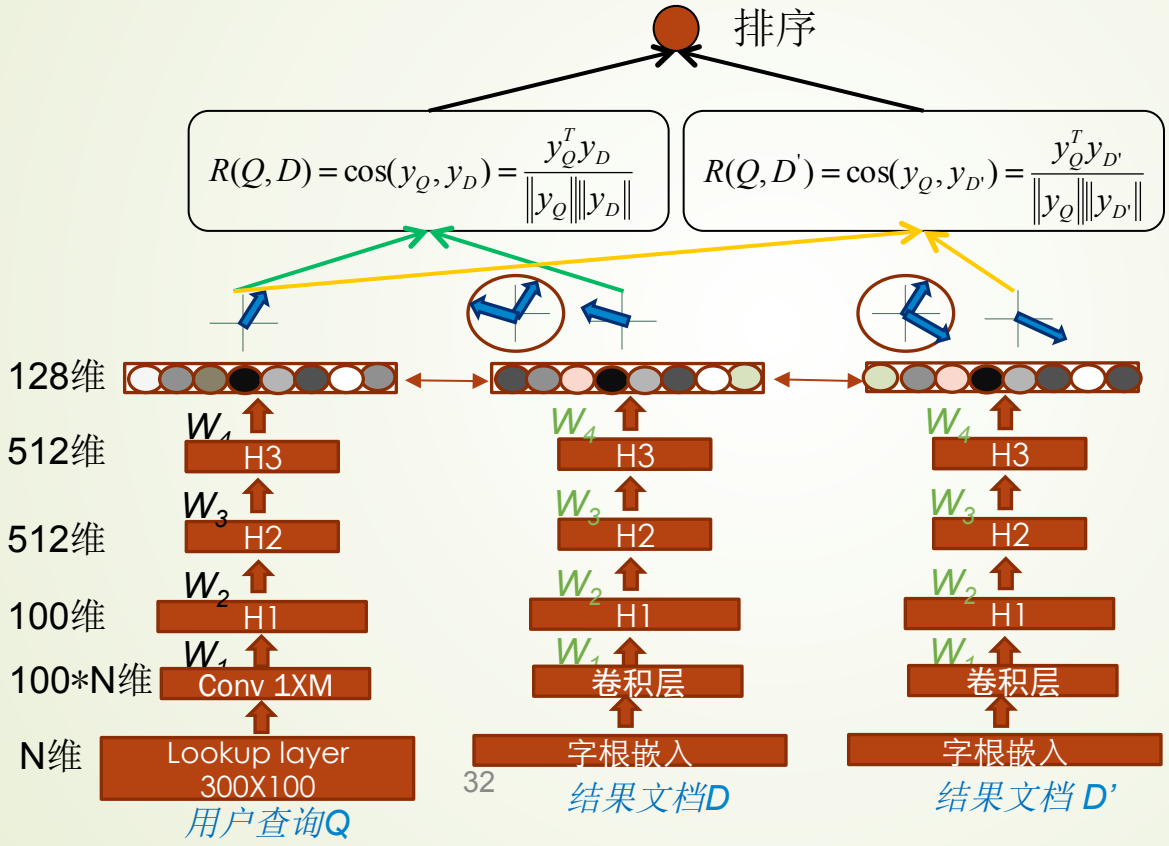
DataSet: 5.5M train set, 122K test set. data from (Emerson, 2005)

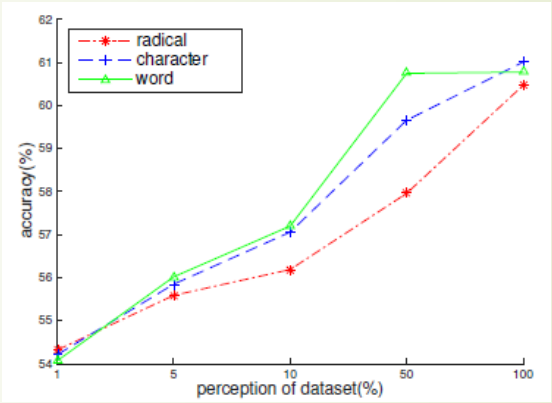
- **Results show radical embedding performs almost the same good as word embedding.**

[Shi X. et al. 2015]

Common Neural Network based Word Embedding Approaches

- **Experiment of Radical Embedding on Chinese Word Segmentation**





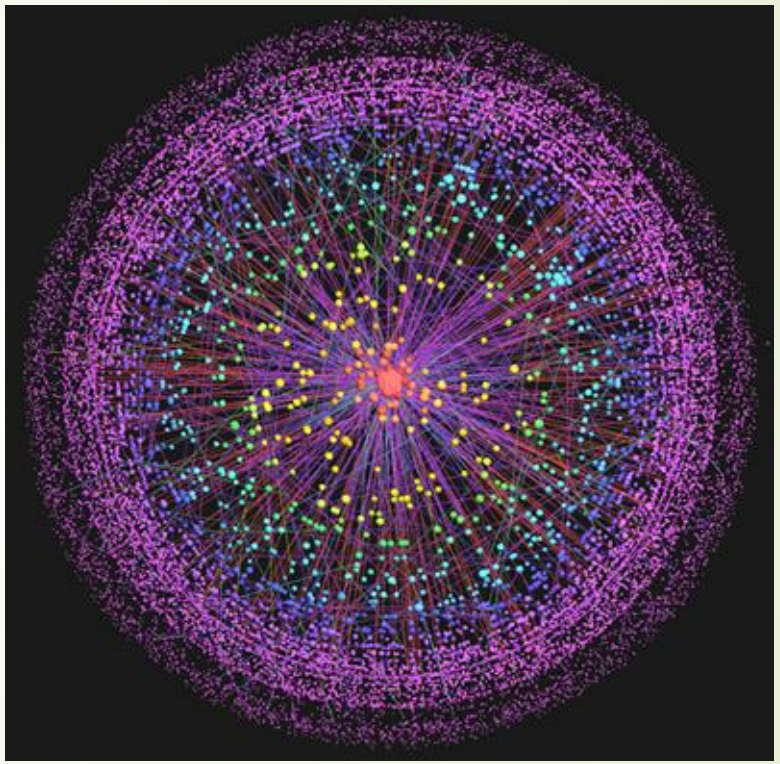
DataSet: 95M train set, 137K test set, data from user clickthrough log.

- **Results show except that radical embedding performs almost the same good as word embedding, radical embedding need more data to learning.**

[Shi X. et al. 2015]

Common Neural Network based Word Embedding Approaches

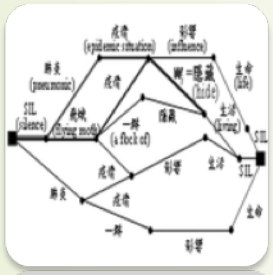
- Thinking
  - Flexible granularity: from radical-ngram embedding to character-ngram embedding to word-ngram embedding for Chinese NLP.
  - More information, such as morphology, synonym, syntactic, etc. needs to consider



Outline

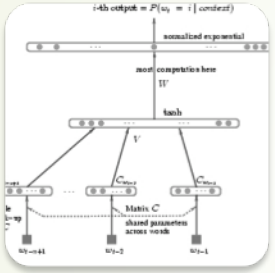
- Background of Deep Learning
- Deep learning for Chinese NLP
  - Neural Language Model
  - Feed-forward Neural Network
  - Recurrent Neural Network

The Roadmap of the Development of Chinese NLP



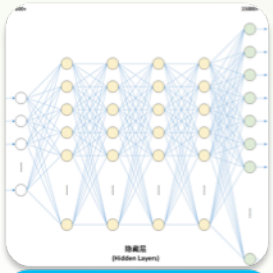
PLM(Probabilistic Language Model)

- Sparse representation
- Sparse feature, imbalanced learning
- Curse of dimension
- Lack of Semantic information
- High dimension Unfit for Neural network



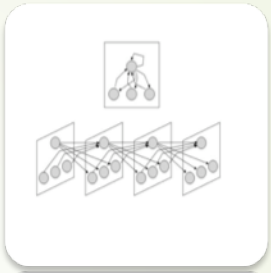
NLM(Neural Language Model)

- Dense representation
- Sparse feature , imbalanced learning
- Curse of dimension
- Semantic information
- Fit for neural network



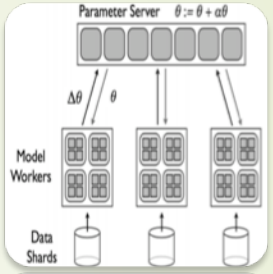
FNN(Feedforward Neural Network)

- Robust
- Capable for modeling Complex problem
- Arbitrary complexity of model
- Avoid the feature engineering
- Unfit for sequence task



RNN(Recurrent Neural network)

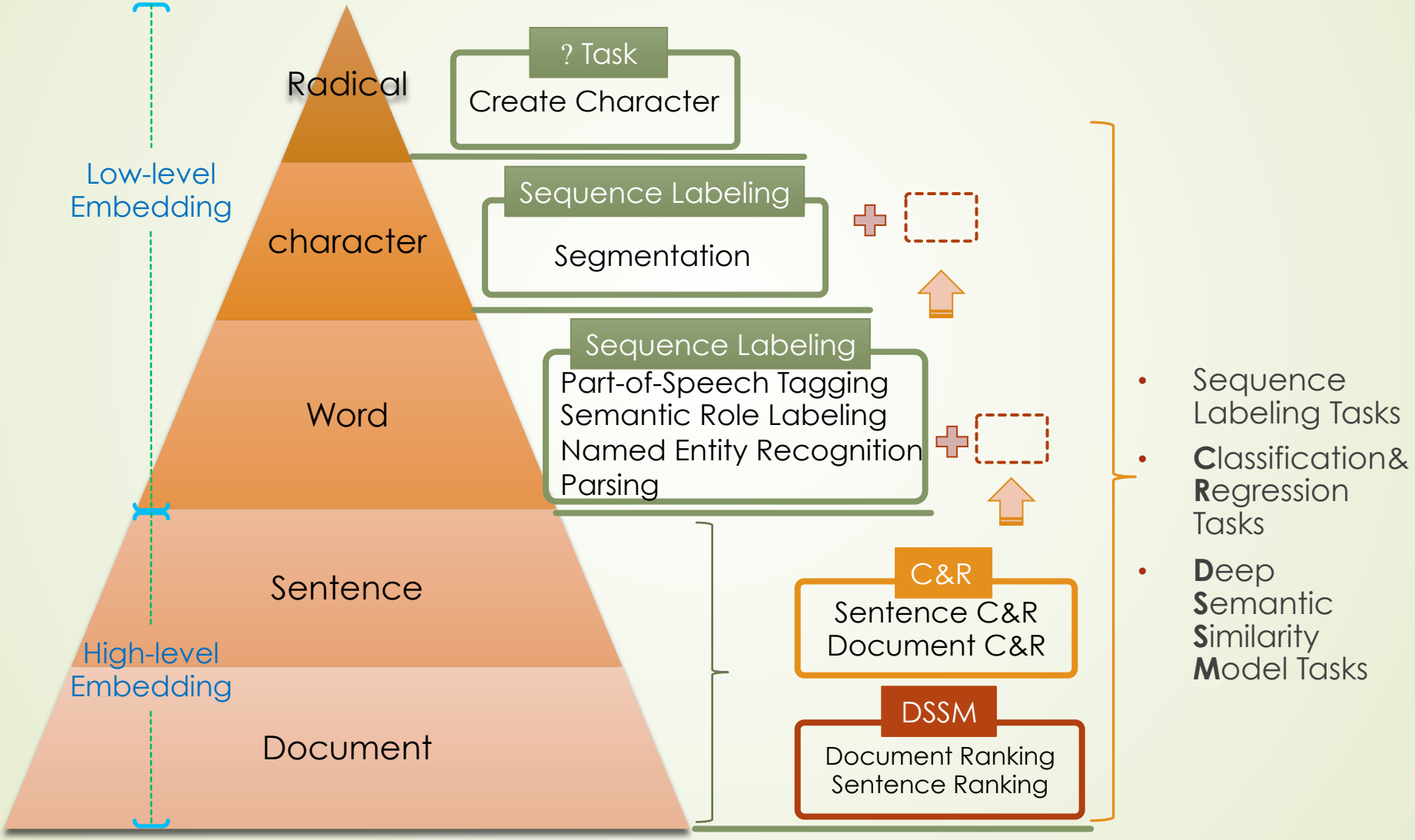
- Robust
- Capable for modeling Complex problem
- Arbitrary complexity of model
- Avoid the feature engineering
- Born with sequence property



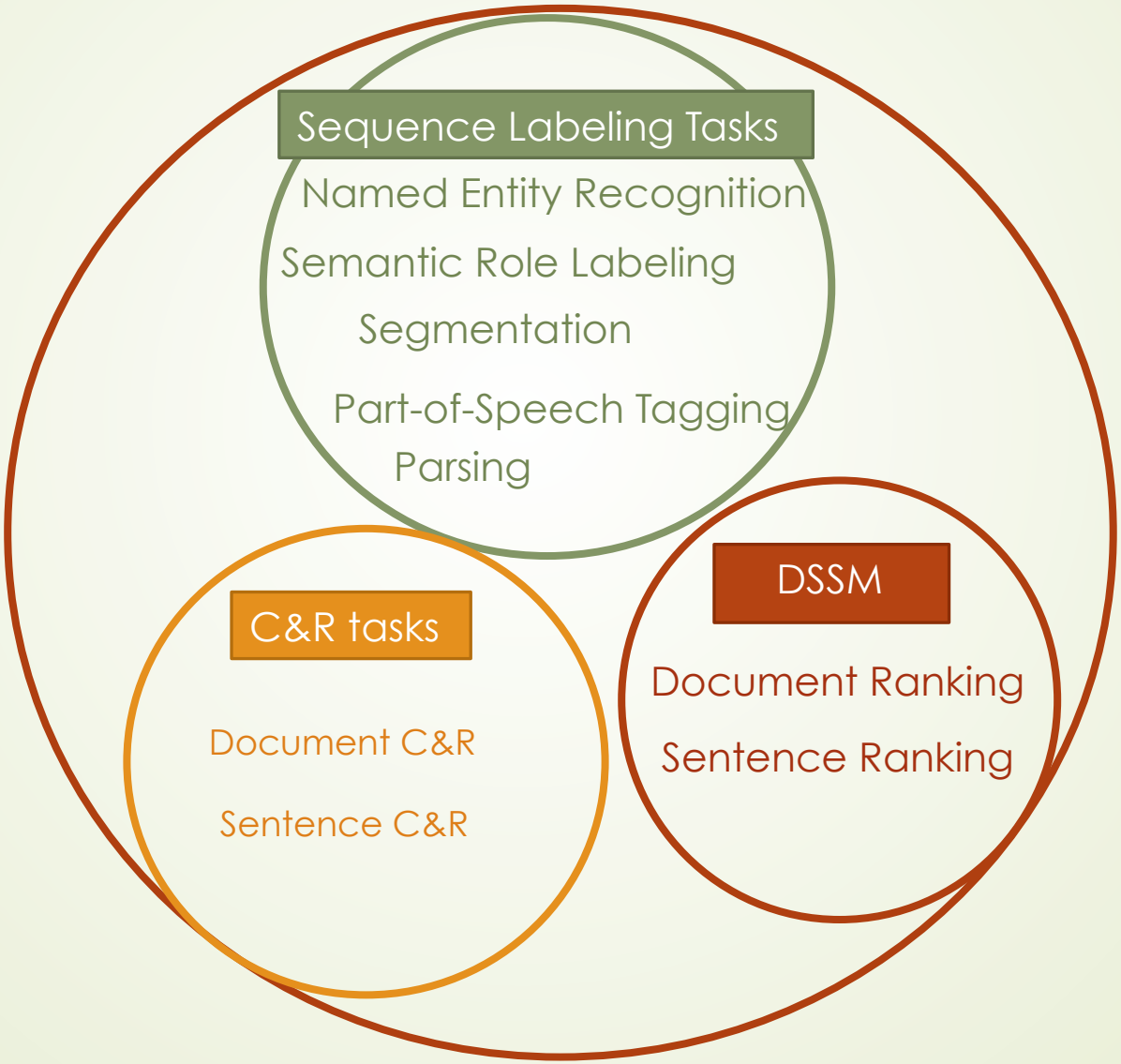
PDP(Parallel Distributed Platform)

- Large data set
- Parallel computing
- Widely algorithm supporting
- Scalable capacity

Main Tasks of the Pyramid of Chinese NLP



Main Tasks of Chinese NLP

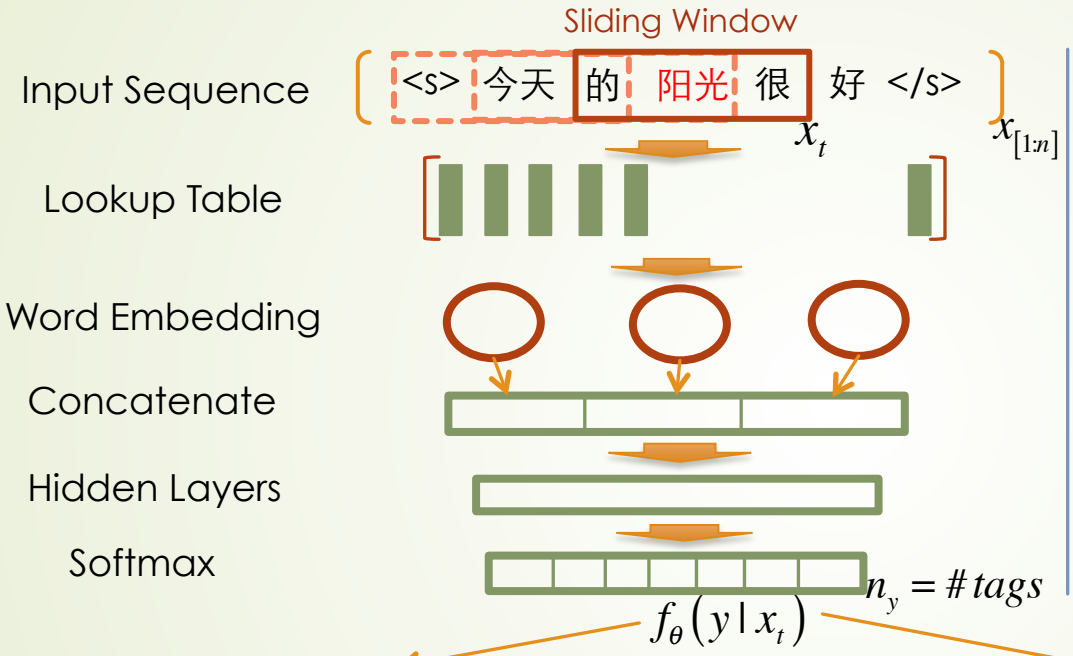


Outline

- Background of Deep Learning
- Deep learning for Chinese NLP
  - Neural Language Model
  - Feed-forward Neural Network
    - The Common Models for Natural Language Processing
      - Sequence Labeling
      - Classification & Regression
      - Deep Semantic Similarity Model
    - The Typical Applications and Experiments
  - Recurrent Neural Network

The Common Models for Natural Language Processing

- Sequence Labeling Tasks



Word-level likelihood

Each word in a sequence is considered independently!

$$p(y \mid x_t, \theta) = \frac{e^{f_\theta(y \mid x_t)}}{\sum_i e^{f_\theta(i \mid x_t)}}$$

So the word-level log-likelihood:

$$\log p(y \mid x_t, \theta) = f_\theta(y \mid x_t) - \log \sum_i e^{f_\theta(i \mid x_t)}$$

Sentence-level likelihood

Consider the dependency between word tags!

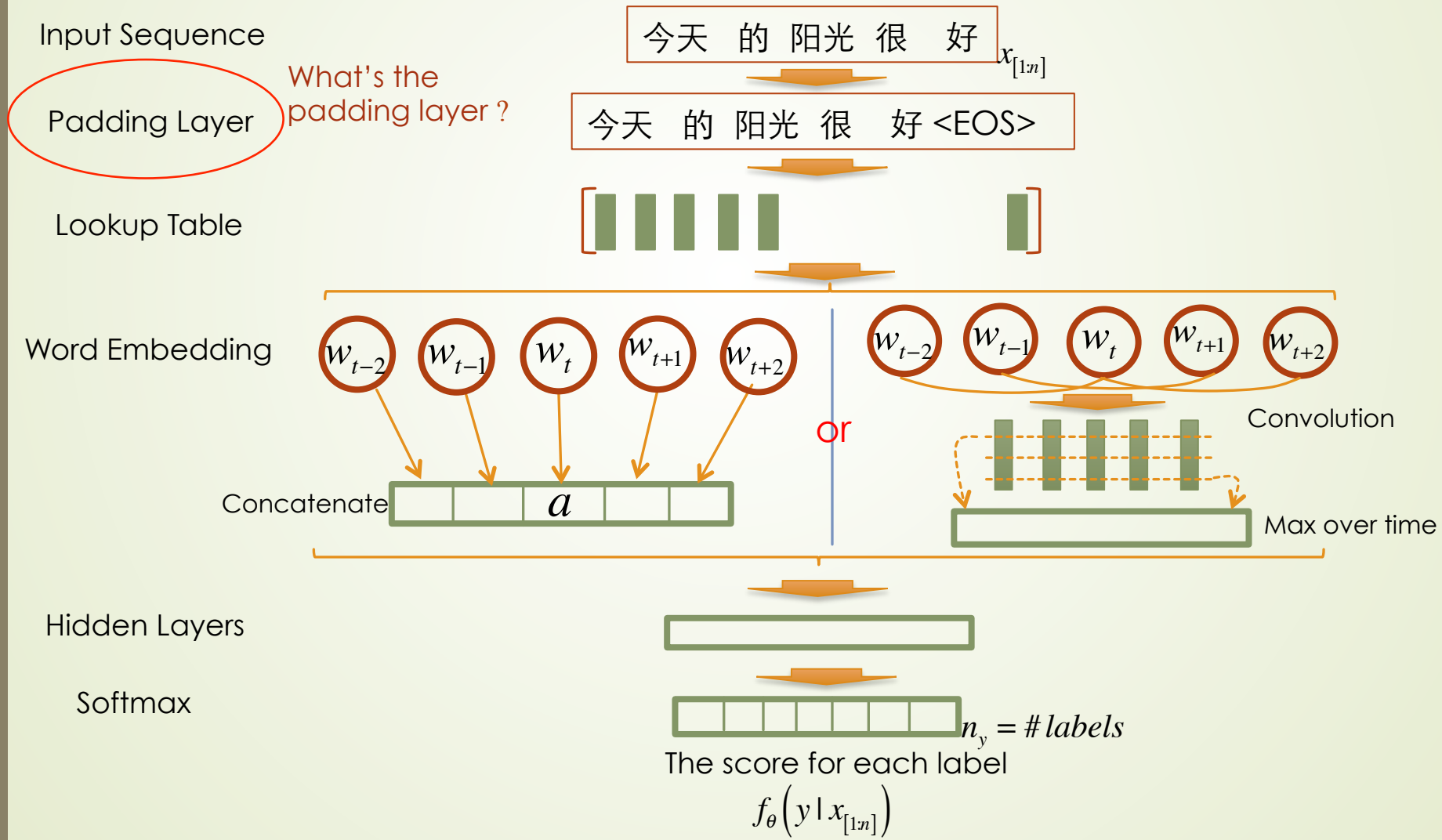
$$s(x_{[1:n]}, y_{[1:n]}, \theta) = \sum_{t=1}^{n} \left(A_{y_{t-1} y_t} + f_\theta(y_t \mid x_t) \right)$$

So the sentence-level log-likelihood:

$$\log p(y_{[1:n]}, x_{[1:n]}, \theta) = s(x_{[1:n]}, y_{[1:n]}, \theta) - \log \sum_{\forall i_{[1:n]}} s(x_{[1:n]}, i_{[1:n]}, \theta)$$

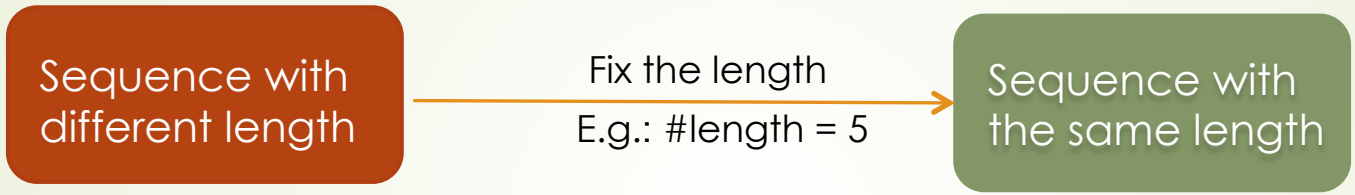
The Common Models for Natural Language Processing

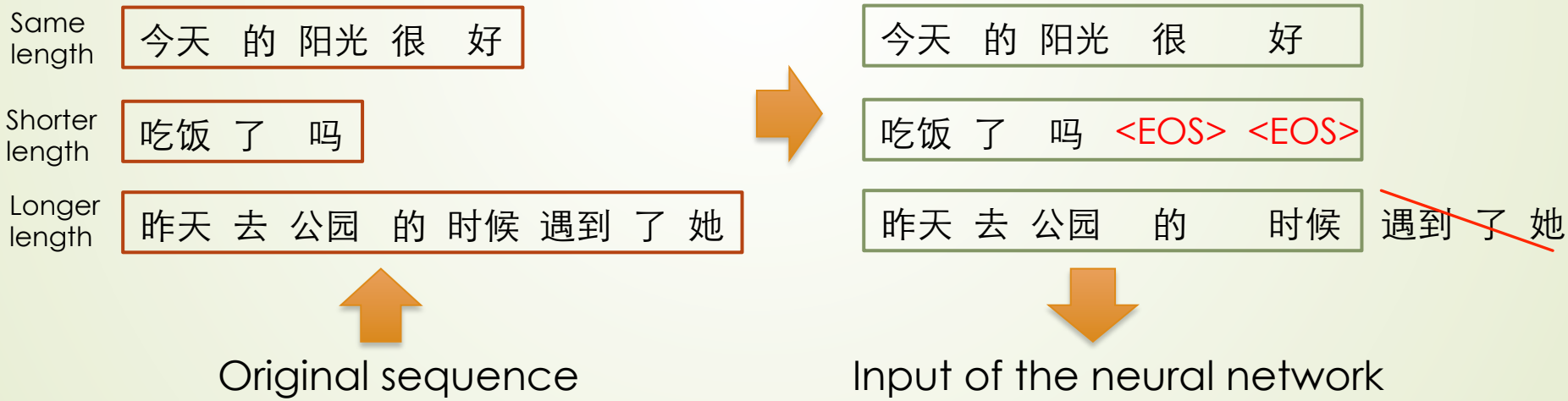
- Classification & Regression Tasks



The Common Models for Natural Language Processing

- Padding

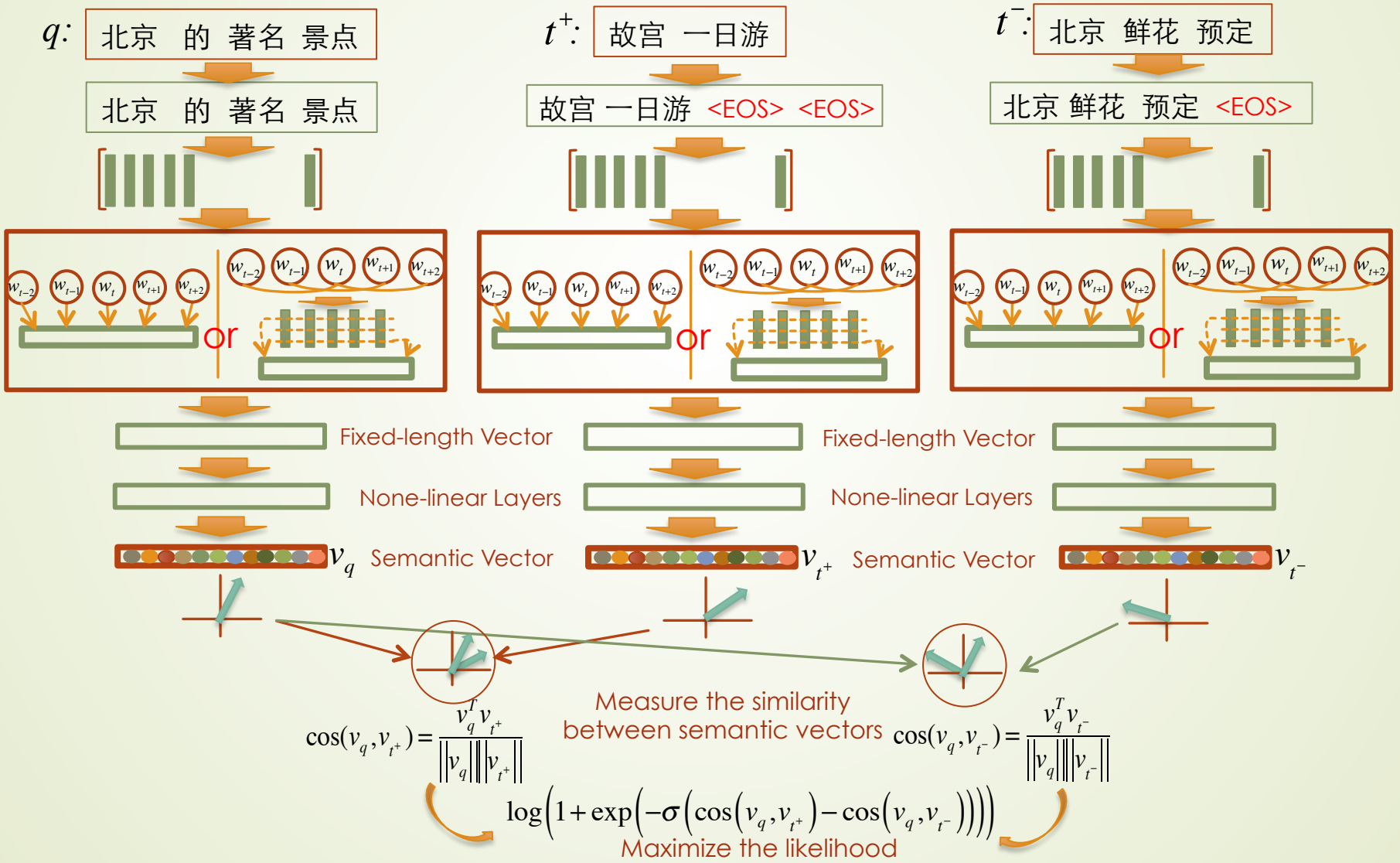




Make it available for batch learning !

The Common Models for Natural Language Processing

- DSSM

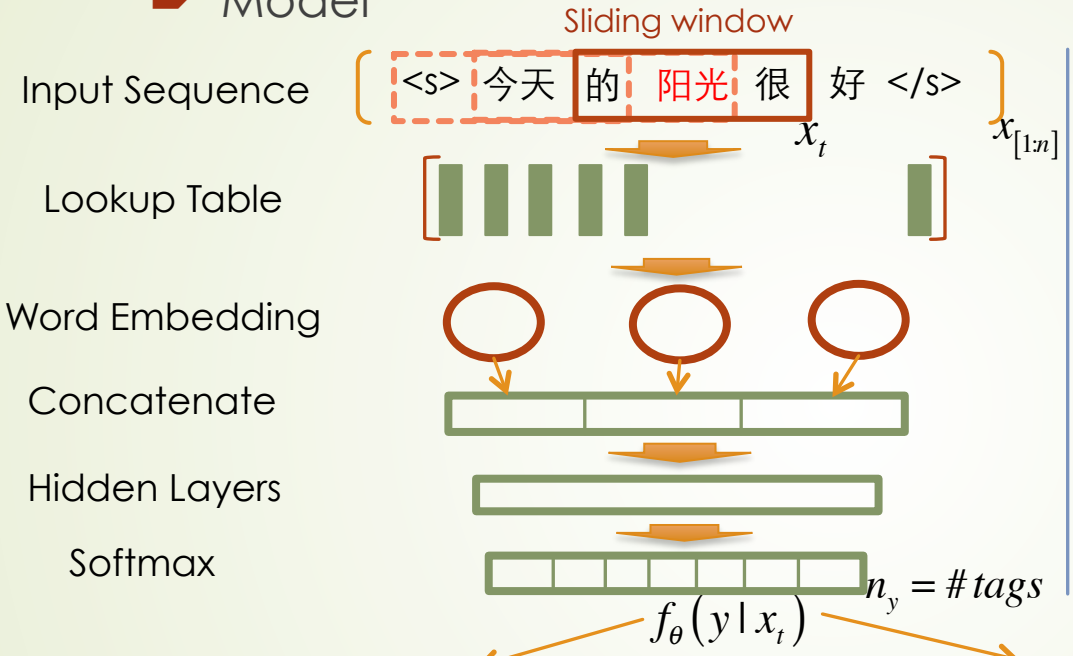


Outline

- Background of Deep Learning
- Deep learning for Chinese NLP
  - Neural Language Model
  - Feed-forward Neural Network
    - The Common Models for Natural Language Processing
    - The Typical Applications and Experiments
  - Recurrent Neural Network

The Typical Applications: Sequence Labeling

- Model



Word-level likelihood

$$p(y \mid x_t, \theta) = \frac{e^{f_\theta(y \mid x_t)}}{\sum_i e^{f_\theta(i \mid x_t)}}$$

So the word-level log-likelihood:

$$\log p(y \mid x_t, \theta) = f_\theta(y \mid x_t) - \log \sum_i e^{f_\theta(i \mid x_t)}$$

Sentence-level likelihood

$$s(x_{[1:n]}, y_{[1:n]}, \theta) = \sum_{t=1}^{n} \left(A_{y_{t-1} y_t} + f_\theta(y_t \mid x_t) \right)$$

So the sentence-level log-likelihood:

$$\log p(y_{[1:n]}, x_{[1:n]}, \theta) = s(x_{[1:n]}, y_{[1:n]}, \theta) - \log \sum_{\forall i_{[1:n]}} s(x_{[1:n]}, i_{[1:n]}, \theta)$$

[Collobert, R. et al., 2011]

Experiments on Sequence Labeling

- The tasks : Part-Of-Speech tagging (POS), Chunking (CHUNK), Named Entity Recognition (NER) and Semantic Role Labeling (SRL)
- The experimental setup:

Task	Benchmark	Data Set	Training set (#tokens)	Test set (#tokens)
POS	Toutanova et al. (2003)	WSJ	Section 0-18 (129,654)	Sections 22-24 129,654
Chunking	CoNLL 2000	WSJ	Section 15-18 (211,727)	Section 20 47,377
NER	CoNLL 2003	Reuters	Eng.train 203,621	Eng.testb 46,435
SRL	CoNLL 2005	WSJ	Sections 2-21 950,028	Section 23 +3 Brown sections 63,843

[Collobert, R. et al., 2011]

Experiments on Sequence Labeling

- The table reports supervised results with both the word-level log-likelihood (WLL) and the sentence-level log-likelihood (SLL), compared with the state-of-art system

Approach	POS (PWA)	CHUNK (F1)	NER (F1)	SRL (F1)
Bidirectional Graphical Model (Toutanova et al. 2003)	97.24	-	-	-
CRF-based Model (Sha et al. 2003)	-	94.29	-	-
Semi-Supervised Learning (Ando et al. 2005)	-	-	89.31	-
Joint Inference Model (Koomen et al. 2005)	-	-	-	77.92
NN+WLL	96.31	89.13	79.53	55.40
NN+SLL	96.37	90.33	81.47	70.99
NN+WLL+LM1	97.05	91.91	85.68	58.18
NN+SLL+LM1	97.10	93.65	87.58	73.84
NN+WLL+LM2	97.14	92.04	86.96	58.34
NN+SLL+LM2	97.20	93.63	88.67	74.15

LM1: Wikipedia LM2: Wikipedia+Reuters RCV1

- The NN performs a little worse than the state-of-art system but it is a unified model.
- The SLL performs better than the WLL, benefit from the use of context.
- The initialization with LM significantly boosts the generalization performance of the supervised networks .

The Typical Applications: Chinese Word Segmentation

- Models

PSA (Perception-Style Algorithm) based Model [Zheng, X. et al. 2013]

- A modification of Collobert's model, Viterbi algorithm is applied to find the best tag path.
- A perception-style algorithm is applied to speed up.

MMTNN (Max-Margin Tensor Neural Network) [Wen, Z. et al. 2014]

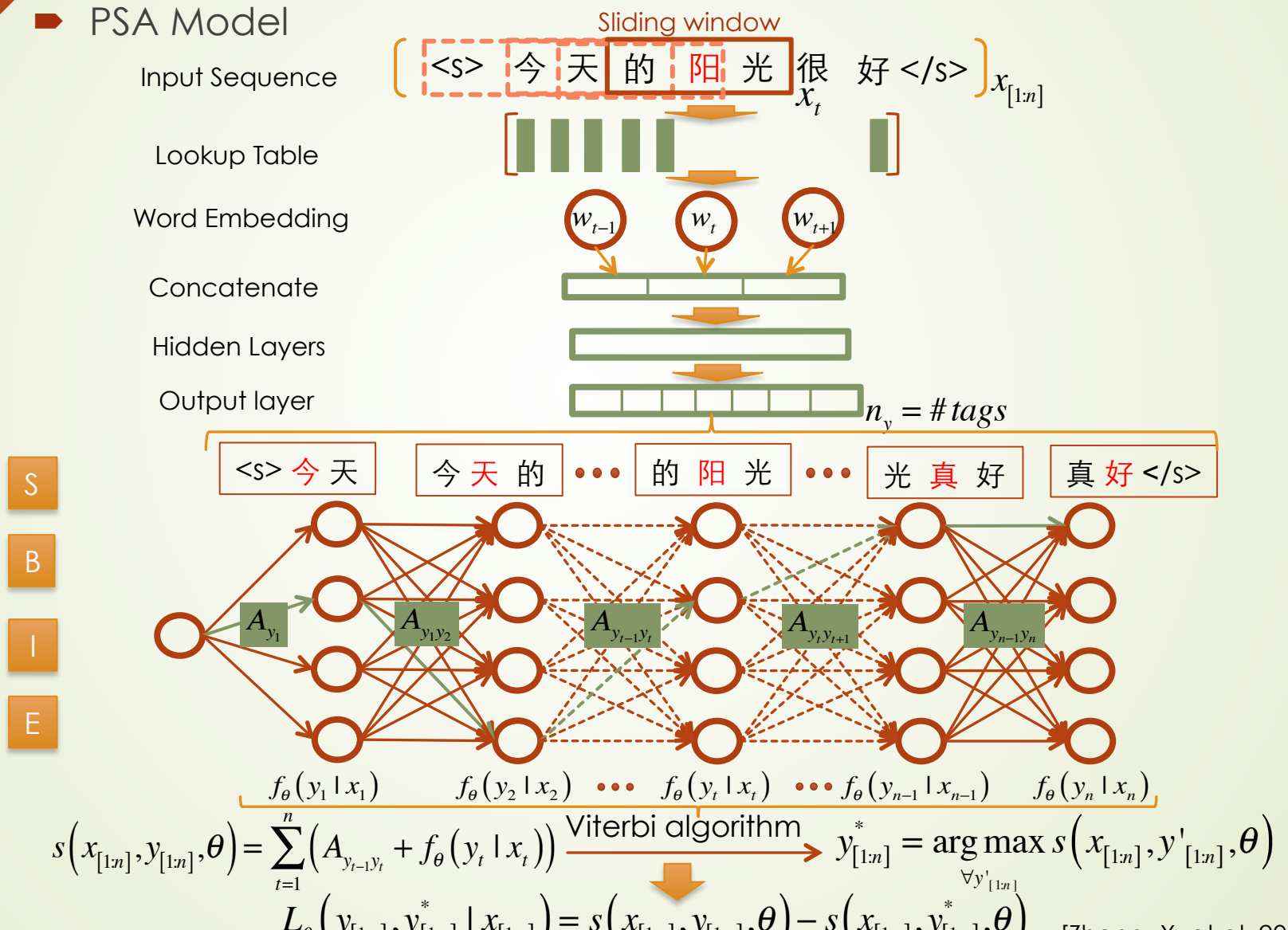
- The tensor-based neural network model the interaction between tags and context characters better.
- The Max-Margin criterion is applied instead of softmax.

RdE (Radical Embedding) based Model [Shi, X. et al. 2015]

- The radical embedding is used instead of word embedding.
- The input space is compacted significantly while the results are comparable.

The Typical Applications: Chinese Word Segmentation

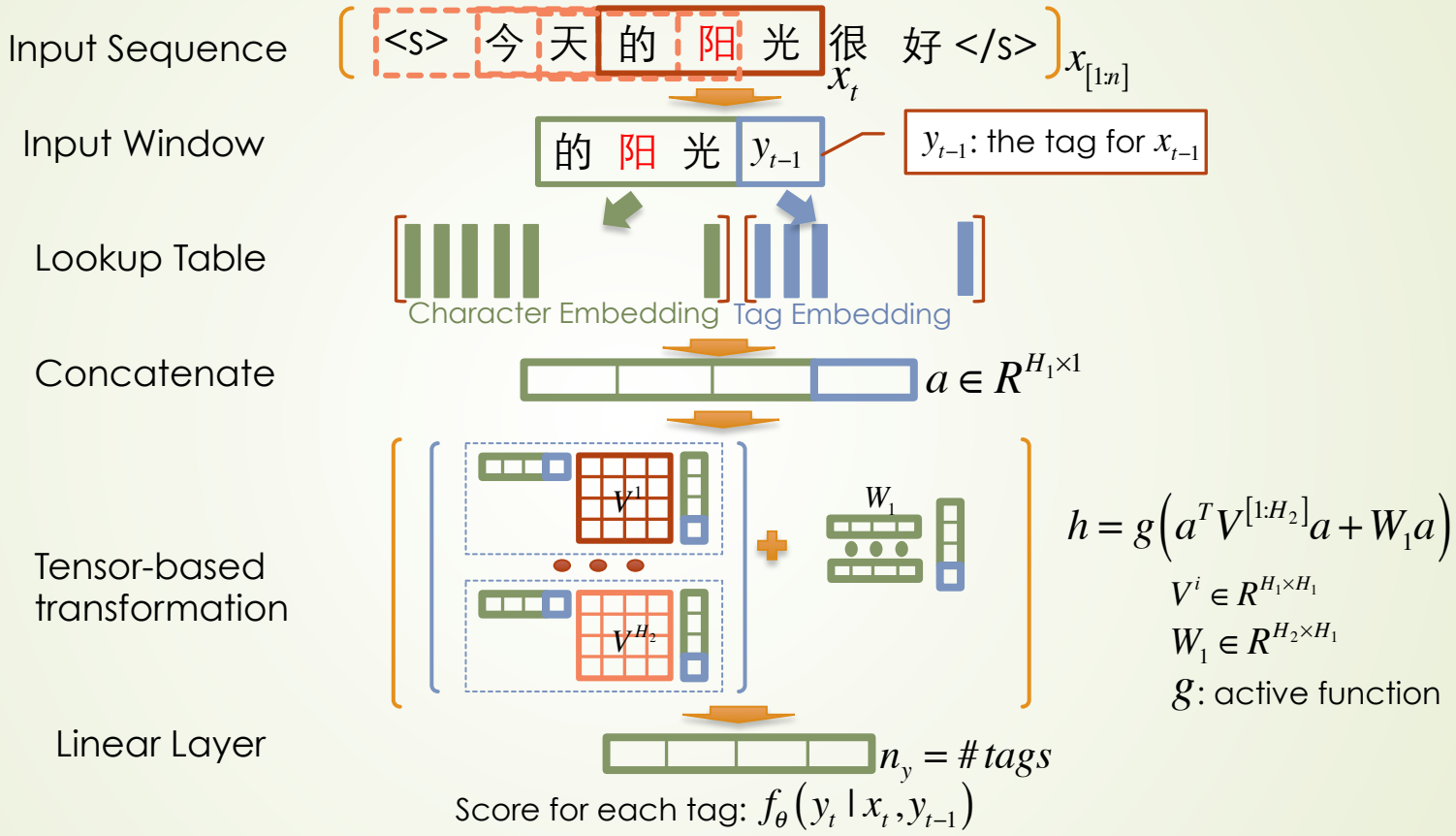
- PSA Model



[Zheng, X. et al. 2013]

The Typical Applications: Chinese Word Segmentation

- MMTNN Model



The score of a tag sequence: $s\left(x_{[1:n]}, y_{[1:n]}, \theta\right) = \sum_{t=1}^{n} f_\theta\left(y_t \mid x_t, y_{t-1}\right)$

Minimize the object: $J(\theta) = \frac{1}{m}\sum_{i=1}^{m} l_i(\theta) + \frac{\lambda}{2}\|\theta\|^2$

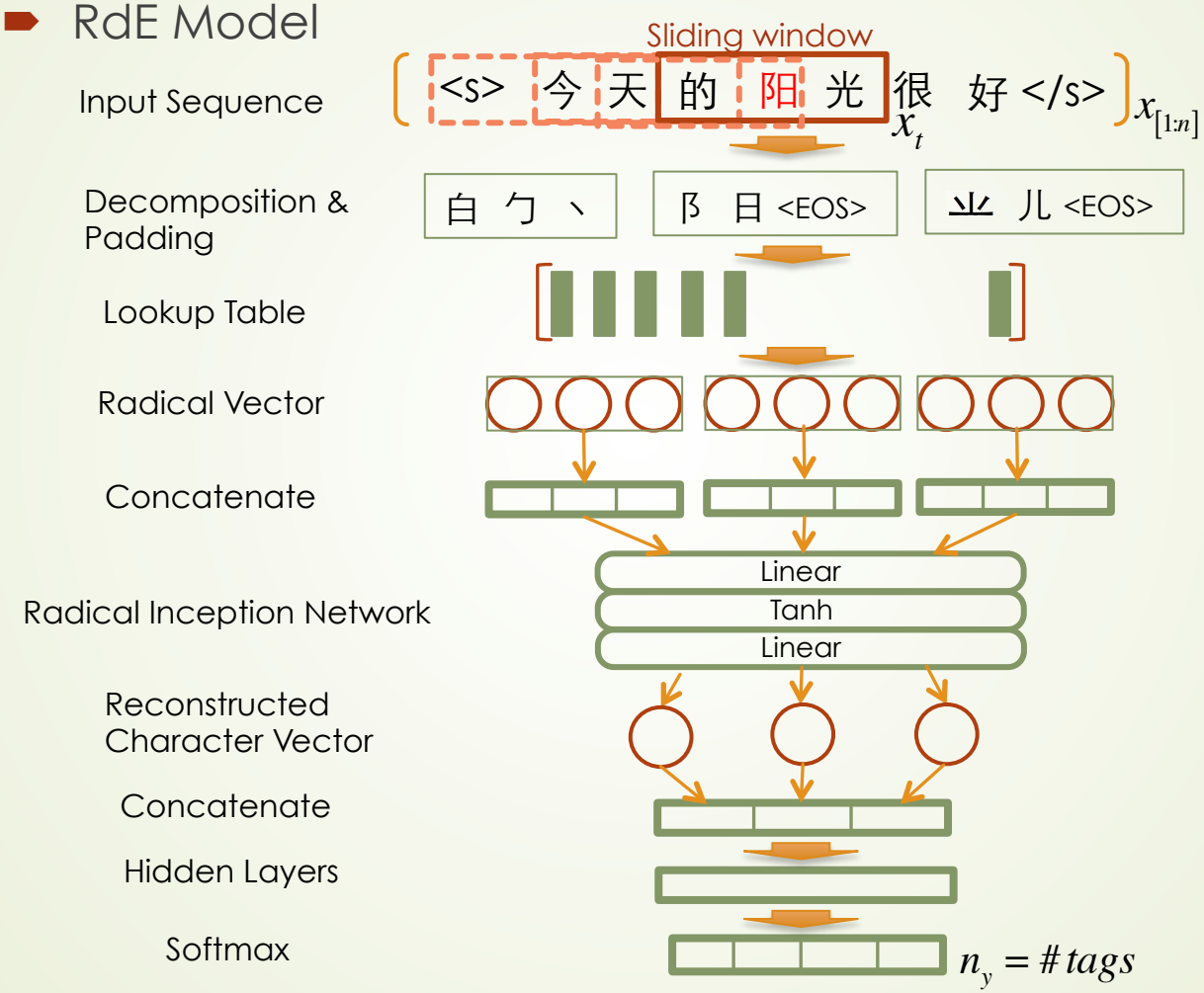
where $l_i(\theta) = \max_{\hat{y} \in Y(x_i)} \left(s\left(x_i, \hat{y}, \theta\right) + \Delta\left(y_i, \hat{y}\right)\right) - s\left(x_i, y_i, \theta\right)$

y_i : correct tag sequence

$\hat{y}$: highest scoring tag sequence

The Typical Applications: Chinese Word Segmentation

- RdE Model



- Given a sequence window, the network outputs the scores of all the possible tags 'S', 'B', 'I', 'E' for the character in the center of the window

[Shi, X. et al. 2015]

Experiments on Chinese Word Segmentation

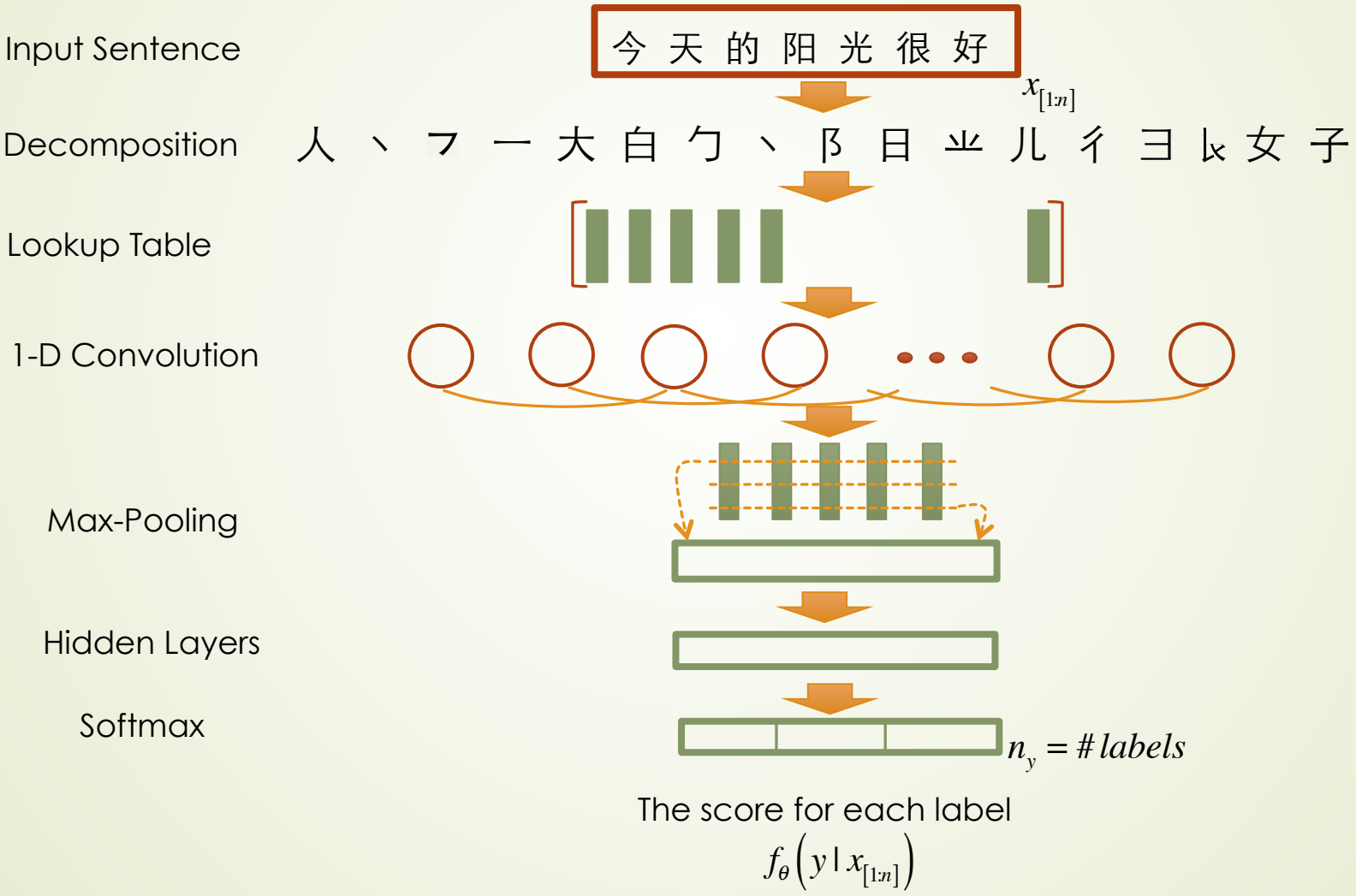
- Dataset: PKU and MSR, as provided by (Emerson, 2005)

Data	Approach	Precision	Recall	F1
PKU	CRF	88.1	86.2	87.1
	PSA Model	92.8	92.0	92.4
	MMTNN Model	**93.7**	**93.4**	**93.5**
	RdE Model	92.6	92.1	92.3
MSR	CRF	89.3	87.5	88.4
	PSA Model	92.9	93.6	92.3
	MMTNN Model	**94.6**	**94.2**	**94.4**
	RdE Model	93.4	93.3	93.3

- Results show that the neural network models perform better than the CRF-based model.
- The MMTNN model outperforms the other two models.
- The model based on radical embedding performs as good as PSA model with less parameters.

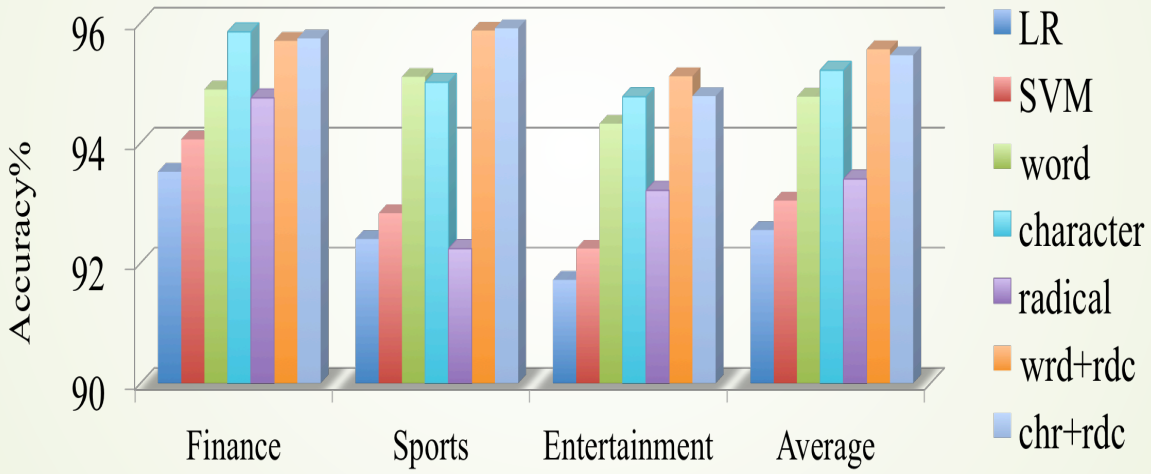
The Typical Applications: Chinese Short-text Classification

- Model



Experiments on Chinese Short-text Classification

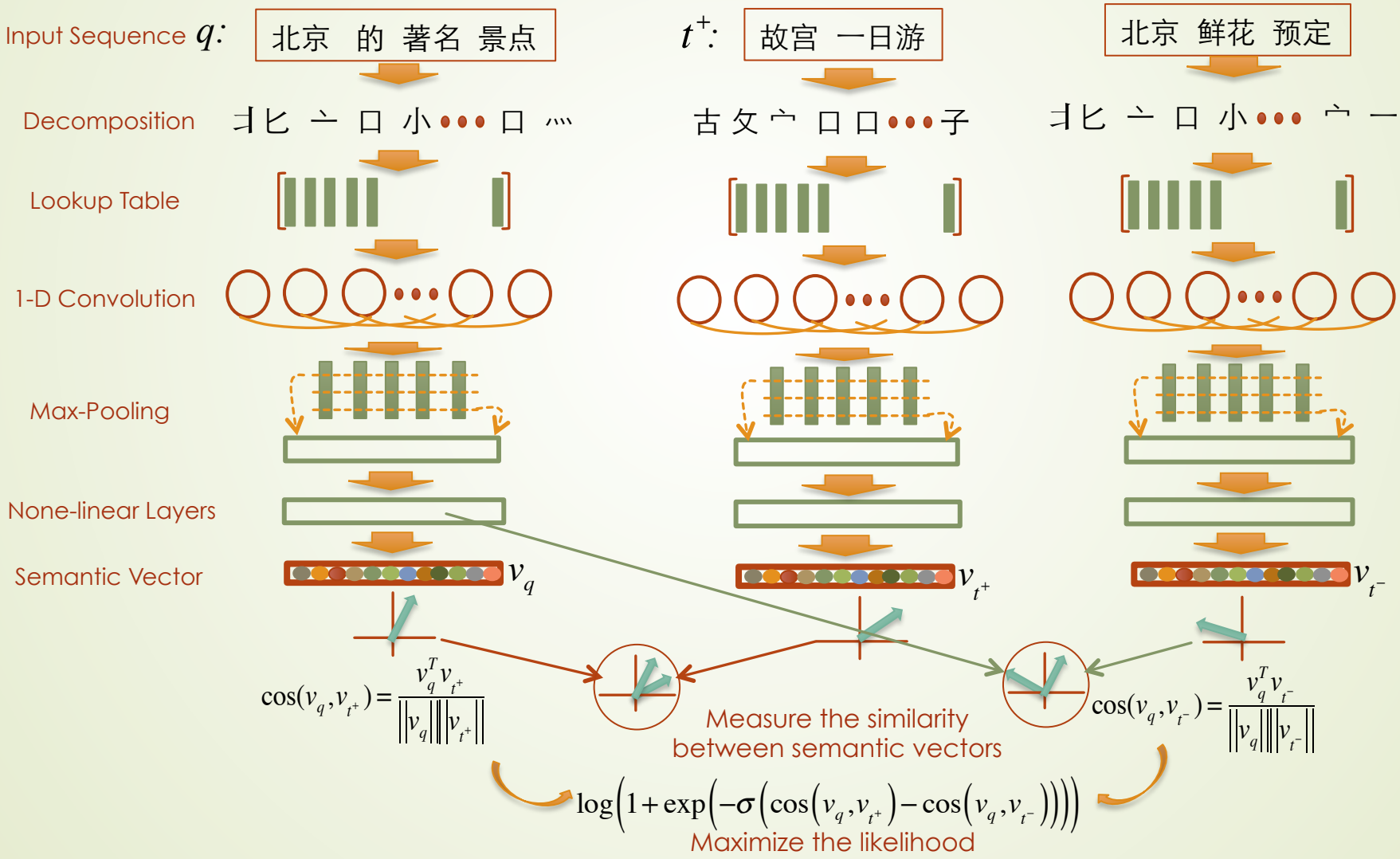
- Dataset: train set 400K， test set 40K. Data from SogouCA， SogouCS news corpus



- Result shows FNN model with embedding performs better than the traditional methods, and the combination of different embedding methods could improve the performance.

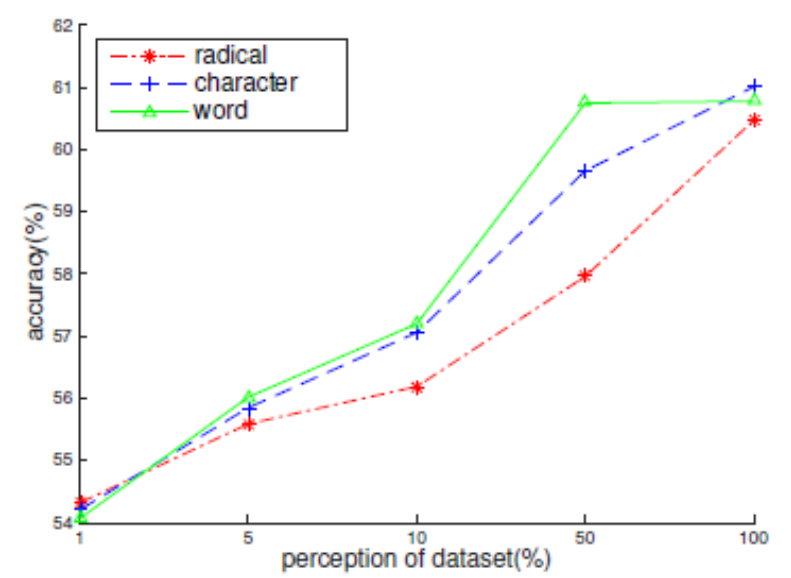
The Typical Applications: Chinese Search Ranking

- Model



Experiments on Chinese Search Ranking

- Dataset: 95M train set, 137K test set, data from user clickthrough log

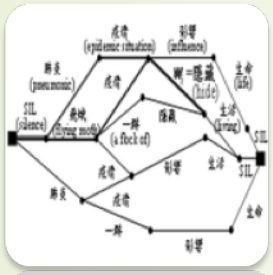


- Result shows that with more data, the three kinds of embedding methods almost achieve the same good performance.

Outline

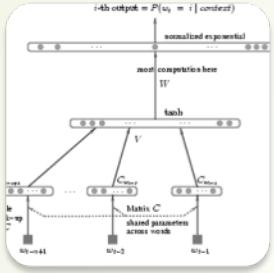
- Background of Deep Learning
- Deep learning for Chinese NLP
  - Neural Language Model
  - Feed-forward Neural Network
  - Recurrent Neural Network

The Roadmap of the Development of Chinese NLP



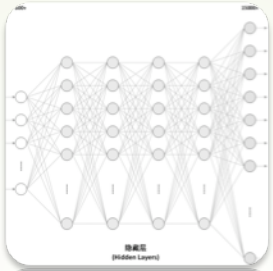
PLM(Probabilistic Language Model)

- Sparse representation
- Sparse feature, imbalanced learning
- Curse of dimension
- Lack of Semantic information
- High dimension Unfit for Neural network



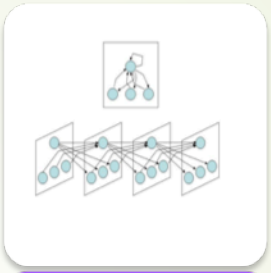
NLM(Neural Language Model)

- Dense representation
- Sparse feature, imbalanced learning
- Curse of dimension
- Semantic information
- Fit for neural network



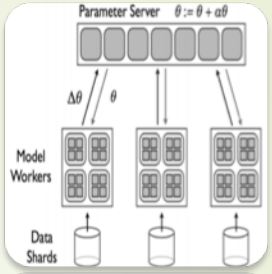
FNN(Feedforward Neural Network)

- Robust
- Capable for modeling Complex problem
- Arbitrary complexity of model
- Avoid the feature engineering
- Unfit for sequence task



RNN(Recurrent Neural network)

- Robust
- Capable for modeling Complex problem
- Arbitrary complexity of model
- Avoid the feature engineering
- Born with sequence property



PDP(Parallel Distributed Platform)

- Large data set
- Parallel computing
- Widely algorithm supporting
- Scalable capacity

From Feed-forward Neural Network to Recurrent Neural Network

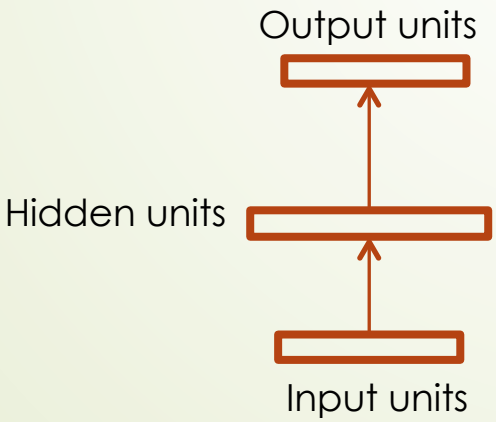
- Motivation

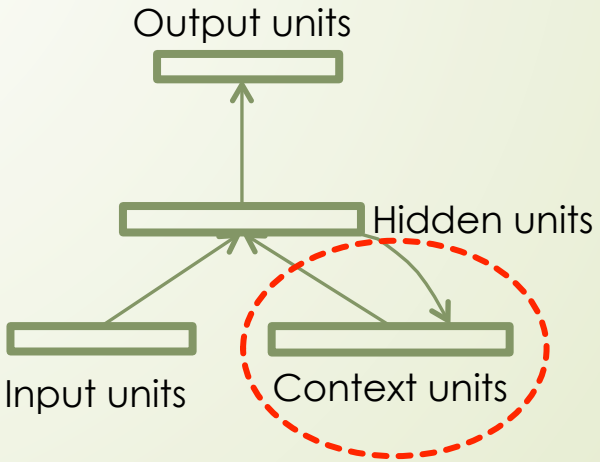
FNN

- Has to use fixed length context
- Lack any form of memory

RNN

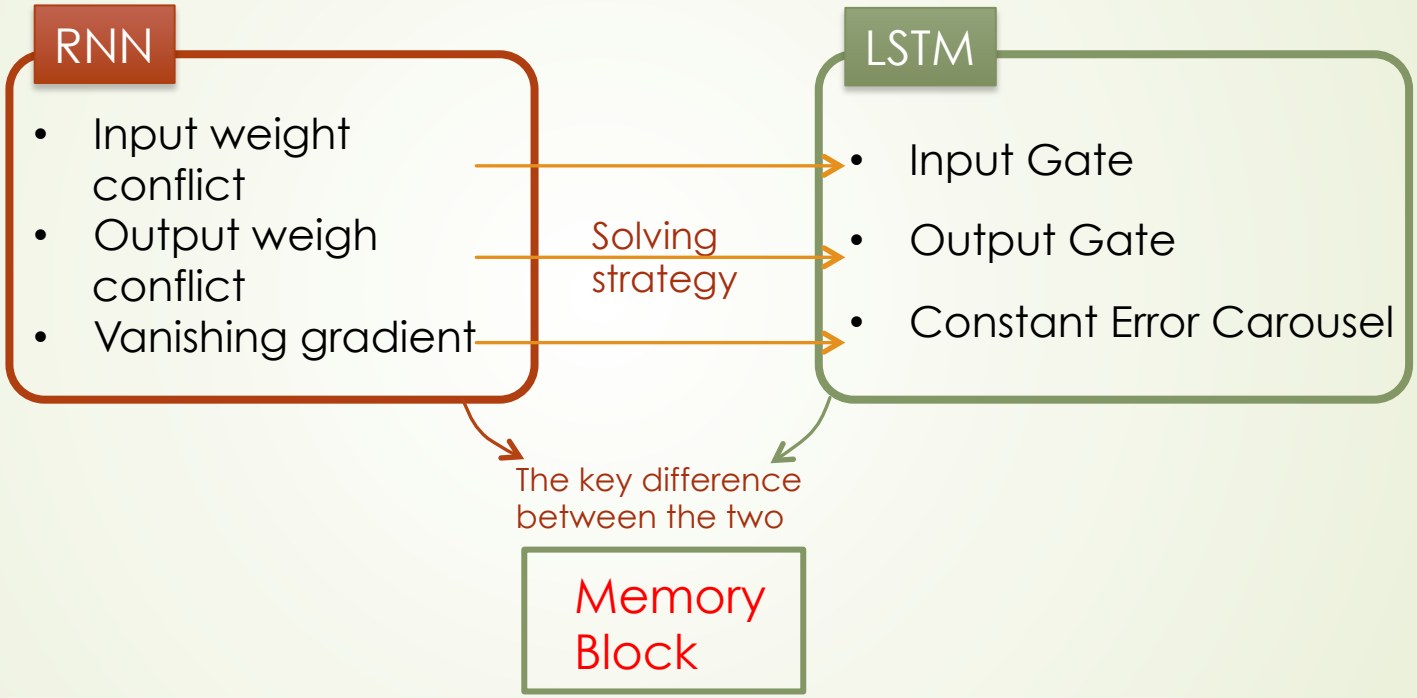
- The context length was extended to indefinite
- The ability to memorize





From RNN to Long Short Term Memory

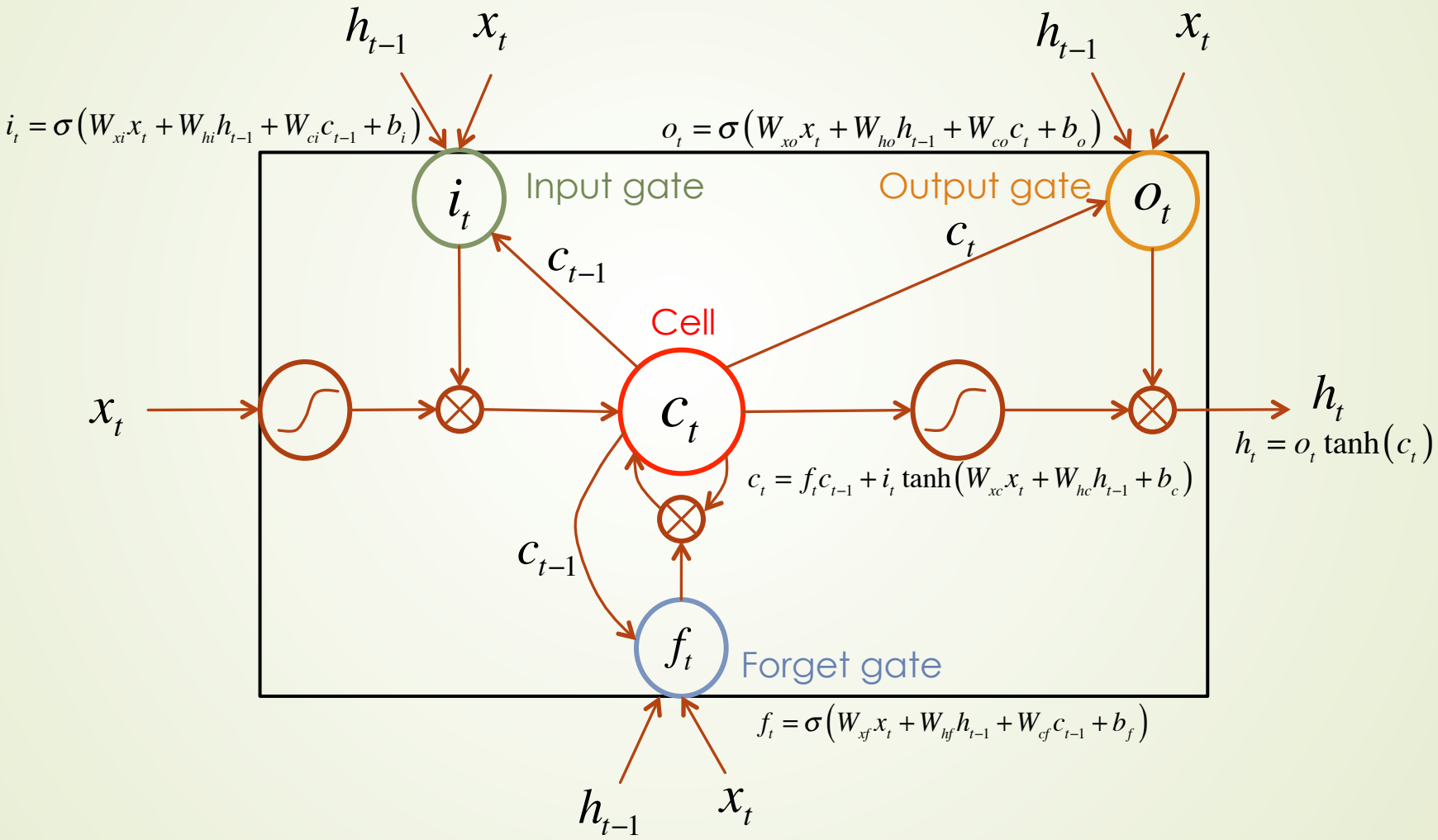
- Motivation



LSTM just replace the hidden units in RNN with the memory block !

[Hochreiter and Schmidhuber ,1997]

Memory Block in LSTM



σ: usually the logistic sigmoid

From LSTM to Bidirectional LSTM

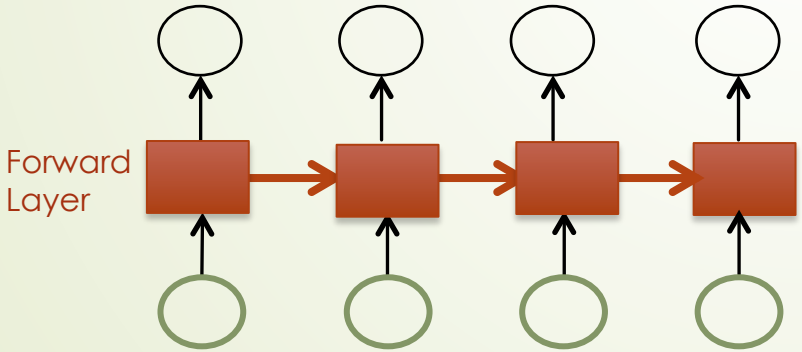
- Motivation

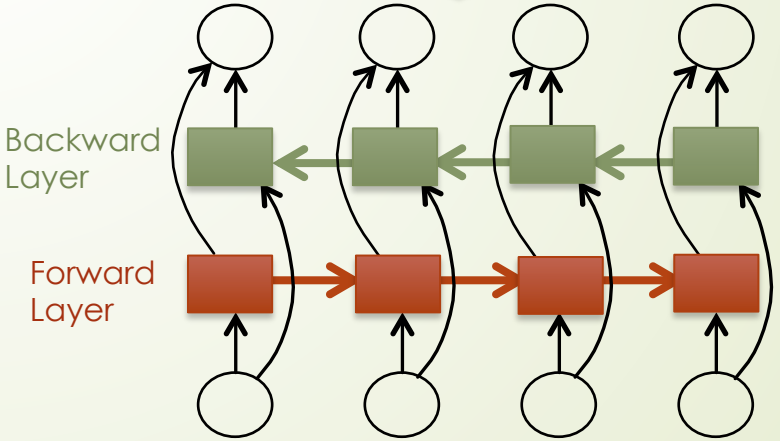
LSTM

- Only able to make use of previous context!

Bidirectional LSTM

- Make use of the contextual information!

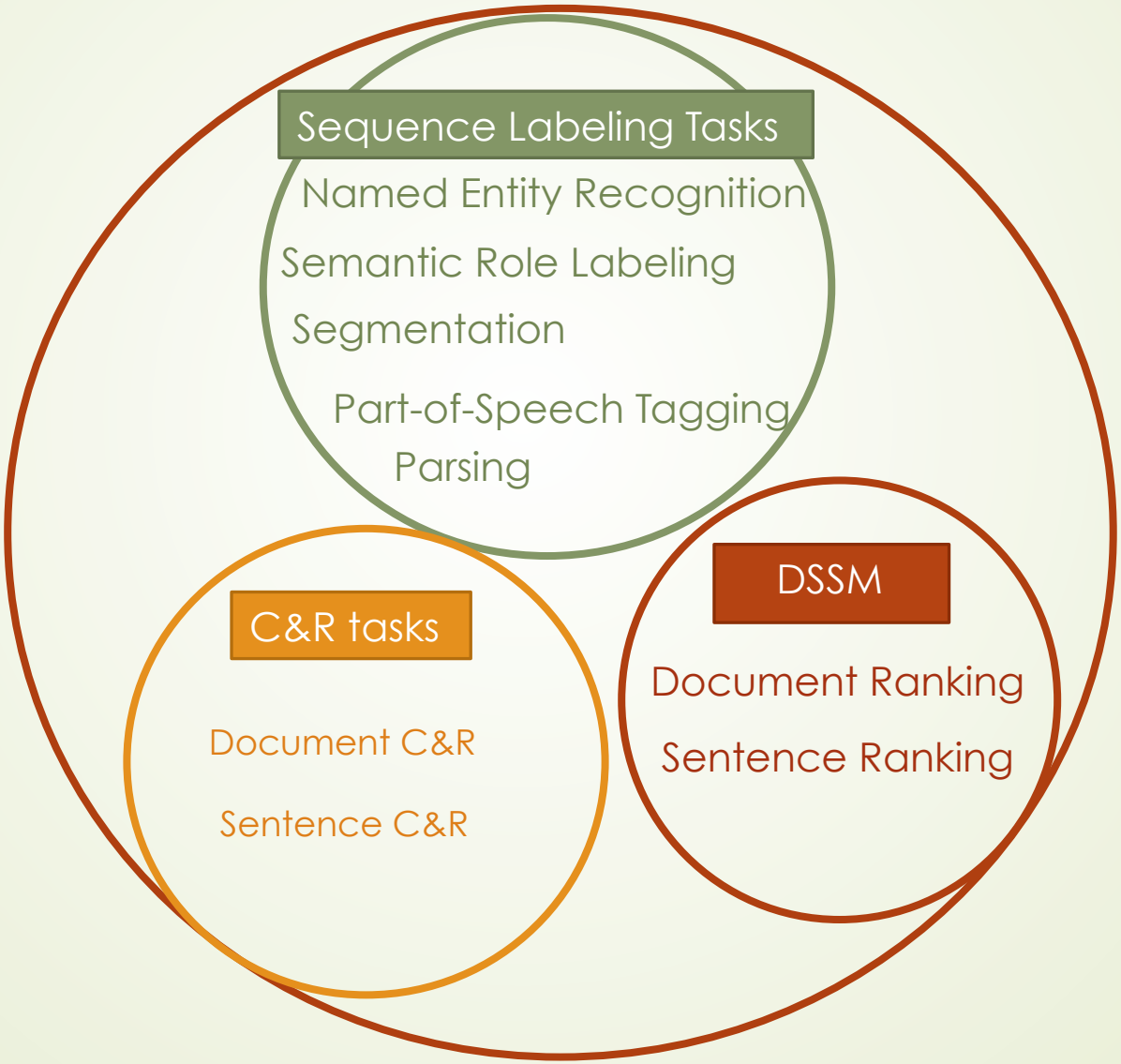




Outline

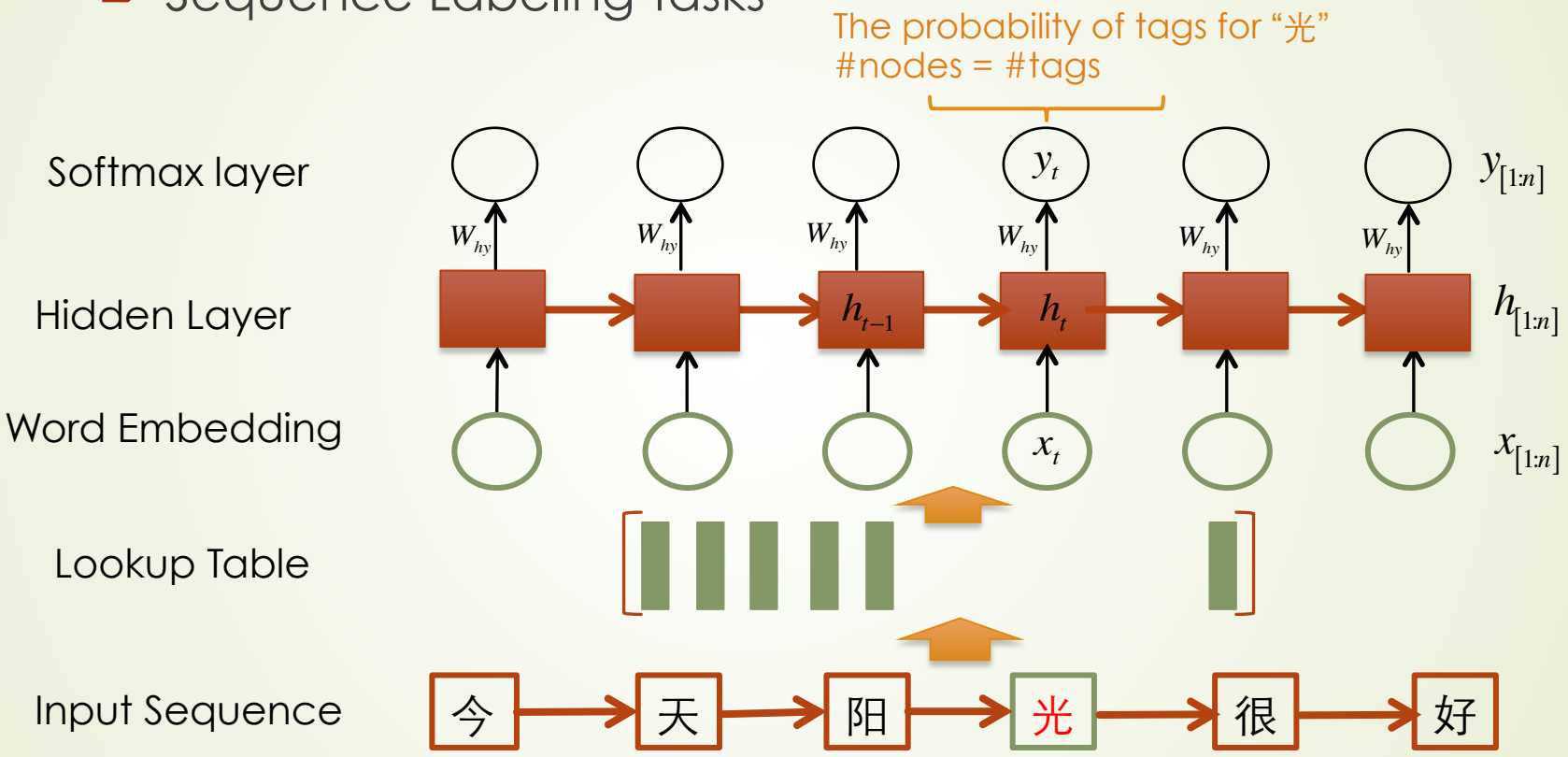
- Background of Deep Learning
- Deep learning for Chinese NLP
  - Neural Language Model
  - Feed-forward Neural Network
  - Recurrent Neural Network
    - Common Models for Natural Language Processing
    - The Typical Applications and Experiments

Main Tasks of Chinese NLP



Common RNN Models for Natural Language Processing

- Sequence Labeling Tasks

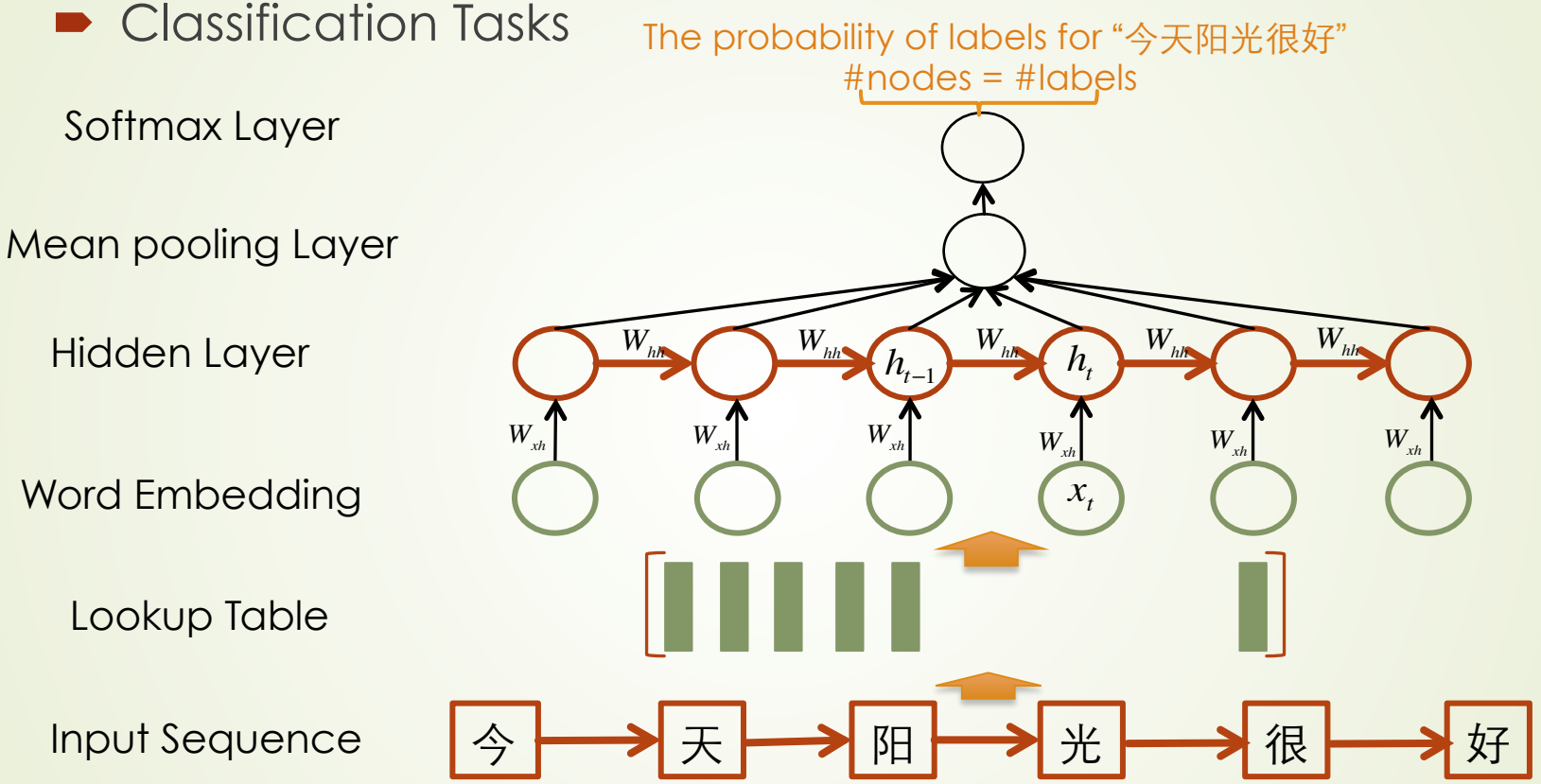


For t=1 to n, do $h_t = H\left(x_t, h_{t-1}, c_{t-1}\right)$

$y_t = f\left(W_{hy} h_t + b_y\right)$ f: sigmoid function

Common RNN Models for Natural Language Processing

- Classification Tasks



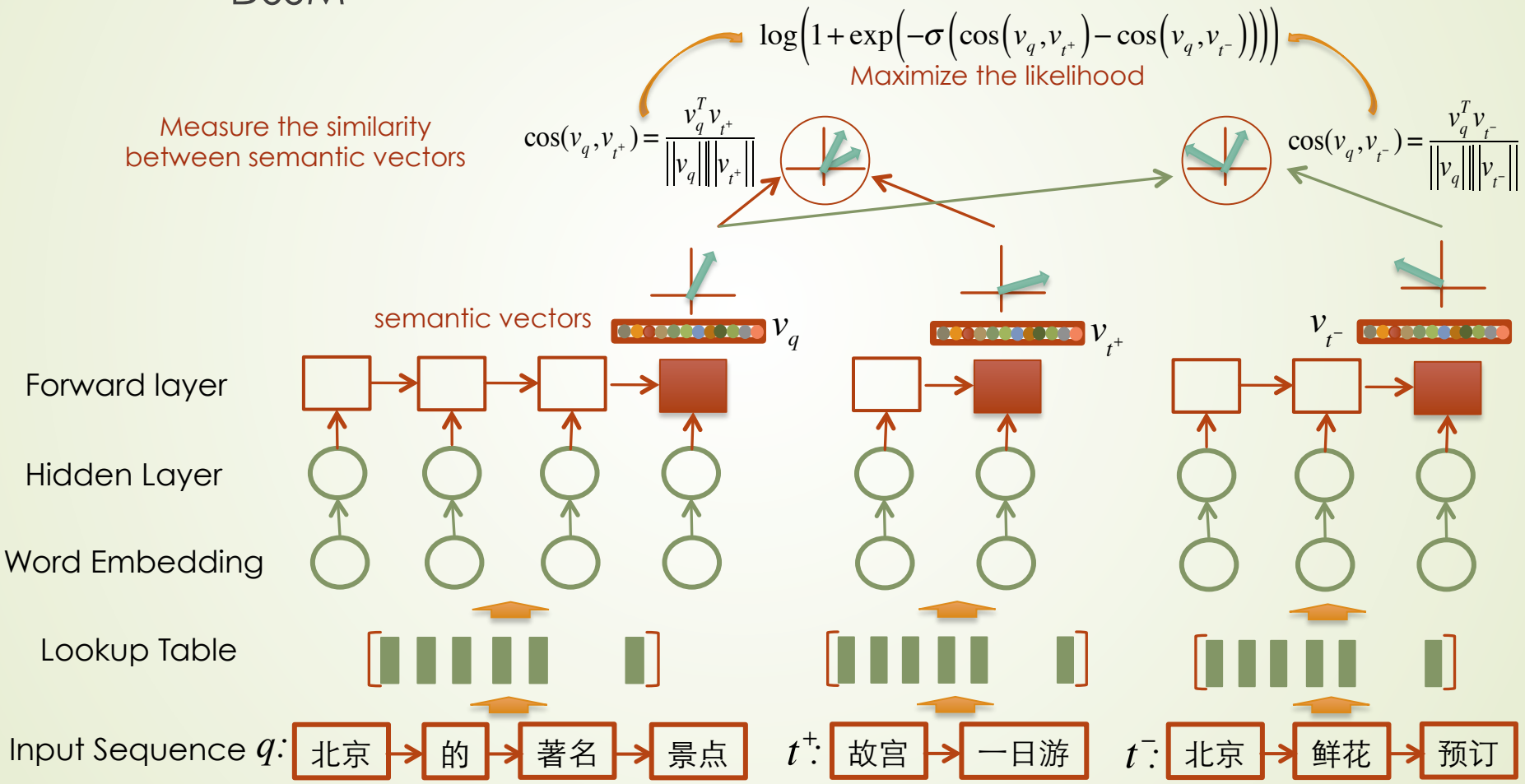
For t=1 to n, do $h_t = H\left(W_{xh}x_t + W_{hh}h_{t-1} + b_h\right)$

$$y = f\left(W_{hy}h_n + b_y\right)$$

H : tanh or Relu f: sigmoid function

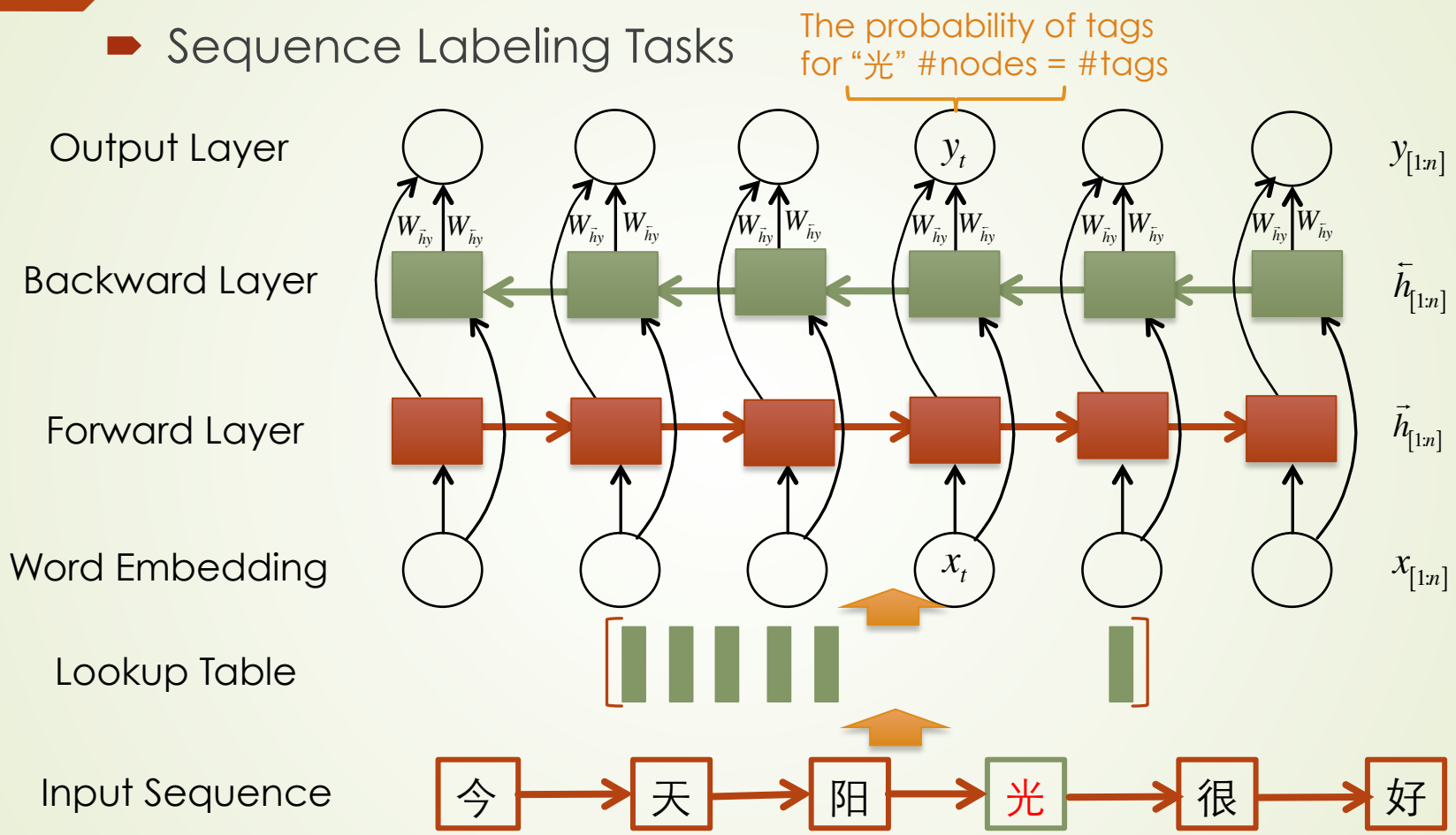
Common RNN Models for Natural Language Processing

- DSSM



Common RNN Models for Natural Language Processing ——Bidirectional LSTM

- Sequence Labeling Tasks



For t=1 to n, do $\vec{h}_t = H\left(x_t, \vec{h}_{t-1}, c_{t-1}\right)$ For t=n to 1, do $\overleftarrow{h}_t = H\left(x_t, \overleftarrow{h}_{t+1}, c_{t-1}\right)$

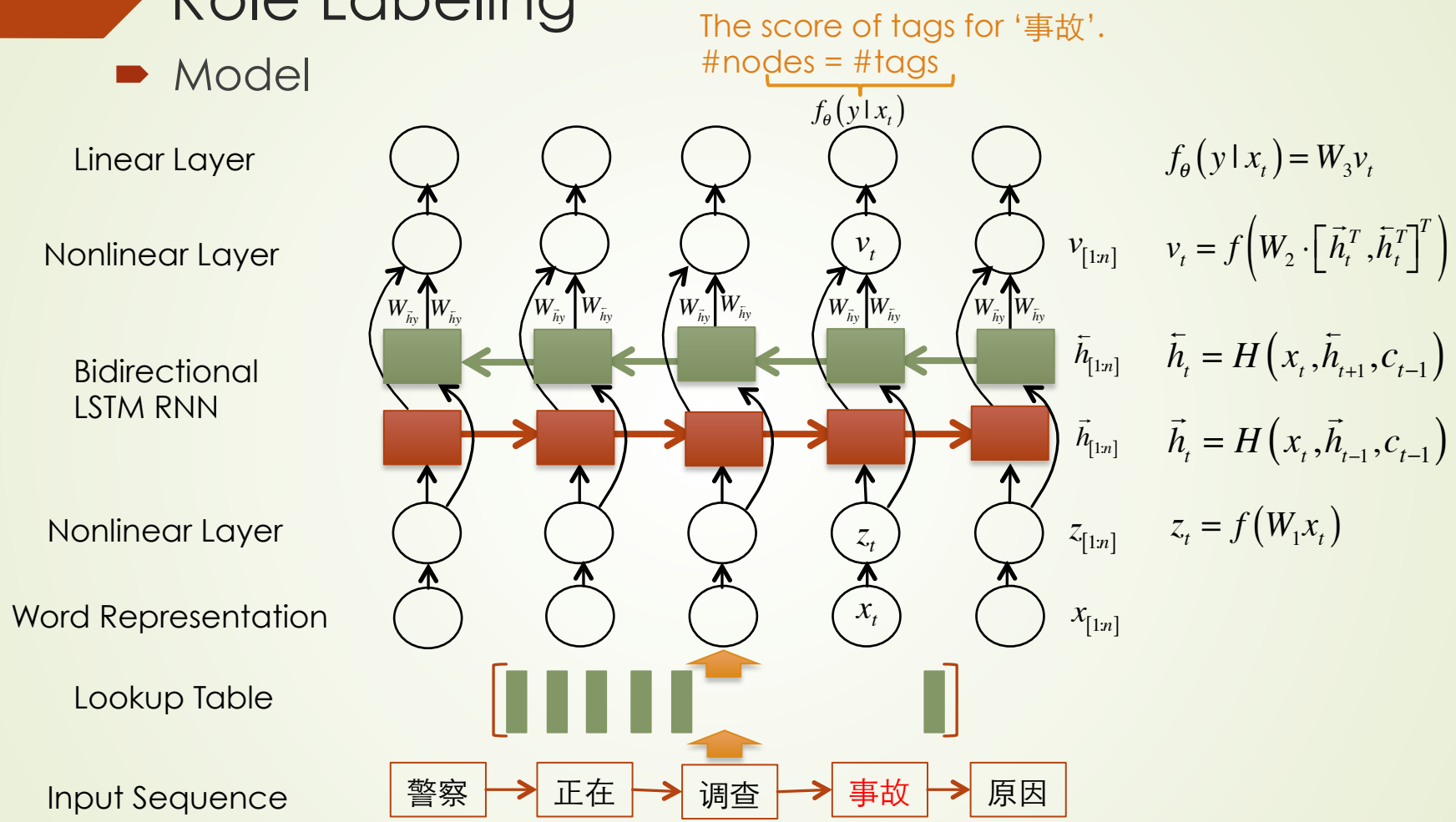
For all t, do $y_t = f\left(W_{\vec{h}y}\vec{h}_t + W_{\overleftarrow{h}y}\overleftarrow{h}_t + b_y\right)$ where f is a softmax function

Outline

- Background of Deep Learning
- Deep learning for Chinese NLP
  - Neural Language Model
  - Feed-forward Neural Network
  - Recurrent Neural Network
    - Common Models for Natural Language Processing
    - The Typical Applications and Experiments

Typical Applications: Chinese Semantic Role Labeling

- Model



The score of $x_{[1:n]}$ along the path $y_{[1:n]}$: $s\left(x_{[1:n]}, y_{[1:n]}, \theta\right) = \sum_{t=1}^{n} f_\theta\left(y_t \mid x_t\right)$

So the sentence-level log-likelihood: $\log p\left(y_{[1:n]}, x_{[1:n]}, \theta\right) = s\left(x_{[1:n]}, y_{[1:n]}, \theta\right) - \log \sum_{\forall i_{[1:n]}} s\left(x_{[1:n]}, i_{[1:n]}, \theta\right)$

[Zhen W., 2015]

Experiments on Chinese Semantic Role Labeling

- Dataset: CPB 1.0 for Chinese Semantic Role Labeling

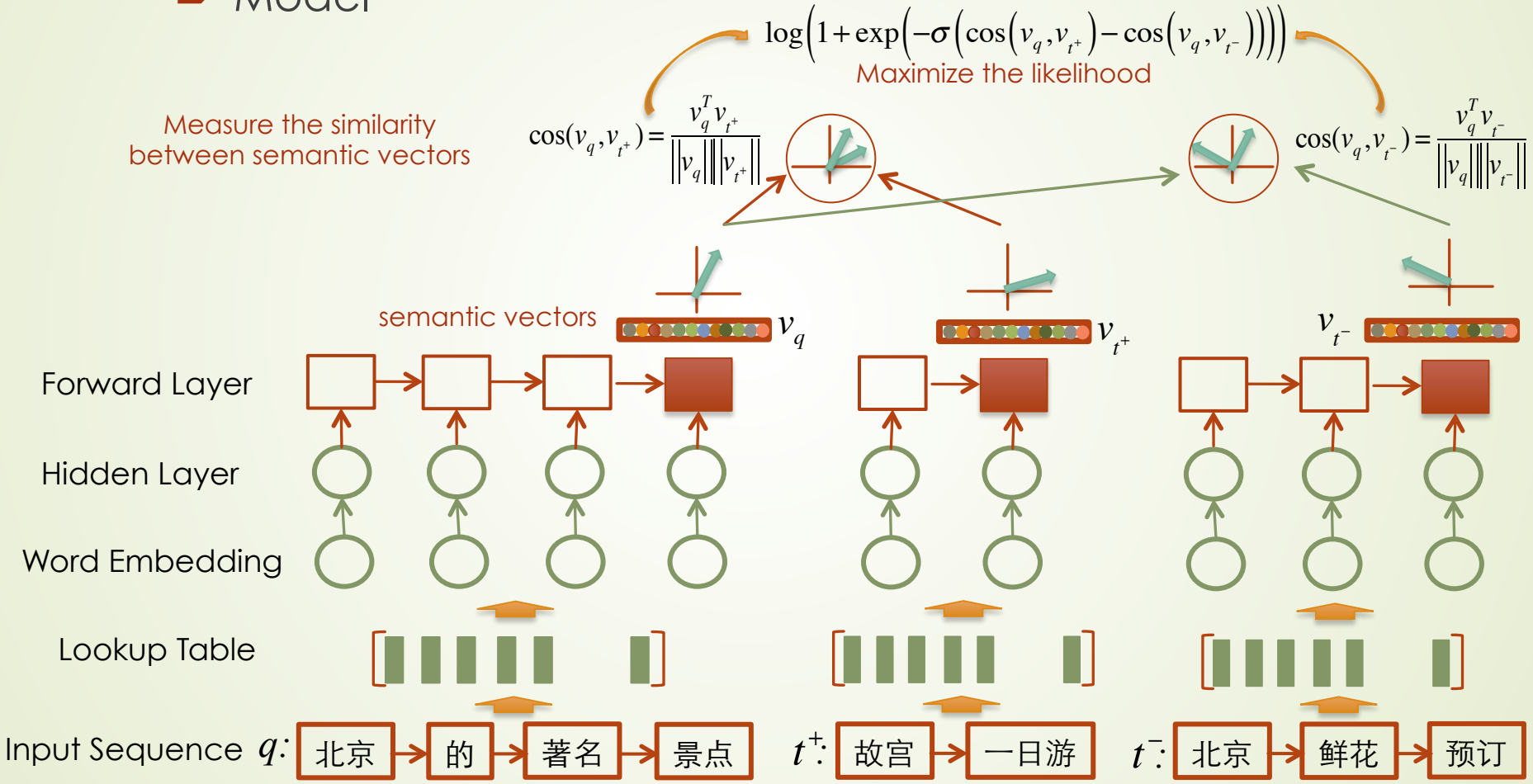
Method	F1(%)
Syntactic Model [Xue, 2008]	71.90
CNN Model [Collobert and Weston, 2008]	74.05
Shallow Parsing Model [Sun et al. 2009]	74.12
Multi-predicate Model [Yang and Zong, 2014]	75.31
BRNN+Random Initialization	77.09
BRNN+Standard Pre-training	77.21

- The BRNN model significantly outperforms previous state-of-art methods even with all parameters randomly initialized. Also, pre-training has a good effect on the performance.

[Zhen W., 2015]

Typical Applications: Web Document Retrieval

- Model



[Cooke A., 2015]

Experiments on Web Document Retrieval

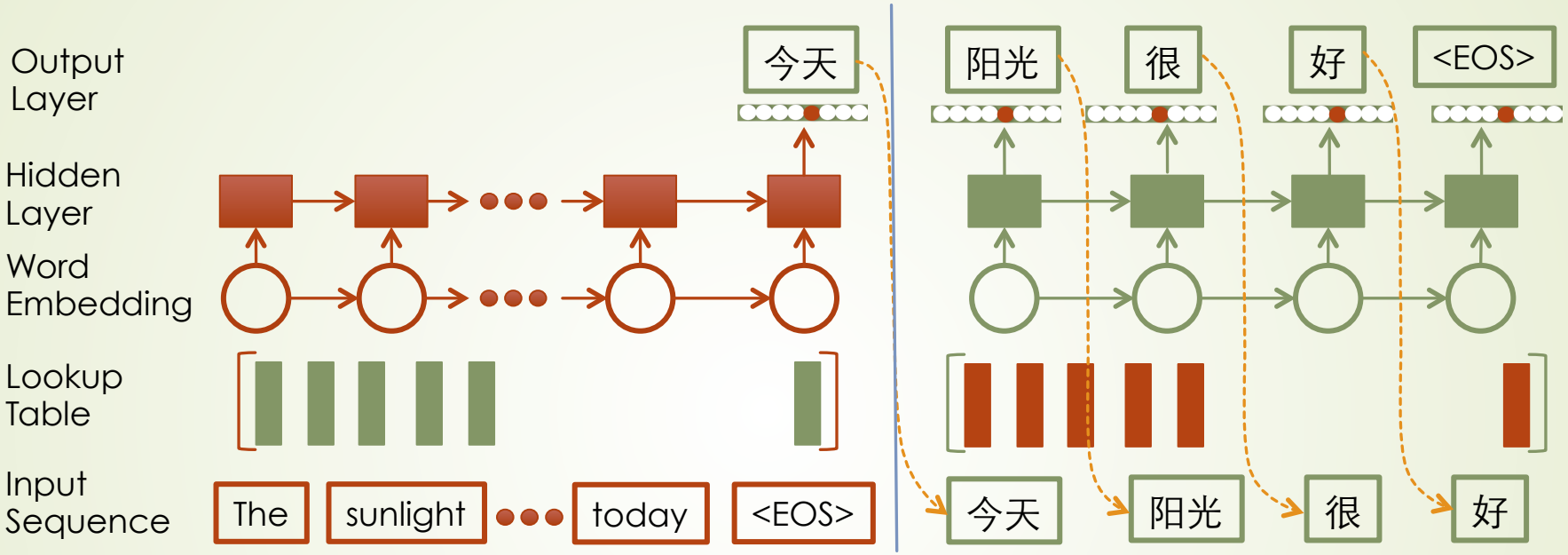
- Task: Web Document Retrieval Task, evaluating the ranking performance

Model	NDCG@1	NDCG@3	NDCG@10
ULM	30.4%	32.7%	38.5%
BM25	30.5%	32.8%	38.8%
PLSA	30.8%	33.7%	40.2%
DNN DSSM (nhid=288/96), 2 Layers	31.0%	34.4%	41.7%
CLSM (nhid=288/96), 2 Layers	31.8%	35.1%	42.6%
RNN DSSM (nhid=288), 1 Layer	31.7%	35.0%	42.3%
LSTM DSSM (ncell=96), 1 Layer	**33.1%**	**36.5%**	**43.6%**

- The LSTM DSSM significantly outperforms all the other models.

[Cooke A., 2015]

The Typical Application : Machine Translation



- Step 1: obtaining the fixed-dimensional representation v of the input sequence $(x_1,...,x_T)$ given by the last hidden state of the LSTM.
- Step 2: computing the probability of $y_1,...,y_{T'}$ with a standard LSTM-LM formulation whose initial hidden state is set to the representation v of $x_1,...,x_T$

$$p(y_1,...,y_{T'} \mid x_1,...,x_T) = \prod_{t=1}^{T'} p(y_t \mid v, y_1,...,y_{t-1})$$

[Sutskever, Ilya, et al. 2014]

Training

- The model is trained by maximizing the log of a correct translation T given the source sentence S:

$$\frac{1}{|S|}\sum_{(T,S)\in S}\log p(T \mid S)$$

- Once training is complete, the model produce translation by finding the most likely translation according to the LSTM:

$$\hat{T} = \arg\max_{T} p(T \mid S)$$

- Then the most likely translation can be searched by a simple left-to-right beam search decoder.

[Sutskever, Ilya, et al. 2014]

Experiments on Machine Translation

- Dataset: WMT'14 English to French dataset

Method	Test BLEU score
Baseline System based on Moses	33.30
Single forward LSTM, beam size 12	26.17
Single reversed LSTM, beam size 12	30.59
Ensemble of 5 reversed LSTMs, beam size 1	33.00
Ensemble of 2 reversed LSTMs, beam size 12	33.27
Ensemble of 5 reversed LSTMs, beam size 2	34.50
Ensemble of 5 reversed LSTMs, beam size 12	**34.81**

- The results show that the LSTM system performs better than the system based on Moses.

[Sutskever, Ilya, et al. 2014]

The Typical Application : Machine Translation

- An alternative model