

CCF ADL  
Beijing  
April 8, 2017

# **Deep Learning for Natural Language Processing**

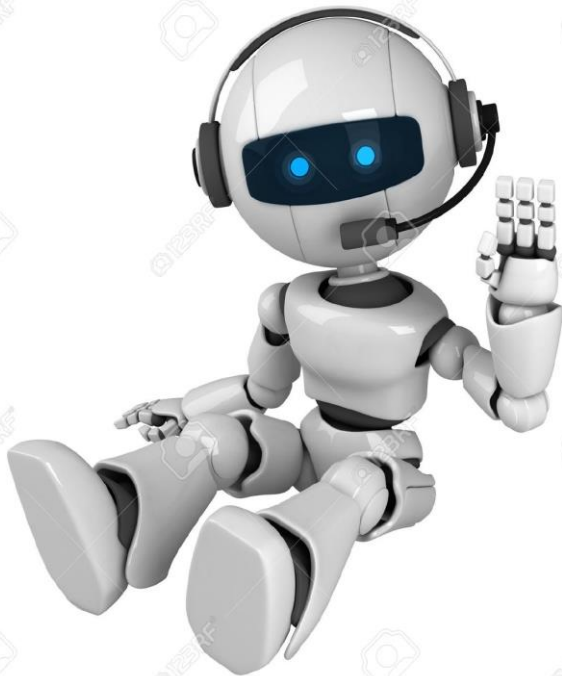
## **深度学习在自然语言处理的应用**

Hang Li  
Noah's Ark Lab  
Huawei Technologies

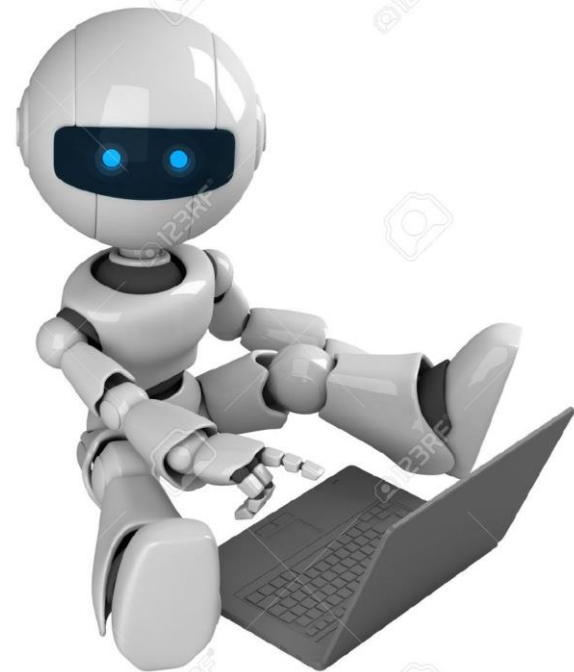
# Outline of Lecture

- *Introduction*
- Basics of DL for NLP
- State of the Art of DL for NLP
- Previous Work at Noah's Ark Lab
- Recent Progress at Noah's Ark Lab
- Advantages and Disadvantages
- Summary

# Ultimate Goal: Natural Language Understanding



Natural Language Dialogue



Text Comprehension

# Natural Language Understanding

- Two definitions:
  - Representation-based: if system creates proper internal representation, then we say it “understands” language
  - Behavior-based: if system follows instruction in natural language, then we say it “understands” language, e.g., “bring me a cup of tea”
- We take the latter definition



# Five Characteristics of Human Language

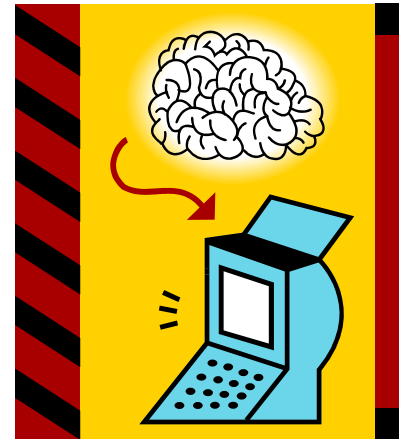
- Incompletely Regular (Both Regular and Idiosyncratic)
- Compositional (or Recursive)
- Metaphorical
- Associated with World Knowledge
- Interactive

# Natural Language Understanding by Computer Is Extremely Difficult

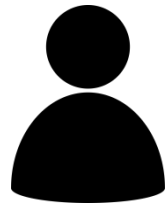
- It is still not clear whether it is possible to realize human language ability on computer
- On modern computer
  - The incomplete regularity and compositionality characteristics imply complex combinatorial computation
  - The metaphor, knowledge, and interaction characteristics imply exhaustive computation
- Big question: can we invent new computer closer to human brain?

# Reason of the Challenge

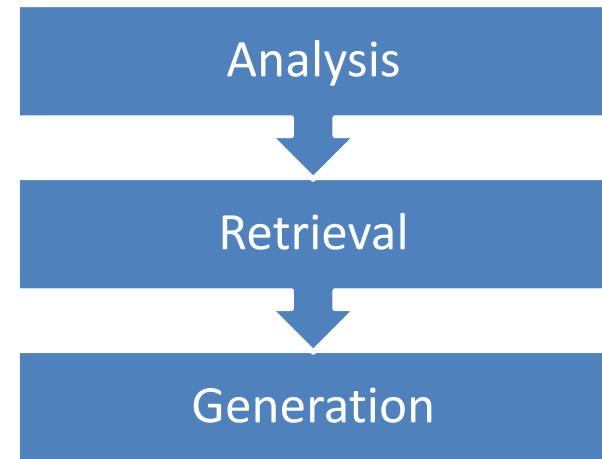
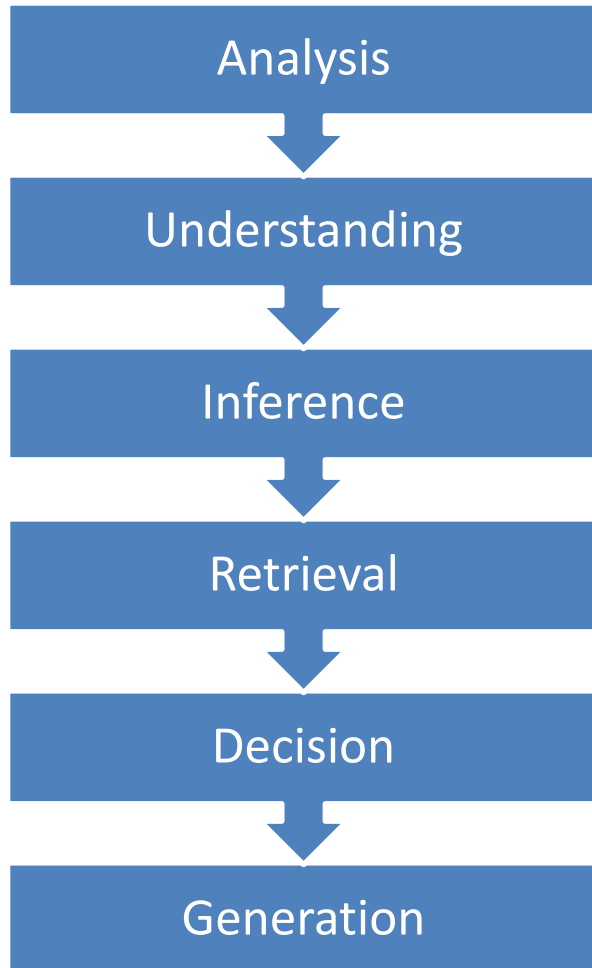
- A computer system must be constructed based on math
- Open question: whether it is possible to process natural language *as humans*, using math models
- Natural language processing is believed to be AI complete



# Simplified Problem Formulation May Work



- Eg., Question Answering



Question answering, including search, can be practically performed, because it is simplified

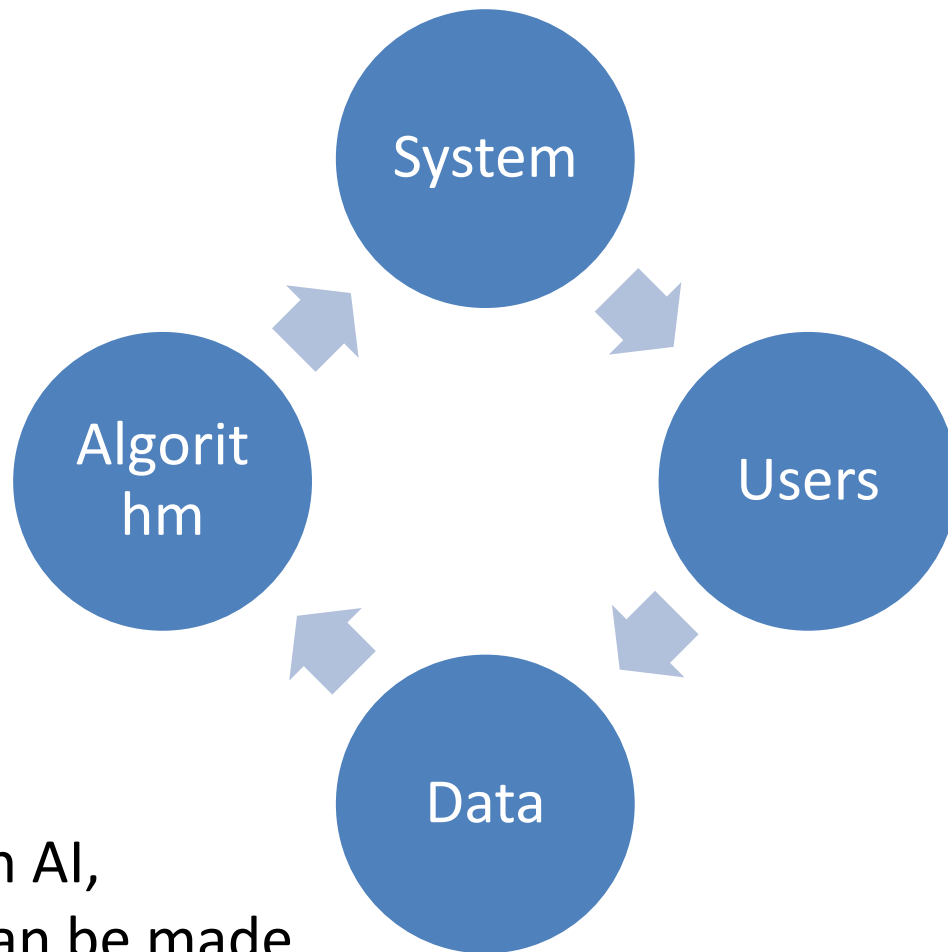


# Data-driven Approach May Work

- Hybrid is most realistic and effective for natural language processing, and AI
  - machine learning based
  - human-knowledge incorporated
  - human brain inspired
- Big data and deep learning provides new opportunity



# AI Loop



Advancement in AI,  
including NLP can be made  
through the closed loop

# Fundamental Problems of Statistical Natural Language Processing

- Classification: assigning a label to a string

$$s \rightarrow c$$

- Matching: matching two strings

$$s, t \rightarrow \mathbf{R}^+$$

- Translation: transforming one string to another

$$s \rightarrow t$$

- Structured prediction: mapping string to structure

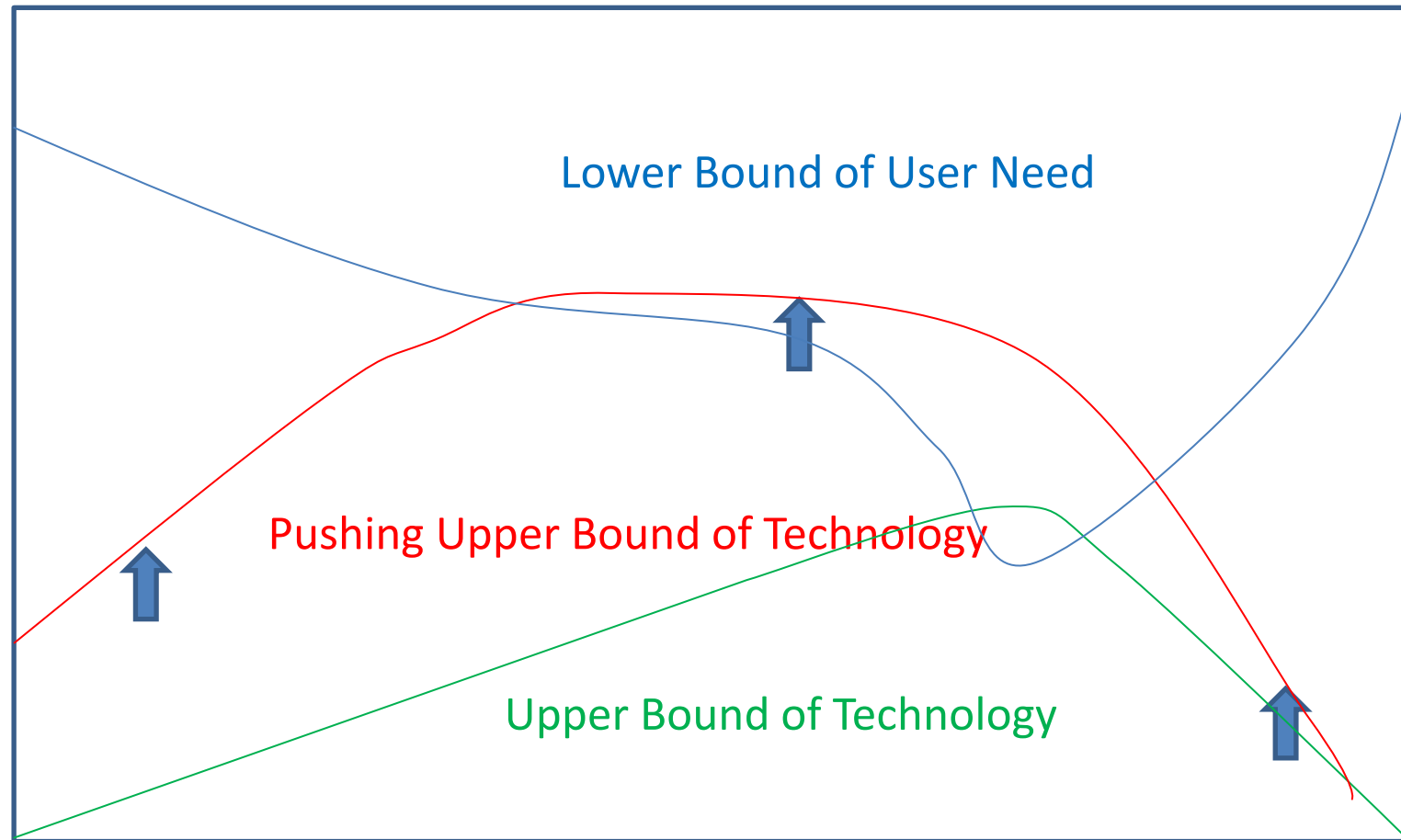
$$\begin{array}{ccc} s & \xrightarrow{\quad} & s' \\ \uparrow & & \\ \mathbf{D} & & \end{array}$$

- Markov decision process: deciding next state given previous state and action

# Fundamental Problems of Statistical Natural Language Processing

- Classification
  - Text classification
  - Sentiment analysis
- Matching
  - Search
  - Question answering
  - Dialogue (single turn)
- Translation
  - Machine translation
  - Speech recognition
  - Hand writing recognition
  - Dialogue (single turn)
- Structured Prediction
  - Named entity extraction
  - Part of speech tagging
  - Sentence parsing
  - Semantic parsing
- Markov Decision Process
  - Dialogue (multi turn, task dependent)

# Lower Bound of User Need vs Upper Bound of Technology



# Deep Learning for Natural Language Processing (DL for NLP)

- Achieved State-of-Art Performances in
  - Classification
  - Matching
  - Translation
  - Structured Prediction
- E.g., Neural Machine Translation outperforms Statistical Machine Translation

# Outline of Lecture

- Introduction
- *Basics of DL for NLP*
- State of the Art of DL for NLP
- Previous Work at Noah's Ark Lab
- Recent Progress at Noah's Ark Lab
- Advantages and Disadvantages
- Summary

# Basics of DL for NLP

- Word Embedding
- Convolutional Neural Network (CNN)
- Recurrent Neural Network (RNN)
- Sequence-to-Sequence Learning



# Word Embedding



# Word Embedding

- Motivation: representing words with low-dimensional real-valued vectors, utilizing them as input to deep learning methods, vs one-hot vectors
- Method: SGNS (Skip-Gram with Negative Sampling)
- Tool: Word2Vec
- Input: words and their contexts in documents
- Output: embeddings of words
- Assumption: *similar* words occur in *similar* contexts
- Interpretation: factorization of mutual information matrix
- Advantage: compact representations (usually 100~ dimensions)

# Skip-Gram with Negative Sampling (Mikolov et al., 2013)

- Input: occurrences between words and contexts

$M$	$c_1$	$c_2$	$c_3$	$c_4$	$c_5$
$w_1$	5		1	2	
$w_2$		2			1
$w_3$	3			1	

- Probability model:
$$P(D = 1 \mid w, c) = \sigma(\vec{w} \cdot \vec{c}) = \frac{1}{1 + e^{-\vec{w} \cdot \vec{c}}}$$
$$P(D = 0 \mid w, c) = \sigma(-\vec{w} \cdot \vec{c}) = \frac{1}{1 + e^{\vec{w} \cdot \vec{c}}}$$

# Skip-Gram with Negative Sampling

- Word vector and context vector: lower dimensional (parameter ) vectors  $\vec{w}, \vec{c}$
- Goal: learning of the probability model from data
- Take co-occurrence data as positive examples
- Negative sampling: randomly sample  $k$  unobserved pairs  $(w, c_N)$  as negative examples
- Objective function in learning


$$L = \sum_w \sum_c \#(w, c) \log \sigma(\vec{w} \cdot \vec{c}) + k \cdot \mathbf{E}_{c_N \sim P} \log \sigma(-\vec{w} \cdot \vec{c}_N)$$

- Algorithm: stochastic gradient descent

# Interpretation as Matrix Factorization (Levy & Goldberg 2014)

- Pointwise Mutual Information Matrix

$M$	$c_1$	$c_2$	$c_3$	$c_4$	$c_5$
$w_1$	3		-.5	2	
$w_2$		1			-0.5
$w_3$	1.5			1	


$$\log \frac{P(w, c)}{P(w)P(c)}$$

# Interpretation as Matrix Factorization

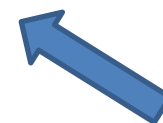
$M$	$c_1$	$c_2$	$c_3$	$c_4$	$c_5$
$w_1$	3		-0.5	2	
$w_2$		1			-0.5
$w_3$	1.5			1	

$$M = WC^T$$



Matrix factorization,  
equivalent to SGNS

$W$	$t_1$	$t_2$	$t_3$
$w_1$	7	0.5	1
$w_2$		2.2	3
$w_3$	1	1.5	1



Word embedding

# Convolutional Neural Network



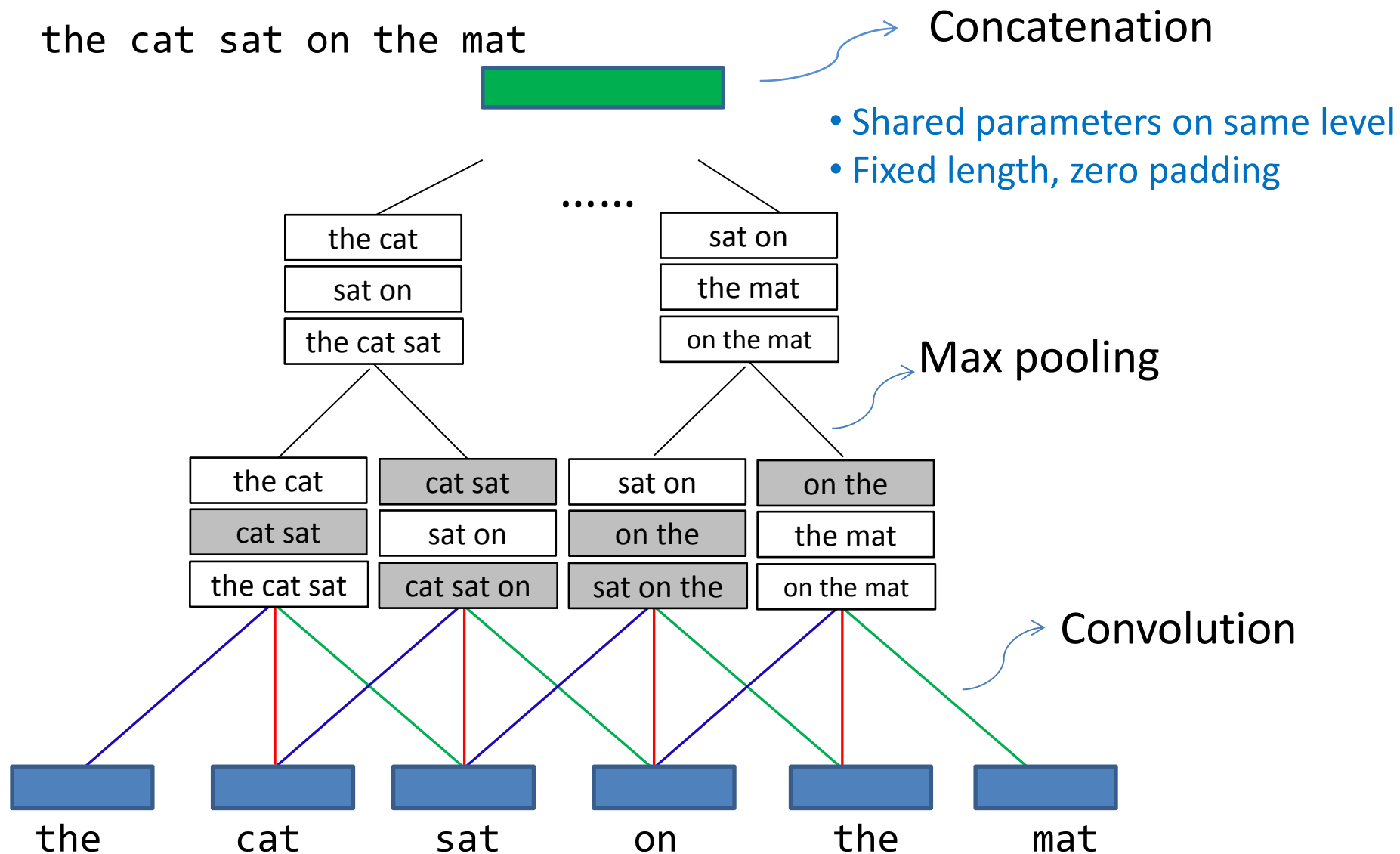
# Convolutional Neural Network

- Motivation: representing sequence of words and utilizing the representation in deep learning methods
- Input: sequence of word embeddings, denoting sequence of words (e.g., sentence)
- Output: representation of input sequence
- Learning of model: stochastic gradient descent
- Advantage: robust extraction of n-gram features; can be used as part of deep model for sequence processing (e.g., sentence classification)



# Convolutional Neural Network (CNN)

(Kim 2014, Blunsom et al. 2014, Hu et al., 2014)

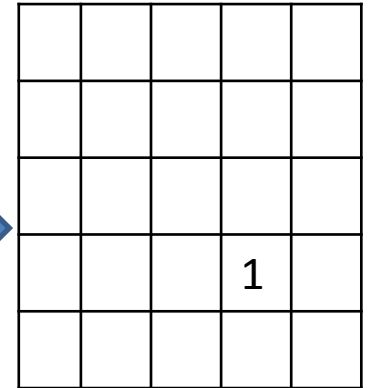
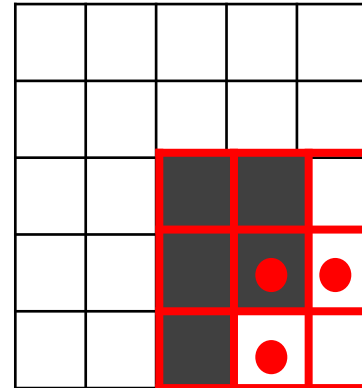
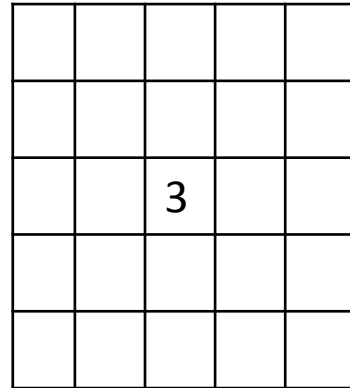
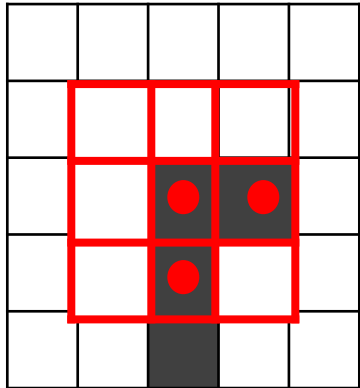
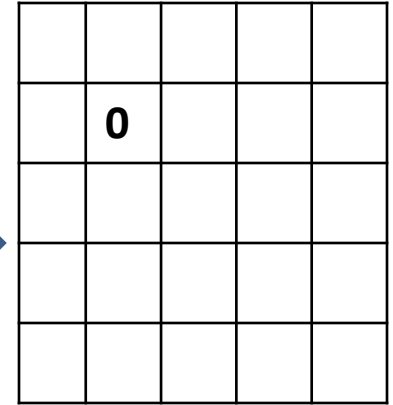
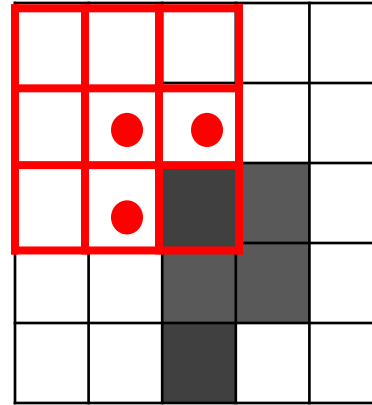
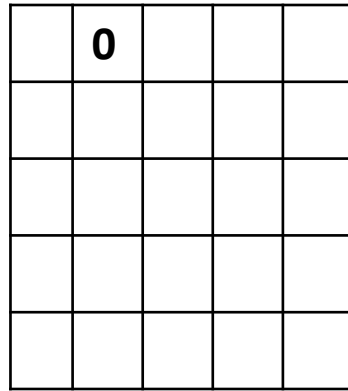
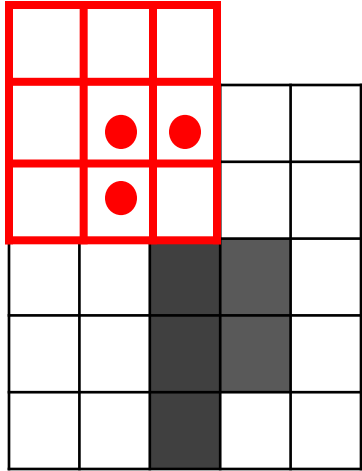


# Example: Image Convolution

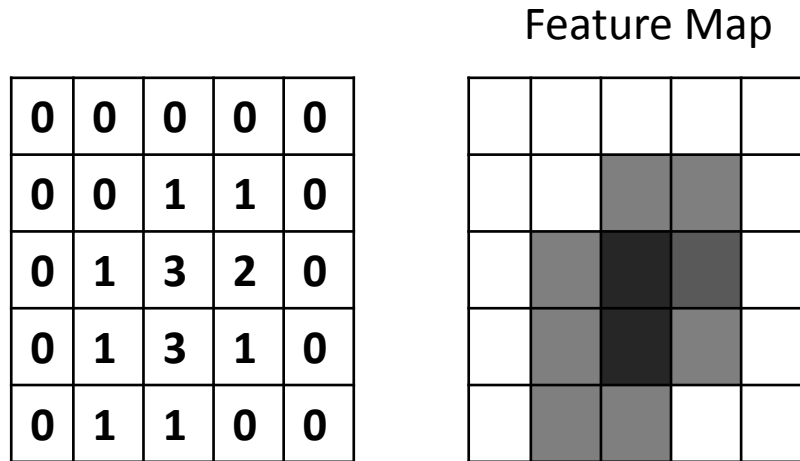
Filter

Dark Pixel Value = 1, Light Pixel Value = 0

Dot in Filter = 1, Others = 0



# Example: Image Convolution



## Convolution Operation

- Scanning image with filter having 3\*3 cells, among them 3 are dot cells
- Counting number of dark pixels overlapping with dot cells at each position
- Creating feature map (matrix), each element represents similarity between filter pattern and pixel pattern at one position
- Equivalent to extracting feature using the filter
- Translation-invariant

# Convolution

$$z_i^{(l,f)} = \sigma(w^{(l,f)} \cdot z_i^{(l-1)} + b^{(l,f)}) \quad f = 1, 2, \dots, F_l$$

$z_i^{(l,f)}$  is output of neuron of type  $f$  for location  $i$  in layer  $l$

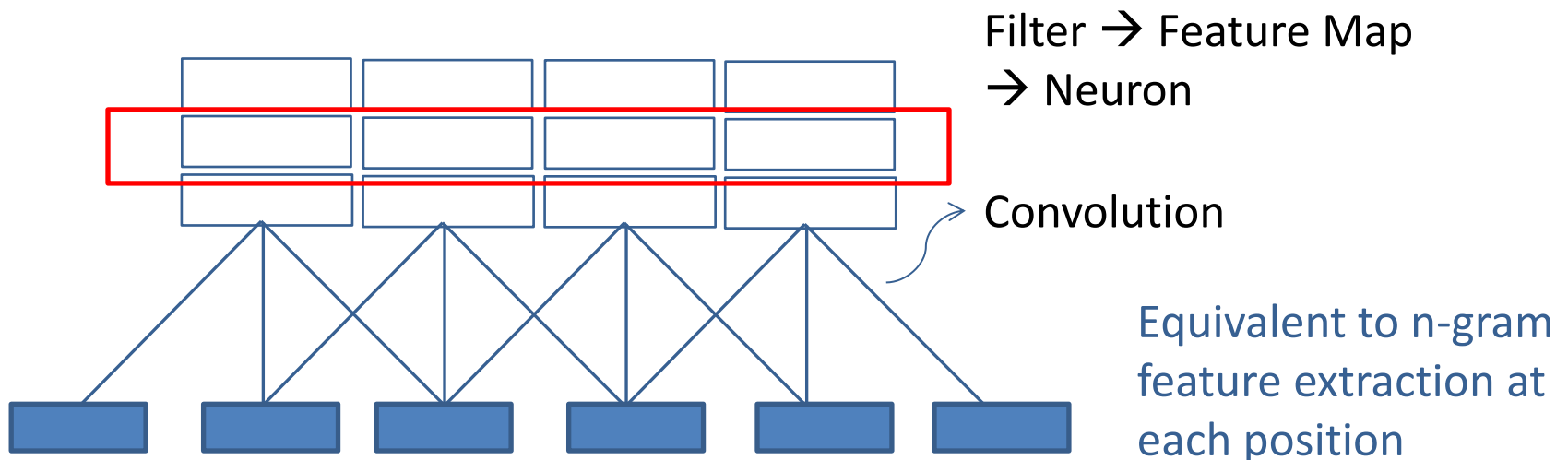
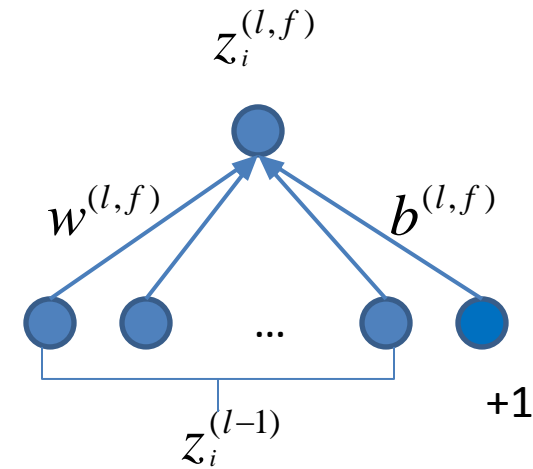
$w^{(l,f)}, b^{(l,f)}$  are parameters of neuron of type  $f$  in layer  $l$

$\sigma$  is sigmoid function

$z_i^{(l-1)}$  is input of neuron for location  $i$  from layer  $l-1$

$z_i^{(0)}$  is input from concatenated word vectors for location  $i$

$$z_i^{(0)} = [x_i^T, x_{i+1}^T, \dots, x_{i+h-1}^T]^T$$

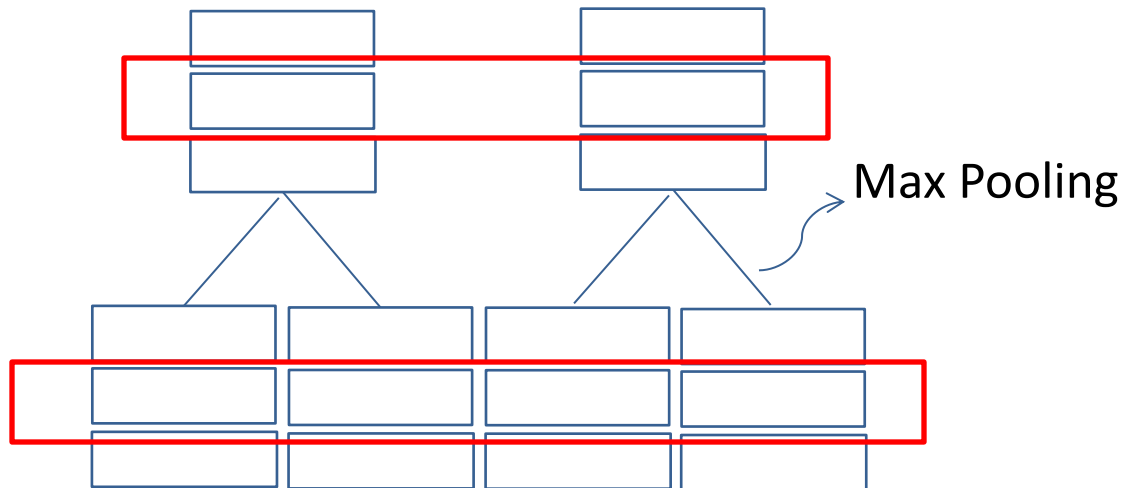


# Max Pooling

$$z_i^{(l,f)} = \max(z_{2i-1}^{(l-1,f)}, z_{2i}^{(l-1,f)})$$

$z_i^{(l,f)}$  is output of pooling of type  $f$  for location  $i$  in layer  $l$

$z_{2i-1}^{(l-1,f)}, z_{2i}^{(l-1,f)}$  are input of pooling of type  $f$  for location  $i$  in layer  $l$



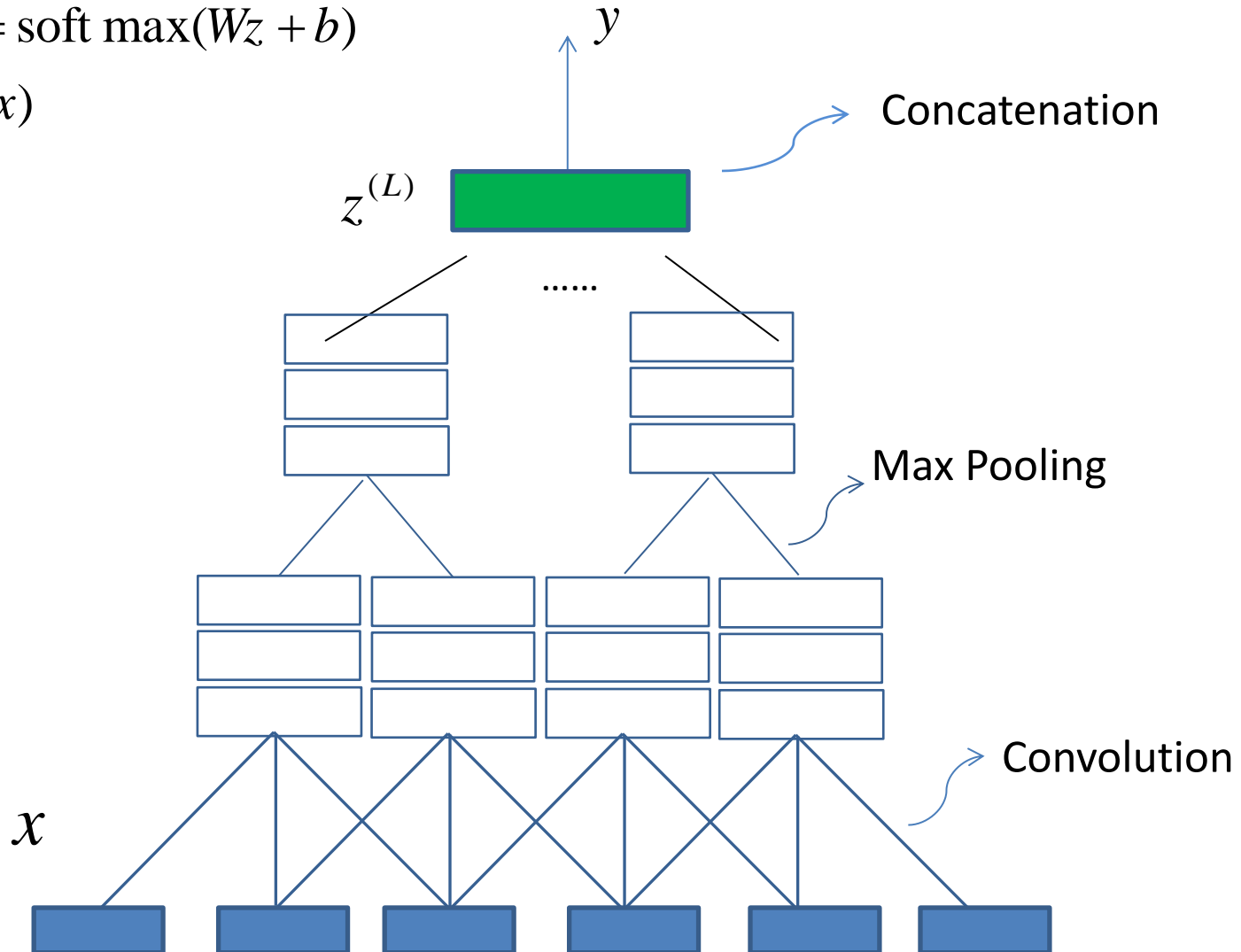
Equivalent to n-gram  
feature selection

# Sentence Classification

## Using Convolutional Neural Network

$$y = f(x) = \text{soft max}(Wz + b)$$

$$z = \text{CNN}(x)$$



# Recurrent Neural Network



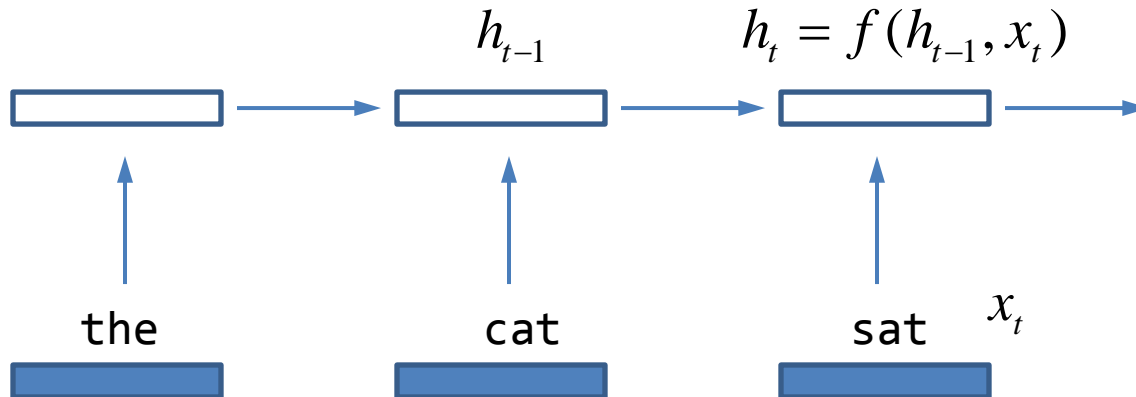
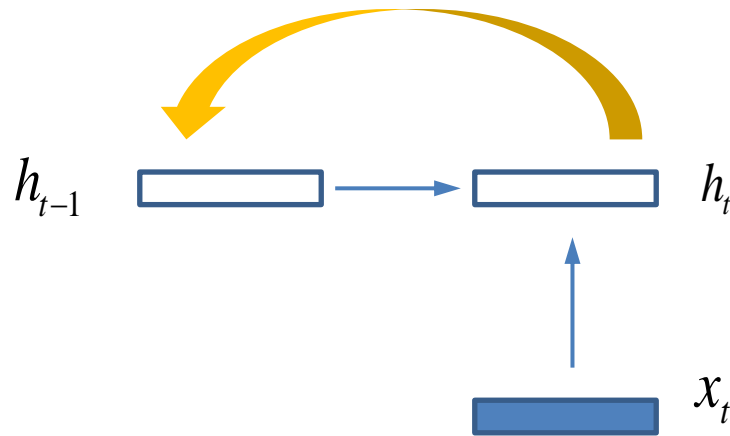
# Recurrent Neural Network

- Motivation: representing sequence of words and utilizing the representation in deep learning methods
- Input: sequence of word embeddings, denoting sequence of words (e.g., sentence)
- Output: sequence of internal representations (hidden states)
- Variants: LSTM and GRU, to deal with long distance dependency
- Learning of model: stochastic gradient descent
- Advantage: handling arbitrarily long sequence; can be used as part of deep model for sequence processing (e.g., language modeling)

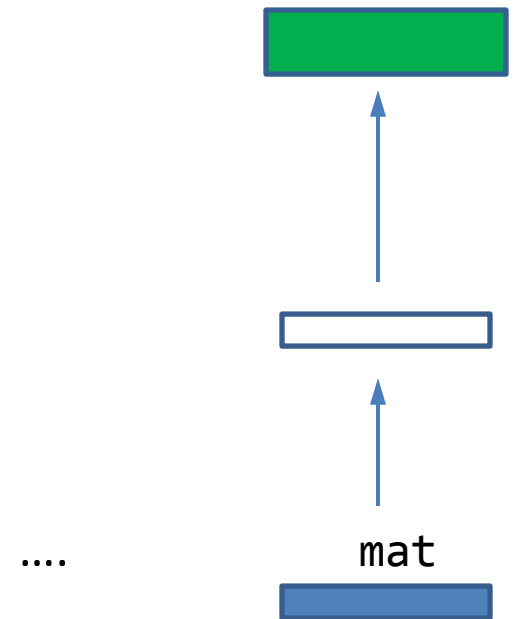


# Recurrent Neural Network (RNN)

(Mikolov et al. 2010)

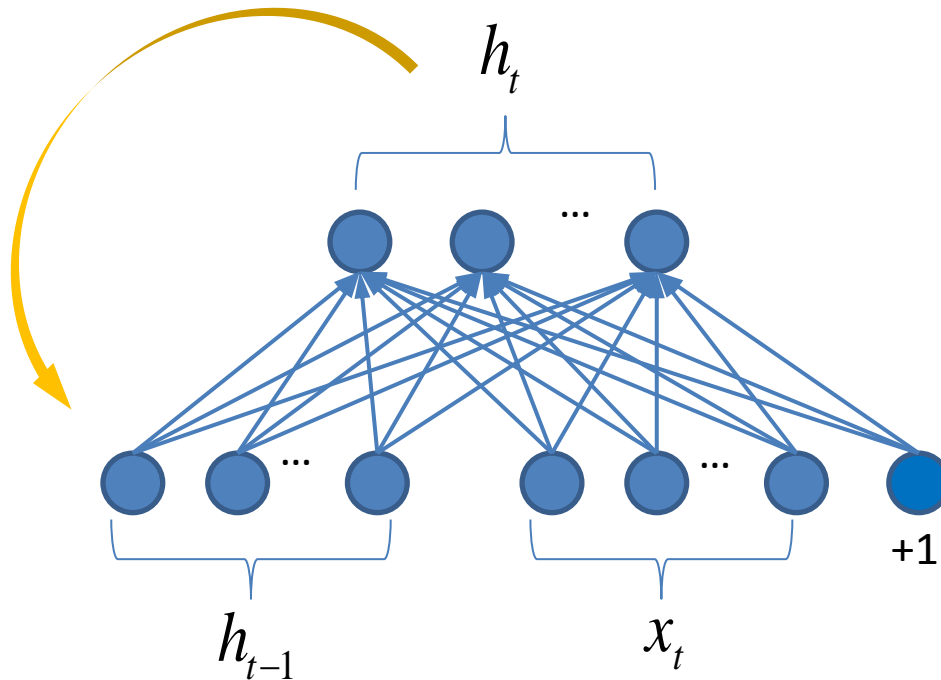


the cat sat on the mat



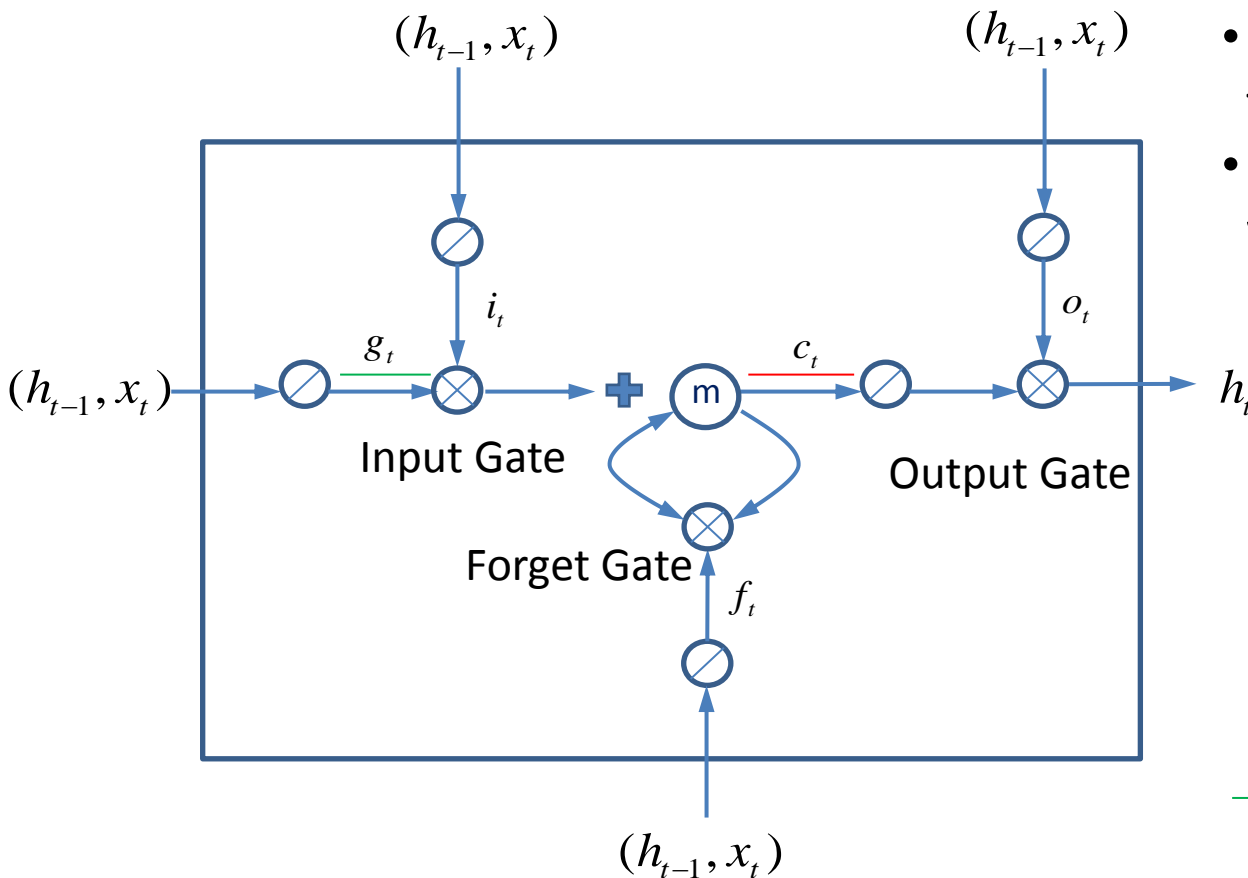
# Recurrent Neural Network

$$h_t = f(h_{t-1}, x_t) = \tanh(W_h h_{t-1} + W_x x_t + b_{hx})$$



# Long Term Short Memory (LSTM)

(Hochreiter & Schmidhuber, 1997)



- A memory (vector) to store values of previous state
- Input gate, output gate, and forget gate to control
- Gate: element-wise product with vector of values in  $[0,1]$

$$i_t = \sigma(W_{ih}h_{t-1} + W_{ix}x_t + b_i)$$

$$f_t = \sigma(W_{fh}h_{t-1} + W_{fx}x_t + b_f)$$

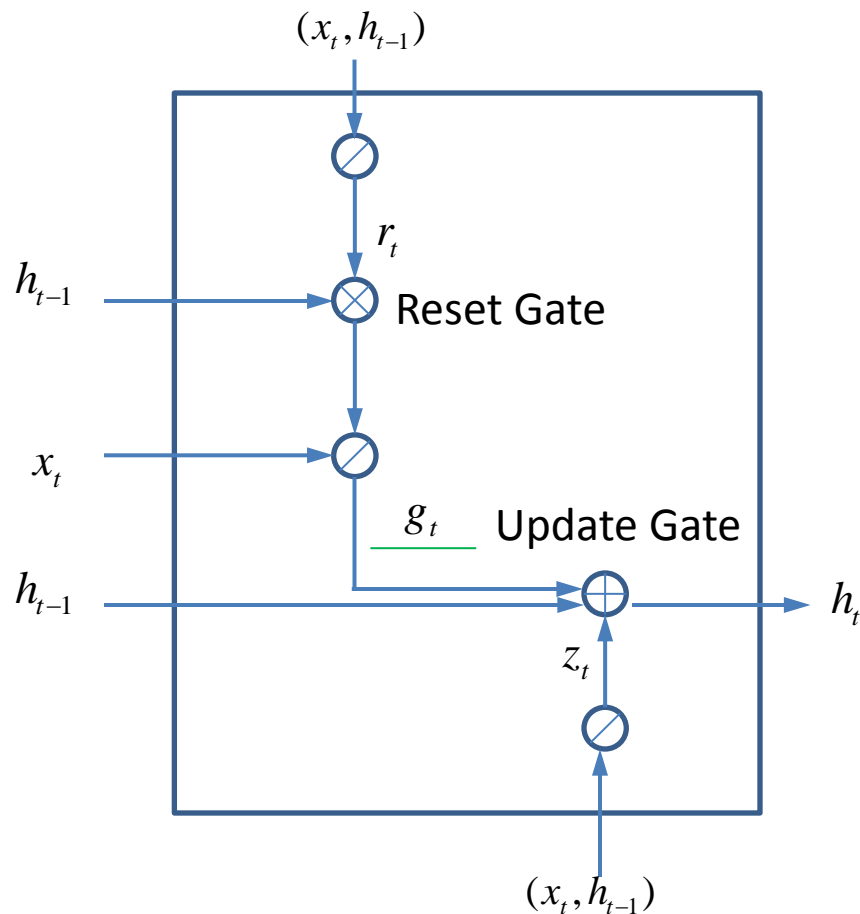
$$o_t = \sigma(W_{oh}h_{t-1} + W_{ox}x_t + b_o)$$

$$g_t = \tanh(W_{gh}h_{t-1} + W_{gx}x_t + b_g)$$

$$c_t = i_t \otimes g_t + f_t \otimes c_{t-1}$$

$$h_t = o_t \otimes \tanh(c_t)$$

# Gated Recurrent Unit (GRU) (Cho et al., 2014)



- A memory (vector) to store values of previous state
- Reset gate and update gate to control

$$r_t = \sigma(W_{rh}h_{t-1} + W_{rx}x_t + b_r)$$

$$z_t = \sigma(W_{zh}h_{t-1} + W_{zx}x_t + b_z)$$

$$g_t = \tanh(W_{gh}(r_t \otimes h_{t-1}) + W_{gx}x_t + b_g)$$

$$h_t = z_t \otimes h_{t-1} + (1 - z_t) \otimes g_t$$

# Recurrent Neural Network Language Model

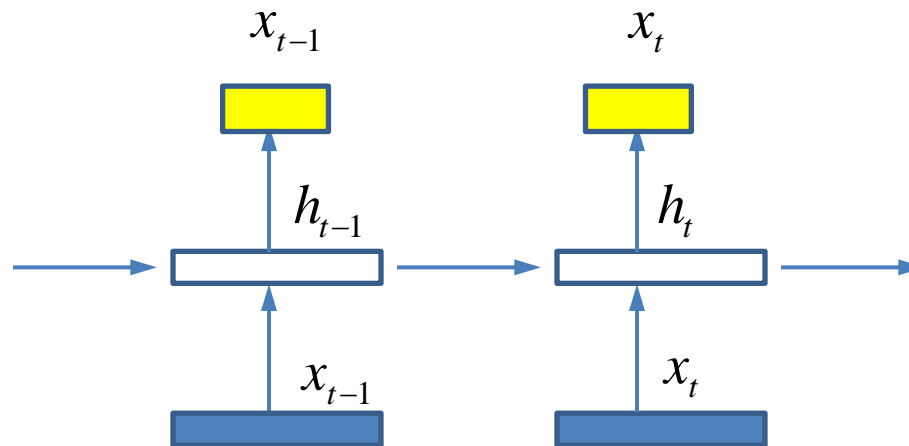
Model

$$h_t = \tanh(W_h h_{t-1} + W_x x_t + b_{hx})$$

$$p_t = P(x_t \mid x_1 \cdots x_{t-1}) = \text{soft max}(Wh_t + b)$$

Objective of Learning

$$\frac{1}{T} \sum_{t=1}^T -\log \hat{p}_t$$

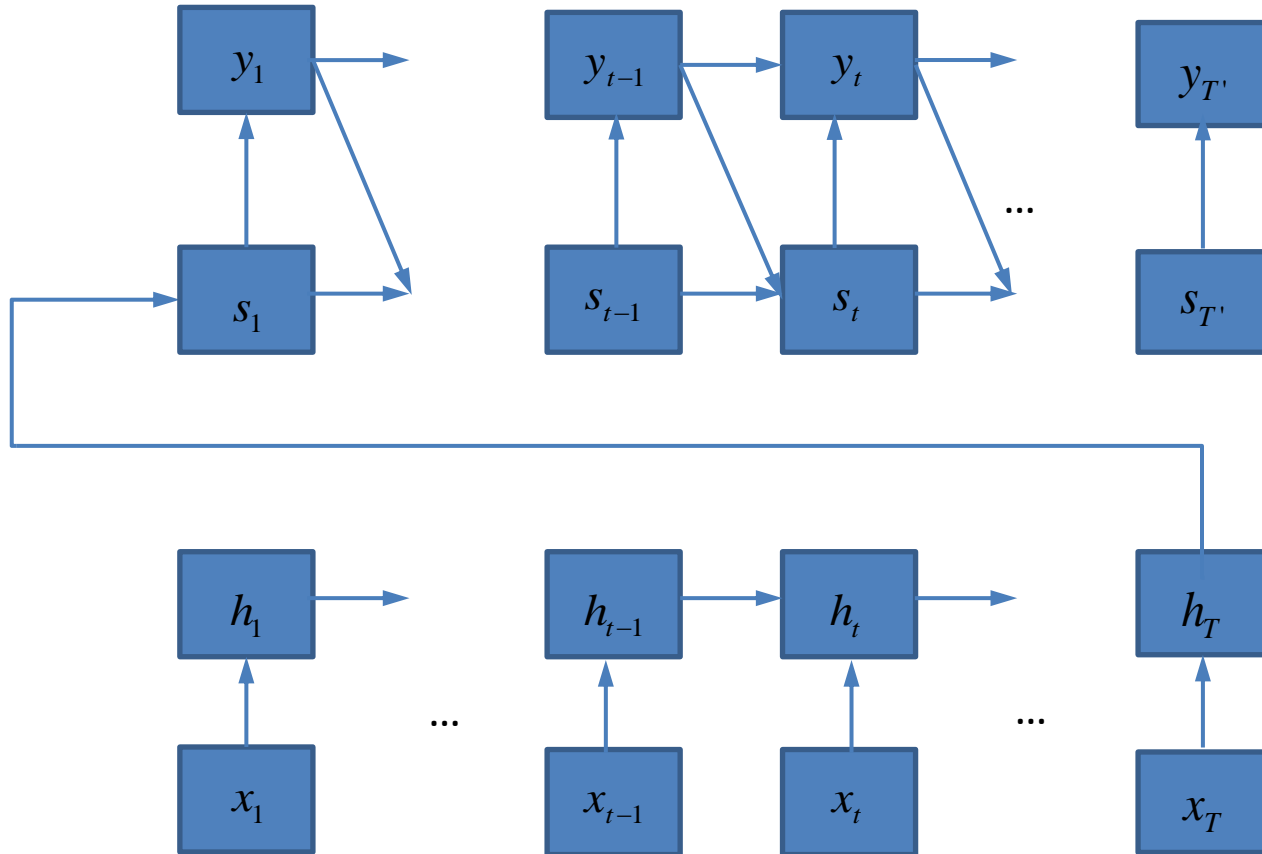


- Input one sequence and output another
- In training, input sequence is same as output sequence

# Sequence to Sequence Learning



# Translation: Sequence to Sequence Learning



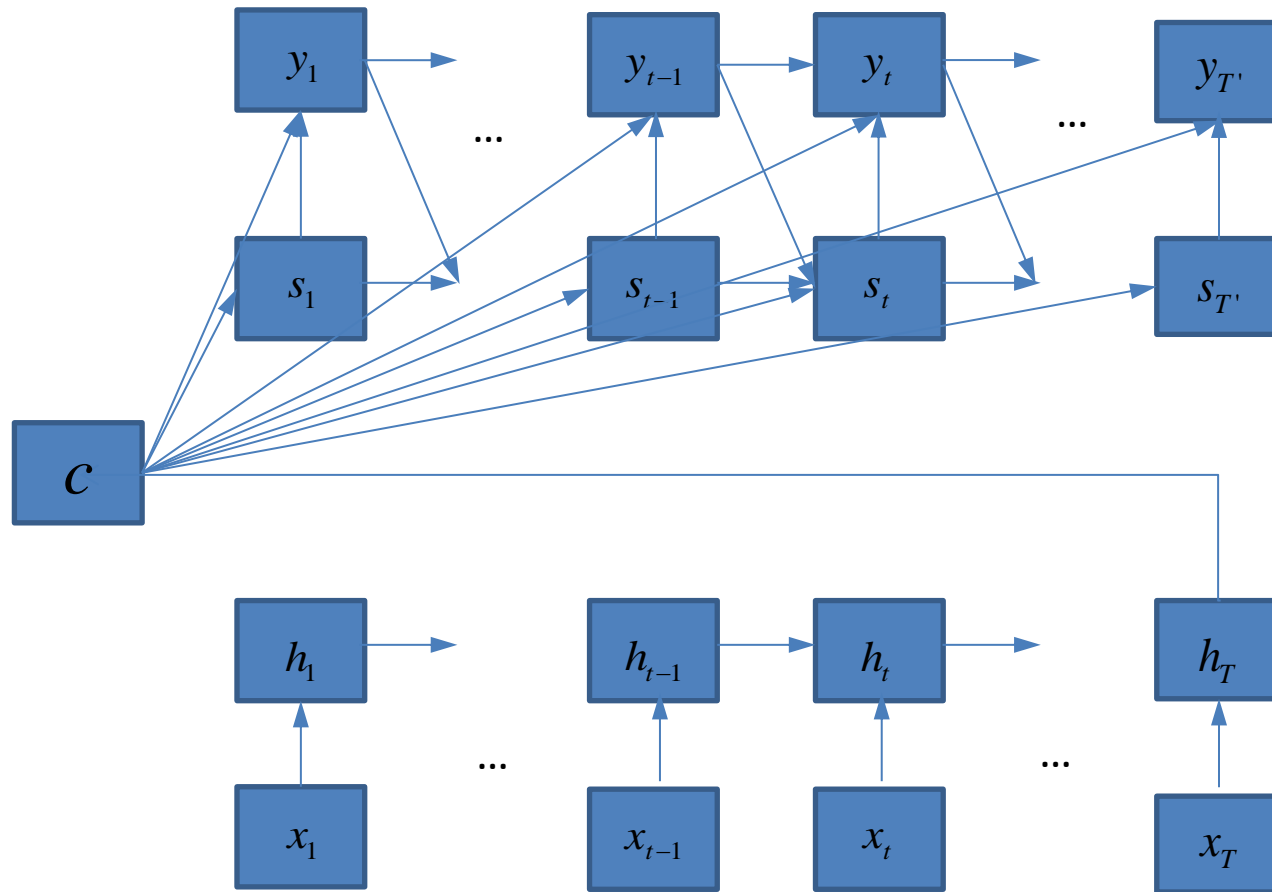
- Hierarchical LSTM
- Different LSTM models for encoder and decoder
- Reverse order of words in source sentence

$$P(y_t \mid y_1 \cdots y_{t-1}, \mathbf{x}) = g(y_{t-1}, s_t)$$

$$h_t = f_e(x_t, h_{t-1}), s_t = f_d(y_{t-1}, s_{t-1})$$

- Sutskever et al. 2014

# Translation: RNN Encoder-Decoder



- Context vector represents source sentence
- GRU is used

$$P(y_t | y_1 \cdots y_{t-1}, \mathbf{x}) = g(y_{t-1}, s_t, c), c = h_T$$

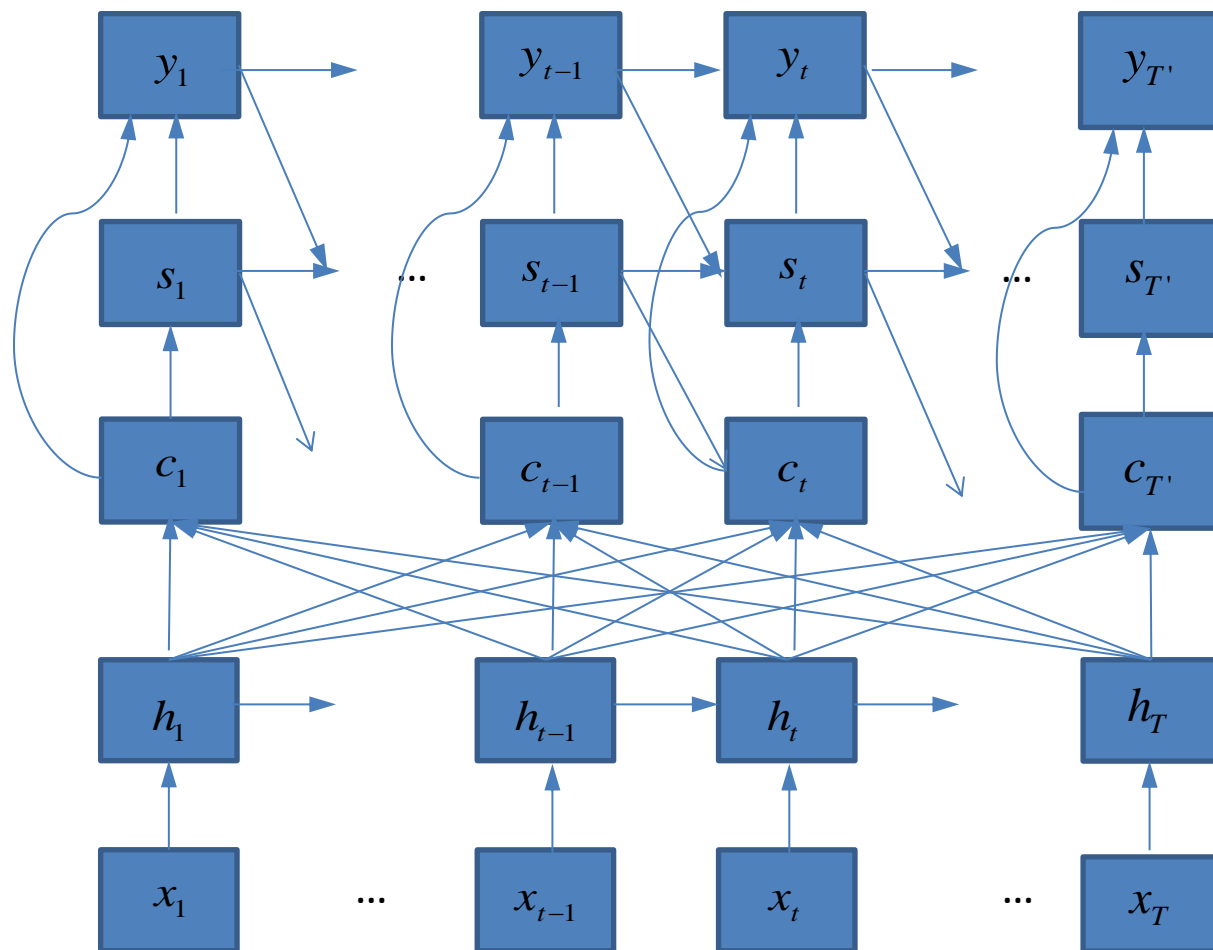
$$s_t = f_d(y_{t-1}, s_{t-1}, c)$$

$$h_t = f_e(x_t, h_{t-1})$$

- Cho et al. 2014



# Translation: Attention Mechanism



- Context vector represents attention
- Corresponds to alignment relation
- Encoder: Bidirectional RNN

$$P(y_t | y_1 \cdots y_{t-1}, \mathbf{x}) = g(y_{t-1}, s_t, c_t)$$

$$s_t = f_d(y_{t-1}, s_{t-1}, c_t)$$

$$h_t = f_e(x_t, h_{t-1})$$

$$c_t = \sum_{j=1}^T \alpha_{tj} h_j$$

$$\alpha_{tj} = q(s_{t-1}, h_j)$$

Bahdanau, et al. 2014

# Outline of Lecture

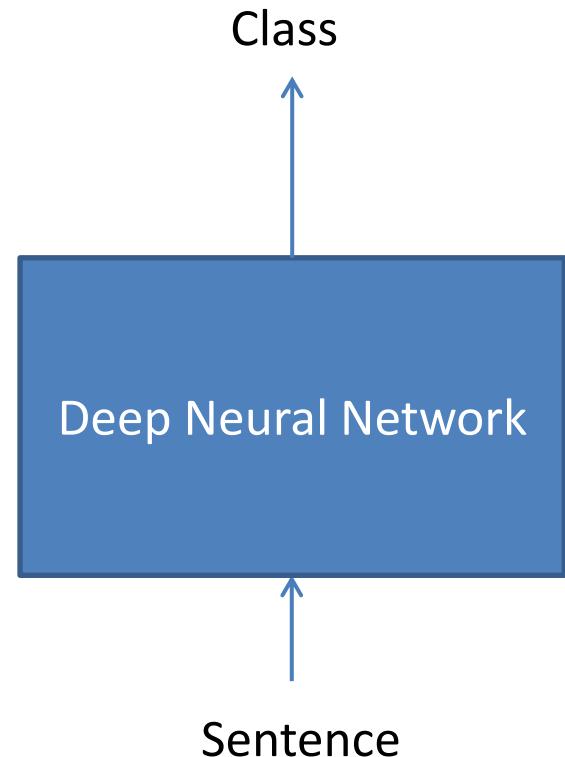
- Introduction
- Basics of DL for NLP
- *State of the Art of DL for NLP*
- Previous Work at Noah's Ark Lab
- Recent Progress at Noah's Ark Lab
- Advantages and Disadvantages
- Summary

# State of the Art of DL for NLP

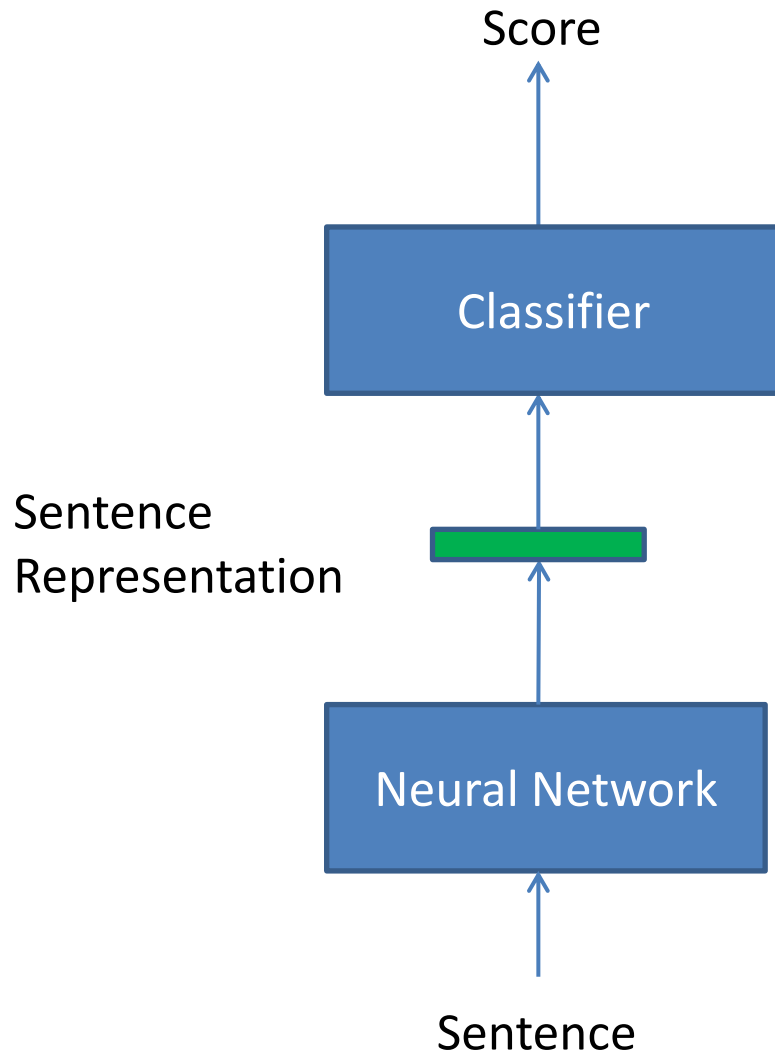
- Classification
  - Matching
  - Translation
  - Structured Prediction
- 
- References can be found at Hang Li, Zhengdong Lu, SIGIR 2016 Tutorial

# Classification

- Examples of Tasks
  - **Search:** query classification, document classification
  - **Question Answering:** question classification, answer classification
- Approaches
  - World Level Model
  - Character Level Model
  - Hierarchical Model (for document classification)



# Sentence Classification: Word Level Model



**Classifier:**

Softmax

**Neural Network:**

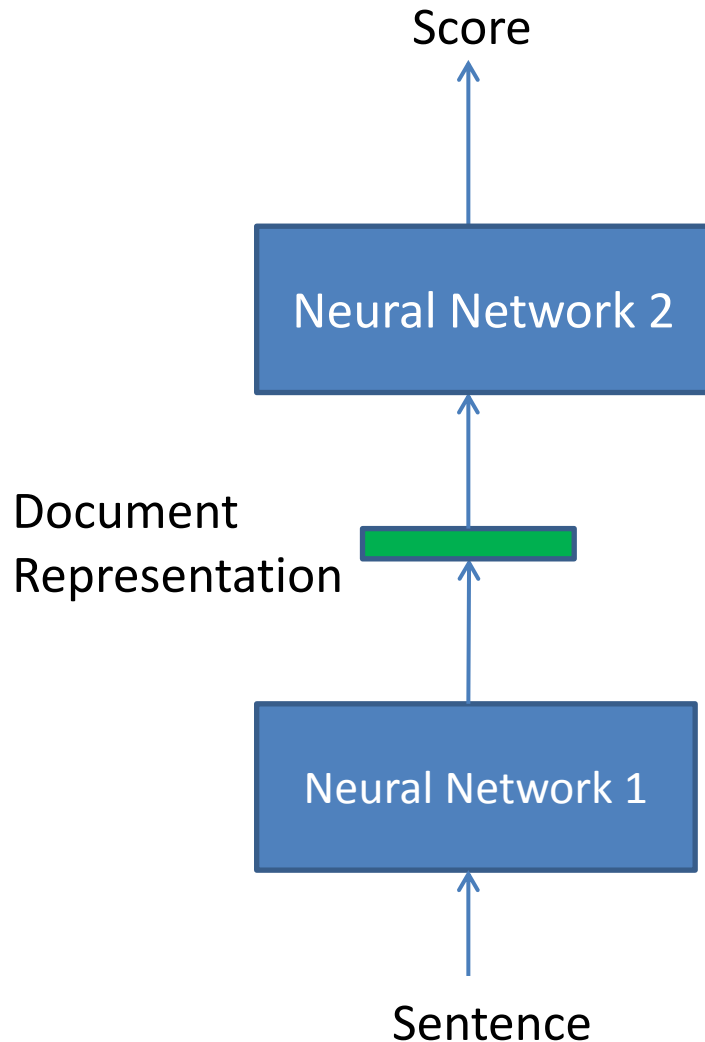
Convolutional Neural Network,  
Deep Neural Network

**Input:**

Continuous Word Embedding,  
Discrete Word Embedding  
(one-hot)

- Kim 2014
- Blunsom et al. 2014
- Johnson & Zhang 2015
- Iyer et al. 2015

# Document Classification: Character Level Model



## **Neural Network 1:**

*Deep Convolutional Neural Network*

## **Neural Network 2:**

3-Layer Fully-Connected Neural Network

## **Input:**

Character Embedding

## **Data:**

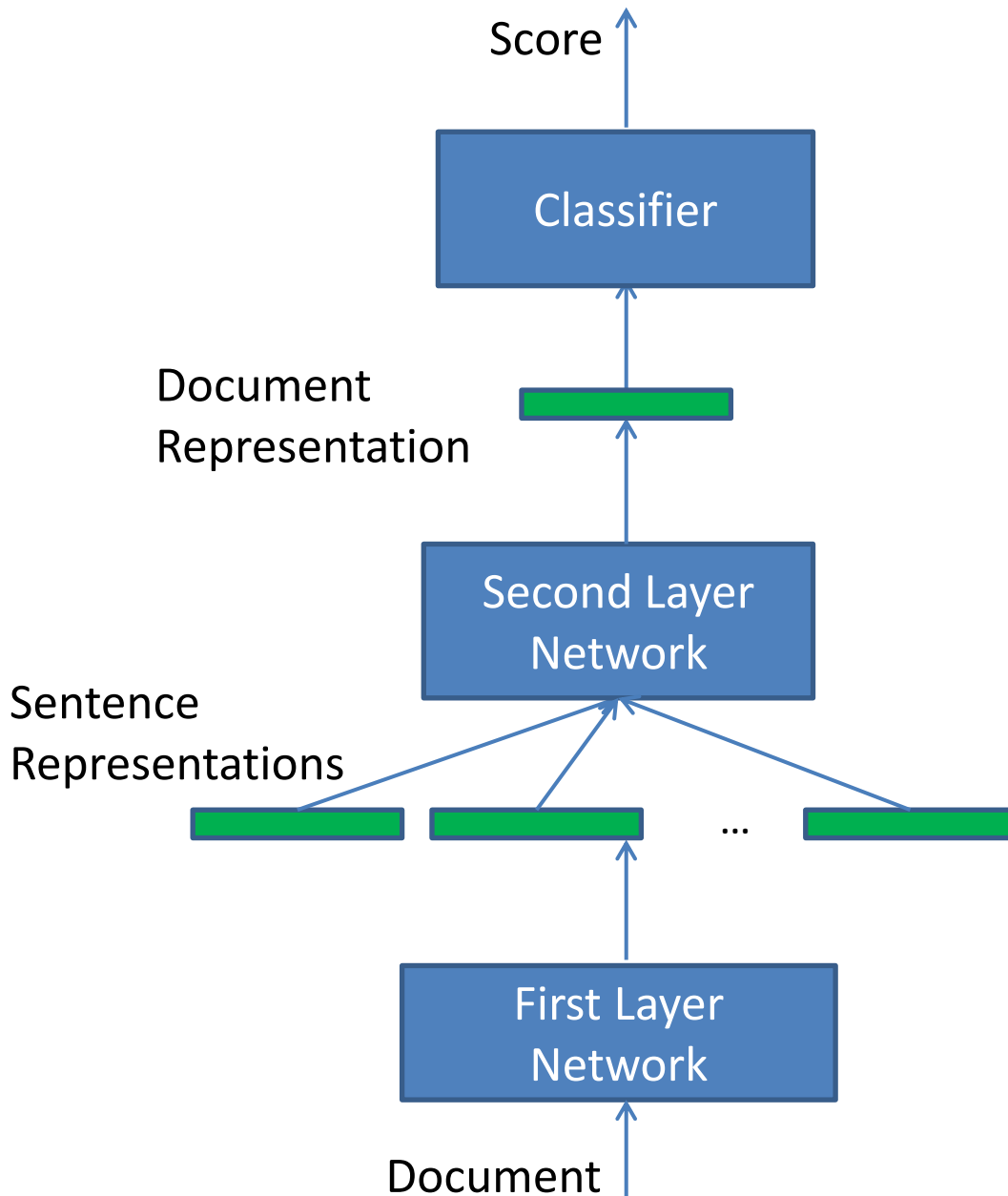
Large Scale Training Dataset

## **Class:**

Semantic Topics

• Zhang et al. 2016

# Document Classification: Hierarchical Model



**Classifier:**

Softmax

**First Layer Network:**

Recurrent Neural  
Network (LSTM, GRU)

**Second Layer Network:**

Recurrent Neural  
Network (LSTM, GRU)

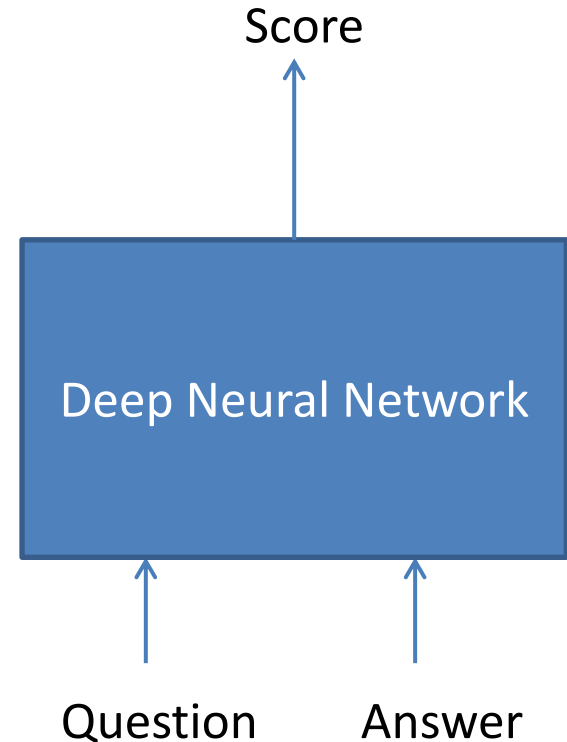
**Attention:**

Can be Employed  
between Two Layers

- Tang et al. 2015
- Lai et al. 2015
- Yang et al. 2016

# Matching

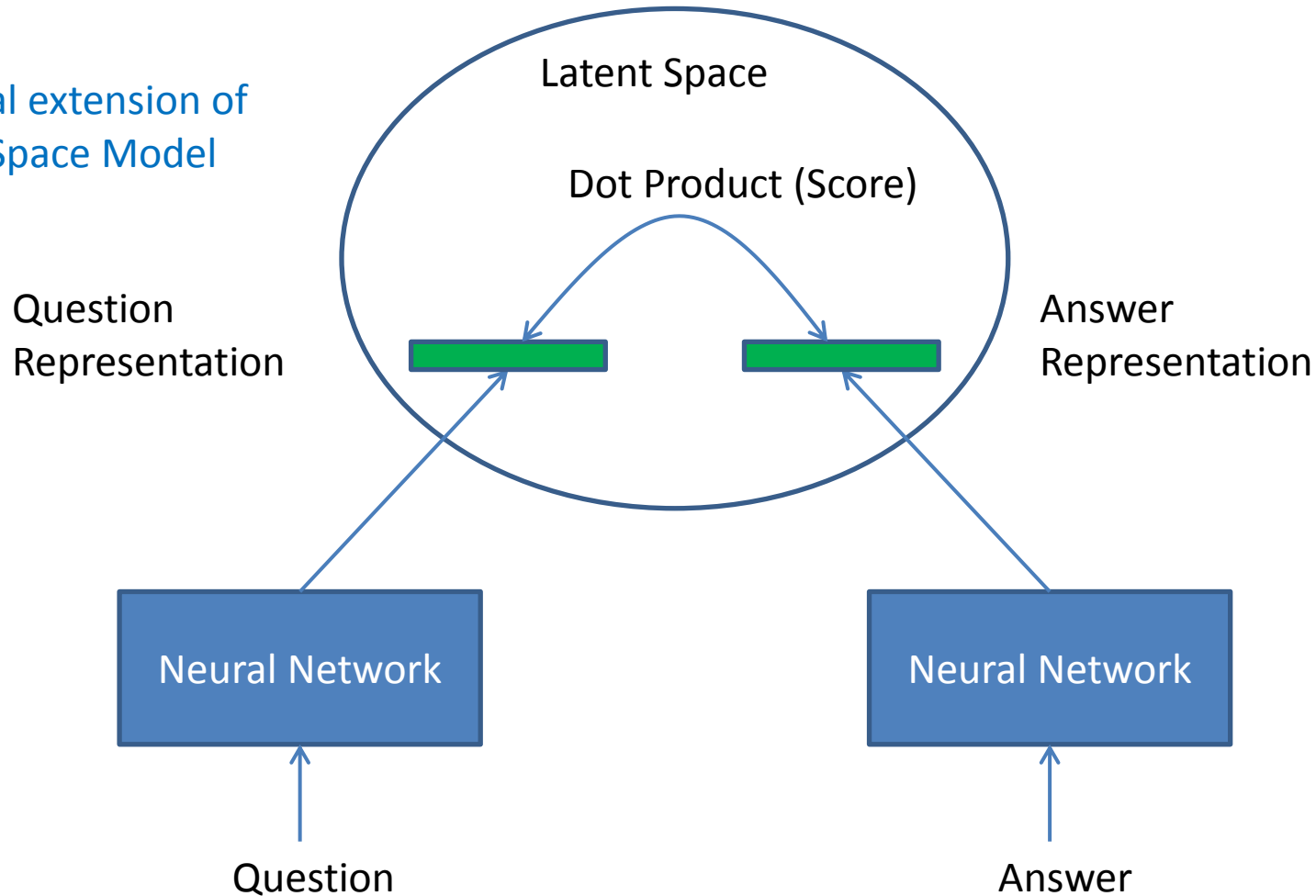
- Examples of Tasks
  - **Search:** query-document (title) matching, similar query finding
  - **Question Answering:** question answer matching
- Approaches
  - Projection to Latent Space
  - One Dimensional Matching
  - Two Dimensional Matching
  - Tree Matching





# Matching: Projection to Latent Space

- Natural extension of Vector Space Model

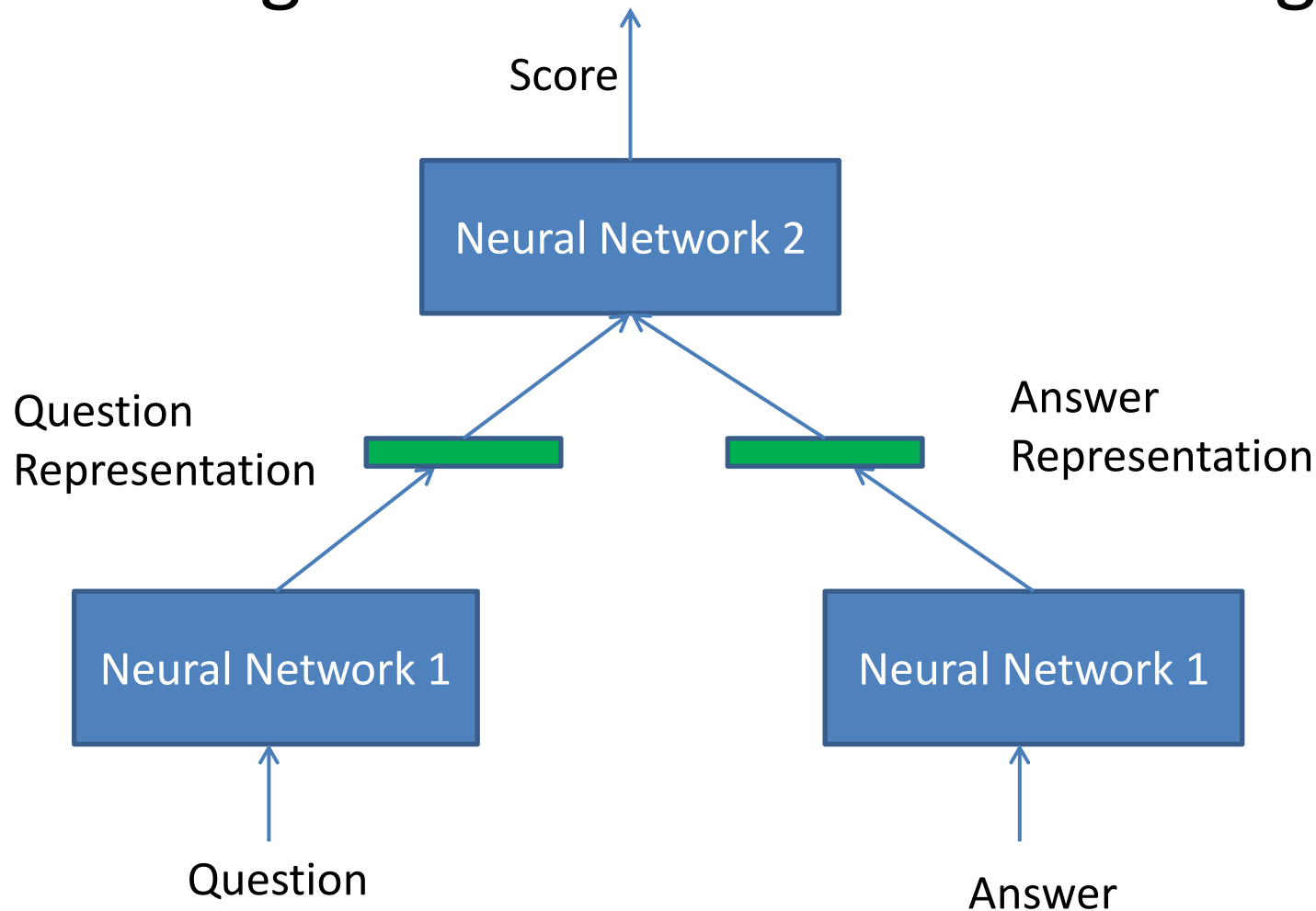


## Neural Networks:

Convolutional Neural Network  
Deep Neural Network  
Recurrent Neural Network

- Huang et al. 2013
- Shen et al. 2014
- Severyn & Moschitti 2015

# Matching: One Dimensional Matching



## Neural Network 1:

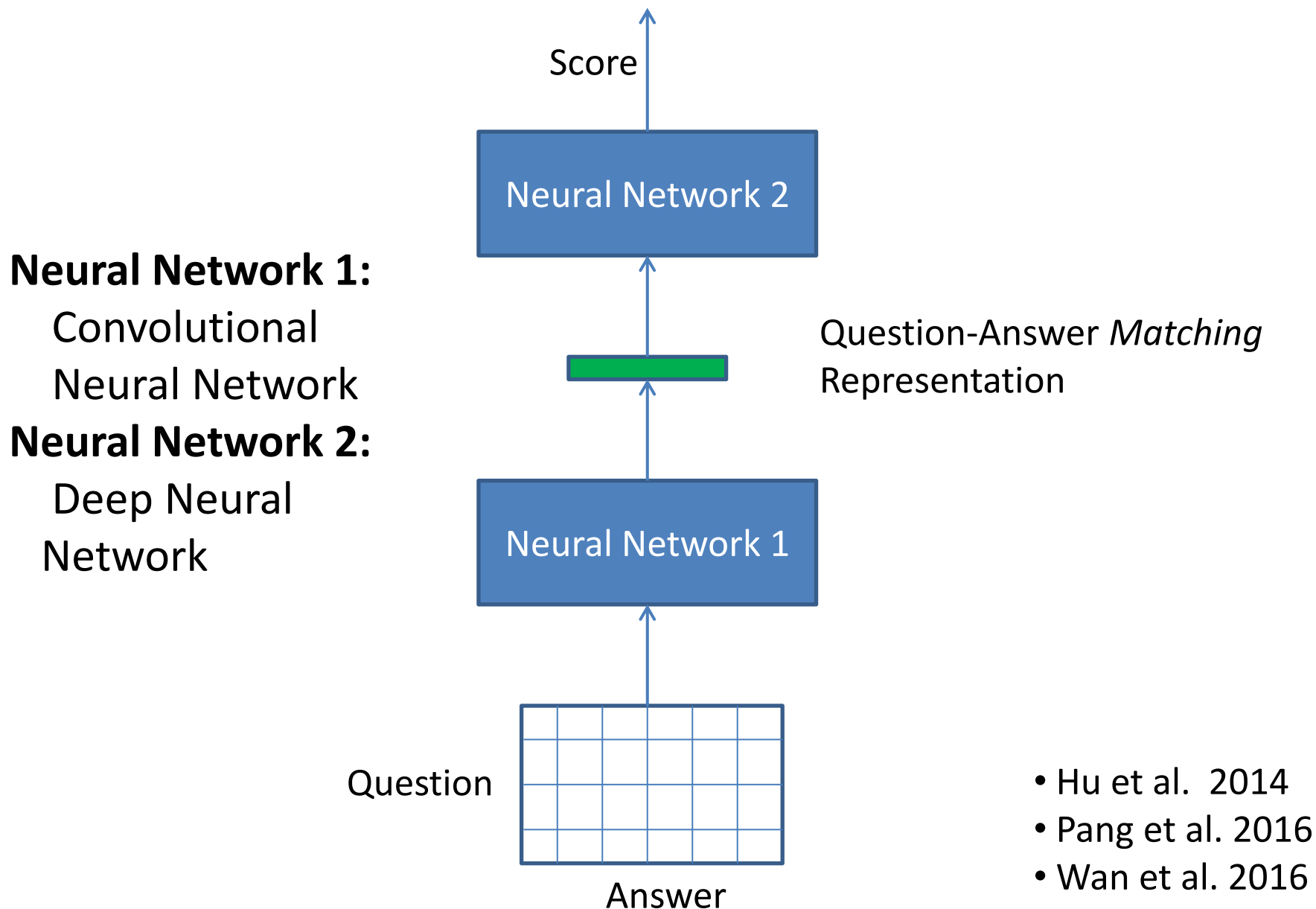
Convolutional Neural Network

## Neural Network 2:

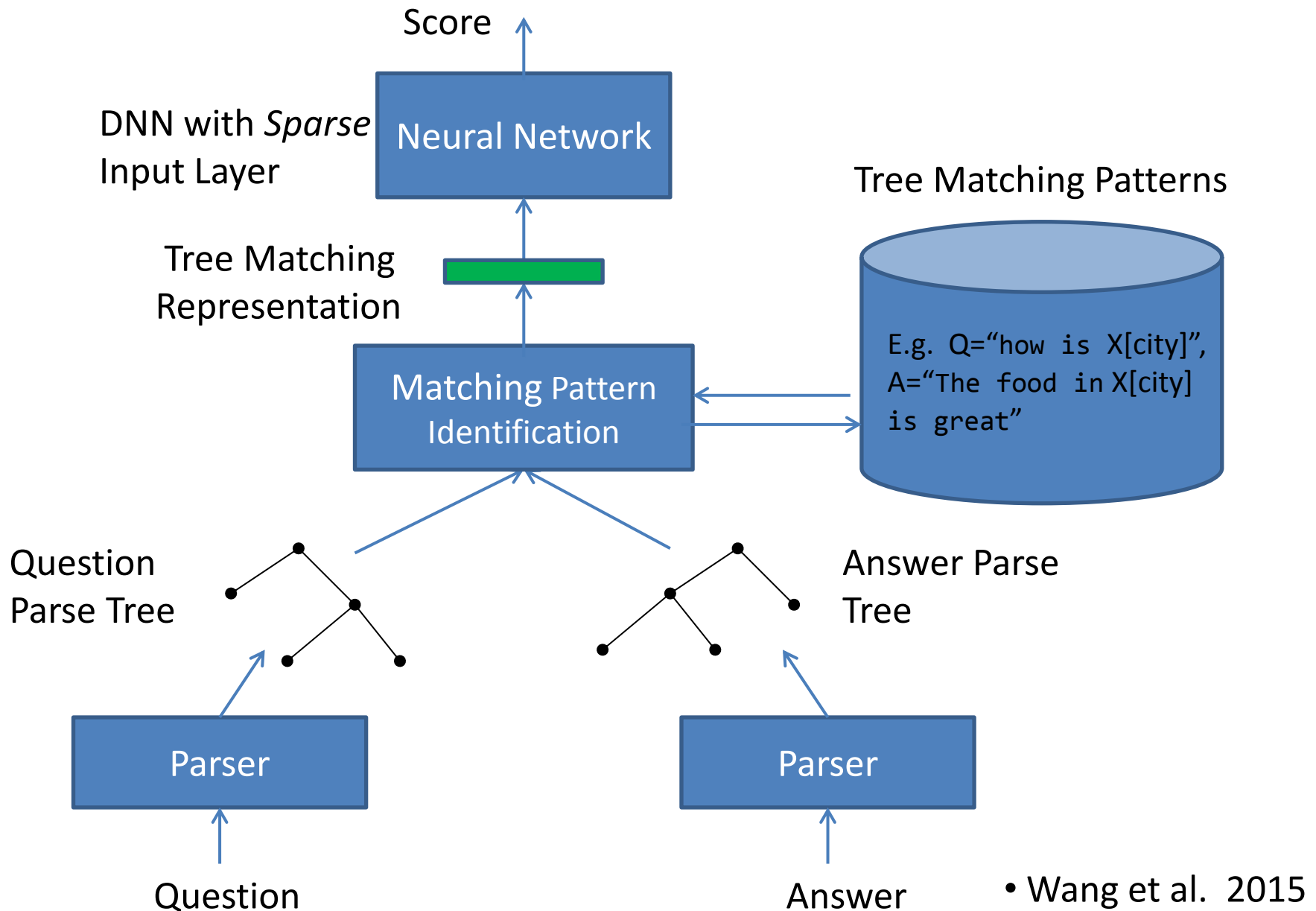
Deep Neural Network, Tensor Network

- Hu et al. 2014
- Qiu & Huang 2015

# Matching: Two Dimensional Matching

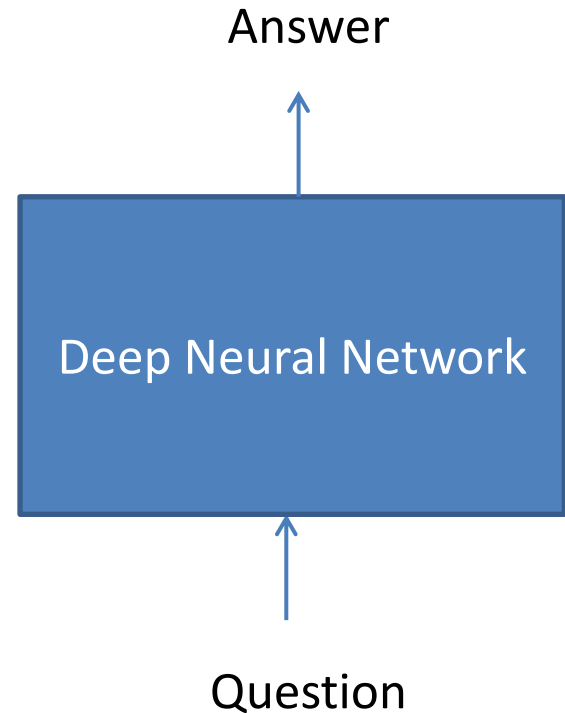


# Matching: Tree Matching

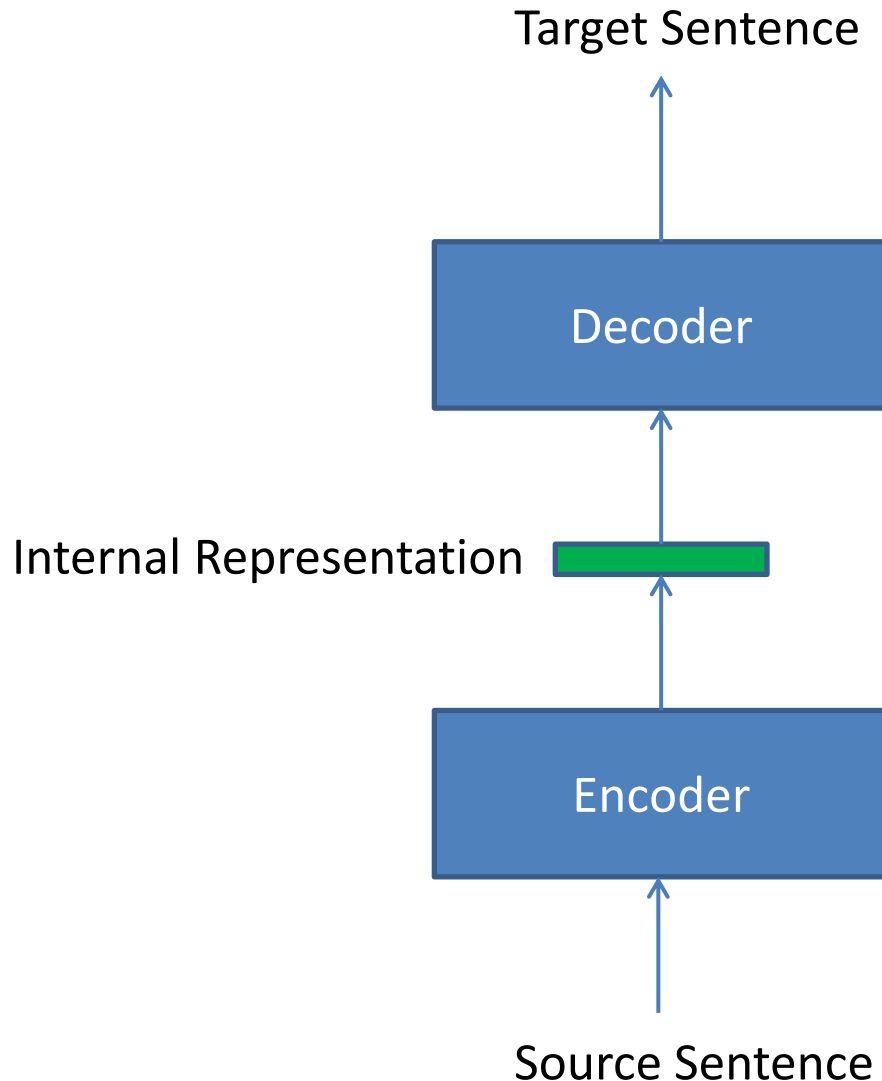


# Translation

- Examples of Tasks
  - **Machine Translation:**  
sentence level translation
  - **Question Answering:**  
answer generation from question
  - **Search:** similar query generation
- Approaches
  - Sequence-to-Sequence Learning
  - RNN Encoder-Decoder
  - Attention Mechanism



# Translation: Sequence-to-Sequence Learning (Same for RNN Encoder-Decoder)



**Encoder:**

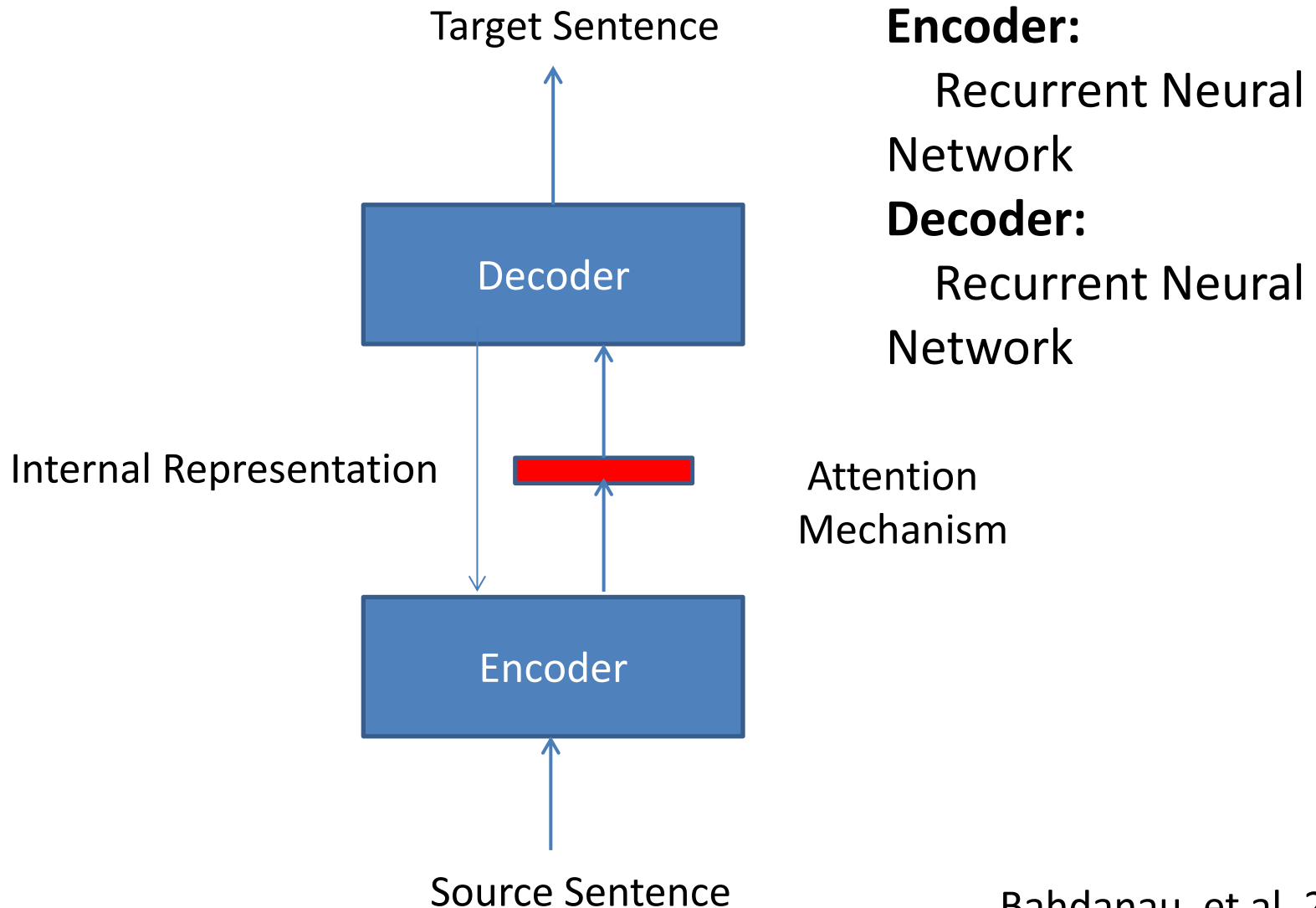
Recurrent Neural  
Network

**Decoder:**

Recurrent Neural  
Network

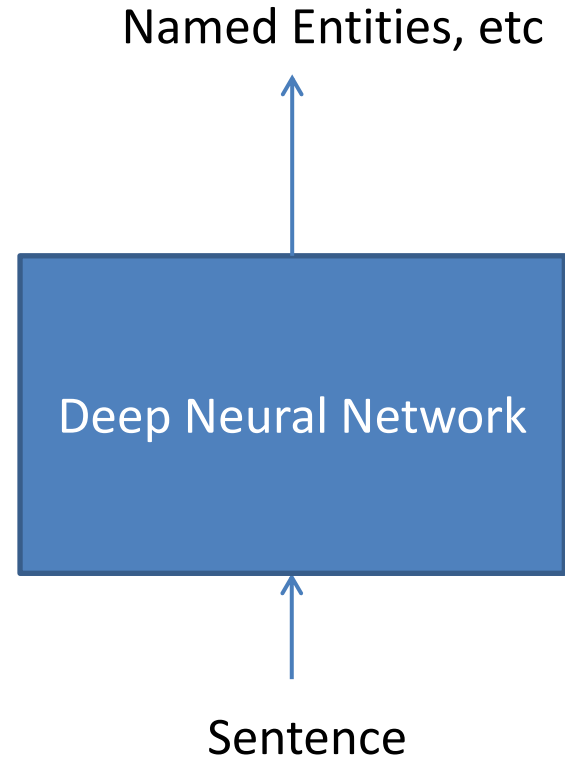
- Sutskever et al. 2014
- Cho et al. 2014

# Translation: Sequence-to-Sequence Learning



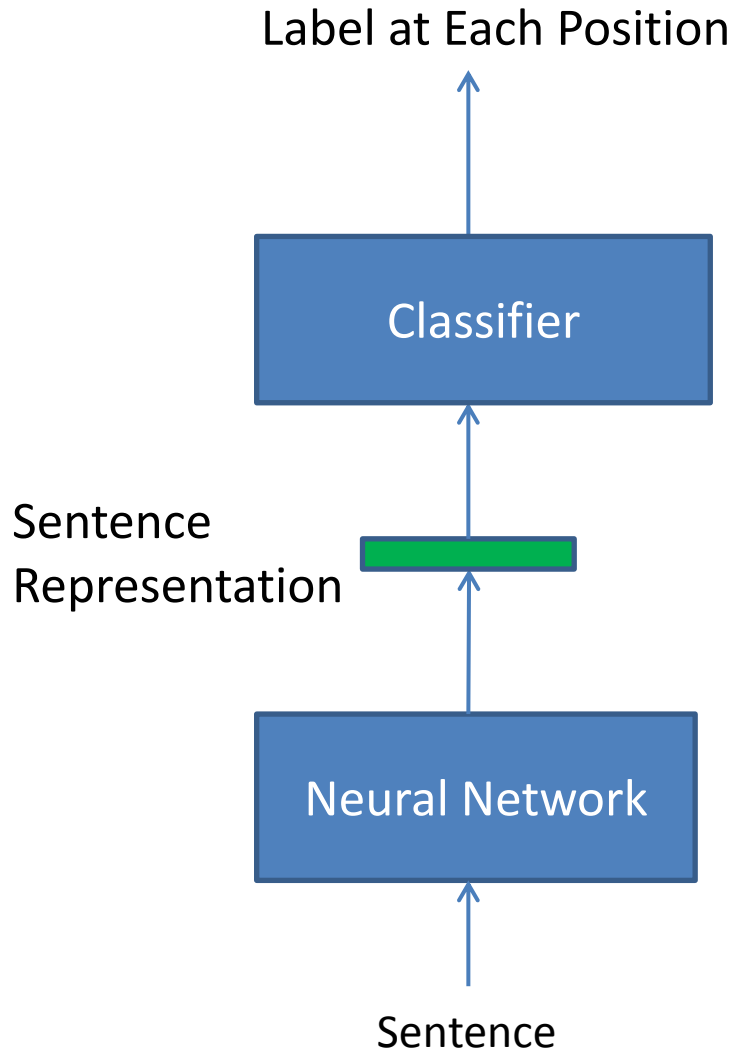
# Structured Prediction

- Examples of Tasks
  - **Search:** named entity recognition in query and document
  - **Question Answering:** named entity recognition in question and answer
- Approaches
  - CNN
  - Sequence-to-Sequence Learning
  - Neural Network based Parsing





# Structured Prediction: CNN



**Classifier at Each Position:**

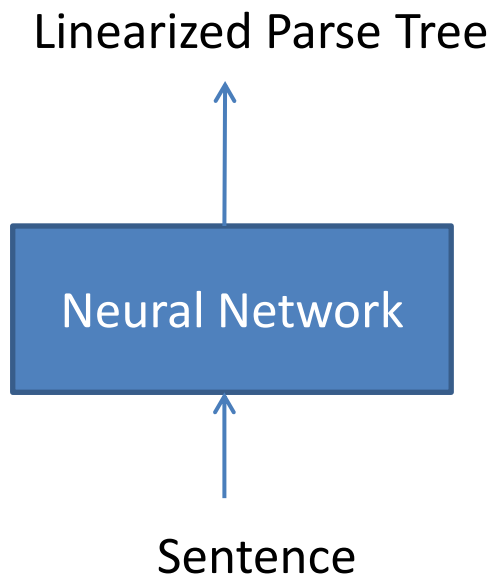
Softmax

**Neural Network:**

Convolutional Neural  
Network

- Collobert et al. 2011

# Structured Prediction: Sequence-to-Sequence Learning



## **Neural Network:**

Sequence-to-Sequence Learning Model

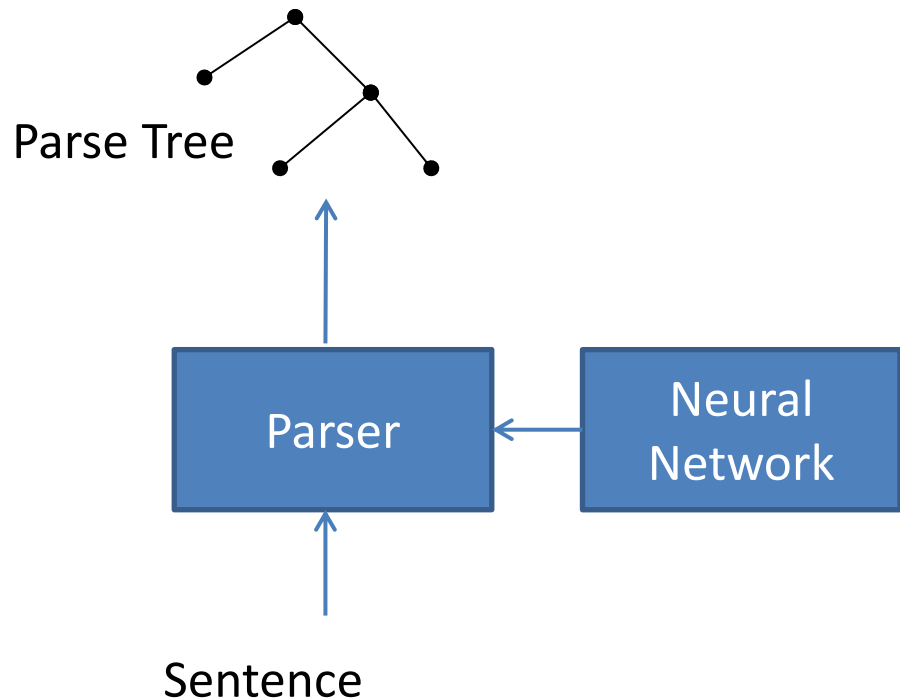
## **Training Data:**

Pairs of Sentence and Linearized Parse Tree

E.g.,

John has a dog .  $\rightarrow$  (S (NP NNP)<sub>NP</sub> (VP VBZ (NP DT NN)<sub>NP</sub> )<sub>VP</sub> . )<sub>S</sub>

# Structured Prediction: Neural Network based Parsing



## **Parser:**

Transition-based Dependency Parser, Constituency Parser, CRF Parser

## **Neural Network:**

Deep Neural Networks

## **Training Data:**

Pairs of Sentence and Parse Tree

- Chen & Manning, 2014
- Durrett & Klein, 2015
- Zhou et al., 2015
- Andor et al., 2016

# Outline of Lecture

- Introduction
- Basics
- State of the Art
- *Previous Work at Noah's Ark Lab*
- Recent Progress at Noah's Ark Lab
- Advantages and Disadvantages
- Summary

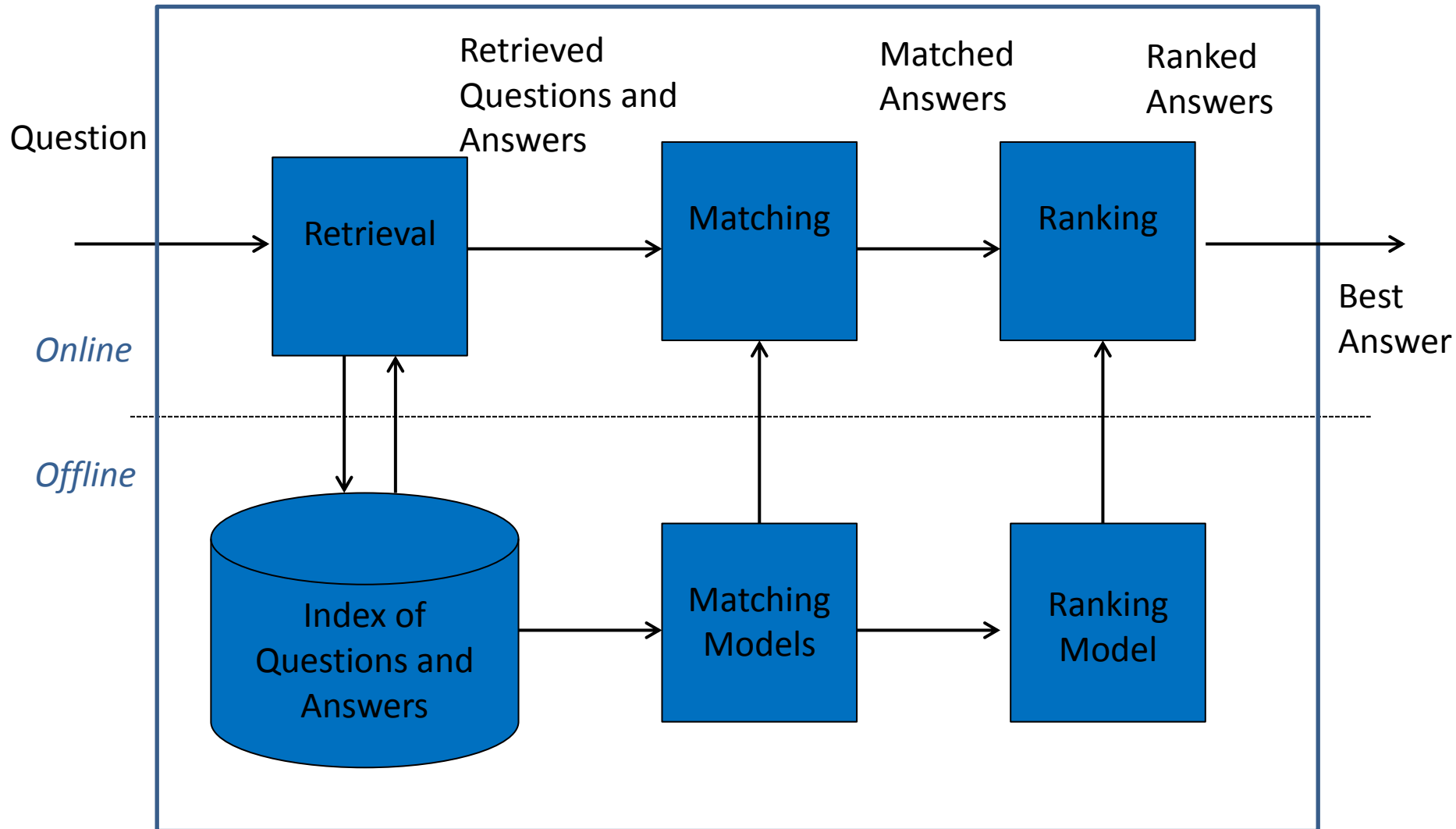
# Question Answering

- DeepMatch CNN

# Demo

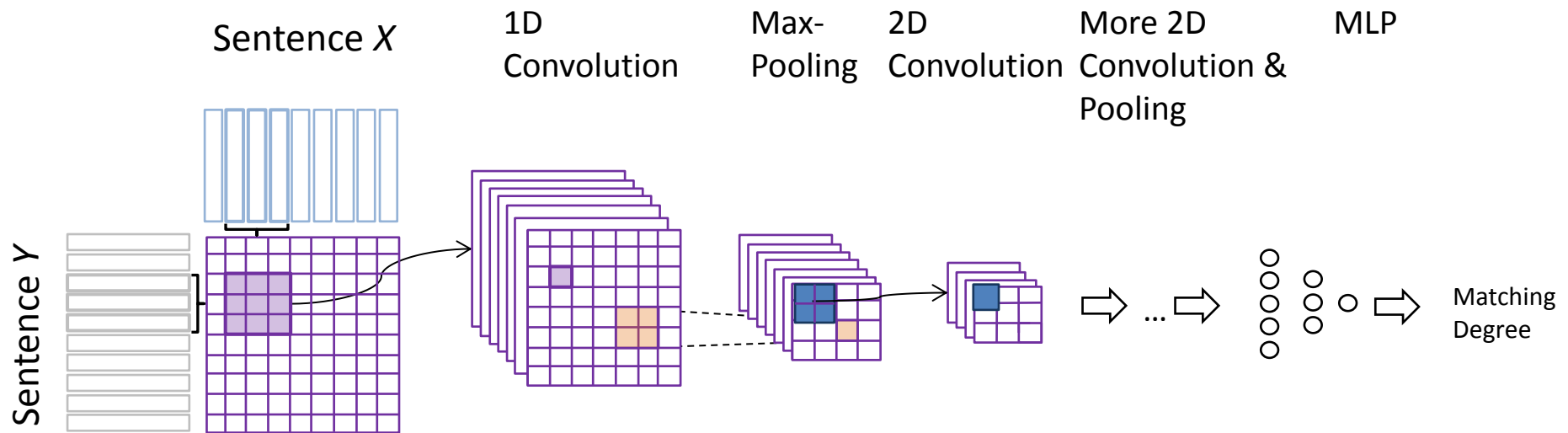


# Retrieval based Question Answering System



# Deep Match Model CNN

- Represent and match two sentences simultaneously
- Two dimensional model
- State of art model for matching in question answering





# Image Retrieval

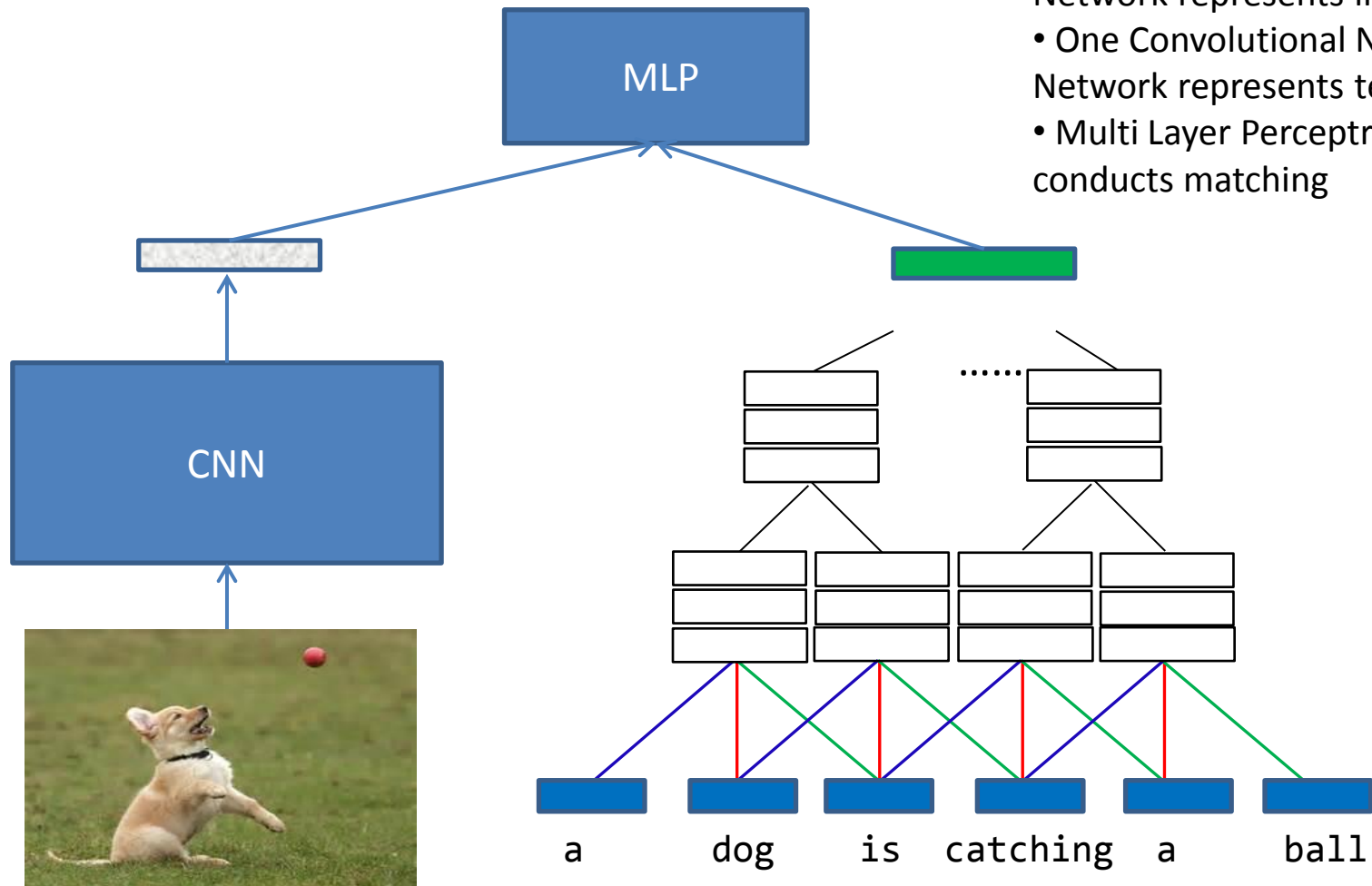
- Multimodal CNN

# Demo



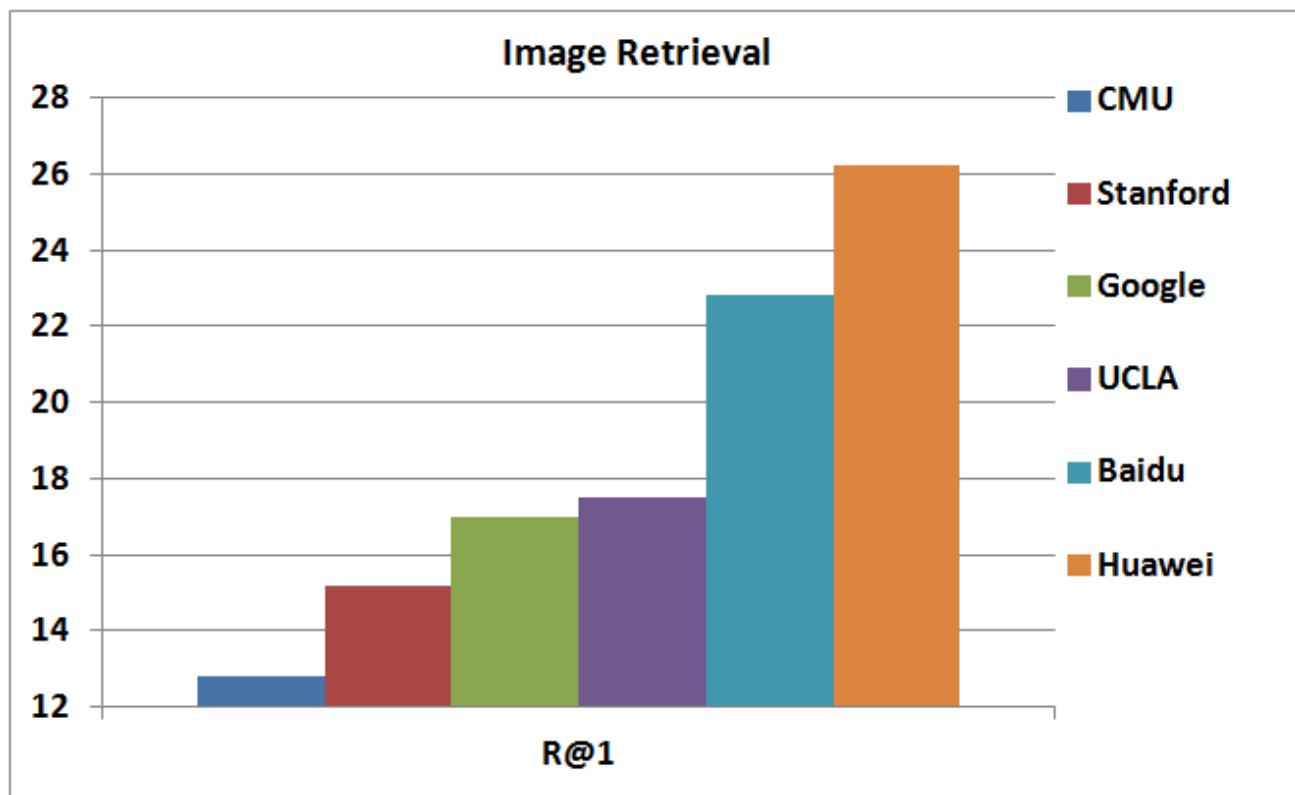
# Multimodal CNN

- One Convolutional Neural Network represents image
- One Convolutional Neural Network represents text
- Multi Layer Perceptron conducts matching



# Experimental Results

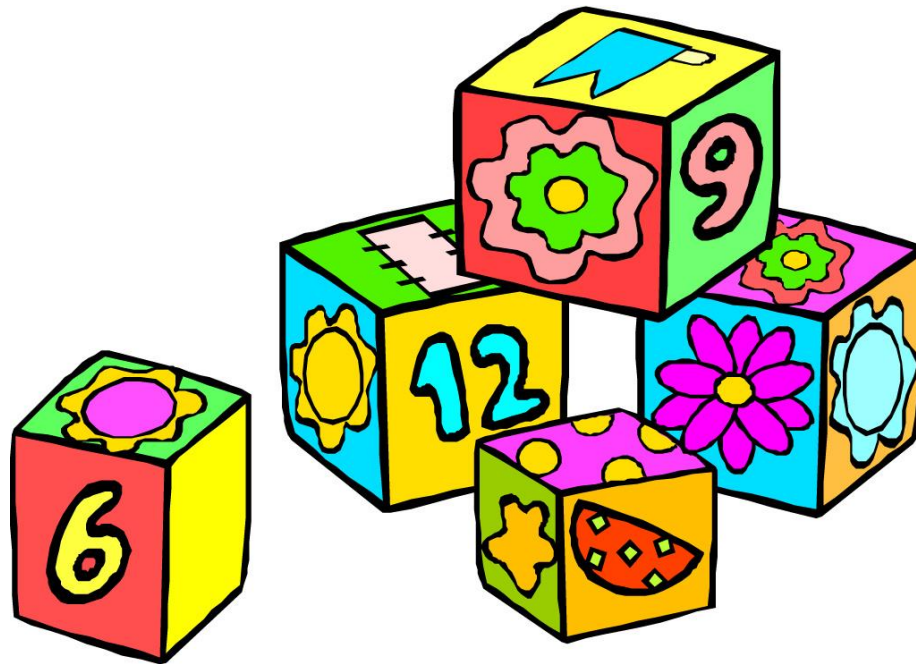
- Experiment
  - Trained with 30K Flickr data
  - Outperforming other state-of-the-art models



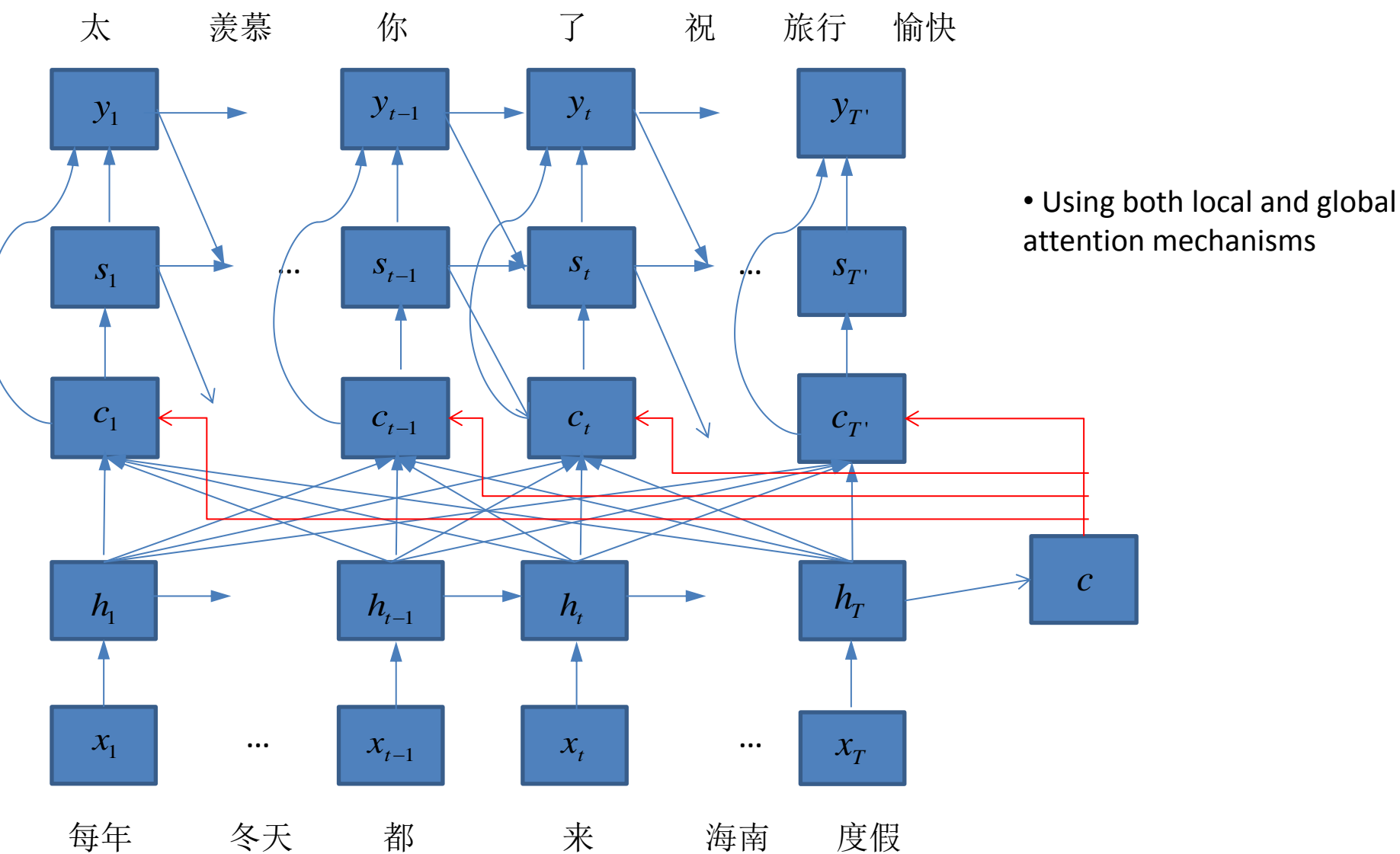
# Natural Language Dialogue

- Neural Responding Machine

# Demo



# Neural Responding Machine



# Experimental Results

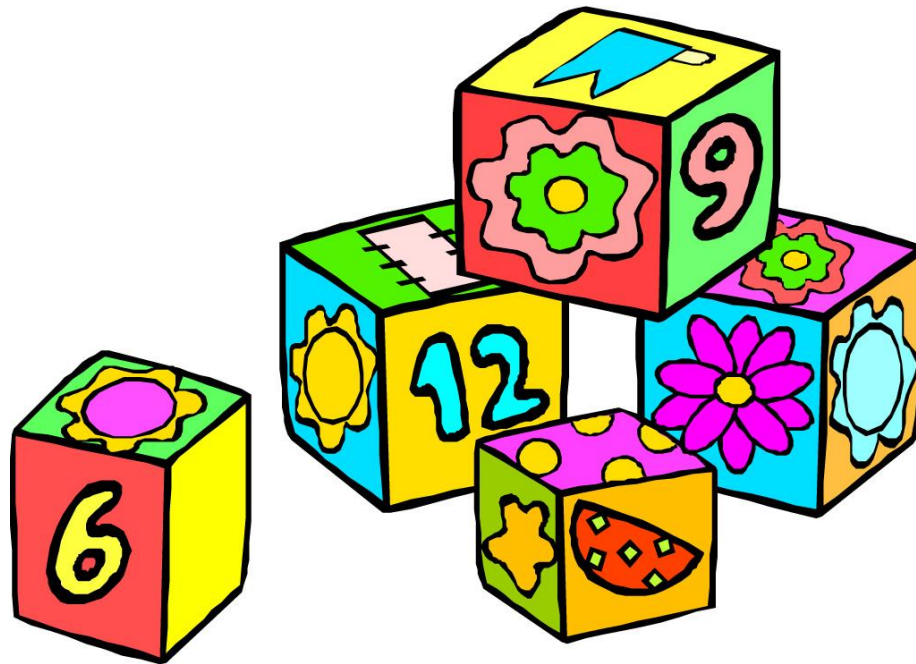
- Experiment
  - Trained with 4.4 million Weibo data (Chinese)
  - 95% of responses are natural, 76% of responses are appropriate as replies

Message	Response
Occupy Central is finally over	Will Lujiazui (finance district in Shanghai) be the next?
I want to buy a Samsung phone	Let us support our national brand.

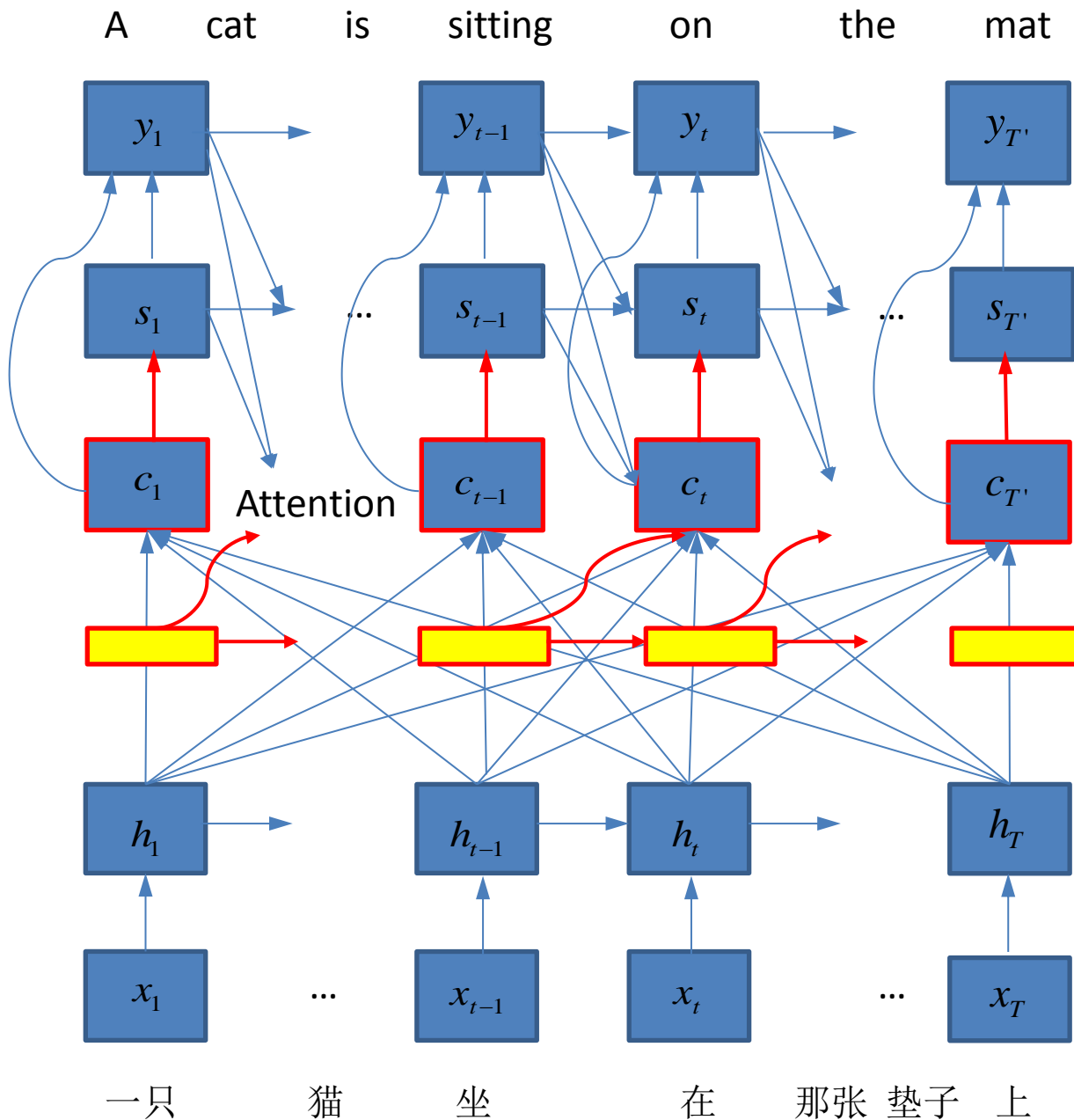


# Neural Machine Translation

# Demo



# Neural Machine Translation

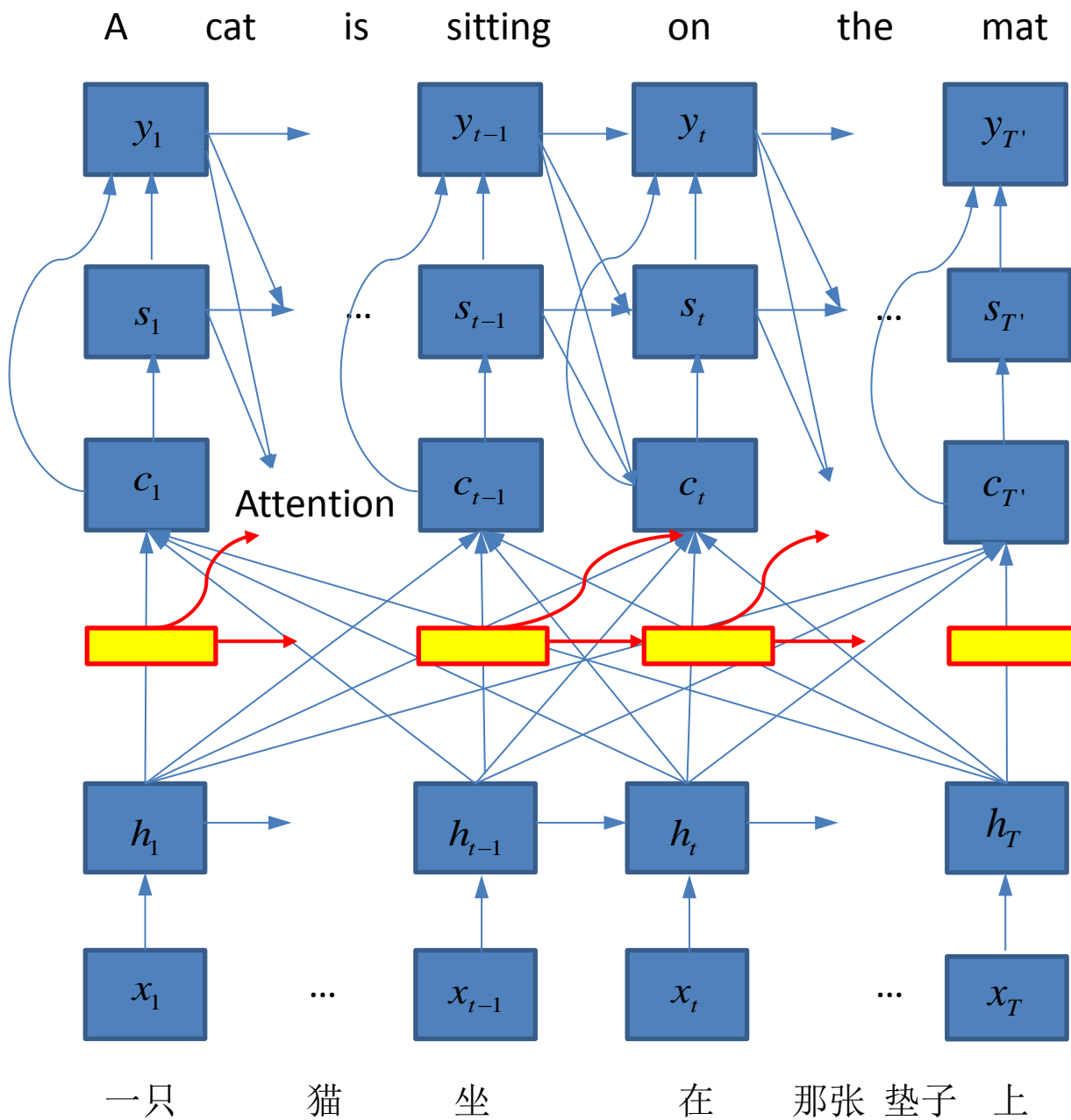


Decoder

- Using coverage vectors to avoid over-translation and under-translation
- Using context gates to dynamically control the impact of attention

Encoder

# Coverage Vector



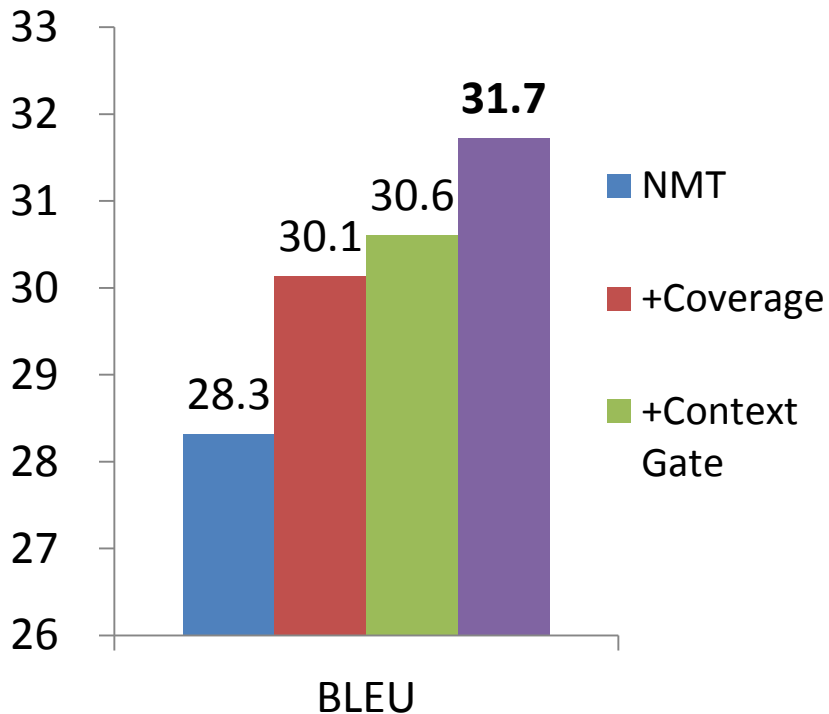
Decoder

- Using coverage vectors to avoid over-translation and under-translation

Encoder

# Experimental Results

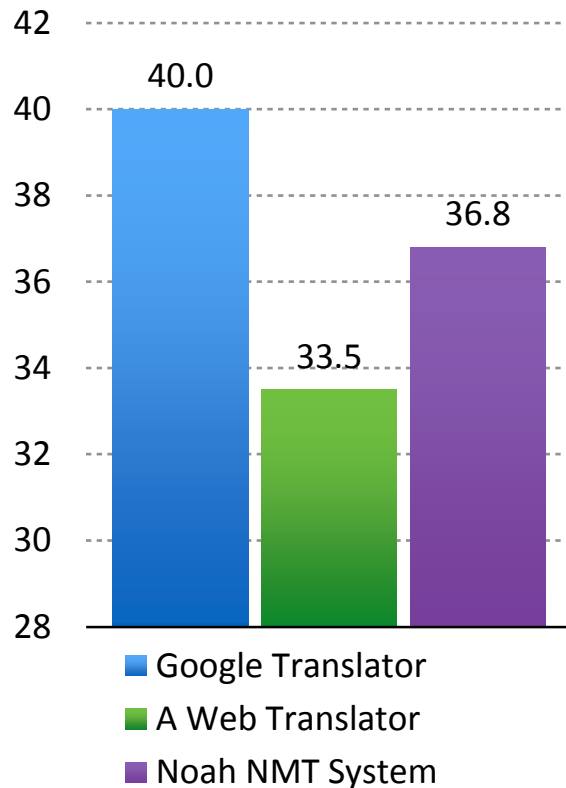
- Experiment
  - Trained with 1.25 million LDC data (Chinese-English)



C-E Translation	有一些恐怖袭击会愈演愈烈。
NMT	There will be some terrorist attacks.
+Both	Some terrorist attacks <i>will become more and more intense.</i>

# Experimental Result

- Google NMT system works better, apparently due to its larger training data and more powerful computing architecture
- Google NMT system also employs coverage mechanism



# Outline of Lecture

- Introduction
- Basics of DL for NLP
- State of the Art of DL for NLP
- Previous Work at Noah's Ark Lab
- *Recent Progress at Noah's Ark Lab*
- Advantages and Disadvantages
- Summary

# Question Answering - Neural Enquirer



# Question Answering from Relational Database

**Q:** How many people participated in the game in Beijing?

**A:** 4,200

**SQL:** *select #\_participants, where city=beijing*

**Q:** When was the latest game hosted?

**A:** 2012

**SQL:** *argmax(city, year)*

Relational Database

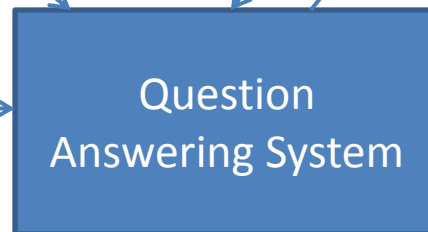
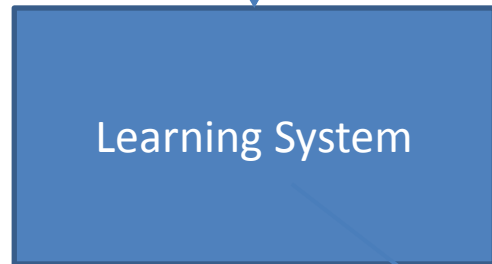
year	city	#_days	#_medals
2000	Sydney	20	2,000
2004	Athens	35	1,500
2008	Beijing	30	2,500
2012	London	40	2,300

Learning System

Question  
Answering System

**Q:** Which city hosted the longest Olympic game before the game in Beijing?

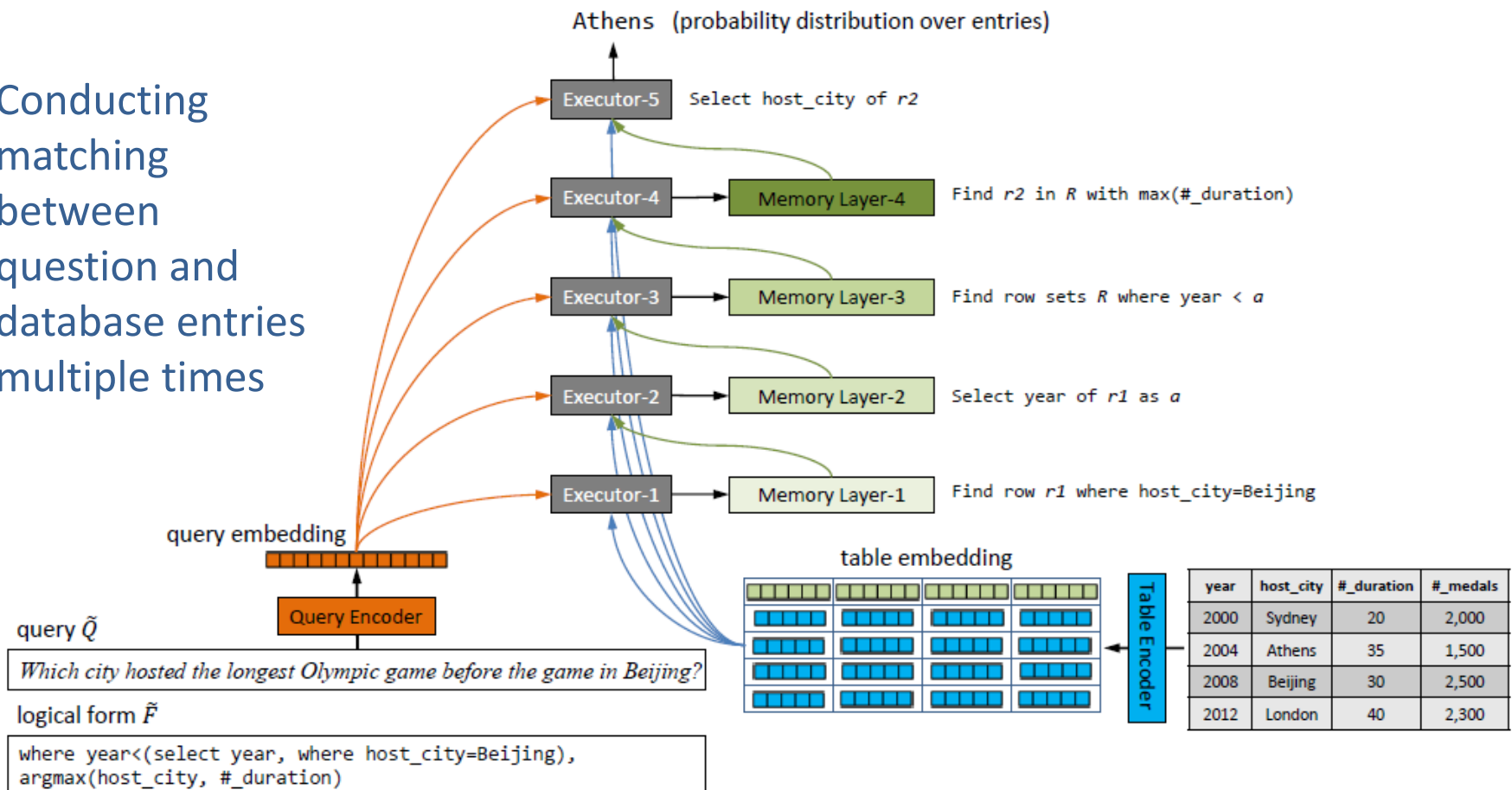
**A:** Athens



# Neural Enquirer

- Query Encoder: encoding query
- Table Encoder: encoding entries in table
- Five Executors: executing query against table

Conducting matching between question and database entries multiple times



# Query Encoder and Table Encoder

Query Encoder

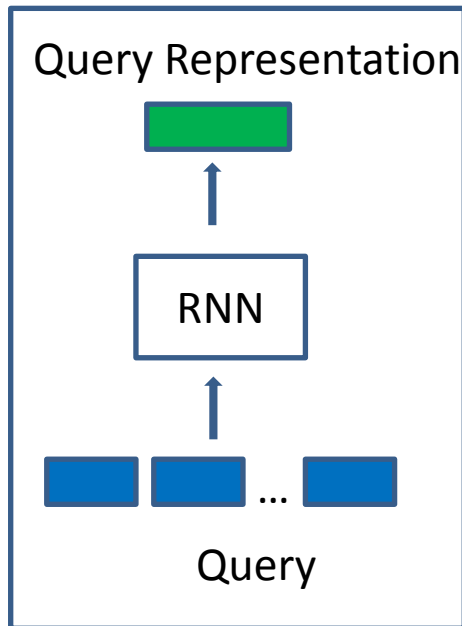


Table Encoder

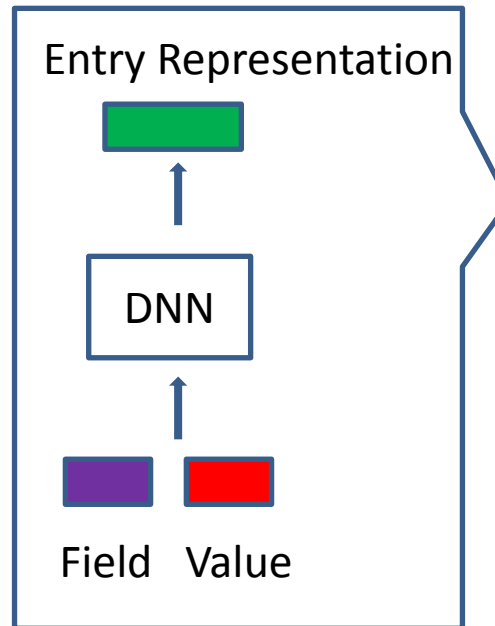




















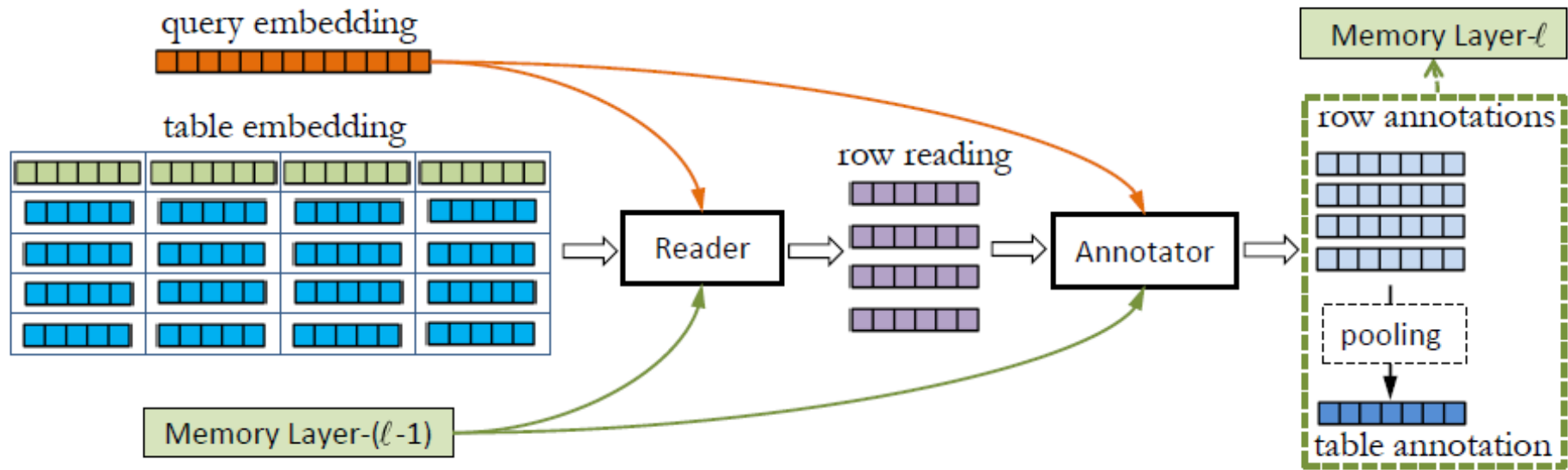


Table Representation

- Creating query embedding using RNN
- Creating table embedding for each entry using DNN

# Executors



- Five layers, except last layer, each layer has reader, annotator, and memory
- Reader fetches important representation for each row, e.g., city=beijing
- Annotator encodes result representation for each row, e.g., row where city=beijing

# Experimental Results

- Experiment
  - Olympic database
  - Trained with 25K and 100K synthetic data
  - Accuracy: 84% on 25K data, 91% on 100K data
  - Significantly better than SemPre (semantic parser)
  - Criticism: data is synthetic

25K Data			100K Data		
Semantic Parser	End-to-End	Step-by-Step	Semantic Parser	End-to-End	Step-by-Step
65.2%	84.0%	96.4%	NA	90.6%	99.9%

# Question Answering

## - GenQA

# Question Answering from Knowledge Graph

**Q:** How tall is Yao Ming?

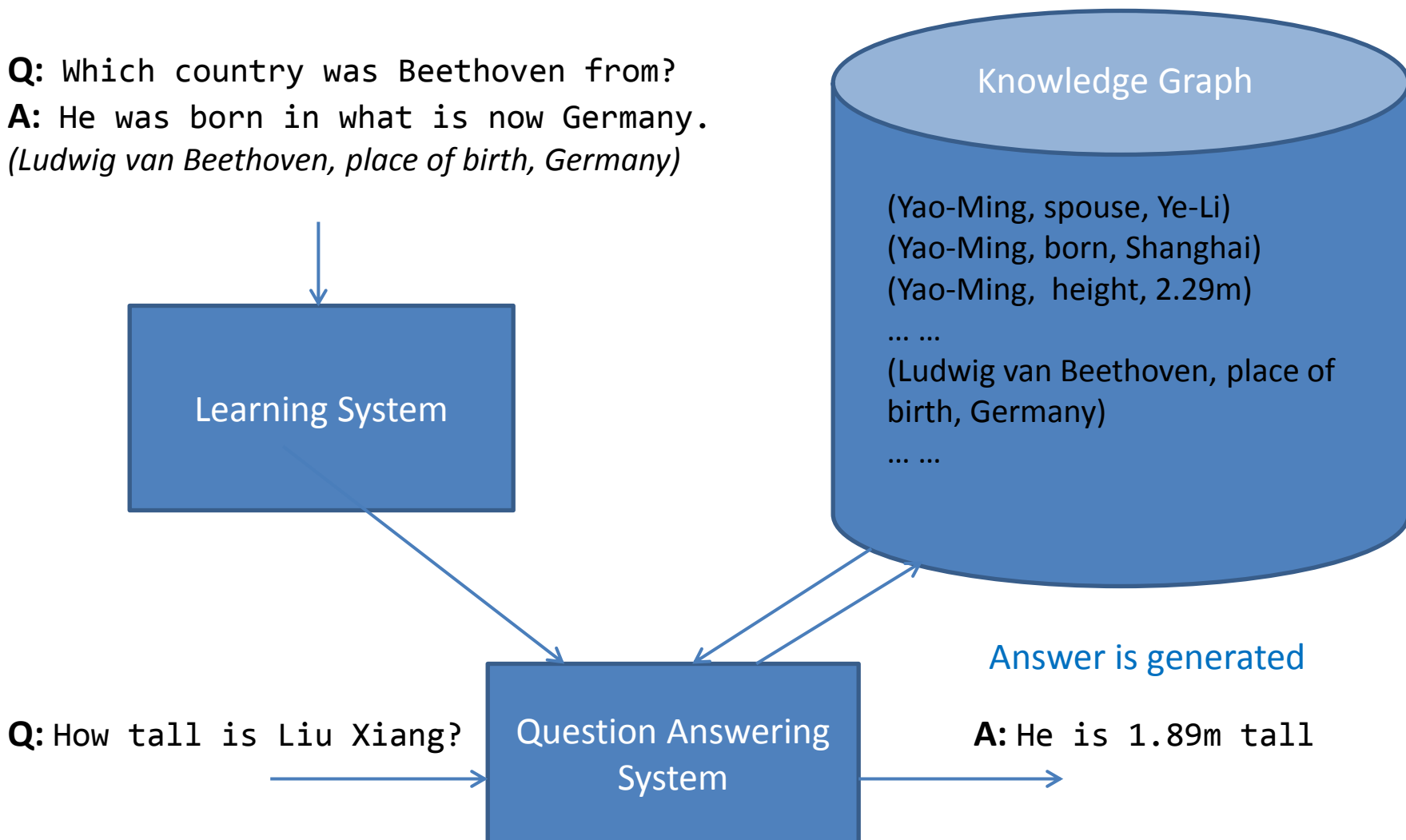
**A:** He is 2.29m tall and is visible from space.

*(Yao Ming, height, 2.29m)*

**Q:** Which country was Beethoven from?

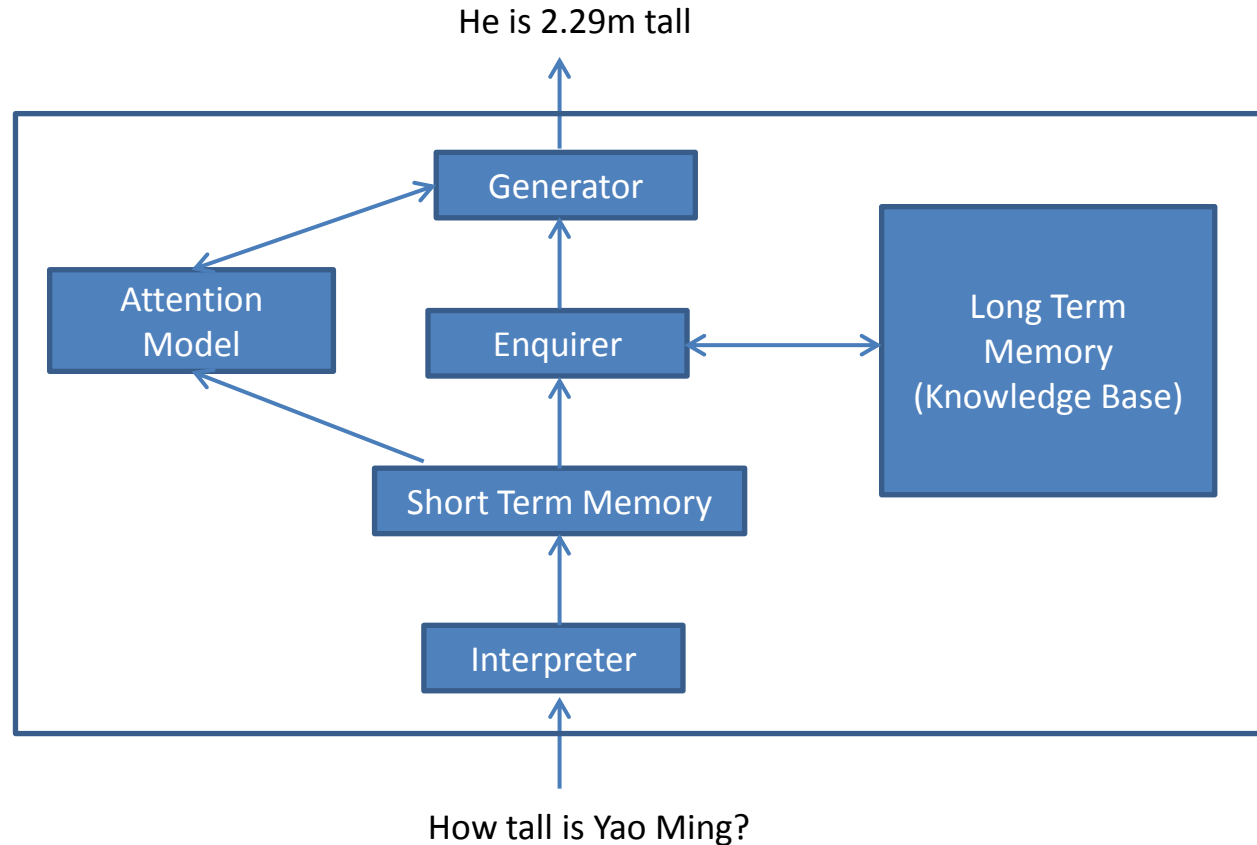
**A:** He was born in what is now Germany.

*(Ludwig van Beethoven, place of birth, Germany)*



# GenQA

- **Interpreter:** creates representation of question using RNN
- **Enquirer:** retrieves top k triples with highest matching scores using CNN model
- **Generator:** generates answer based on question and retrieved triples using attention-based RNN
- **Attention model:** controls generation of answer

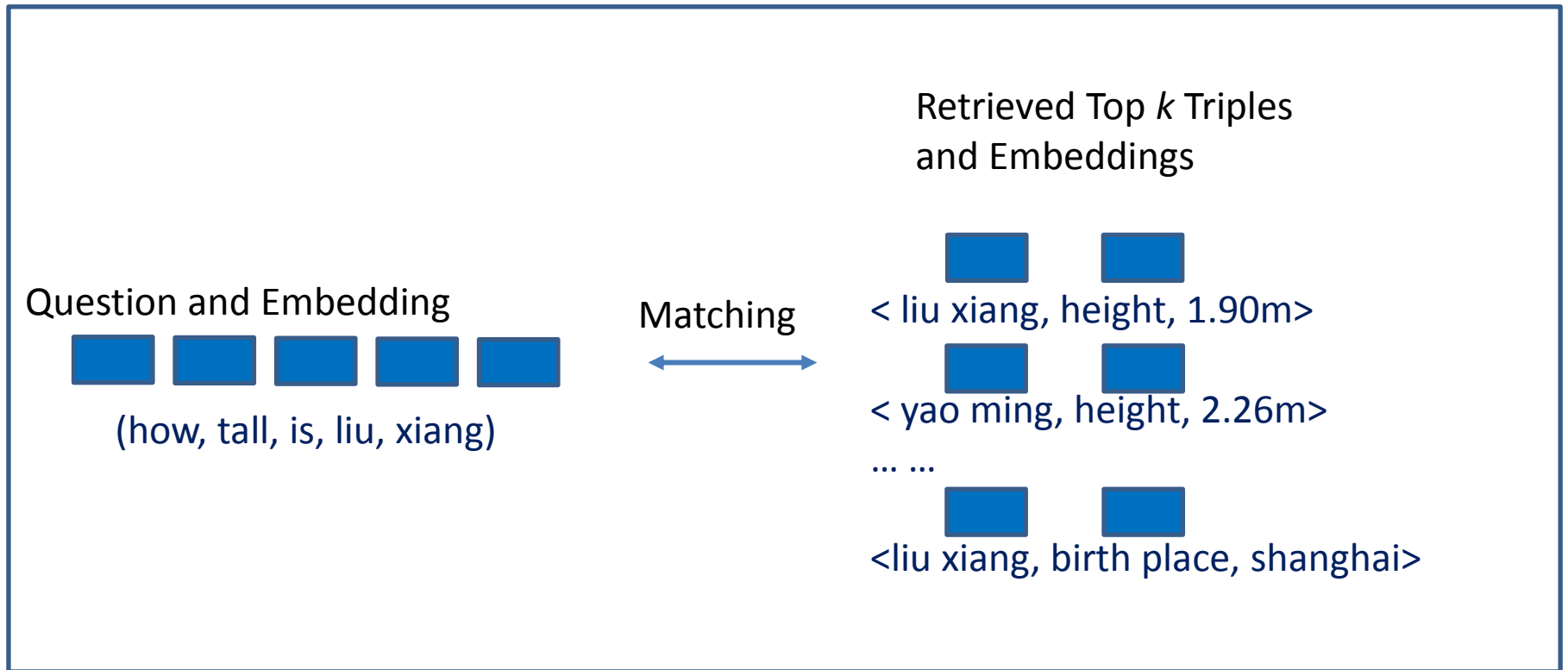


## Key idea:

- Generation of answer based on question and retrieved result
- Combination of neural processing and symbolic processing

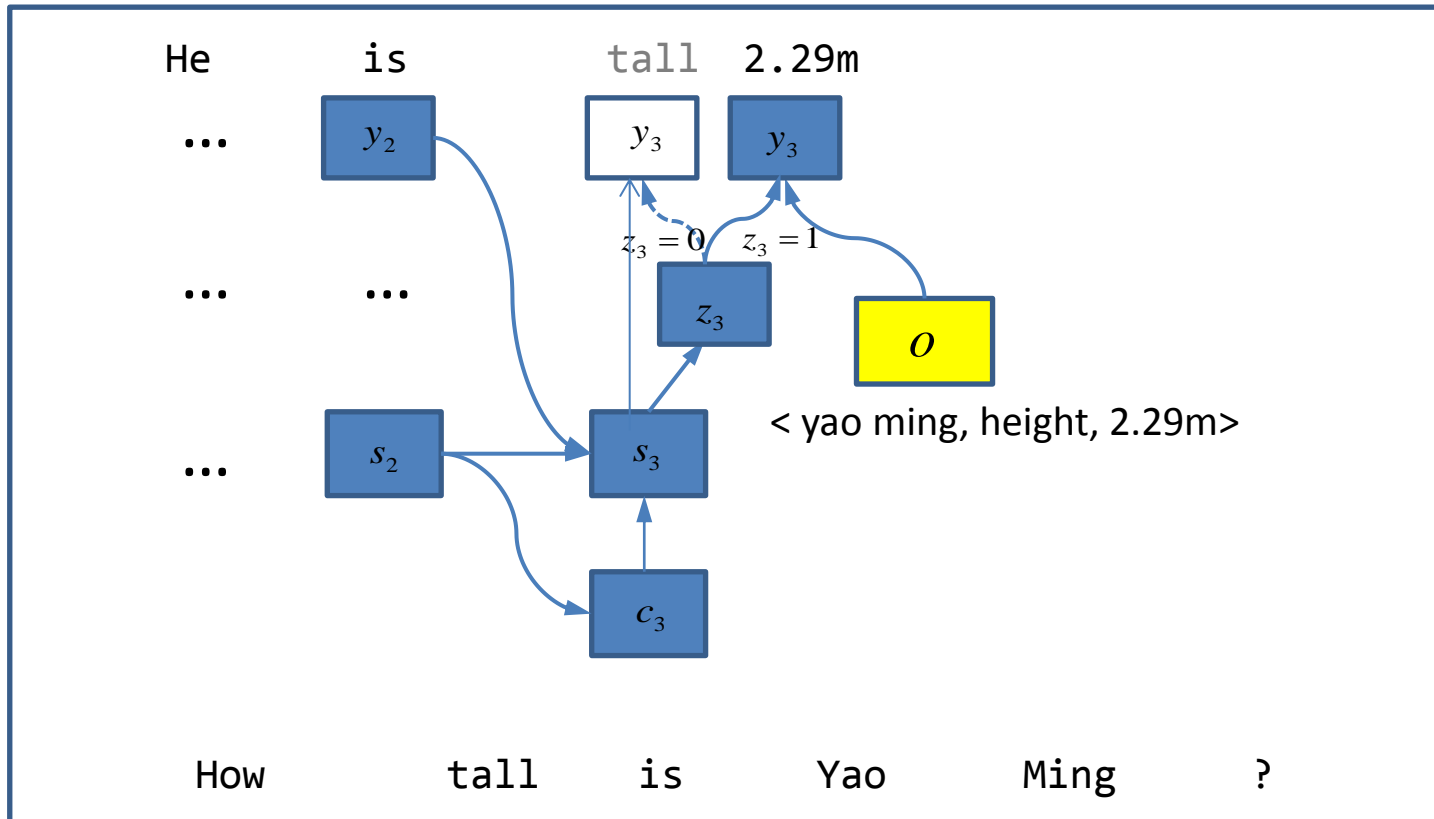


# Enquirer: Retrieval and Matching



- Retaining both symbolic representations and vector representations
- Using question words to retrieve top  $k$  triples
- Calculating matching scores between question and triples using CNN model
- Finding best matched triples

# Generator: Answer Generation



- Generating answer using attention mechanism
- At each position, a variable decides whether to generate a word or use the object of top triple

# Experimental Results

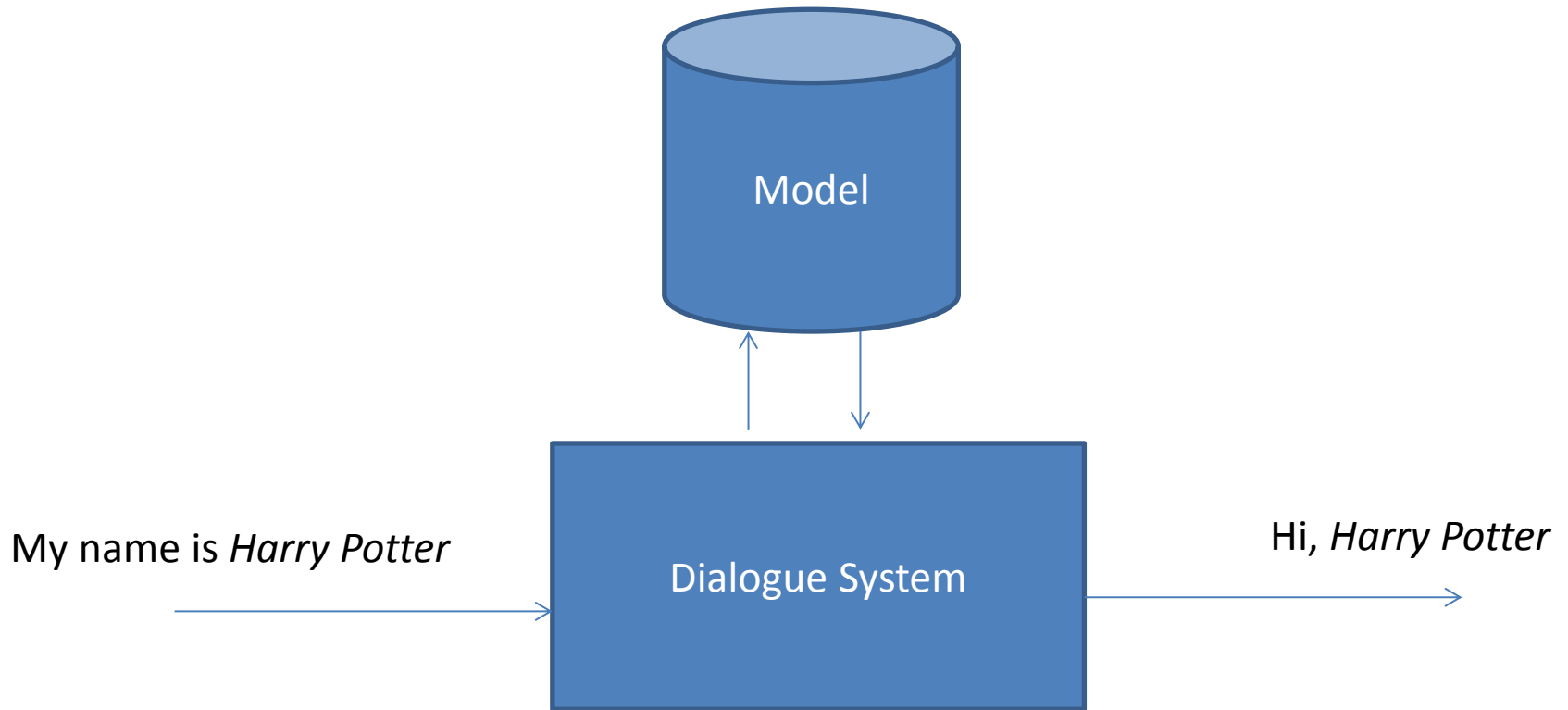
- Experiment
  - Trained with 720K question-answer pairs (Chinese) associated with 1.1M triples in knowledge-base, *data is noisy*
  - Accuracy = 52%
  - Data is still noisy

Question	Answer	
Who wrote the Romance of the Three Kingdoms?	Luo Guanzhong in Ming dynasty	correct
How old is Stefanie Sun this year?	Thirty-two, he was born on July 23, 1978	wrong
When will Shrek Forever After be released?	Release date: Dreamworks Pictures	wrong

# Natural Language Dialogue

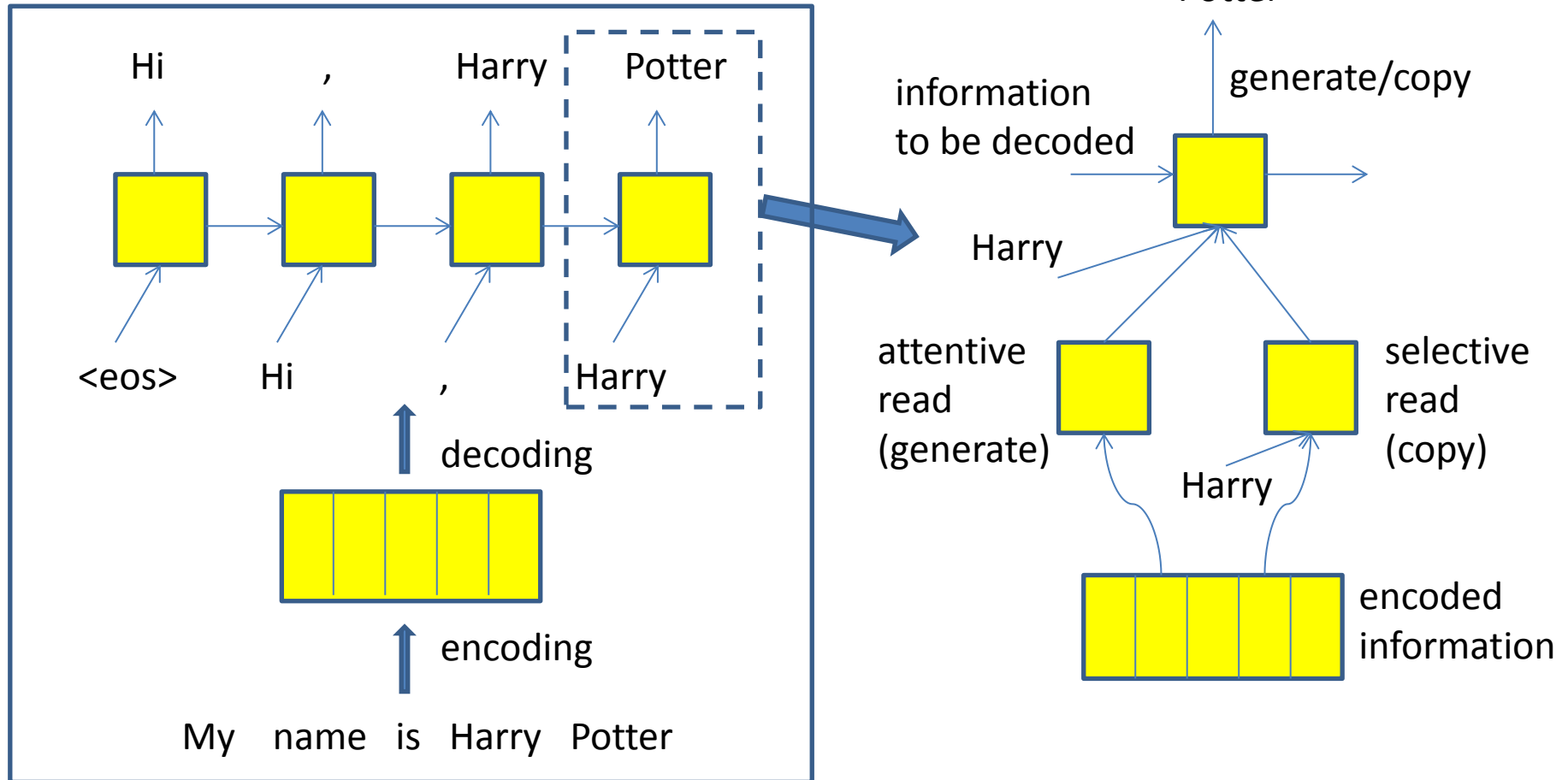
## - CopyNet

# Single Turn Dialogue with Generating and Copying Mechanism



Dialogue system can not only generate response, but also copy from given message

# Architecture: CopyNet



CopyNet can either generate word based on attentive read, or copy word based on selective read

# Characteristics: CopyNet

- Decoder can both generate and copy
- Mixture model of generating and copying
- Attentive read: find suitable word to influence generation of word in target sequence
- Selective read: find location of word to be copied from source sequence
- Model is fully differentiable
- Training: maximum likelihood of target sequence given source sequence

# Experimental Results

- Experiment
  - Summarization of short text in Chinese
  - Trained with 2.4M text-summary pairs
  - Tested with 9.3K text-summary pairs

Model	ROUGE-1	ROUGE-2	ROUGE-L
RNN -C	29.9	17.4	27.2
RNN -W	26.8	16.1	24.1
CopyNet -C	<b>34.4</b>	<b>21.6</b>	<b>31.3</b>
CopyNet -W	<b>35.0</b>	<b>22.3</b>	<b>32.0</b>



# Outline of Lecture

- Introduction
- Basics
- State of the Art
- Previous Work at Noah's Ark Lab
- Recent Progress at Noah's Ark Lab
- *Advantages and Disadvantages*
- Summary

# Advantages and Disadvantages of DL

- Strength

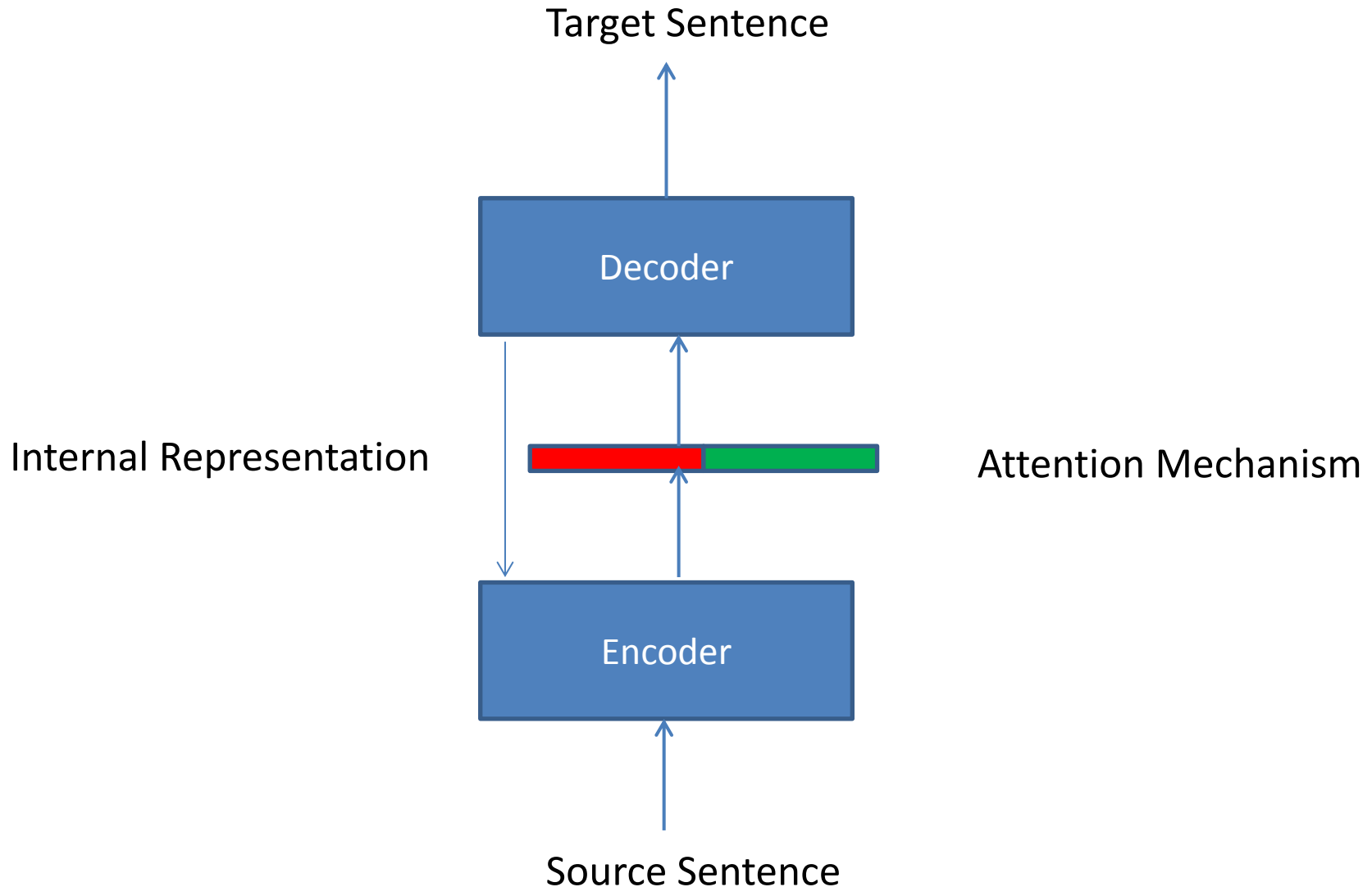
- Good at *pattern recognition* problems
- Data-driven, performance is high in many tasks
- End-to-end training, little or no domain knowledge is needed in system construction
- Representation learning, possible in cross modal processing
- Gradient-based learning, learning algorithm is simple
- Powerful for supervised learning setting

- Weakness

- Not good at *inference and decision* problems
- Data-hungry and thus is not suitable when data size is small
- Difficult to handle tail phenomena
- Model is usually a black box and is difficult to understand
- Computational cost of learning is high
- Unsupervised learning methods are needed
- Still lack of theoretical foundation

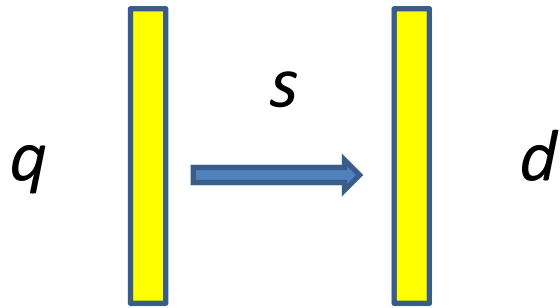
# End-to-End Learning

# Generation-based Dialogue

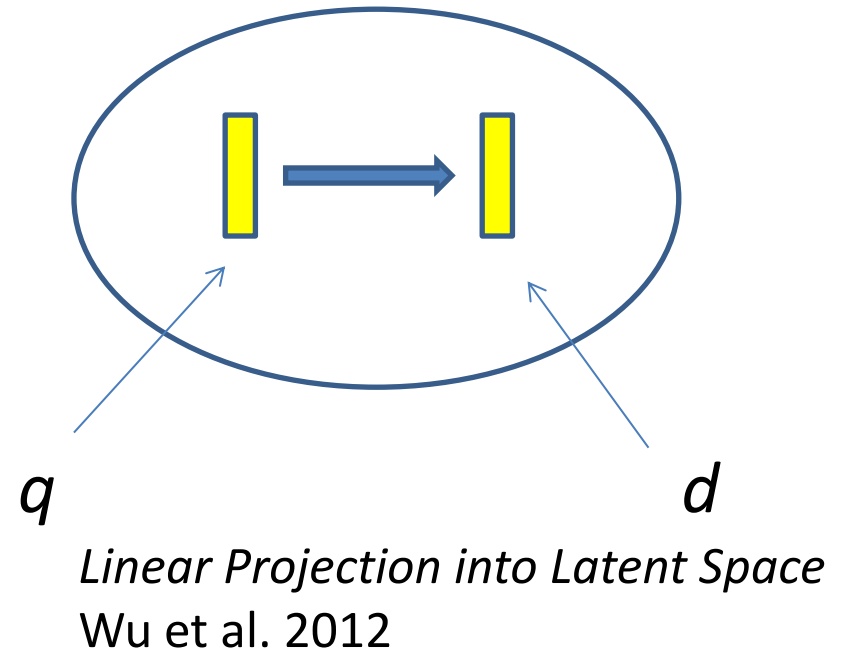


# Representation Learning

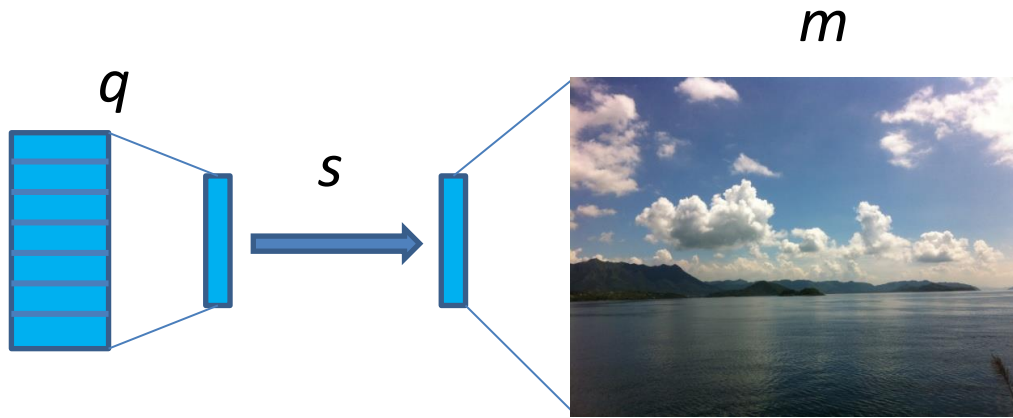
# Symbolic Matching Models



*Vector Space Model,  
BM25, Language Model for IR*



# Neural Matching Models



Neural matching models  
are natural extension of  
symbolic matching models

*Multimodal Match Model (CNN),*  
Ma et al. 2015

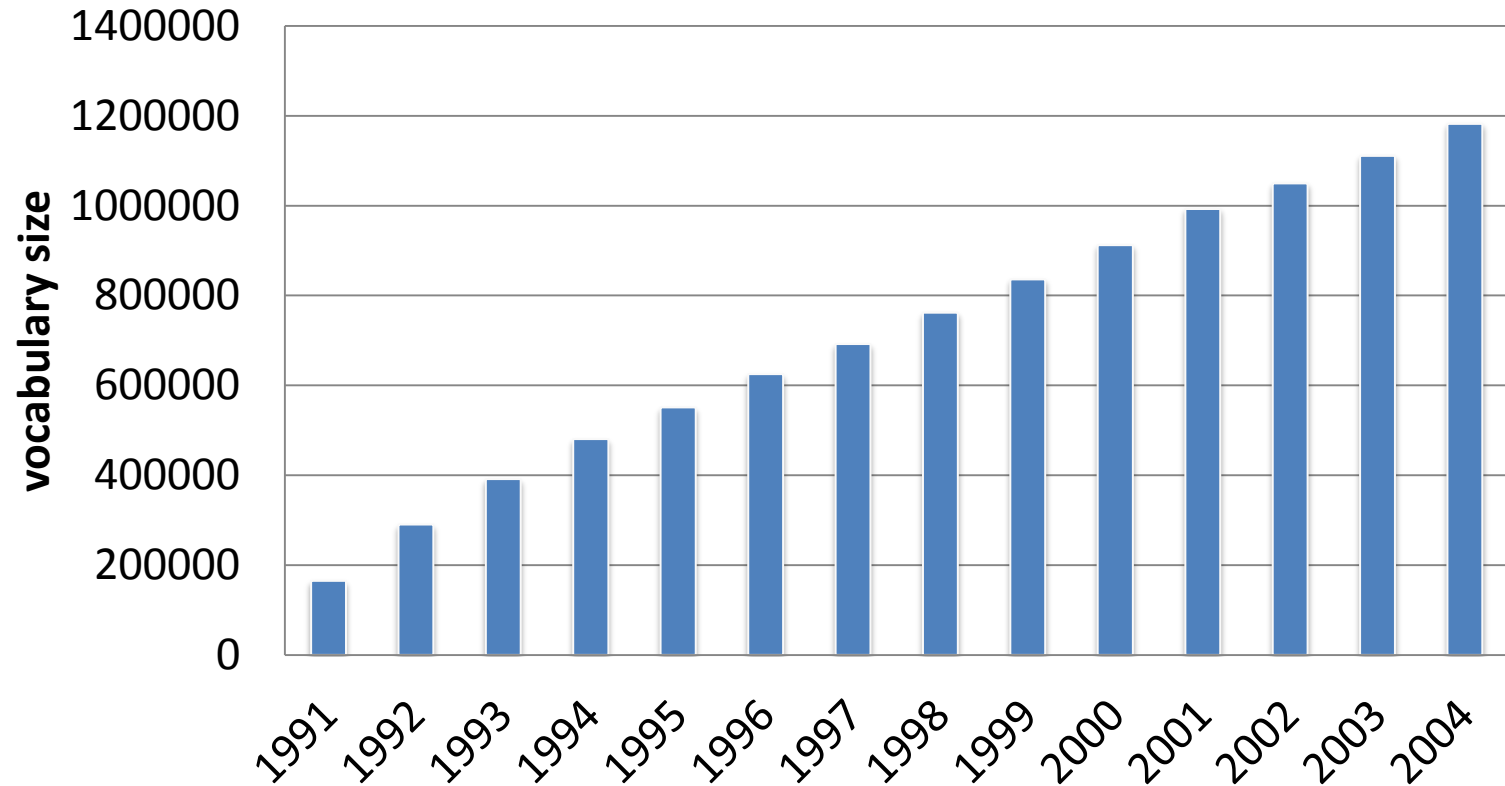
# Challenge in Tail



# Natural Language Processing Problems are Non-Parametric

How to deal with the long tail is challenging issue

## Xinhua News Data



Vocabulary size increases when data size increases

# Theoretical Analysis

# Generalization Ability of Deep Learning

- In practice, usually both training errors and test errors are small, i.e., no over-fitting
- Neural networks can “memorize” training instances
- On the other hand, neural networks can over-fit (i.e., test errors are large although training errors are small), if random noise is injected into training data
- Number of parameters is larger than number of training instances
- Many open questions about the learning ability of deep learning

# Inference and Decision

# Comparison between Single-turn QA and Multi-turn QA by Humans

## Single-turn QA

- **Q:** How tall is Yao Ming?
- **A:** He is 2.29m tall.

## Multi-turn QA

- **Q:** How tall is Yao Ming?
- **A:** He is 2.29m tall.
- **Q:** Who is taller, Yao Ming or Liu Xiang?
- **A:** He is taller, and I think that Liu Xiang is only 1.89m tall.

- Single turn QA is only related to fact retrieval and answer generation.
- Multi-turn QA needs fact retrieval and answer generation, as well as other *mental processing*. More modules in human brain are involved.

# Deep Learning and Multi-turn Dialogue

- DL may not be enough for natural language dialogue
- Key is dialogue management, including dialogue control and dialogue modeling
- Involvement of multiple “modules”, each having multiple “states”
- Recent work tries to use reinforcement learning
- There are many open questions

# Outline of Lecture

- Introduction
- Basics of DL for NLP
- State of the Art of DL for NLP
- Previous Work at Noah's Ark Lab
- Recent Progress at Noah's Ark Lab
- Advantages and Disadvantages
- *Summary*

# Summary

- Deep Learning brings high performance in fundamental language processing problems, particularly translation
- Basic models: Word Embedding, CNN, RNN, Sequence to Sequence Learning
- Deep neural network models are state of the art for question answering, image retrieval, generation based dialogue, machine translation
- Recent progress, combination of symbolic and neural processing
- Advantages and limitations



# References

1. Zhaopeng Tu, Yang Liu, Zhengdong Lu, Xiaohua Liu, and Hang Li. Context Gates for Neural Machine Translation. Transactions of the Association for Computational Linguistics (TACL 2017).
2. Xing Wang, Zhengdong Lu, Zhaopeng Tu, Hang Li, Deyi Xiong, and Min Zhang. Neural Machine Translation Advised by Statistical Machine Translation. The 31st AAAI Conference on Artificial Intelligence (AAAI 2017), 3330-3336, 2017.
3. Zhaopeng Tu, Yang Liu, Lifeng Shang, Xiaohua Liu, and Hang Li. Neural Machine Translation with Reconstruction. The 31st AAAI Conference on Artificial Intelligence (AAAI 2017), 3097-3103. 2017
4. Fandong Meng, Zhengdong Lu, Hang Li, Qun Liu. Interactive Attention for Neural Machine Translation. Proceedings of the 26th International Conference on Computational Linguistics (COLING'16), 2174-2185, 2016.
5. Mingxuan Wang, Zhengdong Lu, Hang Li, Qun Liu. Memory-enhanced Decoder for Neural Machine Translation. Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP'16), 278-286, 2016.
6. Pengcheng Yin, Zhengdong Lu, Hang Li, Ben Kao, Neural Enquirer: Learning to Query Tables in Natural Language. IEEE Data Engineering Bulletin 39(3): 63-73, 2016.
7. Jiatao Gu, Zhengdong Lu, Hang Li, Victor O. K. Li. Incorporating Copying Mechanism in Sequence-to-Sequence Learning. Proceedings of the 54th Annual Meeting of Association for Computational Linguistics (ACL'16), 2016.
8. Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu and Hang Li. Modeling Coverage for Neural Machine Translation. Proceedings of the 54th Annual Meeting of Association for Computational Linguistics (ACL'16), 2016.
9. Jun Yin, Xin Jiang, Zhengdong Lu, Lifeng Shang, Hang Li, Xiaoming Li. Neural Generative Question Answering. Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI'16), 2972-2978, 2016.

# References

10. Pengcheng Yin, Zhengdong Lu, Hang Li, Ben Kao. Neural Enquirer: Learning to Query Tables with Natural Language. Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI'16), 2308-2314, 2016.
11. Lin Ma, Zhengdong Lu, Hang Li. Learning to Answer Questions From Image Using Convolutional Neural Network. Proceedings of the Thirtieth AAAI Conference (AAAI'16), 2016.
12. Lin Ma, Zhengdong Lu, Lifeng Shang, Hang Li, Multimodal Convolutional Neural Networks for Matching Image and Sentence. Proceedings of 2015 IEEE International Conference on Computer Vision (ICCV), 2623-2631, 2015.
13. Baotian Hu, Zhaopeng Tu, Zhengdong Lu, Hang Li, Qingcai Chen. Context-Dependent Translation Selection Using Convolutional Neural Network. Proceedings of the 53th Annual Meeting of Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (ACL-IJCNLP'15), 536-541, 2015.
14. Lifeng Shang, Zhengdong Lu, Hang Li. Neural Responding Machine for Short Text Conversation. Proceedings of the 53th Annual Meeting of Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (ACL-IJCNLP'15), 1577-1586, 2015.
15. Mingxuan Wang, Zhengdong Lu, Hang Li, Wenbin Jiang, Qun Liu. GenCNN: A Convolutional Architecture for Word Sequence Prediction. Proceedings of the 53th Annual Meeting of Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (ACL-IJCNLP'15), 1567-1576, 2015.
16. Fandong Meng, Zhengdong Lu, Mingxuan Wang, Hang Li, Wenbin Jiang, Qun Liu. Encoding Source Language with Convolutional Neural Network for Machine Translation. Proceedings of the 53th Annual Meeting of Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (ACL-IJCNLP'15), 20-30, 2015.

# References

17. Mingxuan Wang, Zhengdong Lu, Hang Li, Qun Liu. Syntax-based Deep Matching of Short Texts. Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI'15), 1354-1361, 2015.
18. Baotian Hu, Zhengdong Lu, Hang Li, Qingcai Chen. Convolutional Neural Network Architectures for Matching Natural Language Sentences. Proceedings of Advances in Neural Information Processing Systems 27 (NIPS'14), 2042-2050, 2014.
19. Zhengdong Lu, Hang Li. A Deep Architecture for Matching Short Texts. Proceedings of Neural Information Processing Systems 26 (NIPS'13), 1367-1375, 2013.

Thank you!