SIGIR 2016 Tutorial
Pisa Italy
July 17, 2016

Deep Learning for Information Retrieval

Hang Li & Zhengdong Lu
Huawei Noah's Ark Lab

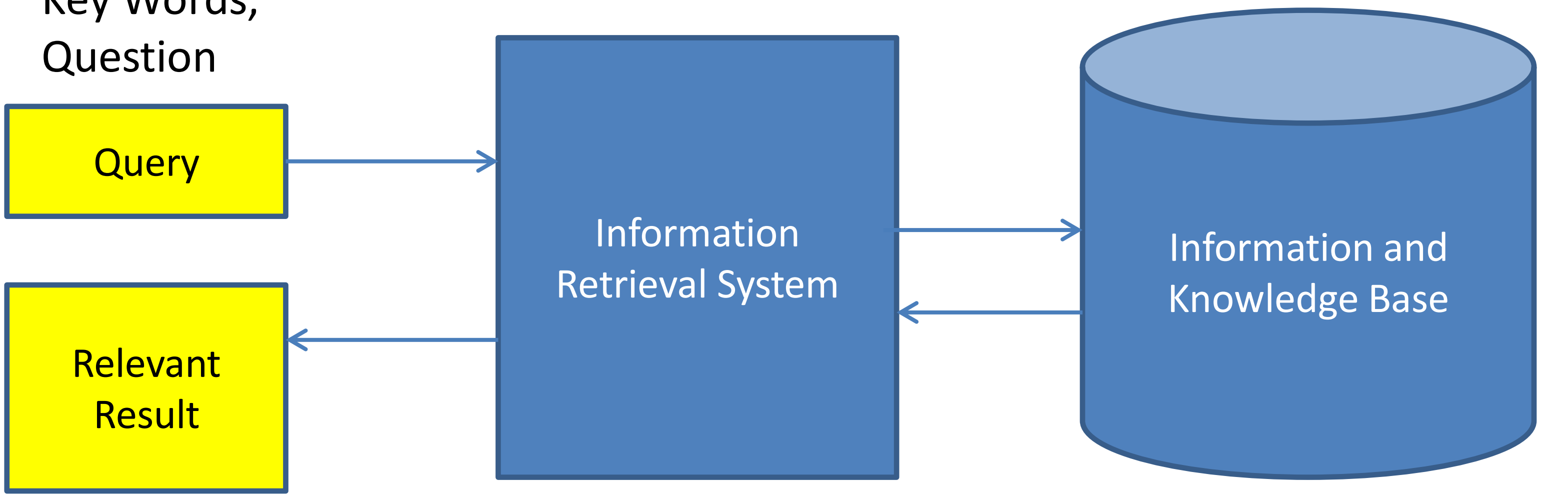
Outline of Tutorial

- Introduction
- Part 1: Basics of Deep Learning
- Part 2: Fundamental Problems in Deep Learning for IR
- Part 3: Applications of Deep Learning to IR
- Summary

Overview of Information Retrieval

Content:
Documents,
Images,
Relational Tables

Intent:
Key Words,
Question



Key Questions: How to Represent Intent and Content, How to Match Intent and Content

- Ranking, indexing, etc are less essential
- Interactive IR is not particularly considered here

Approach in Traditional IR

Document:

```
Star Wars: Episode VII
Three decades after the
defeat of the Galactic
Empire, a new threat arises.
```

Query:

```
star wars the force awakens reviews
```

$$q = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \xrightarrow{f(q,d)} d = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 1 \end{bmatrix}$$

$$f_{VSM}(q,d) = \frac{\langle q, d \rangle}{\| q \| \cdot \| d \|}$$

- Representing query and document as tf-idf vectors
- Calculating cosine similarity between them
- BM25, LM4IR, etc can be considered as non-linear variants

Approach in Modern IR

Query:
`star wars the force awakens reviews`

Document:

```
Star Wars: Episode VII
Three decades after the
defeat of the Galactic
Empire, a new threat arises.
```

`(star wars)`
`(the force awakens)`
`(reviews)`

$$\begin{array}{ccc} q & & d \\ v_{q1} & \xrightarrow{\vec{f}(q,d)} & v_{d1} \\ \vdots & & \vdots \\ v_{qm} & & v_{dn} \end{array}$$

- Conducting query and document understanding
- Representing query and document as feature vectors
- Calculating multiple matching scores between query and document
- Training ranker with matching scores as features using *learning to rank*

“Easy” Problems in IR

- Search
  - Matching between query and document
- Question Answering from Documents
  - Matching between question and answer

- Well studied so far
- Deep Learning may not help so much

“Hard” Problems in IR

- Image Retrieval
  - Matching between text and image
  - Not the same as traditional setting
- Question Answering from Knowledge Base
  - Complicated matching between question and fact in knowledge base
- Generation-based Question Answering
  - Generating answer to question based on facts in knowledge base

- Not well studied so far
- Deep Learning can make a big deal

Hard Problems in IR

Q: `How tall is Yao Ming?`

Question Answering from Knowledge Base

Name	Height	Weight
Yao Ming	2.29m	134kg
Liu Xiang	1.89m	85kg

Q: `A dog catching a ball`

Image Retrieval

(No tag on images)

Q:`How far is sun from earth?`

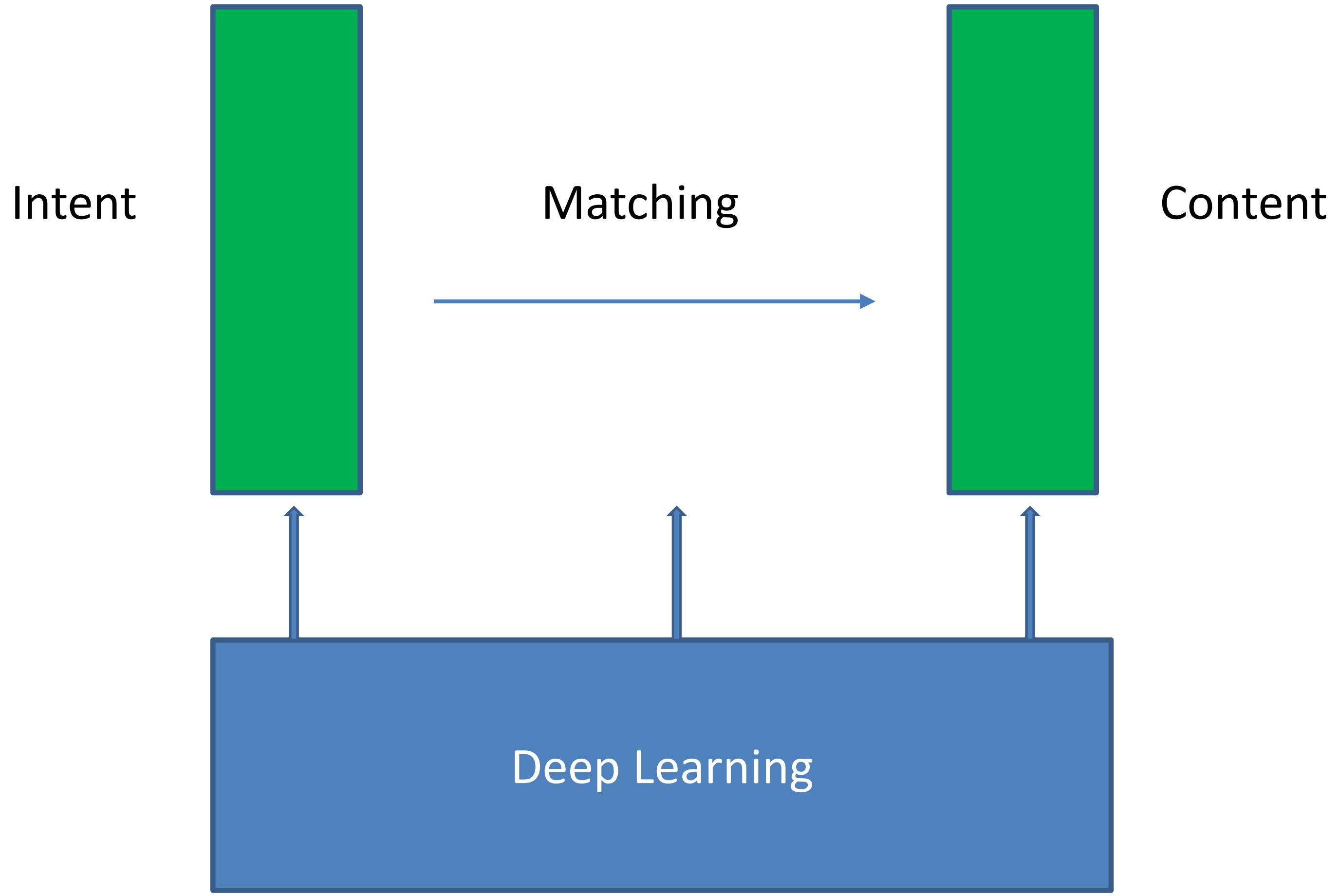
Generation-based Question Answering

The average **distance between the Sun and the Earth** is about 92,935,700 miles.

A: It is about 93 million miles

Key Questions: How to Represent Intent and Content, How to Match Intent and Content

Deep Learning and IR



Recent Progress: Deep Learning Is Particularly Effective for Hard IR Problems

Part 1: Basics of Deep Learning

Outline of Part 1

- Word Embedding
- Recurrent Neural Networks
- Convolutional Neural Networks

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**		**-.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

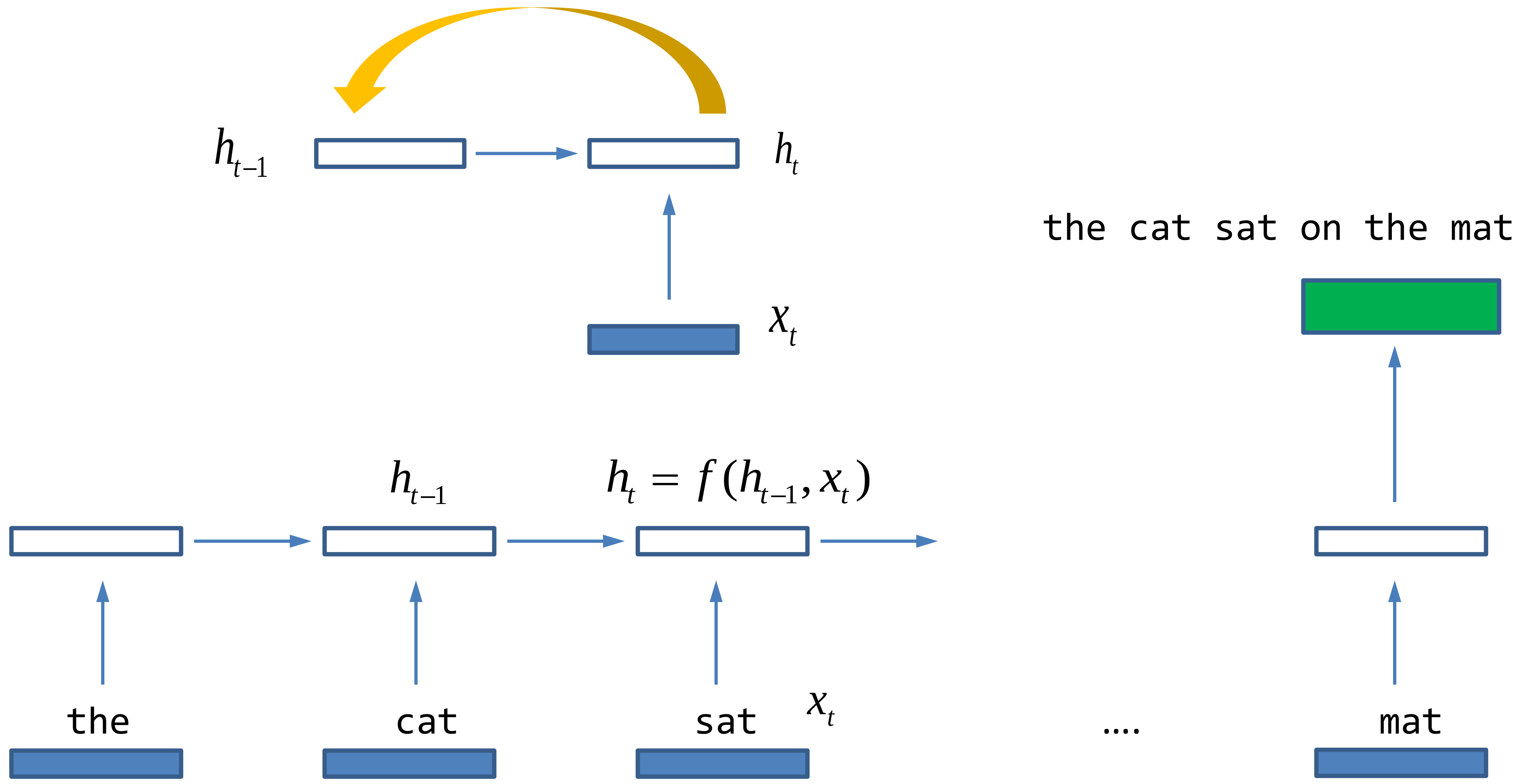
Word embedding

Recurrent Neural Network

Recurrent Neural Network

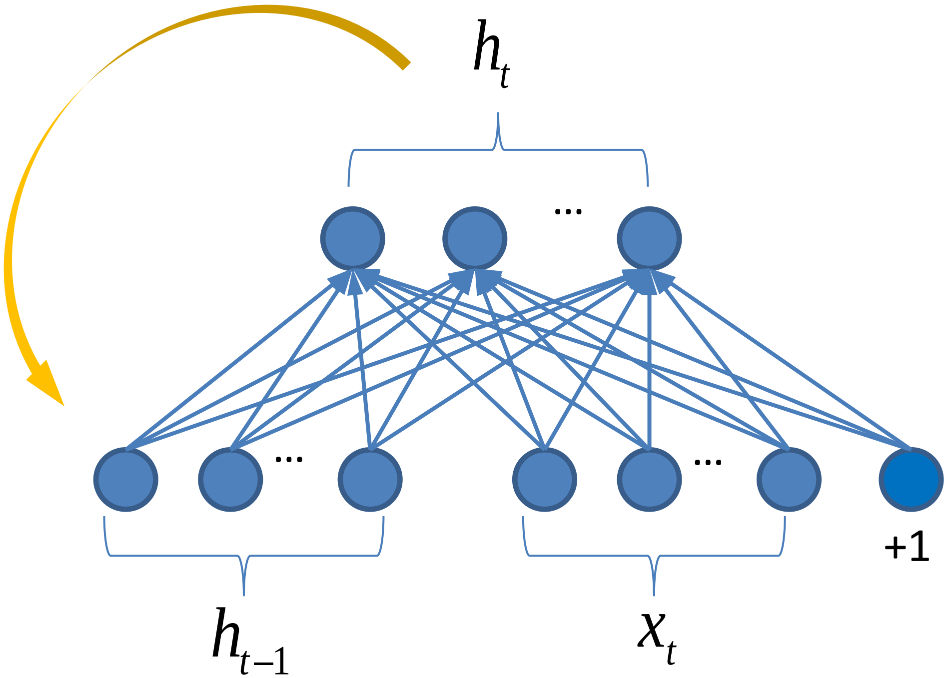
- Motivation: representing sequence of words and utilizing the representation in deep learning methods
- Input: sequence of word embeddings, denoting sequence of words (e.g., sentence)
- Output: sequence of internal representations (hidden states)
- Variants: LSTM and GRU, to deal with long distance 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)

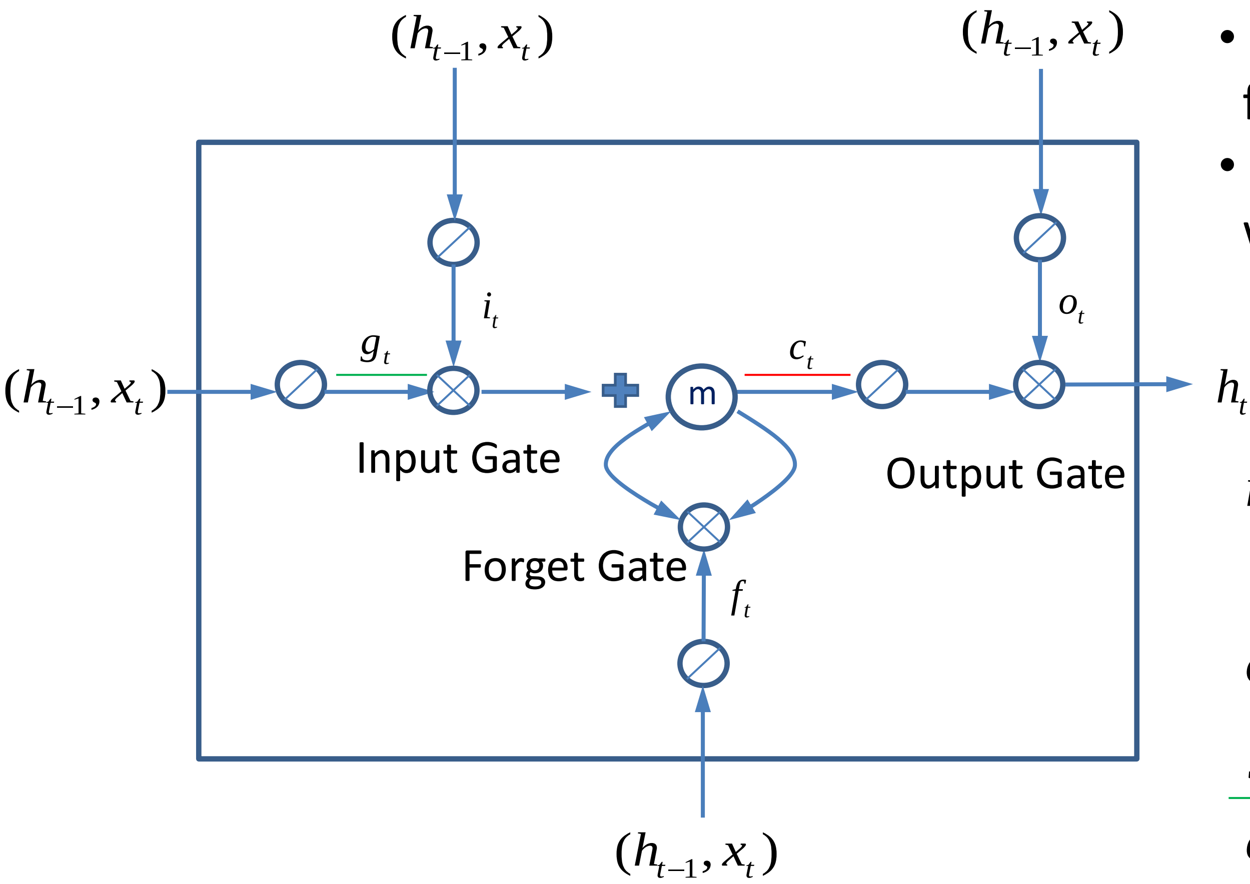


Recurrent Neural Network

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



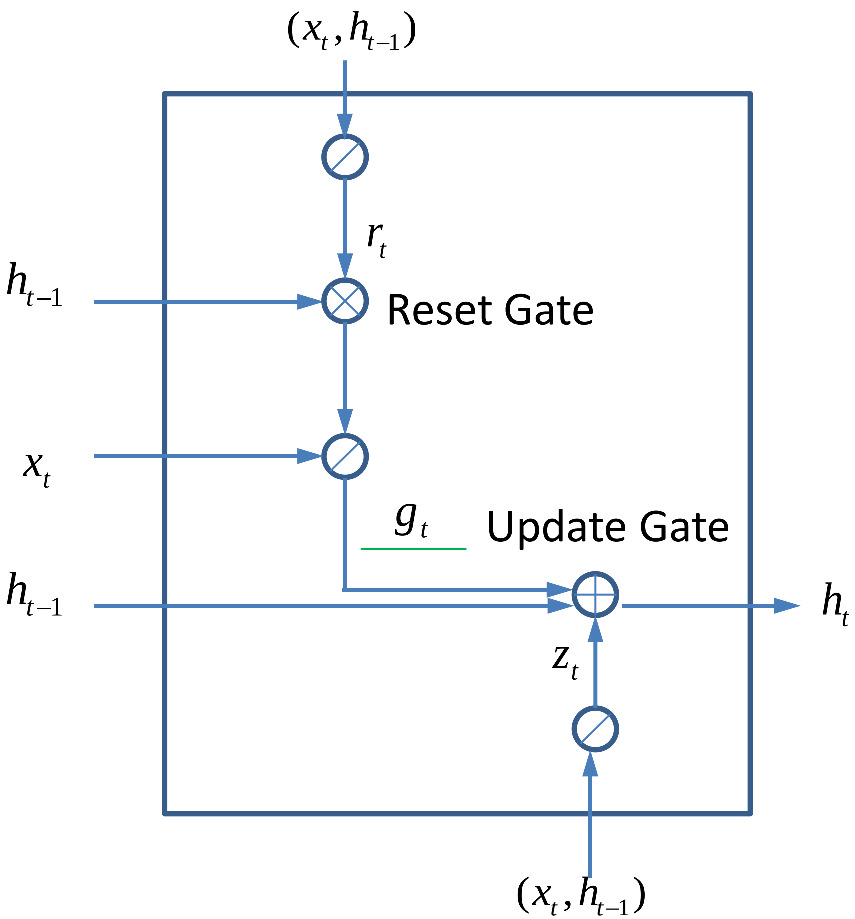
Long Term Short Memory (LSTM)
(Hochreiter & Schmidhuber, 1997)



- 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

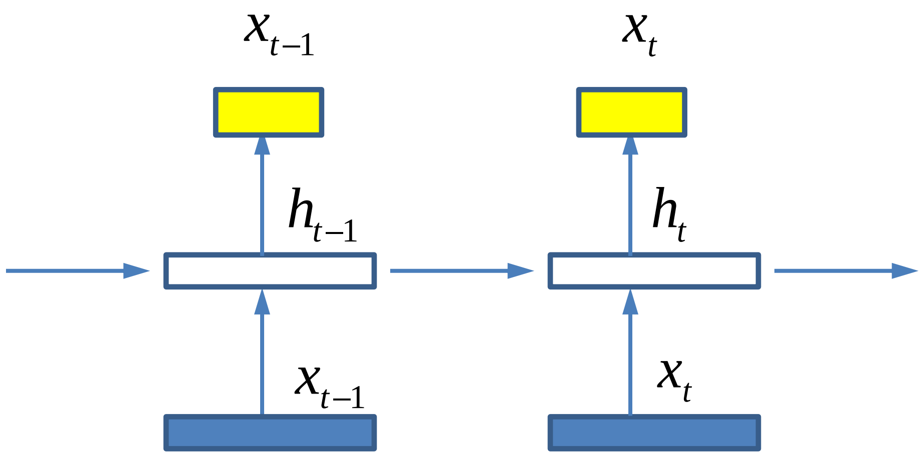
Model

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

$$p_t = P(x_t \mid x_1 \cdots x_{t-1}) = \mathrm{softmax}(W h_t + b)$$

Objective of Learning

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



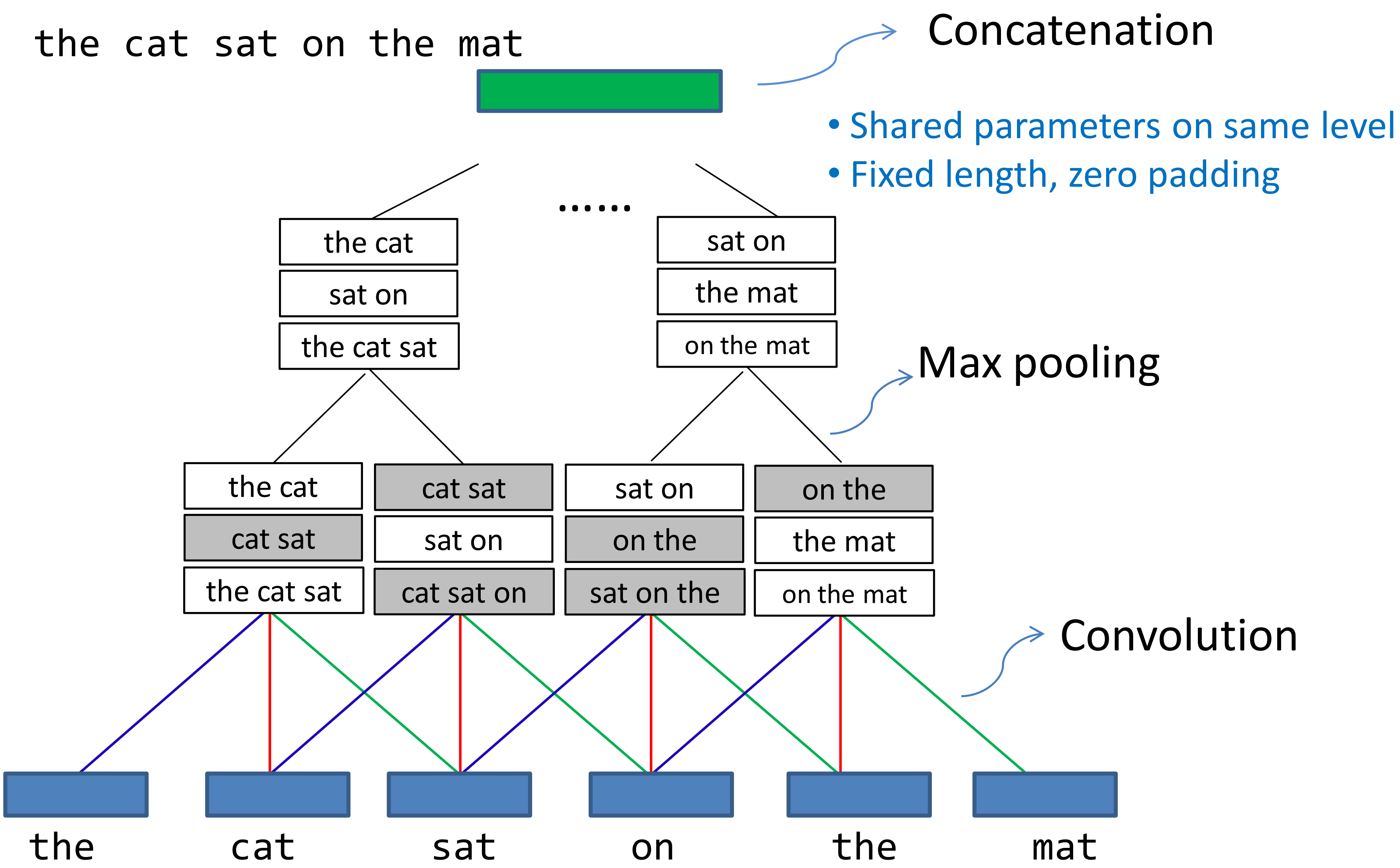
- Input one sequence and output another
- In training, input sequence is same as output sequence

Convolutional Neural Network

Convolutional Neural Network

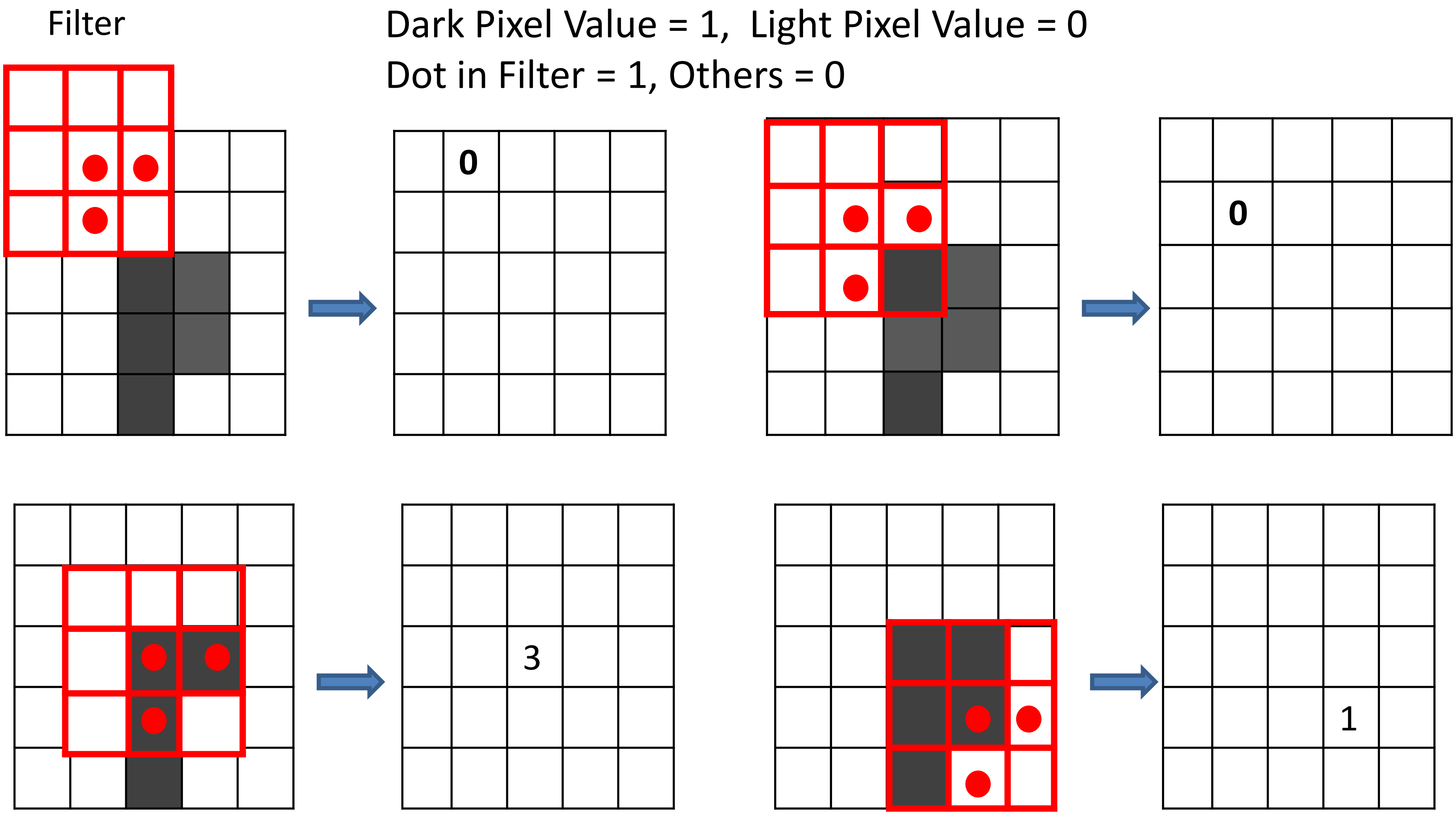
- Motivation: representing sequence of words and utilizing the representation in deep learning methods
- Input: sequence of word embeddings, denoting sequence of words (e.g., sentence)
- Output: representation of input sequence
- Learning of model: stochastic gradient descent
- Advantage: robust extraction 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

Dark Pixel Value = 1, Light Pixel Value = 0
Dot in Filter = 1, Others = 0

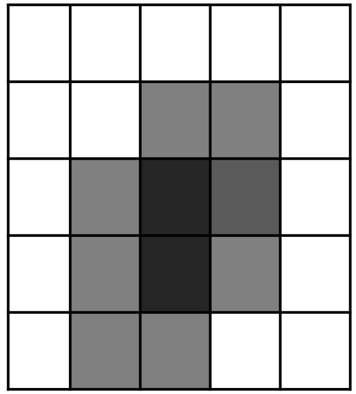


Leow Wee Kheng

Example: Image Convolution

0	**0**	**0**	**0**	**0**
0	**0**	**1**	**1**	**0**
0	**1**	**3**	**2**	**0**
0	**1**	**3**	**1**	**0**
0	**1**	**1**	**0**	**0**

Feature Map



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 filer pattern and pixel pattern at one position
- Equivalent to extracting feature using the filter
- Translation-invariant

Convolution

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

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

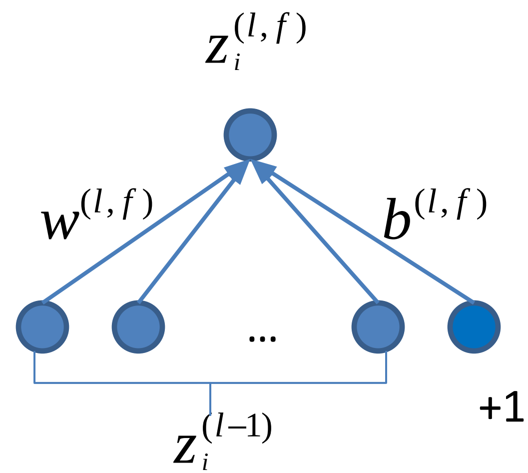
$w^{(l,f)}, b^{(l,f)}$ are parameters of neuron of type f in layer l

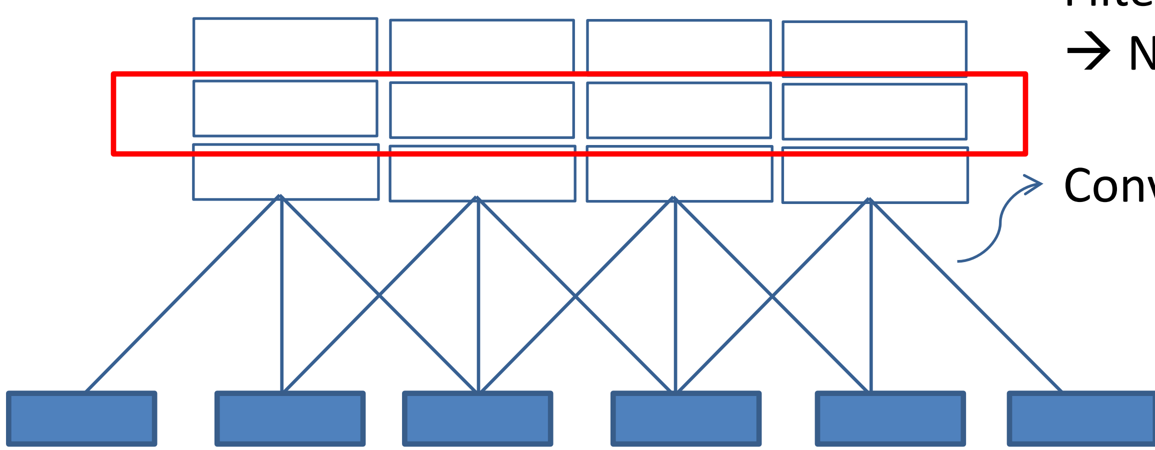
σ is sigmoid function

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

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

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





Filter → Feature Map → Neuron

Convolution

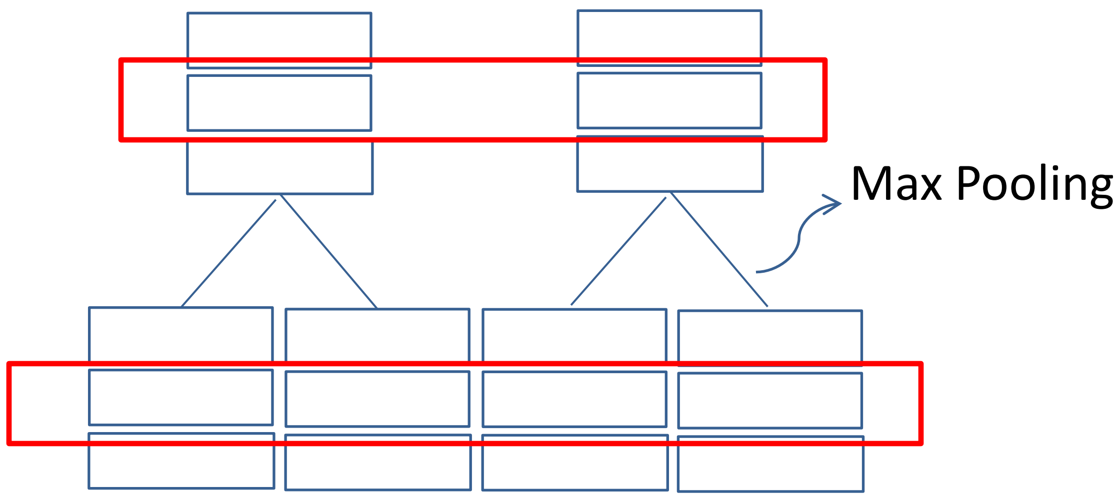
Equivalent to n-gram feature extraction at each position

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

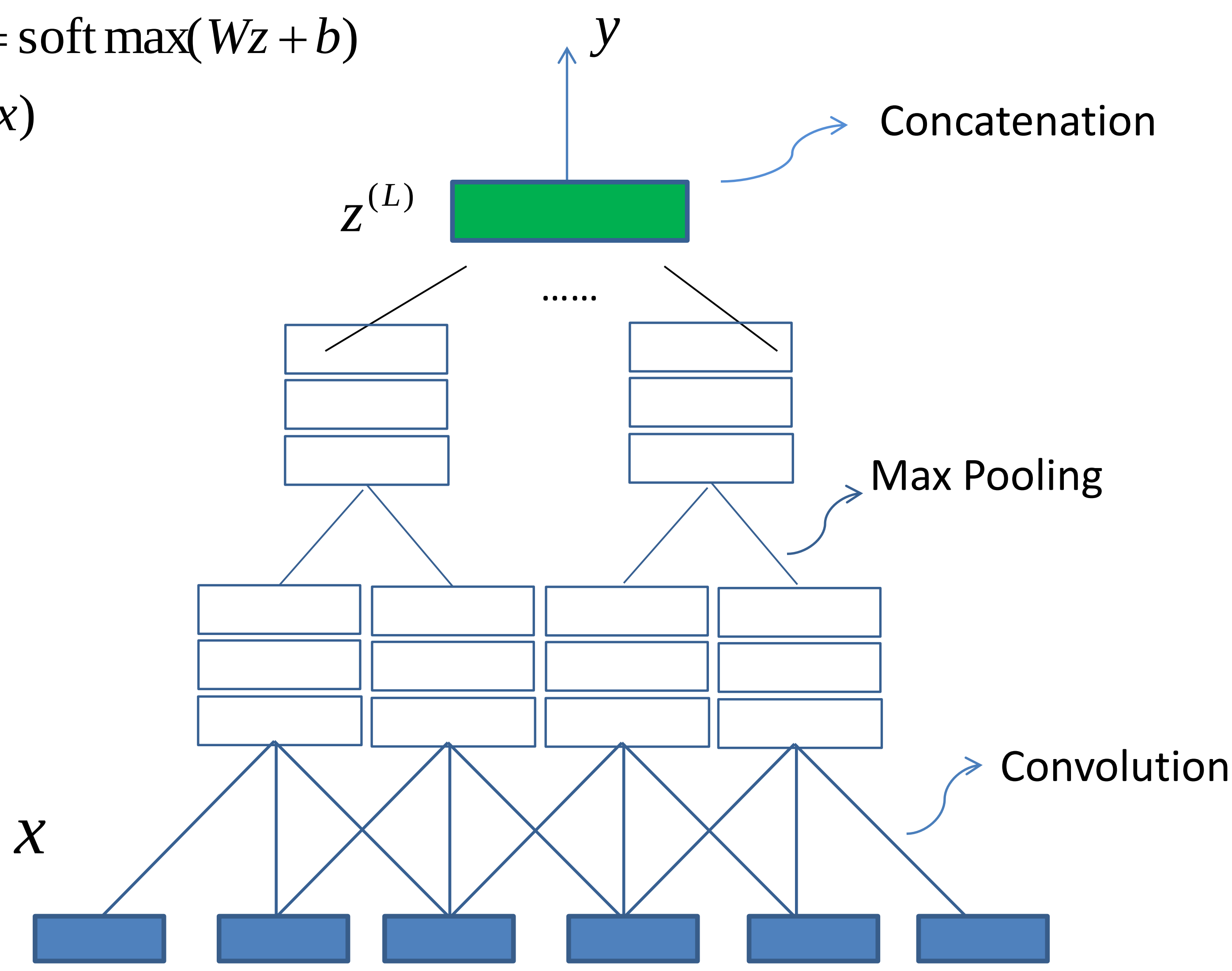


Equivalent to n-gram feature selection

Sentence Classification Using Convolutional Neural Network

$$y = f(x) = \mathrm{softmax}(Wz + b)$$

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



References

- T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed Representations of Words and Phrases and Their Compositionality. *NIPS* 2013.
- T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient Estimation of Word Representations in Vector Space. *arXiv:*1301.3781, 2013.
- O. Levy, Y. Goldberg, and I. Dagan. Improving Distributional Similarity with Lessons Learned from Word Embeddings. *TACL* 2015.
- T. Mikolov, M. Karafiát, L. Burget, J. Cernocký, and S. Khudanpur. Recurrent Neural Network based Language Model. *InterSpeech* 2010.
- S. Hochreiter and J. Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8), 1997.
- K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. *EMNLP*, 2014.
- Y. Kim. Convolutional Neural networks for Sentence Classification. *EMNLP* 2014.
- B. Hu, Z. Lu, H. Li, and Q. Chen. Convolutional Neural Network Architectures for Matching Natural Language Sentences. *NIPS* 2014.
- P. Blunsom, E. Grefenstette, and N. Kalchbrenner. A Convolutional neural network for modeling sentences. *ACL* 2014.
- R. Socher, J. Bauer, C. D. Manning, and Andrew Y. Ng. Parsing with Compositional Vector Grammars. *ACL* 2013.
- K. Tai, R. Socher, and C. D. Manning. Improved Semantic Representations from Tree-structured Long Short-term Memory Networks. *arXiv:*1503.00075, 2015.
- H. Zhao, Z. Lu, and P. Poupart. Self-Adaptive Hierarchical Sentence Model. *IJCAI* 2015.

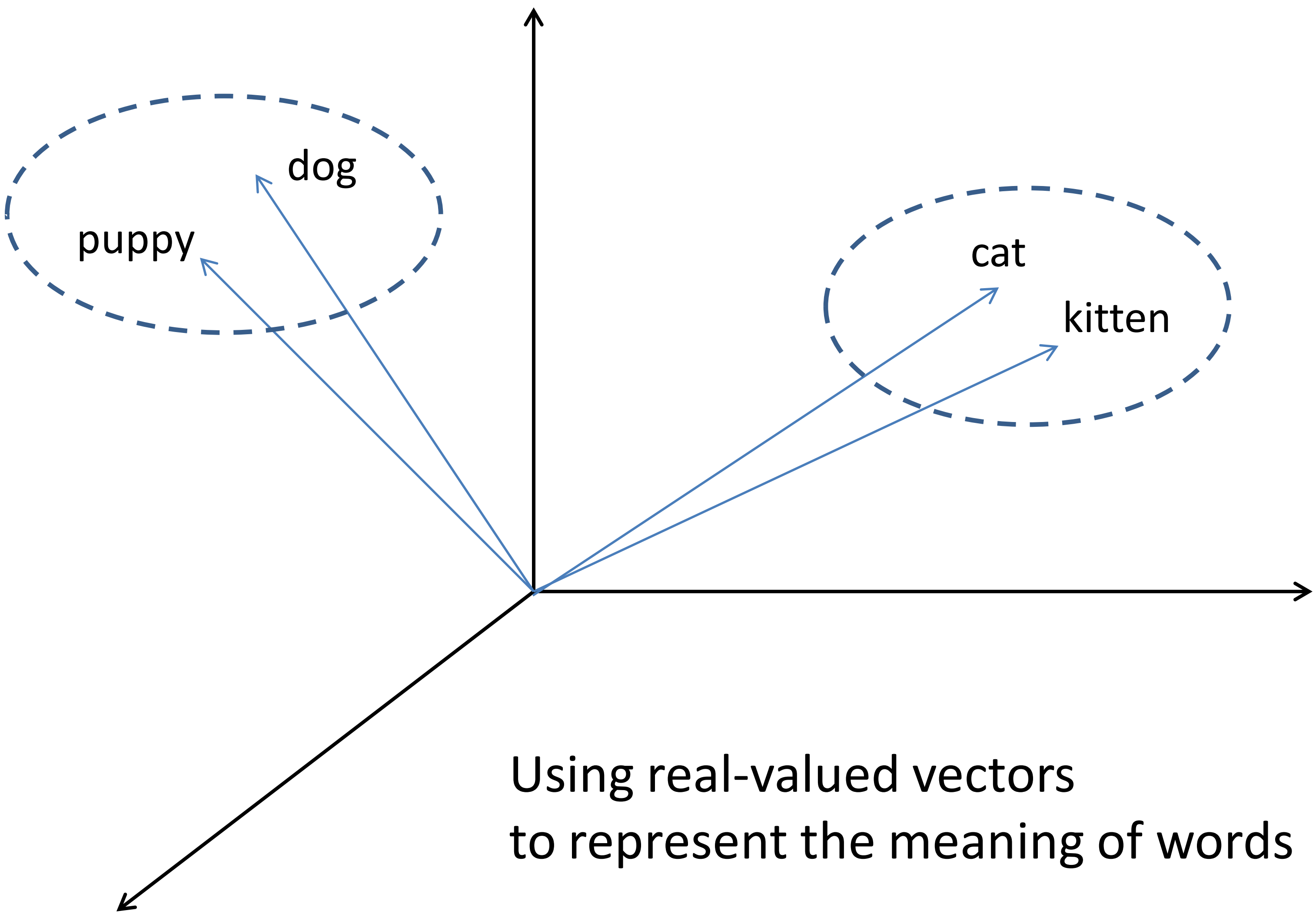
Part 2: Fundamental Problems in Deep Learning for IR

Outline of Part 2

- Representation Learning
- Matching
- Translation
- Classification
- Structured Prediction

Representation Learning

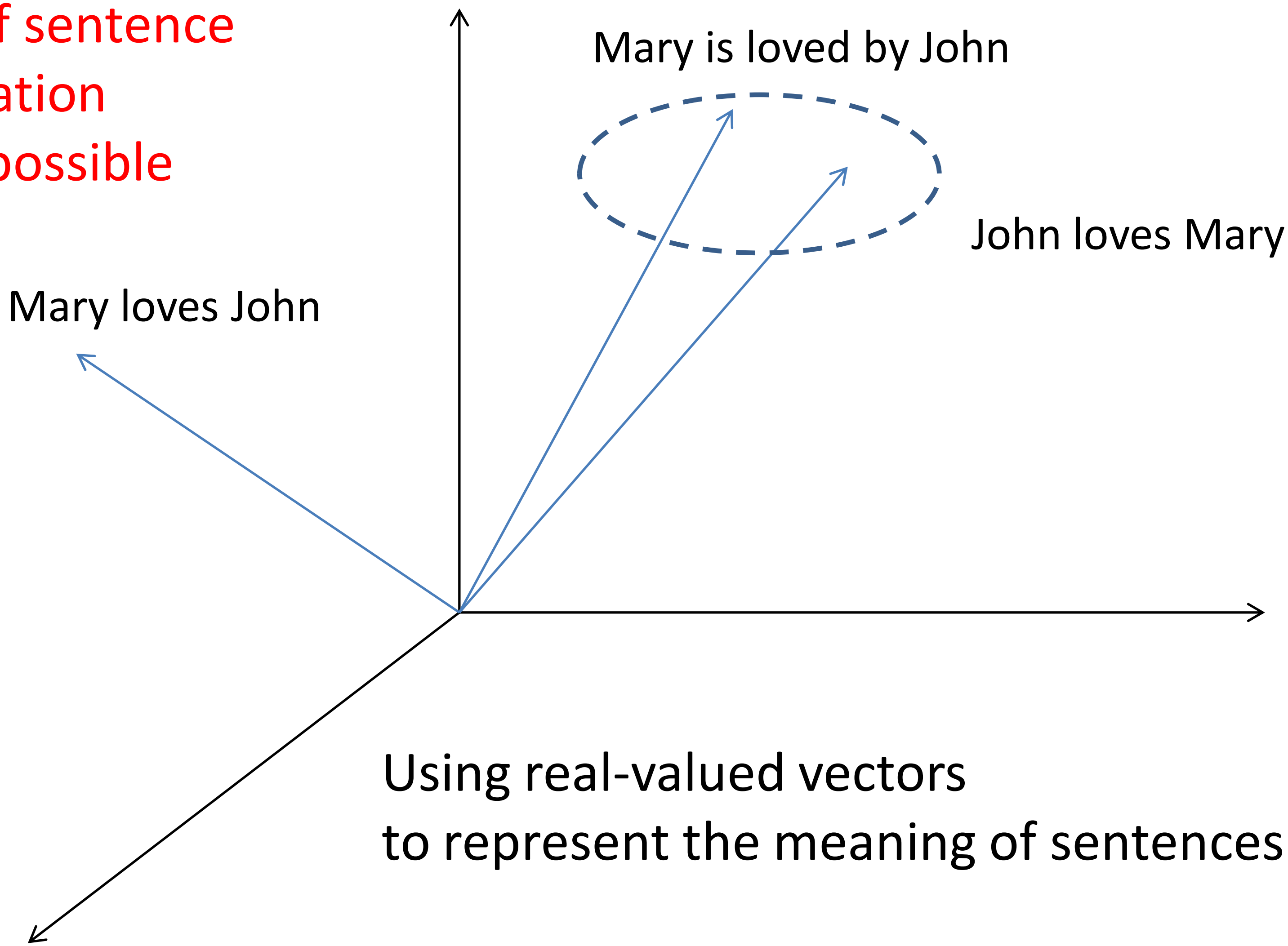
Representation of Word



Using real-valued vectors
to represent the meaning of words

Representation of Sentence

Breakthrough：
learning of sentence
representation
becomes possible



Using real-valued vectors
to represent the meaning of sentences

Learning of Sentence Representation

Task

- Compositional: from words to sentences
- Representing syntax, semantics, and even pragmatics of sentences

Means

- Deep neural networks
- Big data
- Task-dependent
- Error-driven and usually gradient-based training

Fundamental Problems in Information Retrieval (and also 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

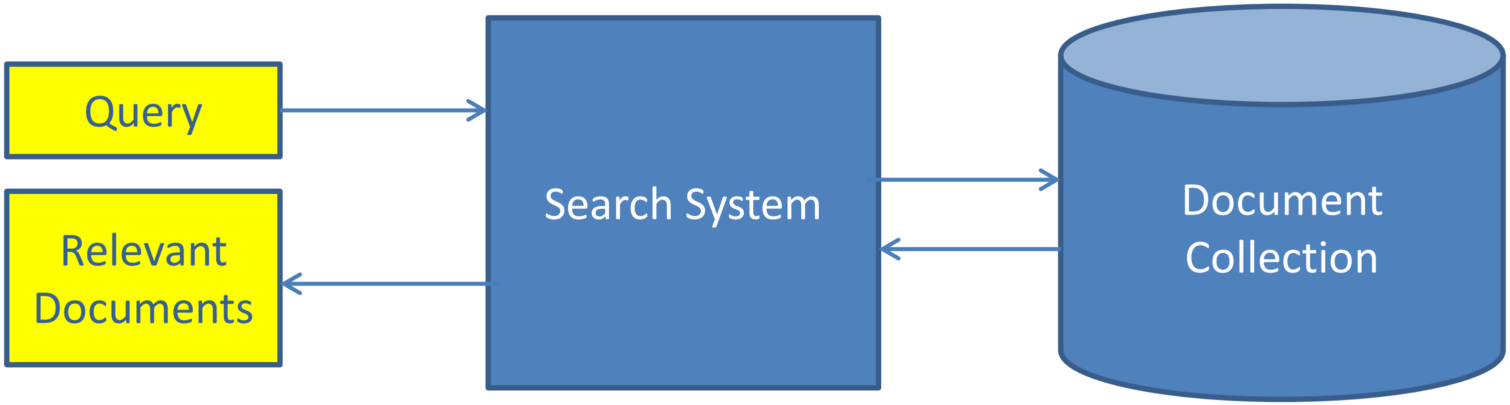
$$s \rightarrow t$$

- Structured prediction: mapping string to structure

$$s \rightarrow s'$$

- In general, *s* and *t* can be any type of data
- Non-interactive setting is mainly considered

Example: Fundamental Problems in Search



- Query Understanding (Classification and Structured Prediction)
  - Query Classification
  - Named entity Recognition in Query
- Document Understanding (Classification and Structured Prediction)
  - Document Classification
  - Named Entity Recognition in Document
- Query Document Matching (Matching)
  - Matching of Query and Document
- Summary Generation (Translation)
  - Generating Summaries of Relevant Documents

Learning of Representations in Fundamental Problems

- Classification

$$s \rightarrow r \rightarrow c$$

- Matching

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

- Translation

$$s \rightarrow r \rightarrow t$$

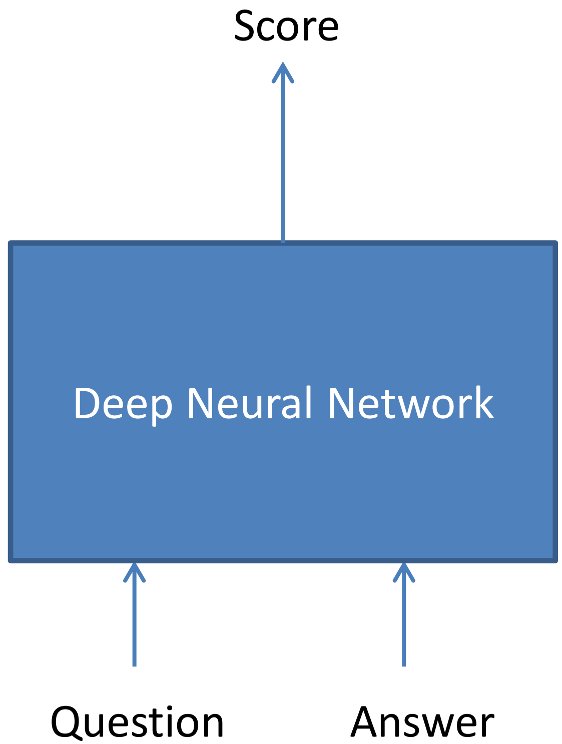
- Structured Prediction

$$s \rightarrow s' + r$$

Matching

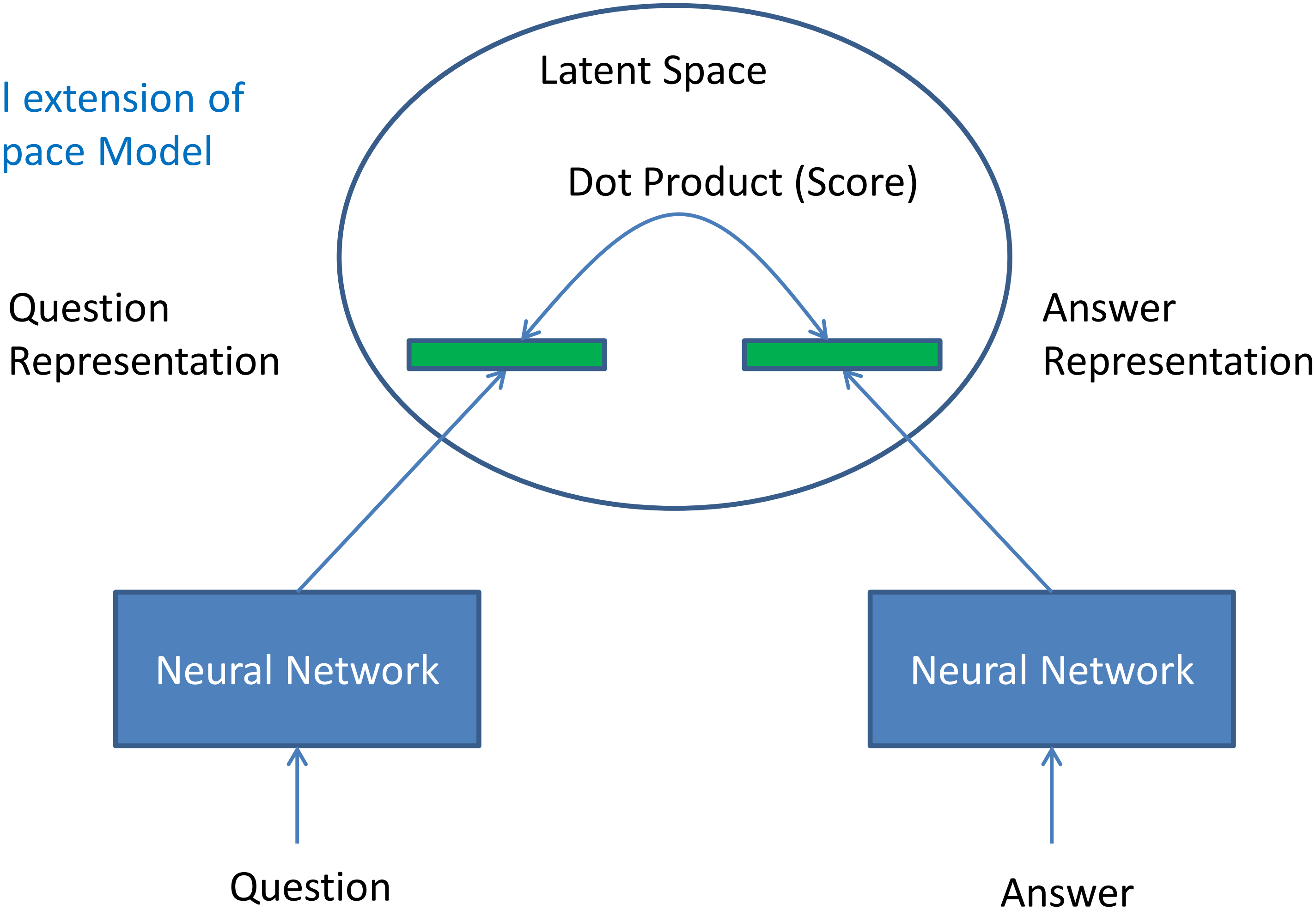
Matching

- 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

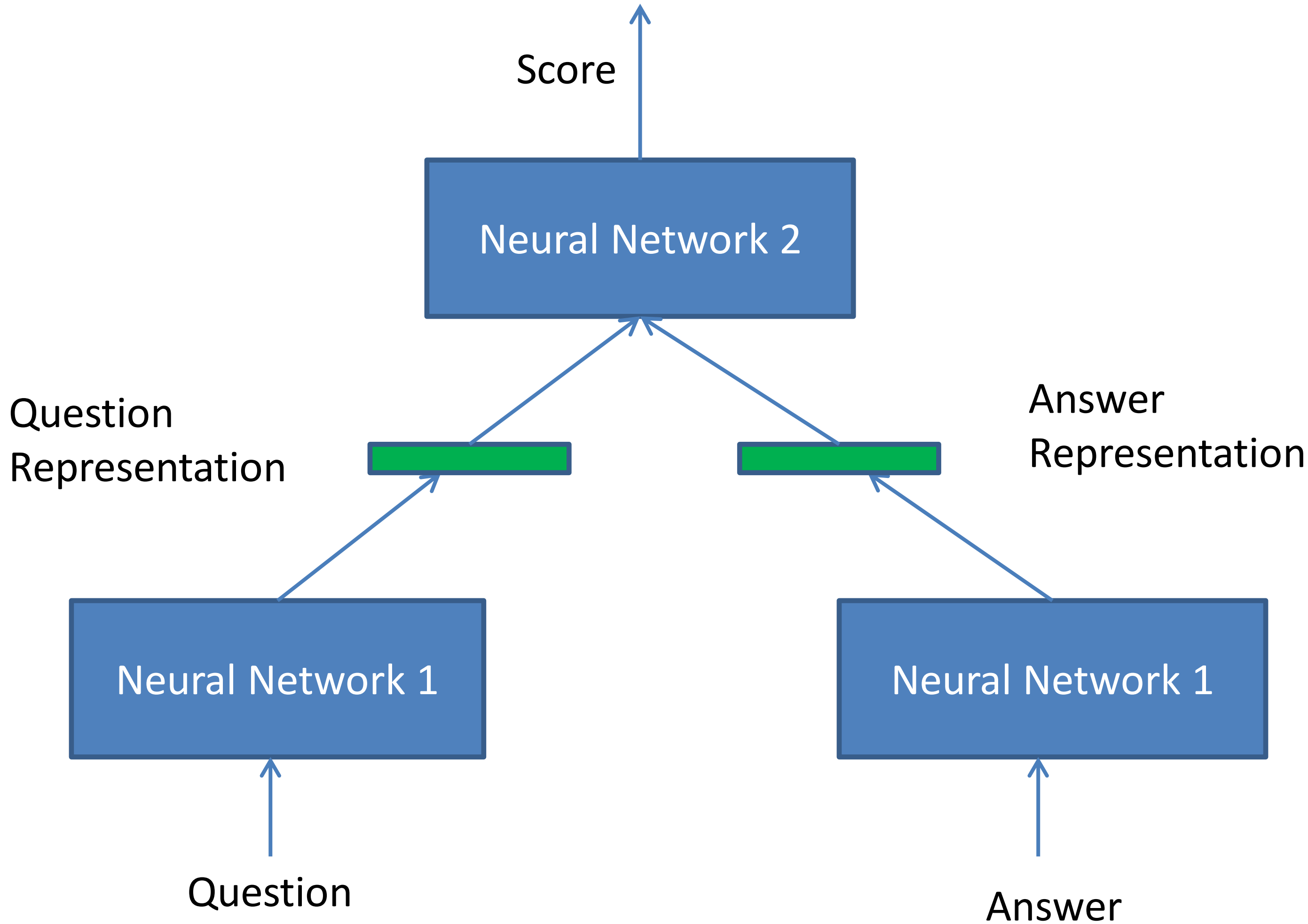


Neural Networks:

Convolutional Neural Network
Deep Neural Network
Recurrent Neural Network

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

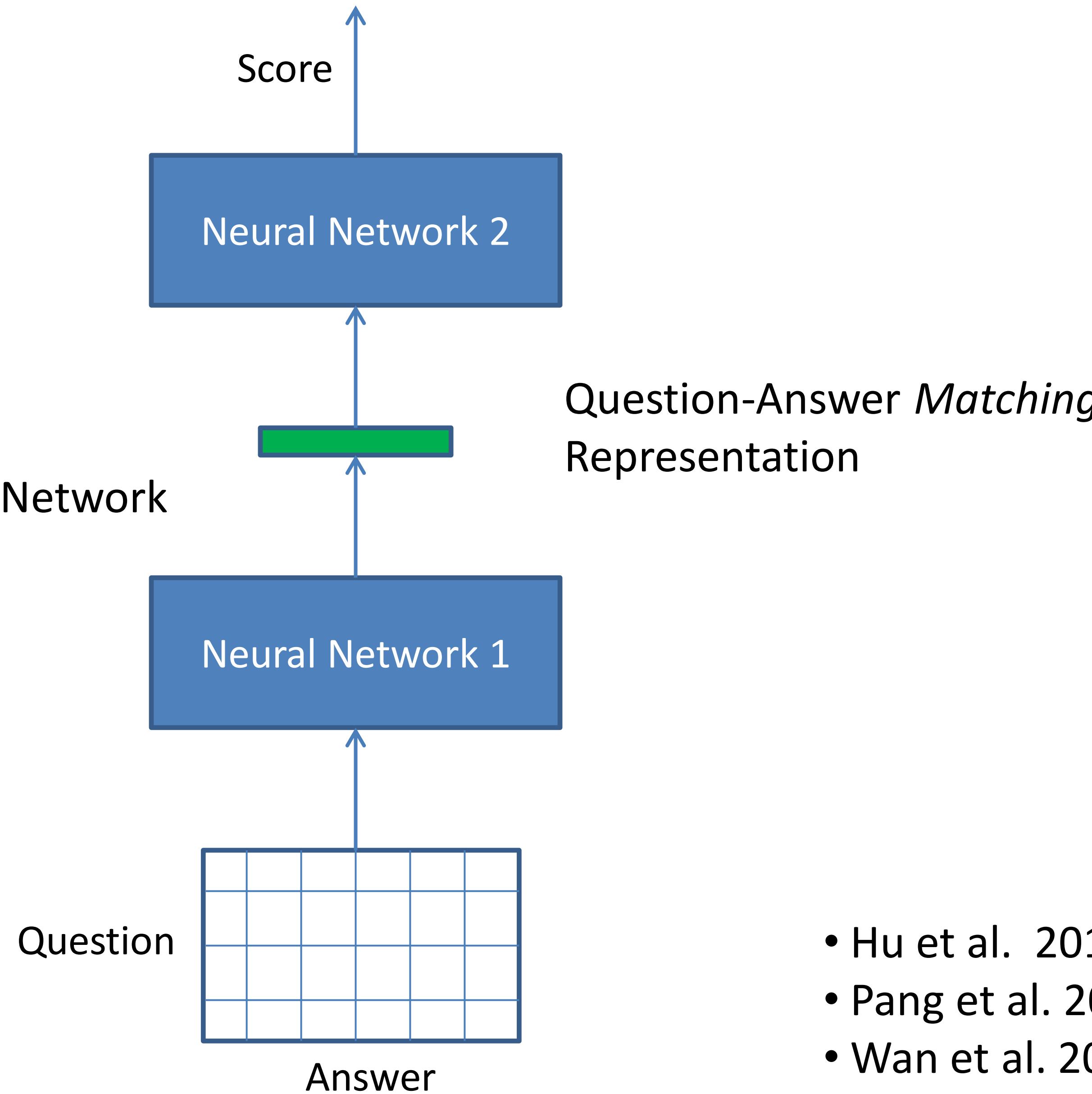
Matching: One Dimensional Matching



Neural Network 1:
Convolutional Neural Network
Neural Network 2:
Deep Neural Network, Tensor Network

- Hu et al. 2014
- Qiu & Huang 2015

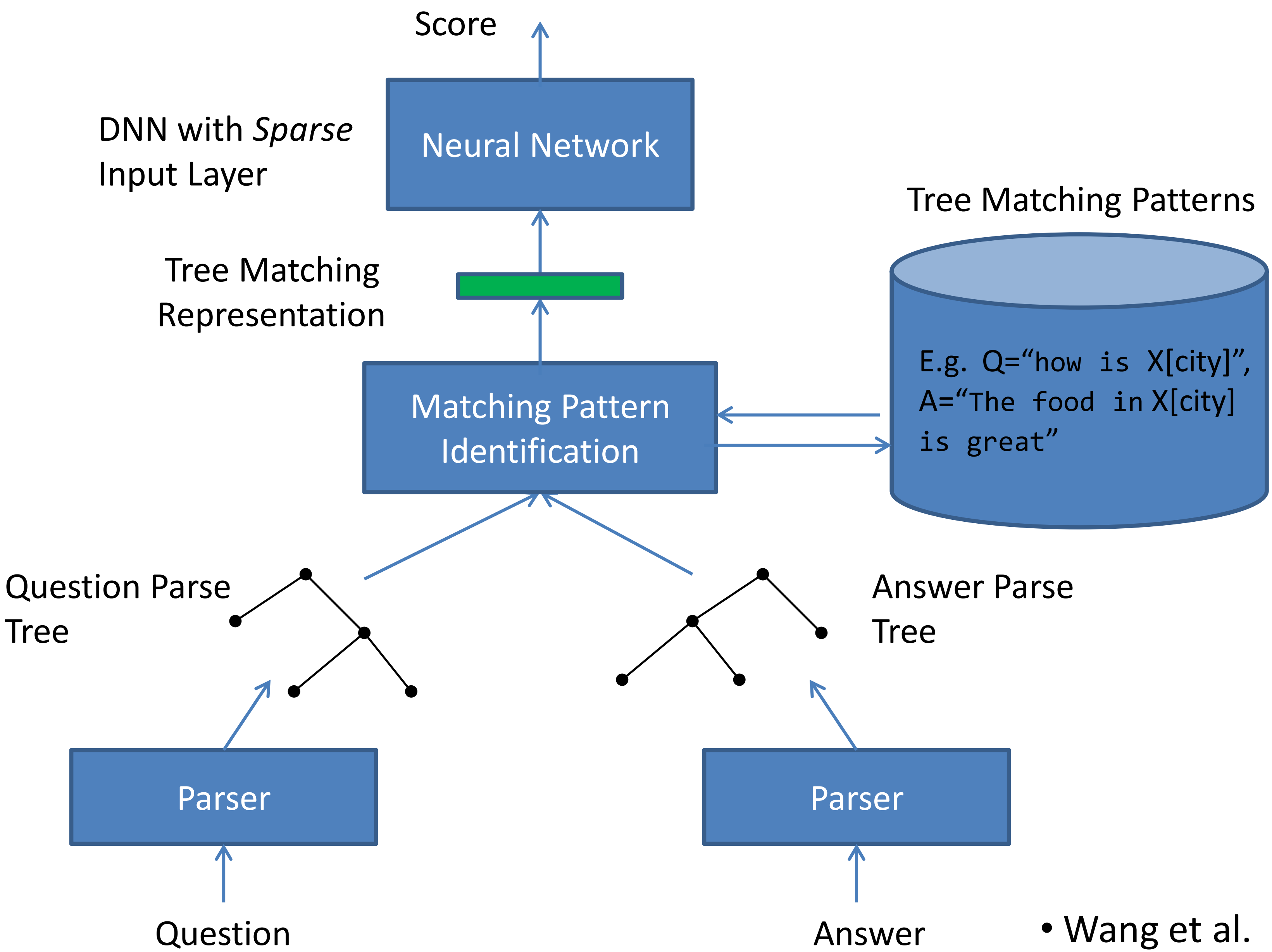
Matching: Two Dimensional Matching



Neural Network 1:
Convolutional Neural Network
Neural Network 2:
Deep Neural Network

- Hu et al. 2014
- Pang et al. 2016
- Wan et al. 2016

Matching: Tree Matching



- Wang et al. 2015

Key Observations

- CNN (Convolutional Neural Networks) usually works better than RNN (Recurrent Neural Networks) for matching (Ma et al.'15)
- 2-dimensional CNN works better than 1-dimensional CNN (Hu et al.'14)
- Representing matched tree patterns in neural network also works well, when there is enough training data (Wang et al.'15)
- Matching scores can be used as features of learning to rank models (Severyn & Moschitti'15)

References

- R. Socher, E. Huang, J. Pennington, A. Ng and C. D. Manning Dynamic Pooling and Unfolding Recursive Autoencoders for Paraphrase Detection. *NIPS* 2011.
- Z. Lu and H. Li. A Deep Architecture for Matching Short Texts. *NIPS* 2013.
- B. Hu, Z. Lu, H. Li, and Q. Chen. Convolutional Neural Network Architectures for Matching Natural Language Sentences. *NIPS* 2014.
- M. Wang, Z. Lu, H. Li, Q. Liu. Syntax-based Deep Matching of Short Texts. *IJCAI* 2015.
- P.-S. Huang, X. He, J. Gao, L. Deng, A. Acero, and L. Heck. Learning Deep Structured Semantic Models for Web Search Using Clickthrough Data. *CIKM* 2013.
- Y. Shen, X. He, J. Gao, L. Deng, and G. Mesnil. A Latent Semantic Model with Convolutional-Pooling Structure for Information Retrieval. *CIKM* 2014.
- H . Palangi, L. Deng, Y. Shen, J. Gao, X. He, J. Chen, X. Song, and R. K. Ward. Deep Sentence Embedding Using the Long Short Term Memory Network: Analysis and Application to Information Retrieval. *CoRR* 1502.06922. 2015.
- A. Severyn, and A. Moschitti. Learning to Rank Short Text Pairs with Convolutional Deep Neural Networks. *SIGIR* 2015.
- A. Severyn, and A. Moschitti. Modeling Relational Information in Question-Answer Pairs with Convolutional Neural Networks. *arXiv:*1604.01178. 2016.
- X. Qiu and X. Huang. Convolutional Neural Tensor Network Architecture for Community-based Question Answering. *IJCAI* 2015.
- W. Yin and H. Schu¨tze. MultiGranCNN: an Architecture for General Matching of Text Chunks on Multiple Levels of Granularity. *ACL* 2015.

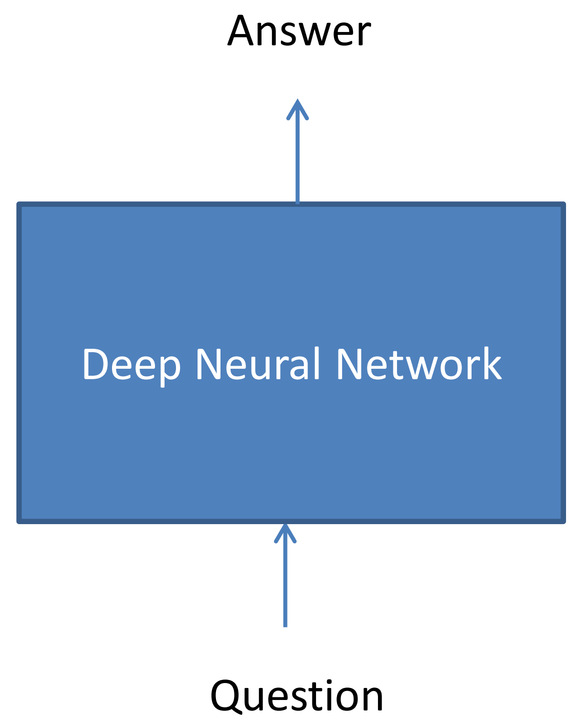
References

- L. Pang, Y. Lan, J. Guo, J. Xu, S. Wan and X. Cheng, X. Text Matching as Image Recognition. *AAAI* 2016.
- S. Wan, Y. Lan, J. Guo, J. Xu, L. Pang , and X. Cheng . A Deep Architecture for Semantic Matching with Multiple Positional Sentence Representations. *AAAI* 2016.
- H. Amiri, P. Resnik, J. Boyd-Graber, and H. Daumé III. Learning Text Pair Similarity with Context-sensitive Autoencoders. *ACL* 2016.
- L. Ma, Z. Lu, L. Shang, Hang Li . Multimodal Convolutional Neural Networks for Matching Image and Sentence. *ICCV* 2015.

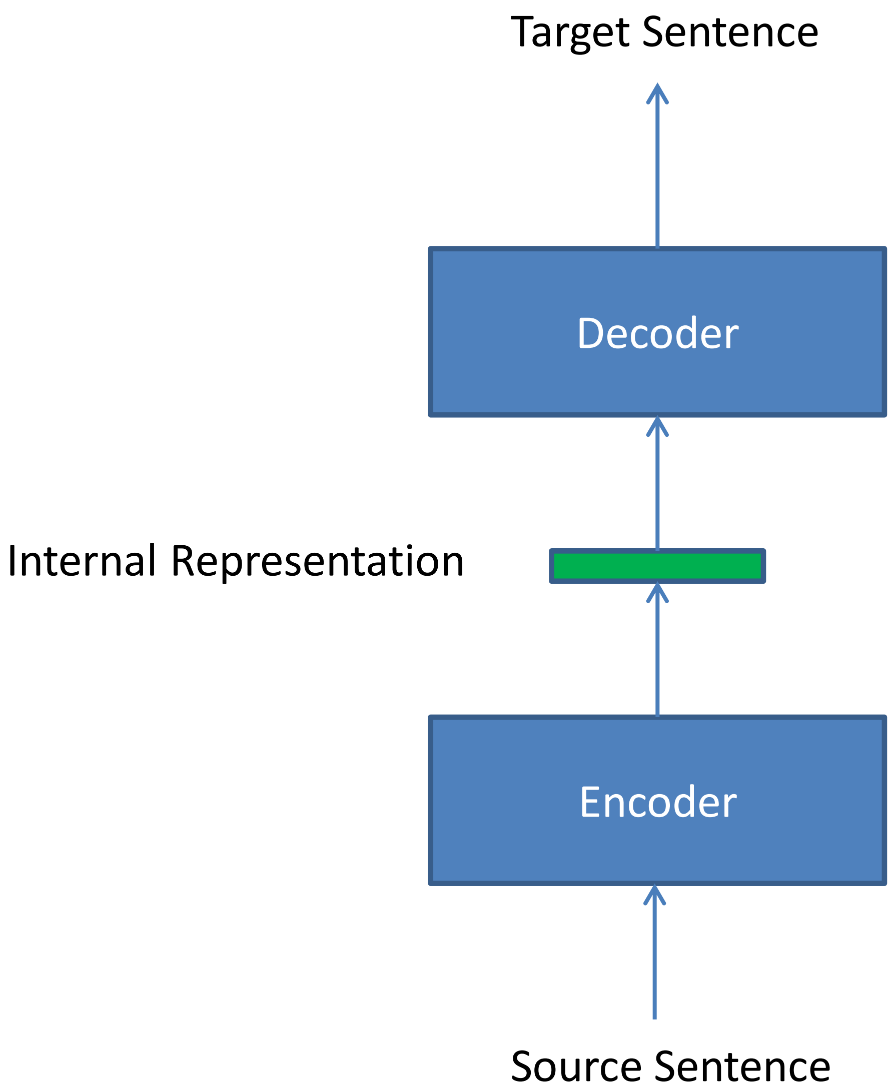
Translation

Translation

- Tasks
  - **Question Answering:** answer generation from question
  - **Search:** similar query generation
- Approaches
  - Sequence-to-Sequence Learning
  - RNN Encoder-Decoder
  - Attention Mechanism

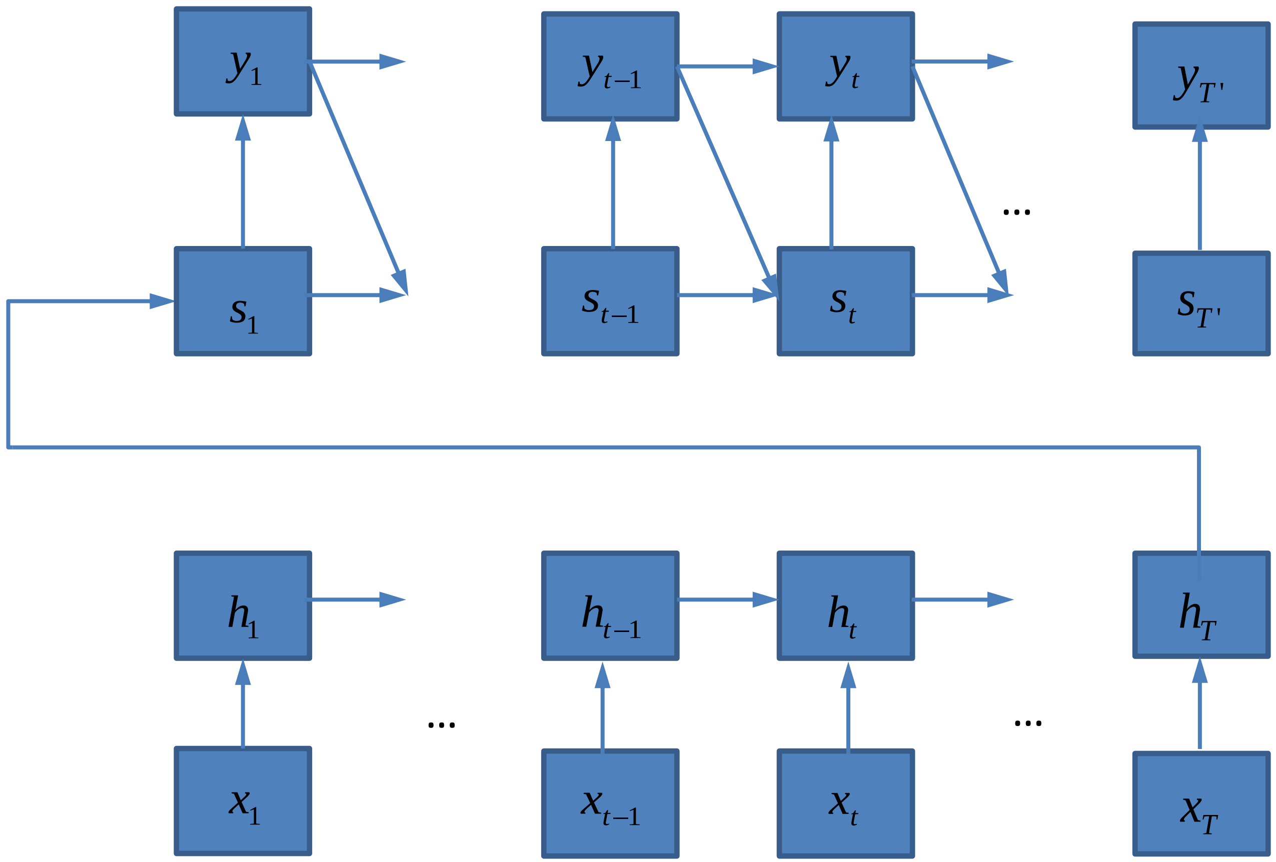


Translation: Sequence-to-Sequence Learning (Same for RNN Encoder-Decoder)



Encoder:
Recurrent Neural Network
Decoder:
Recurrent Neural Network

Translation: Sequence to Sequence Learning



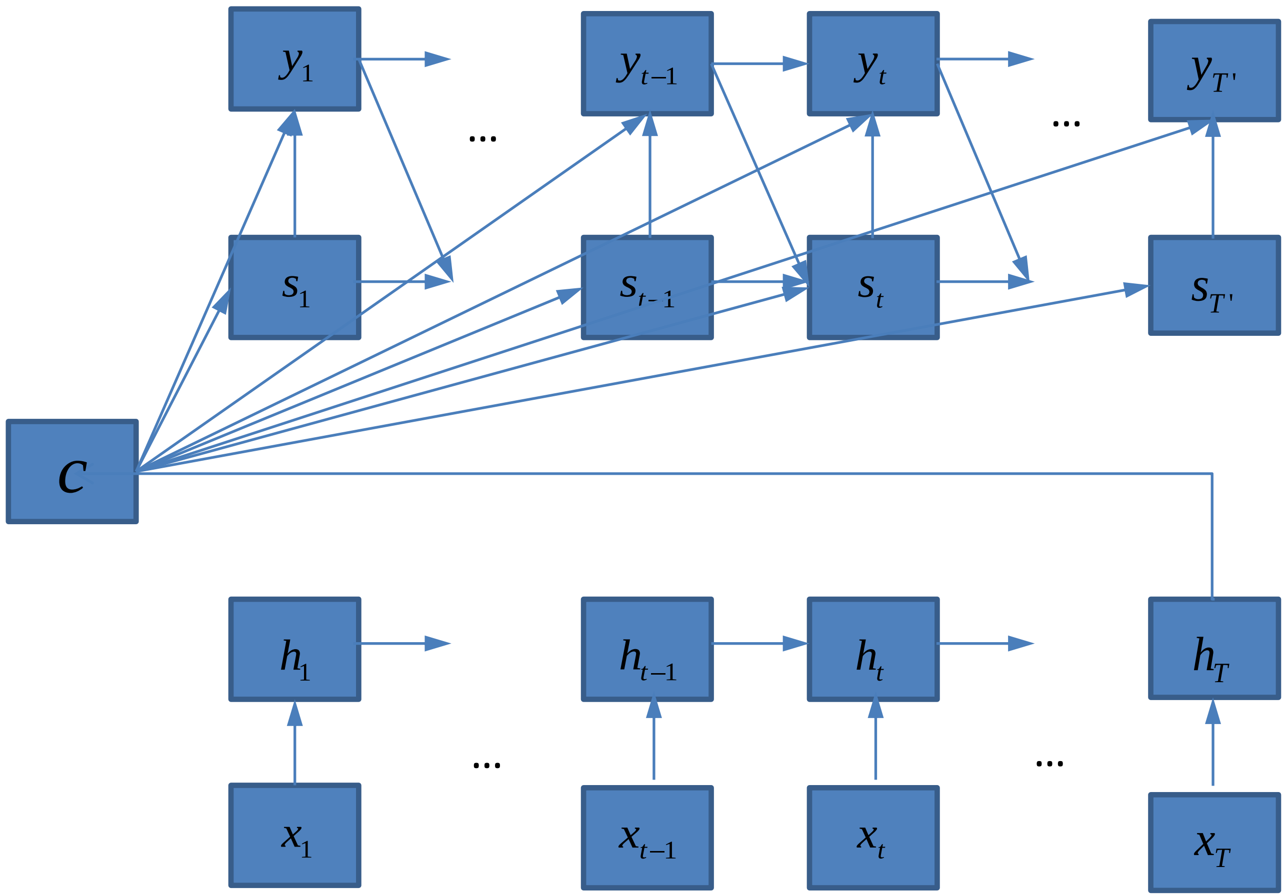
- Hierarchical LSTM
- Different LSTM models for encoder and decoder
- Reverse order of words in source sentence

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

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

- Sutskever et al. 2014

Translation: RNN Encoder-Decoder



- Context vector represents source sentence
- GRU is used

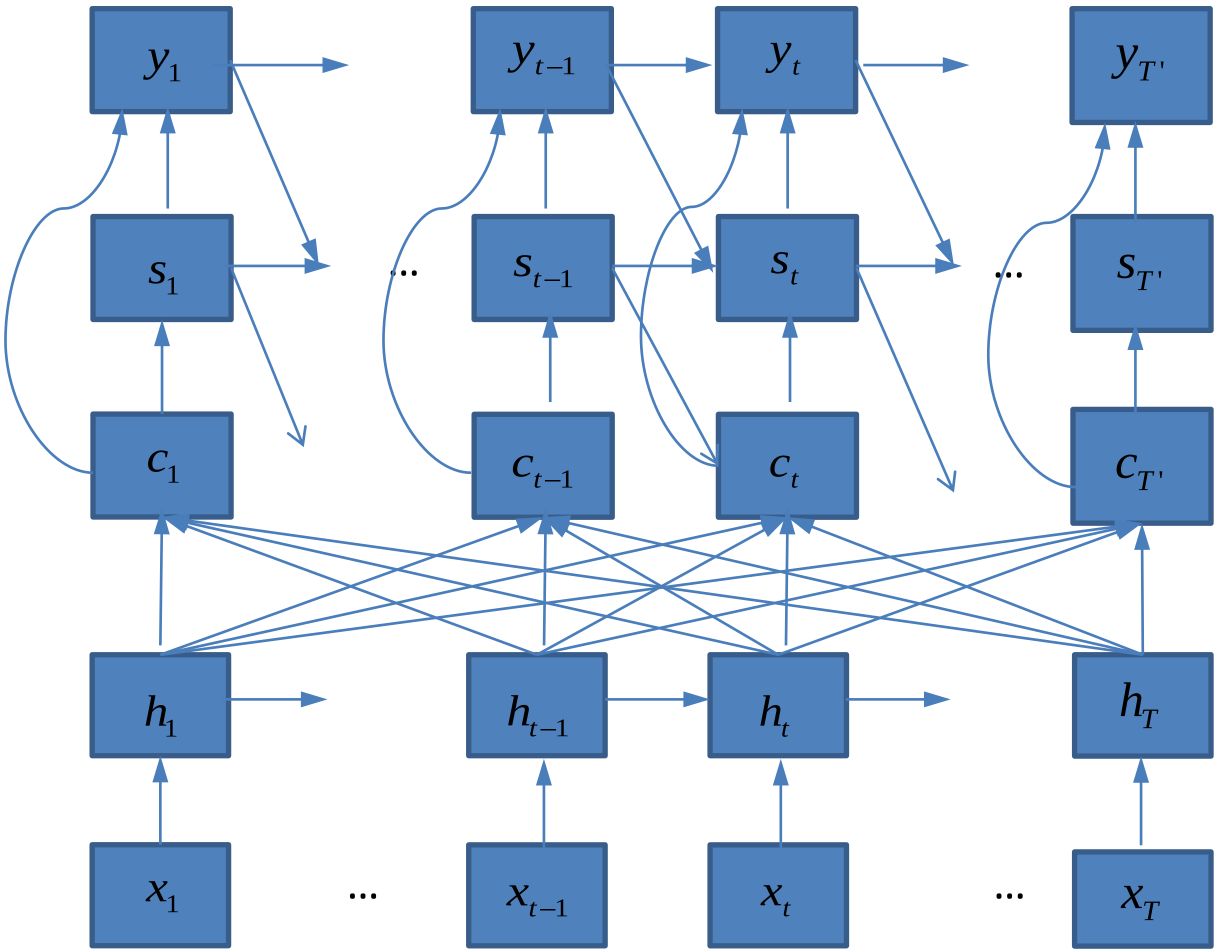
$$P(y_t \mid y_1 \cdots y_{t-1}, \mathbf{x}) = g(y_{t-1}, s_t, c), c = h_T$$

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

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

- Cho et al. 2014

Translation: Attention Mechanism



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

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

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

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

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

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

Bahdanau, et al. 2014

Key Observations

- RNNs (Recurrent Neural Networks) is more suitable for generation or translation
- LSTM and GRU can retain long distance dependency (Cho et al.’14)
- Bidirectional model works better than one-directional model (Bahdanau et al.’15)
- Attention mechanism can improve accuracy and efficiency of RNN models (Bahdanau et al.’15)
- Neural Machine Translation get generate more fluent but less faithful results than Statistical Machine Translation

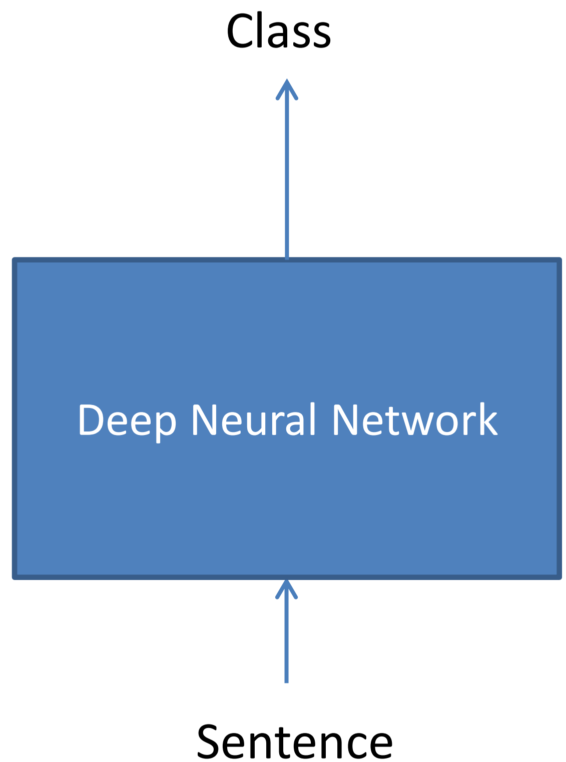
References

- I. Sutskever, O. Vinyals, and Le, Q.V. Le. Sequence to Sequence Learning with Neural Networks. *NIPS* 2014.
- K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. *EMNLP*, 2014.
- D. Bahdanau, K. Cho, and Y. Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. *ICLR*, 2015.
- F. Meng, Z. Lu, Z. Tu, H. Li, Q. Liu. A Deep Memory-based Architecture for Sequence-to-Sequence Learning. *arXiv:*1506.06442, 2015.
- M. Luong, H. Pham, and C. D. Manning. Effective Approaches to Attention-based Neural Machine Translation. *arXiv:*1508.04025, 2015.
- A. Rush, S. Chopra, and J Weston. A Neural Attention Model for Abstractive Sentence Summarization. *arXiv:*1509.00685, 2015.
- K. Hermann, T. Kocisky, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, and P. Blunsom. Teaching Machines to read and Comprehend. *NIPS* 2015.
- L. Shang, Z. Lu, H. Li. Neural Responding Machine for Short Text Conversation. *ACL* 2015.
- O. Vinyals and Q. V. Le. A Neural Conversational Model. *arXiv:*1506.05869, 2015.
- J. Gu, Z. Lu, H. Li & V. O. Li. Incorporating Copying Mechanism in Sequence-to-Sequence Learning. *ACL* 2016.

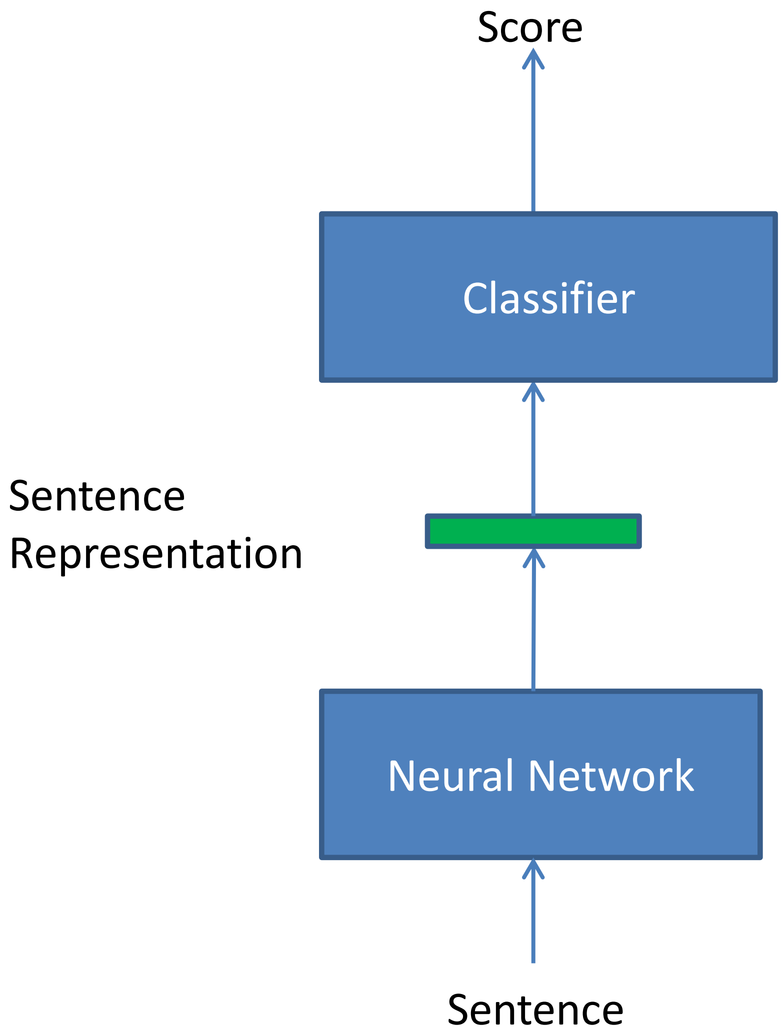
Classification

Classification

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



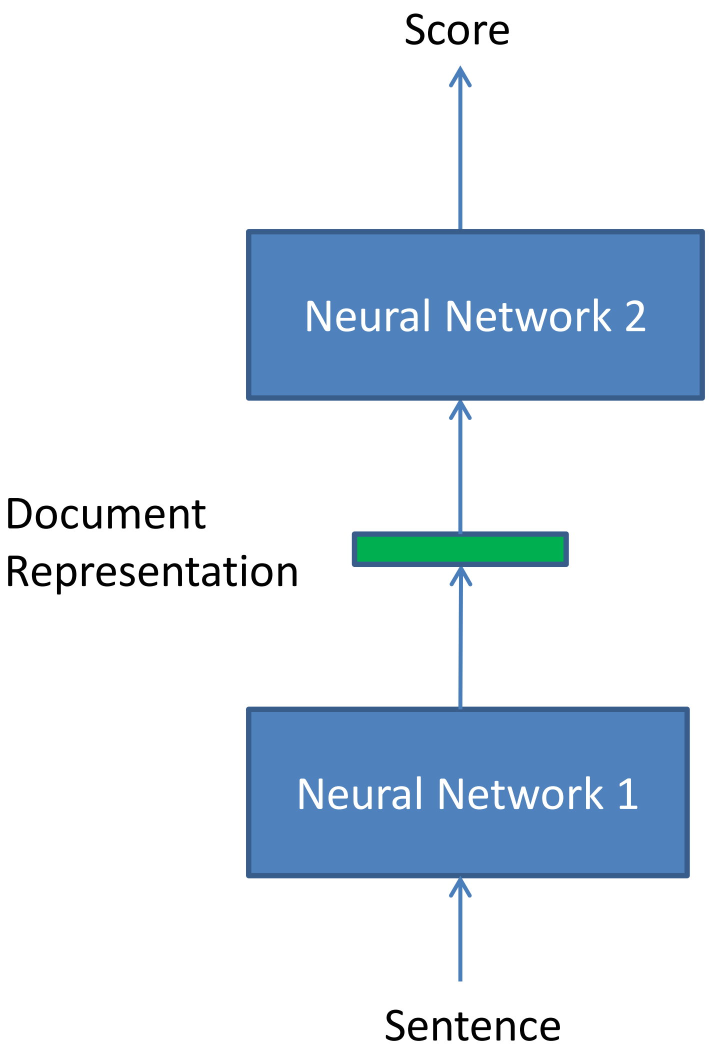
Sentence Classification: Word Level Model



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
- Iyyer et al. 2015

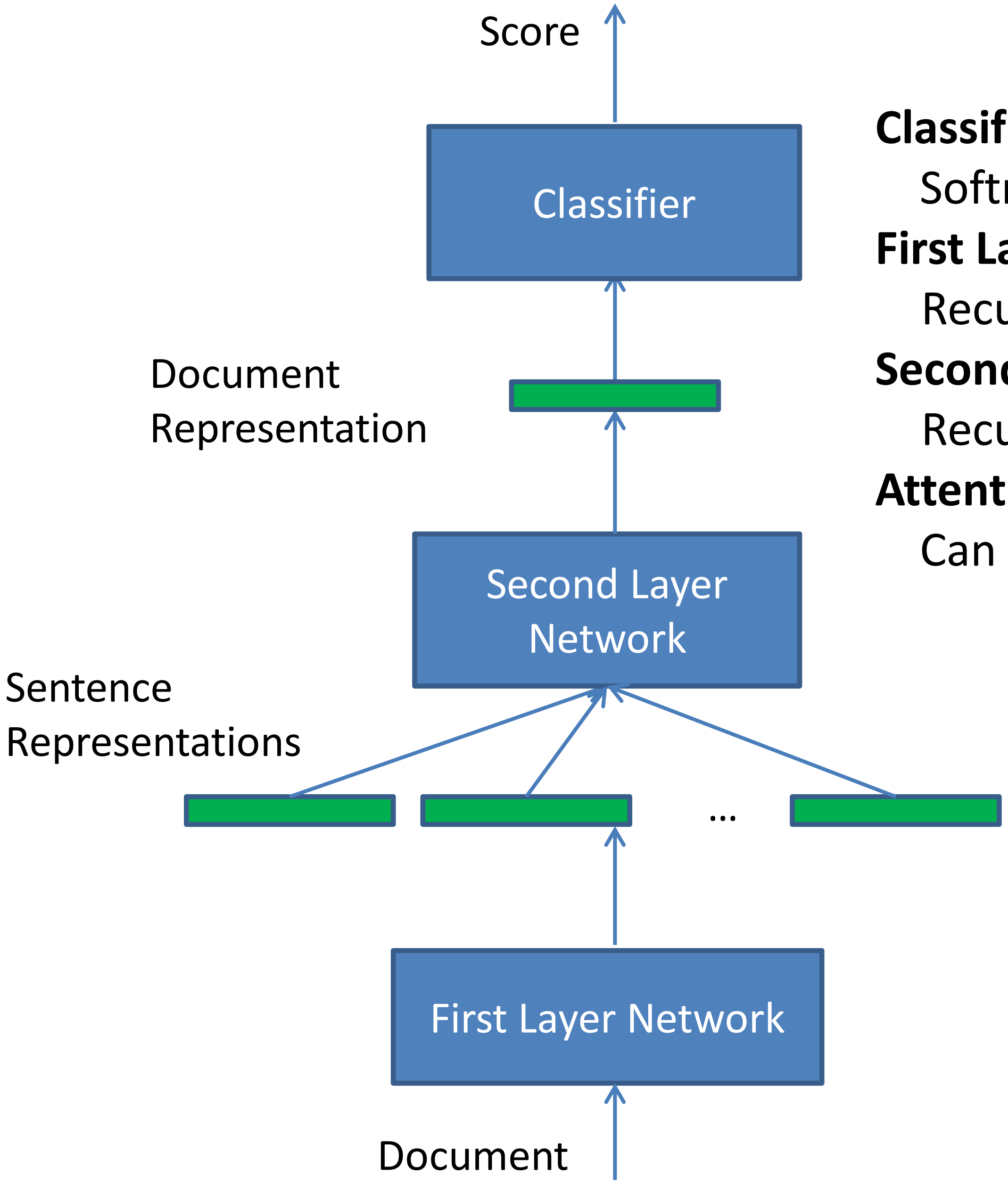
Document Classification: Character Level Model



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

Key Observations

- CNN models are used for both sentence classification and document classification (Kim’14, Blunsom et al.’14, Johnson & Zhang’14, Zhang et al.’15)
- Input can be continuous word embedding (e.g., Kim), discrete word embedding (Johnson & Zhang’14), and even character level embedding (Zhang et al.’15)
- Two-layer models are used for document classification (Yang et al.’16)
- Bag-of-words models work better than syntax aware models (Iyyer et al.’15)

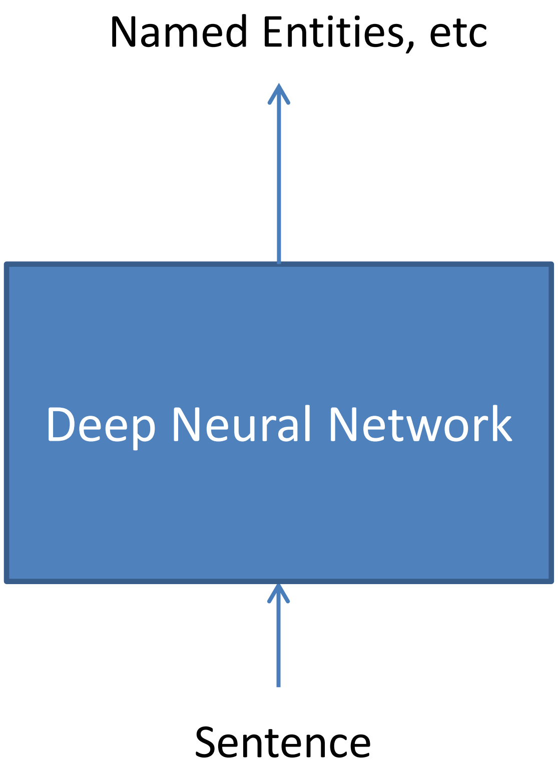
References

- Y. Kim. Convolutional Neural Networks for Sentence Classification. *EMNLP* 2014.
- P. Blunsom, E. Grefenstette, and N. Kalchbrenner. A Convolutional neural network for modeling sentences. *ACL* 2014.
- R. Johnson and T. Zhang. Effective Use of Word Order for Text Categorization with Convolutional Neural Networks. *arXiv:*1412.1058, 2014.
- H. Zhao, Z. Lu, and P. Poupart. Self-Adaptive Hierarchical Sentence Model. *IJCAI* 2015.
- L. Mou, H. Peng, G. Li, Y. Xu, L. Zhang, and Z. Jin. Discriminative Neural Sentence Modeling by Tree-based Convolution. *arXiv:* 1504.01106, 2015.
- D. Tang, B. Qin, and T. Liu. Document Modeling with Gated Recurrent Neural Network for Sentiment Classification. *EMNLP* 2015.
- S. Lai, L. Xu, K. Liu, and J. Zhao. Recurrent Convolutional Neural Networks for Text Classification. *AAAI* 2015.
- X. Zhang, J. Zhao, and Y. LeCun. Character-level Convolutional Networks for Text Classification. *NIPS* 2015.
- M. Iyyer, V. Manjunatha, J. Boyd-Graber, and H. Daumé III. Deep Unordered Composition Rivals Syntactic Methods for Text Classification. *ACL* 2015.
- K. Tai, R. Socher, and C. D. Manning. Improved Semantic Representations from Tree-structured Long Short-term Memory Networks. *arXiv:*1503.00075, 2015.
- Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy. Hierarchical Attention Networks for Document Classification. *NAACL* 2016.

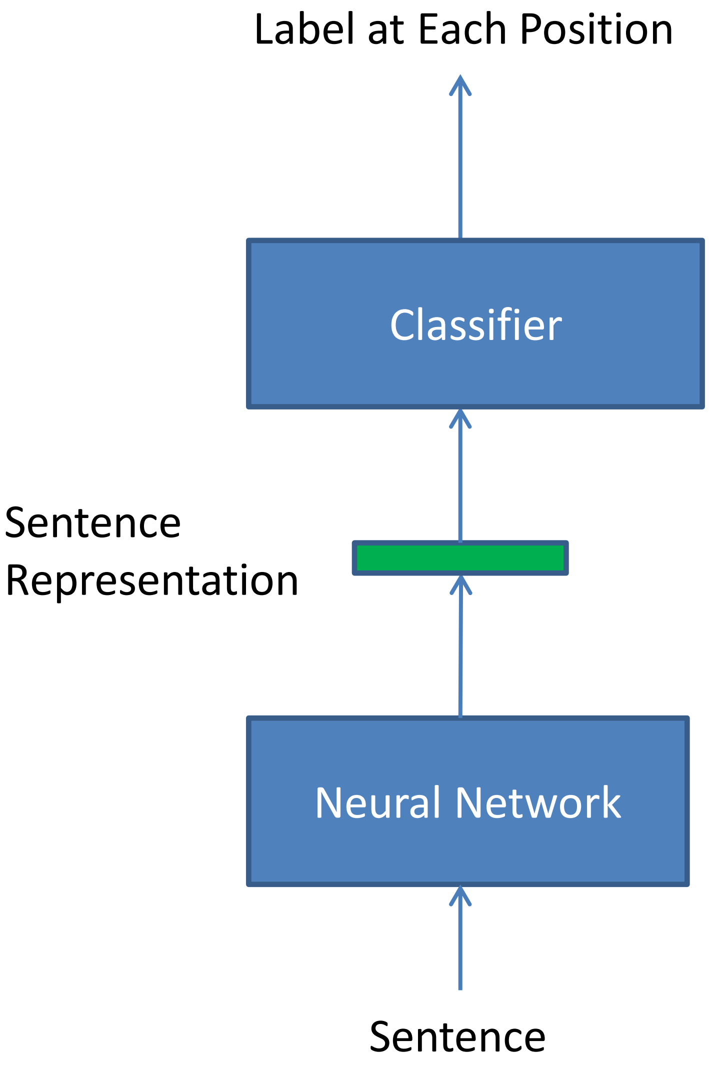
Structured Prediction

Structured Prediction

- 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:

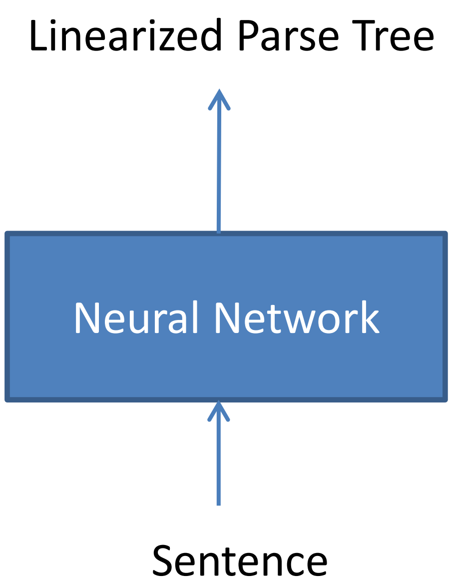
Softmax

Neural Network:

Convolutional Neural Network

- Collobert et al. 2011

Structured Prediction: Sequence-to-Sequence Learning

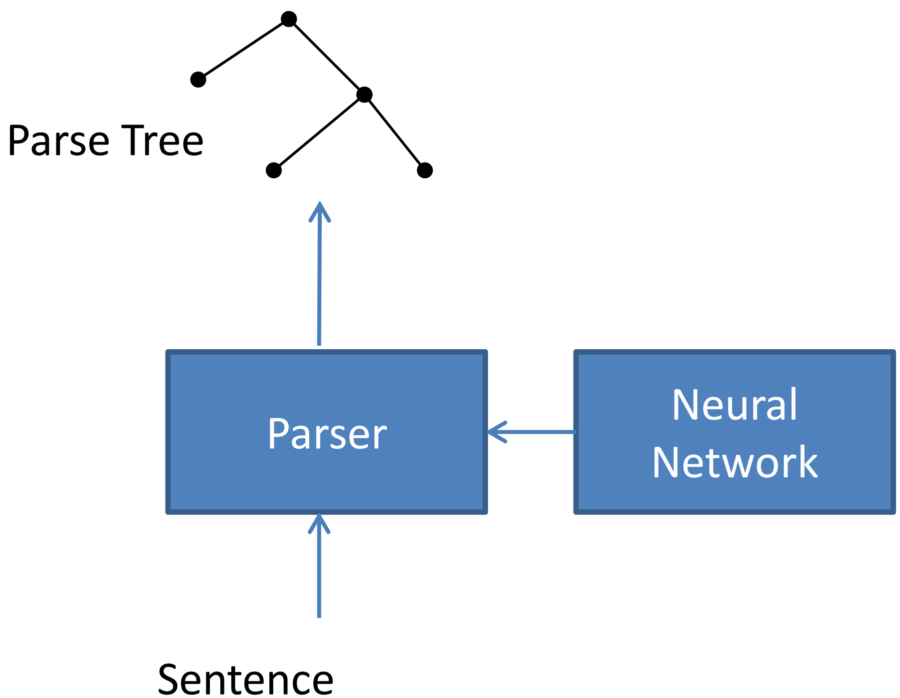


Neural Network:
Sequence-to-Sequence Learning Model
Training Data:
Pairs of Sentence and Linearized Parse Tree

E.g.,
`John has a dog .` → (S (NP NNP)$_{NP}$ (VP VBZ (NP DT NN)$_{NP}$)$_{VP}$.)$_{S}$

- Vinyals et al. 2015

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

Key Observations

- Simplest approach is to employ shallow CNN (Collobert et al.'11)
- Sequence to sequence learning can be employed, when labeled training data is available (Vinyals et al.'15)
- Neural networks based parsers can achieve state-of-the-art performance (Chen & Manning'14, Andor et al., '16)

References

- R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa. Natural Language Processing (Almost) from Scratch. *JMLR*, 2011.
- A . Graves. Neural Networks. In *Supervised Sequence Labeling with Recurrent Neural Networks*, 2012.
- O. Vinyals, Ł. Kaiser, T. Koo, S. Petrov, I. Sutskever, & G. Hinton. Grammar as a Foreign Language. *NIPS* 2015.
- D. Chen & C. D. Manning. A Fast and Accurate Dependency Parser using Neural Networks. *EMNLP* 2014.
- R. Socher, J. Bauer, C. D. Manning, and A. Y. Ng. Parsing with Compositional Vector Grammars. *ACL* 2013.
- K. Yao, B. Peng, G. Zweig, D. Yu, X. Li, and F .Gao. Recurrent Conditional Random Field for Language Understanding. *ICASSP* 2014.
- H . Zhou, Y. Zhang, and J. Chen. A Neural Probabilistic Structured-Prediction Model for Transition-based Dependency Parsing. *ACL* 2015.
- C. Alberti, D. Weiss. G. Coppola, and S. Petrov. Improved Transition-Based Parsing and Tagging with Neural Networks. *EMNLP* 2015.
- D. Weiss, C. Alberti, M. Collins, and S. Petrov. Structured Training for Neural Network Transition-based Parsing. *arXiv*:1506.06158, 2015.
- S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, & P . H. S. Torr. Conditional Random Fields as Recurrent Neural Networks. *ICCV* 2015.
- G. Durrett and D. Klein. Neural CRF Parsing. *arXiv:*1507.03641, 2015.

References

- M. Ballesteros, C. Dyer, and N. A. Smith. Improved Transition-based Parsing by Modeling Characters instead of Words with LSTMs. *arXiv:*1508.00657, 2015.
- C. Dyer, M. Ballesteros, W. Ling, A. Matthews, and N. A. Smith. Transition-based Dependency Parsing with Stack Long Short-term Memory. *arXiv:*1505.08075, 2015.
- T. Watanabe and E. Sumita. Transition-based Neural Constituent Parsing. *ACL* 2015.
- H. Guo, X. Zhu, M. R. Min. A Deep Learning Model for Structured Outputs with High-order Interaction. *arXiv:*1504.08022, 2015.
- D. Belanger and A. McCallum. Structured Prediction Energy Networks. arXiv:1511.06350, 2015.
- D. Andor, C. Alberti, D. Weiss, A. Severyn, A. Presta, K. Ganchev, and M. Collins, M. Globally Normalized Transition-based Neural Networks. arXiv:1603.06042, 2016.

Comparison with State-of-the-Art for Fundamental Problems

	Accuracy	Domain Knowledge
Matching	DL significantly improves	Little is needed
Translation	DL significantly improves, with different flavor	Little is needed
Classification	DL significantly improves	Little is needed
Structured Prediction	DL is comparable	Little is needed

Part 3: Applications of Deep Learning to IR

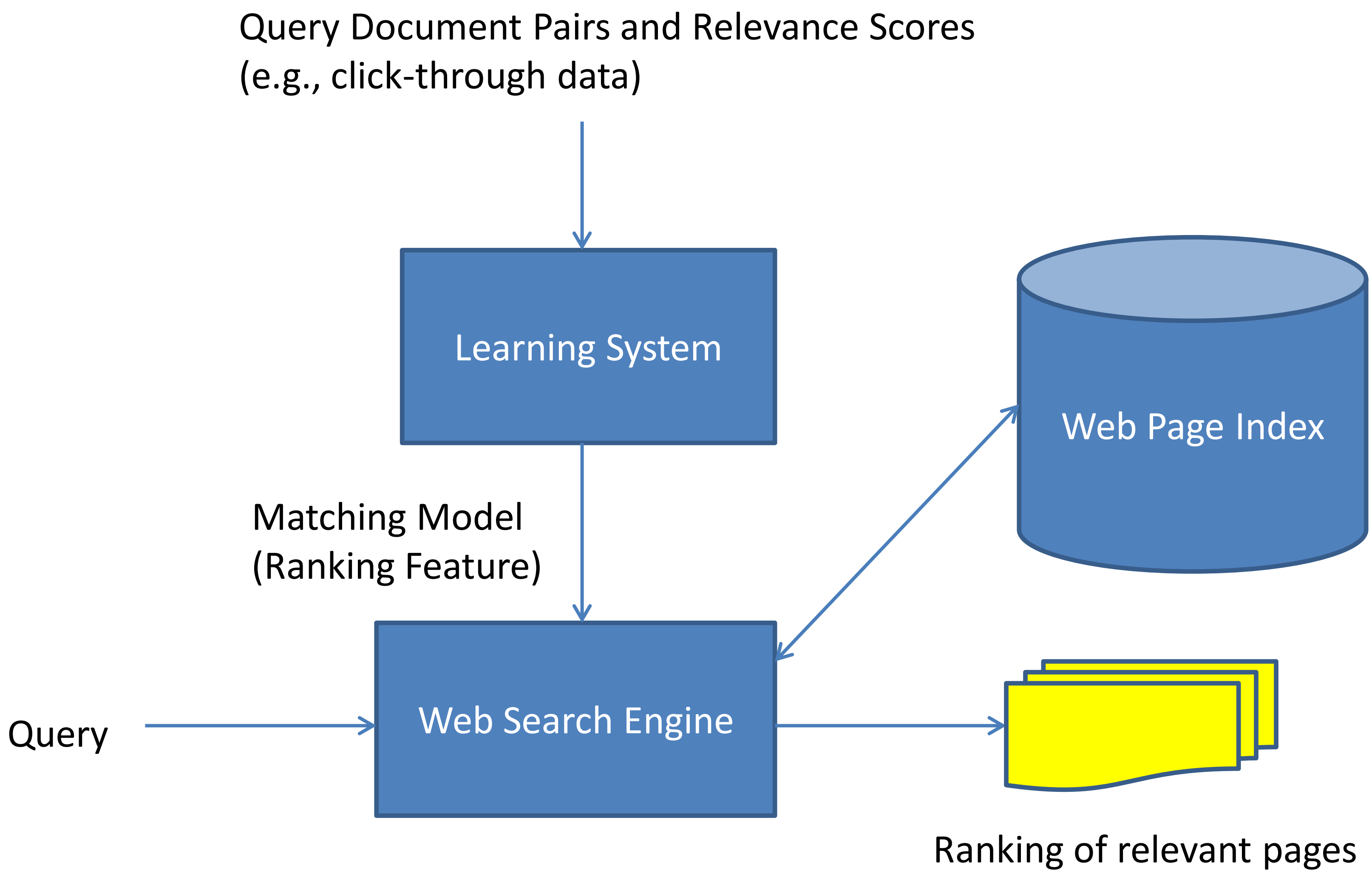
Outline of Part 3

- Document Retrieval
- Retrieval-based Question Answering
- Generation-based Question Answering
- Question Answering from Relational Database
- Question Answering from Knowledge Graph
- Multi-Turn Dialogue
- Image Retrieval

Document Retrieval

Shen et al. 2013

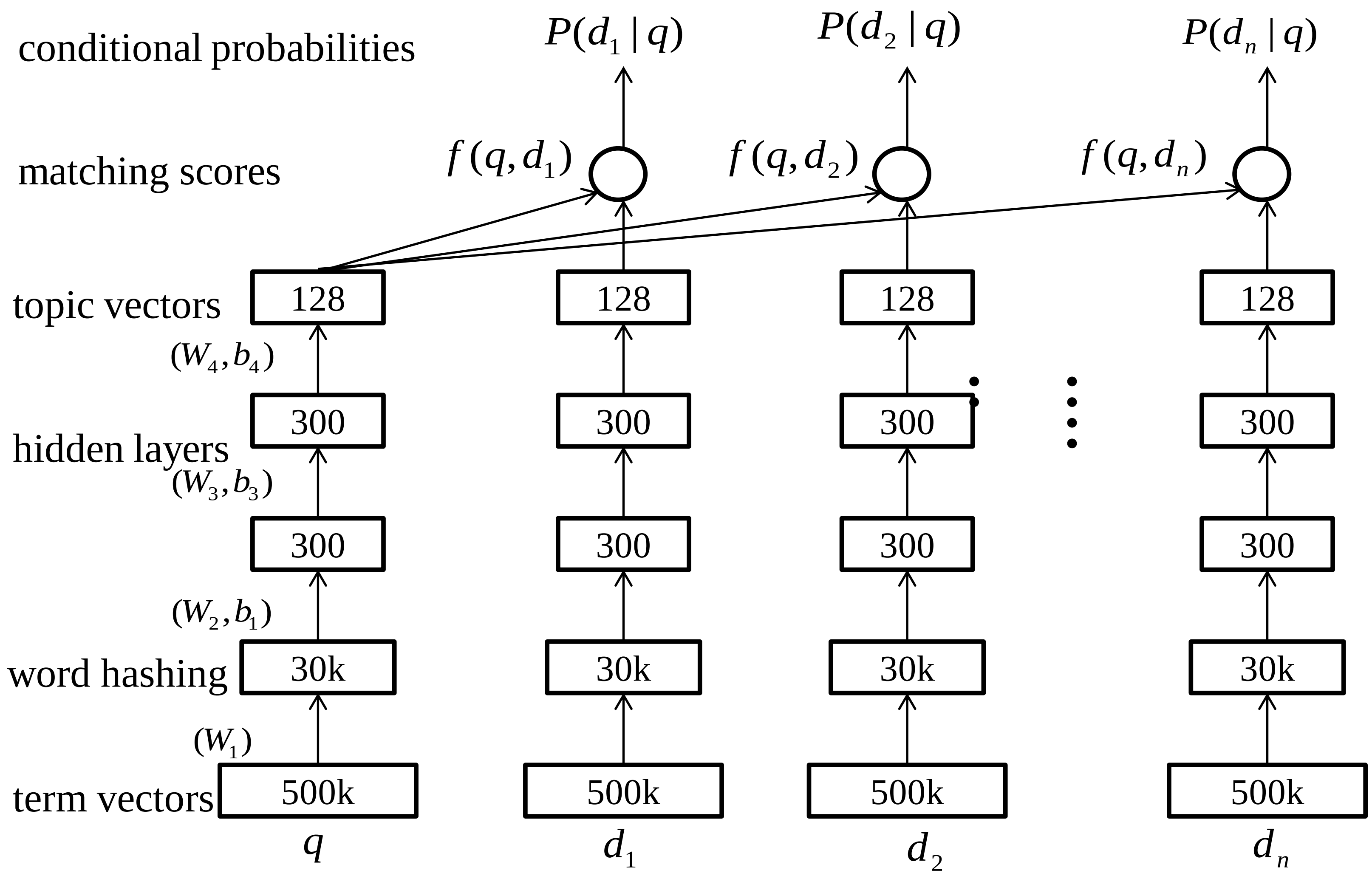
Learning to Match for Document Retrieval



Deep Structured Semantic Model (DSSM)

- Approach: Projection to Latent Space
- DSSM: deep neural network for semantic matching between query and document
- Using click through data as training data
- Tri-letter based word hashing for scalable word representation

System Architecture



Tri-letter Hashing

Representation in vocabulary

$$\text{cat} = \begin{bmatrix} 0 \\ \vdots \\ \vdots \\ 1 \\ \vdots \\ \vdots \\ 0 \end{bmatrix}$$

$$|\,\mathrm{Voc}\,| = 500K$$

Representation with tri-letters

#cat# → #ca, cat, at#

$$\text{cat} = \begin{array}{c} \\ \\ \\ \\ \\ \\ \\ \\ \end{array}\!\!\begin{bmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 1 \\ 1 \\ \vdots \\ 0 \end{bmatrix}\begin{array}{l} \\ \\ \text{at\#} \\ \\ \text{\#ca} \\ \text{cat} \\ \\ \\ \end{array}$$

$$|\,\mathrm{TriL}\,| = 30K$$

- Generalizable to unknown words
- Robust to misspelling, inflection
- Very small collision

Experimental Results

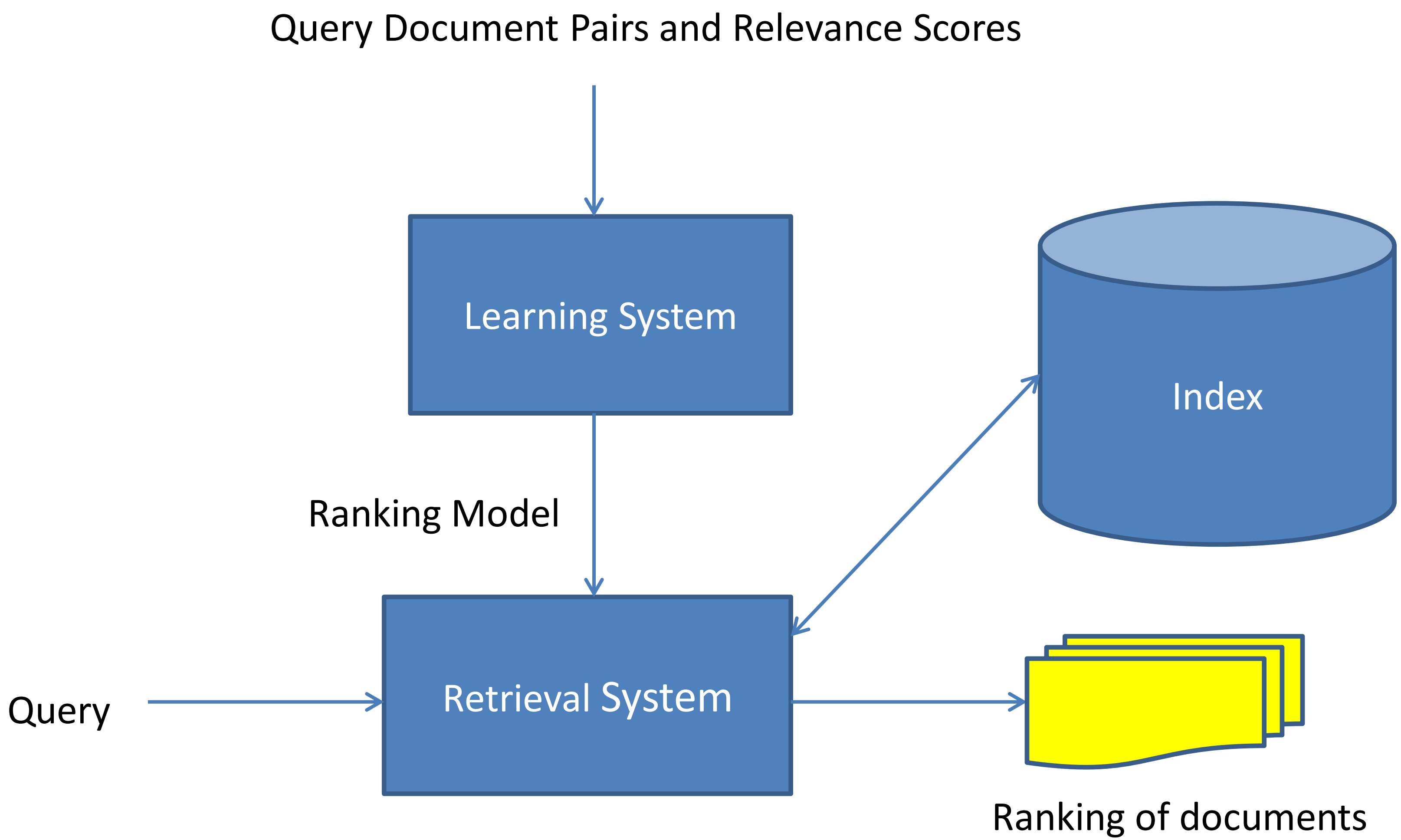
- Experiment
  - Training: 100 million pairs of query-document title in click-through data
  - Testing: 16K queries each associated with about 15 documents

	NDCG@1	NDCG@3	NDCG@10
BM25	30.8	37.3	45.5
LSA	29.8	37.2	45.5
Translation Model	33.2	40.0	47.8
DSSM	**36.2**	**42.5**	**49.8**

Document Retrieval

Severyn & Moschitti 2015

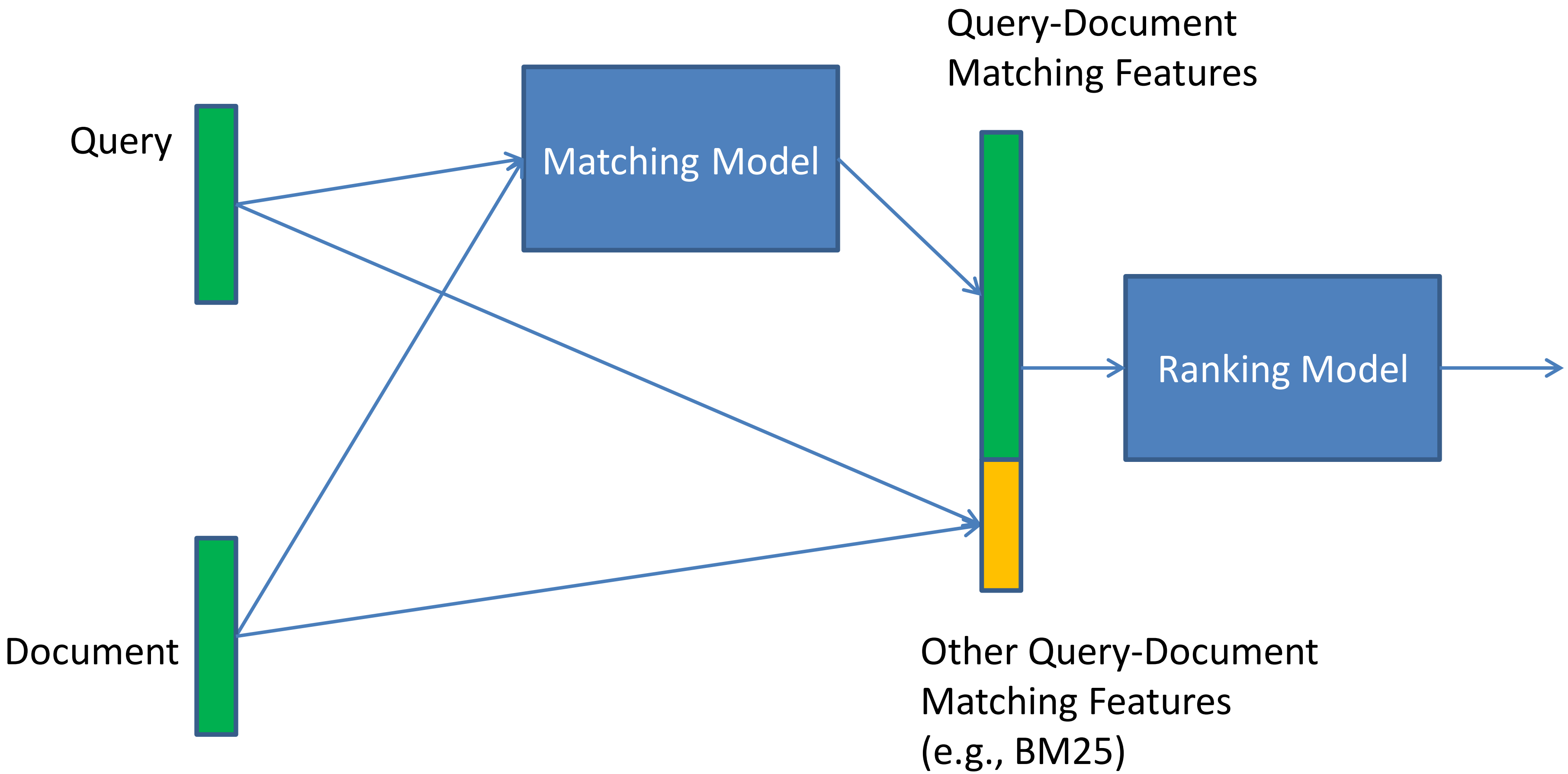
Learning to Rank for Document Retrieval



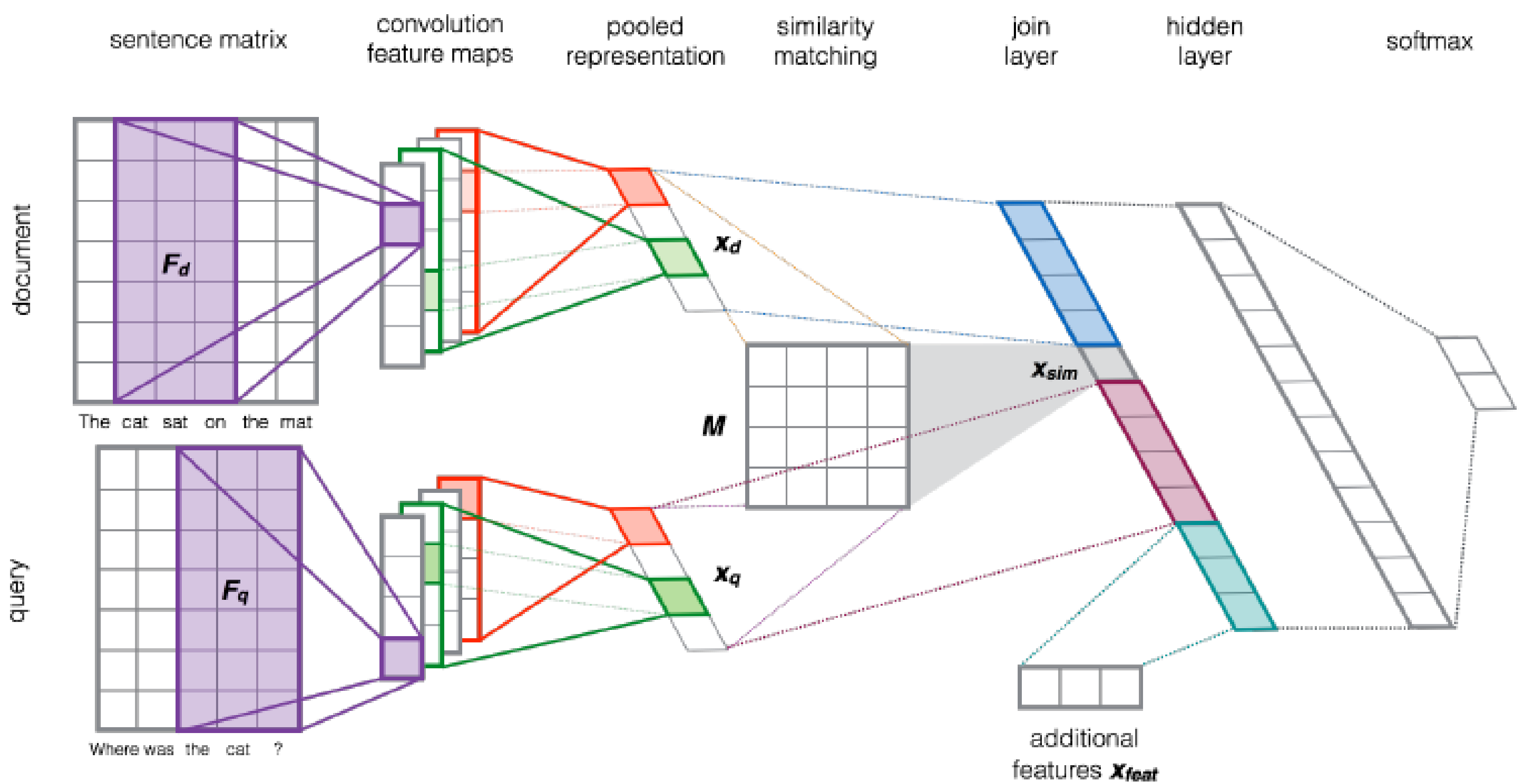
Learning to Rank System Using Neural Network

- Approach: simultaneously learn matching model and ranking model
- Matching model: Projection into Latent Space, Using CNN
- Ranking model: taking matching model output as features, as well as other features, Using DNN

Relation between Matching Model and Ranking Model



System Architecture



Experimental Results

- TREC QA Experiment
  - Training: 53K question answer pairs
  - Test: 13K question answer pairs

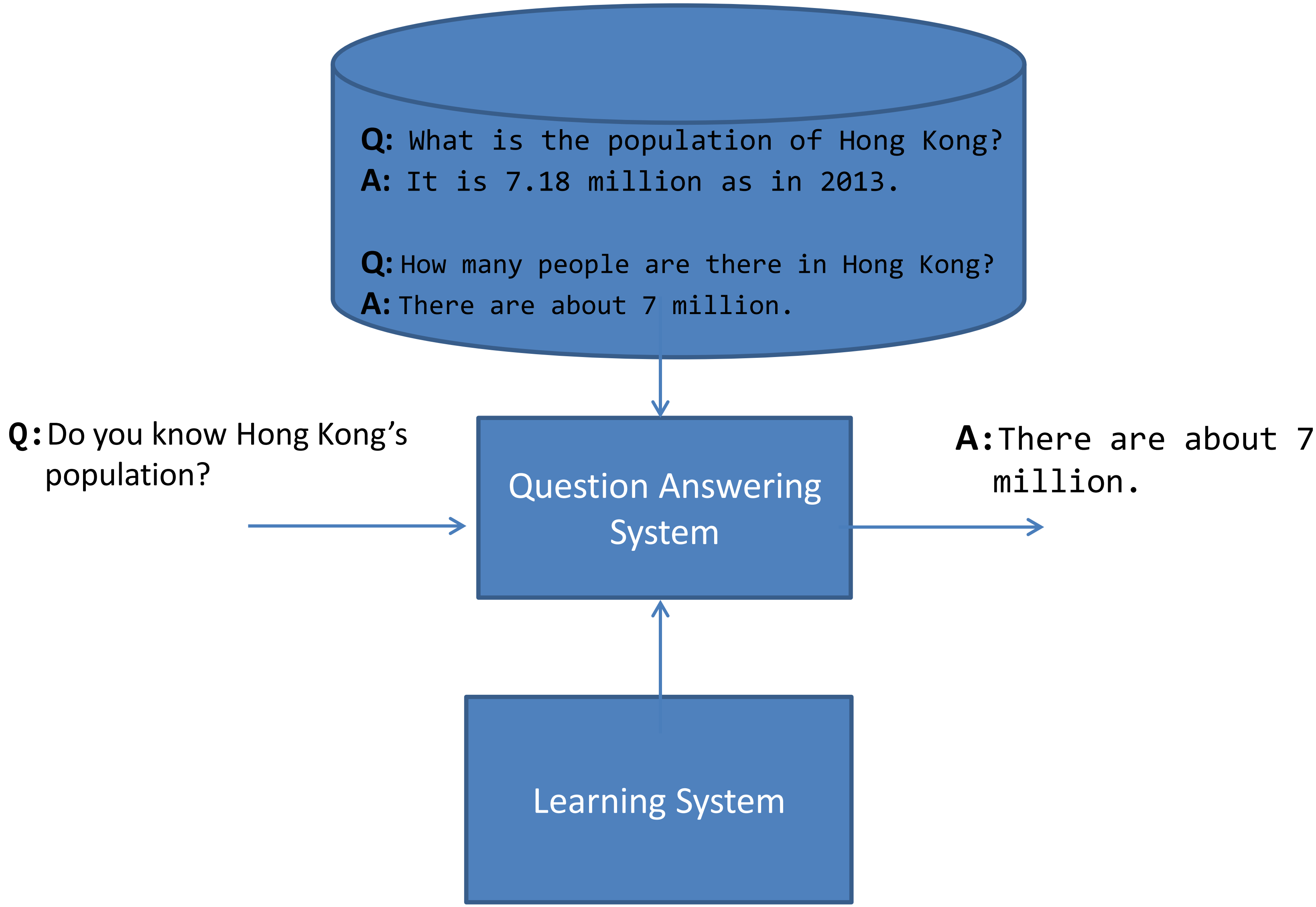
	MAP	MRR
Tree Edit Model (Parsing)	60.9	69.2
Tree Kernel	67.8	73.6
CNN Model	**74.6**	**80.8**

Retrieval based Question Answering

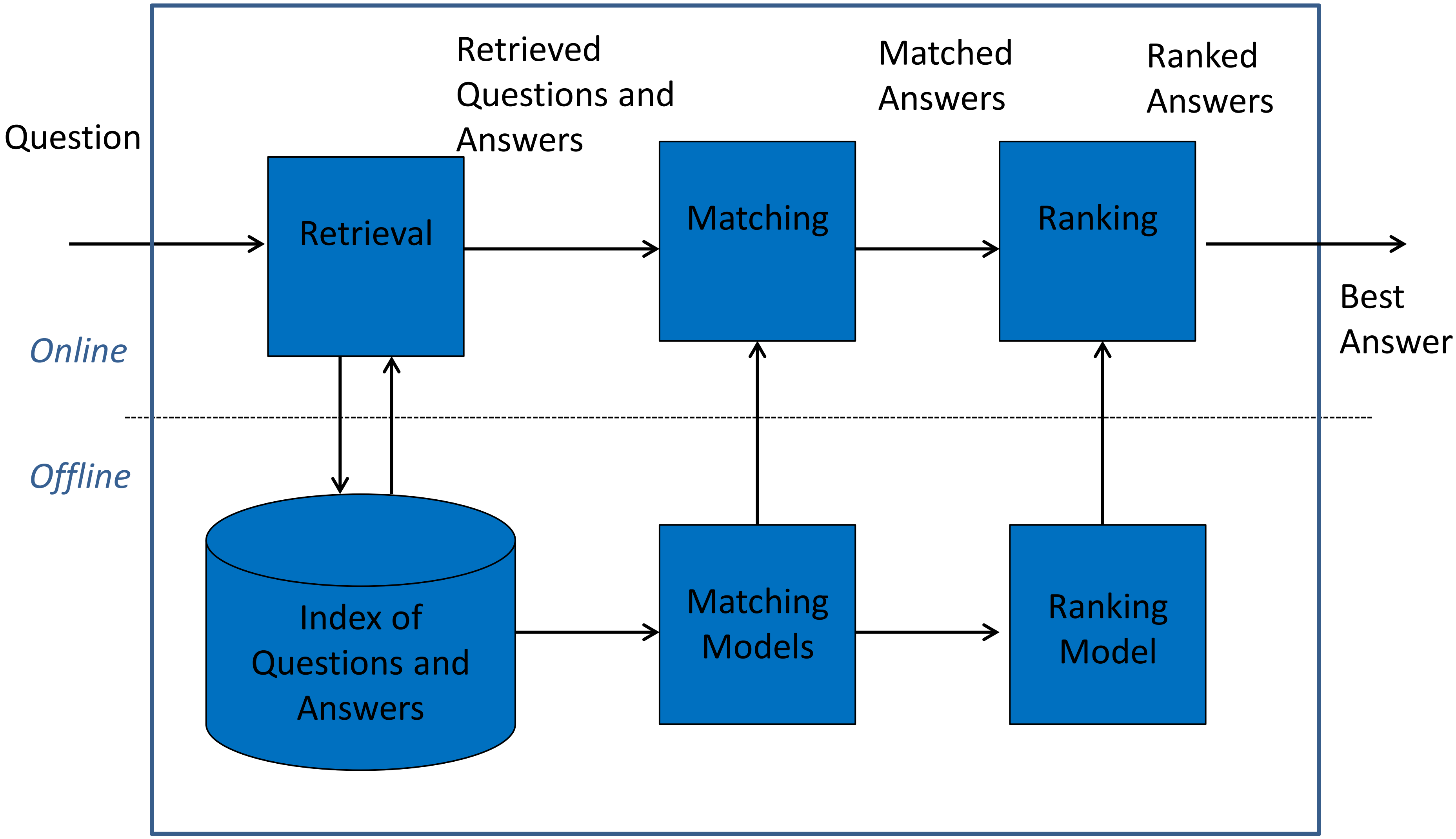
Ji et al. 2014

Hu et al. 2014

Retrieval-based Question Answering

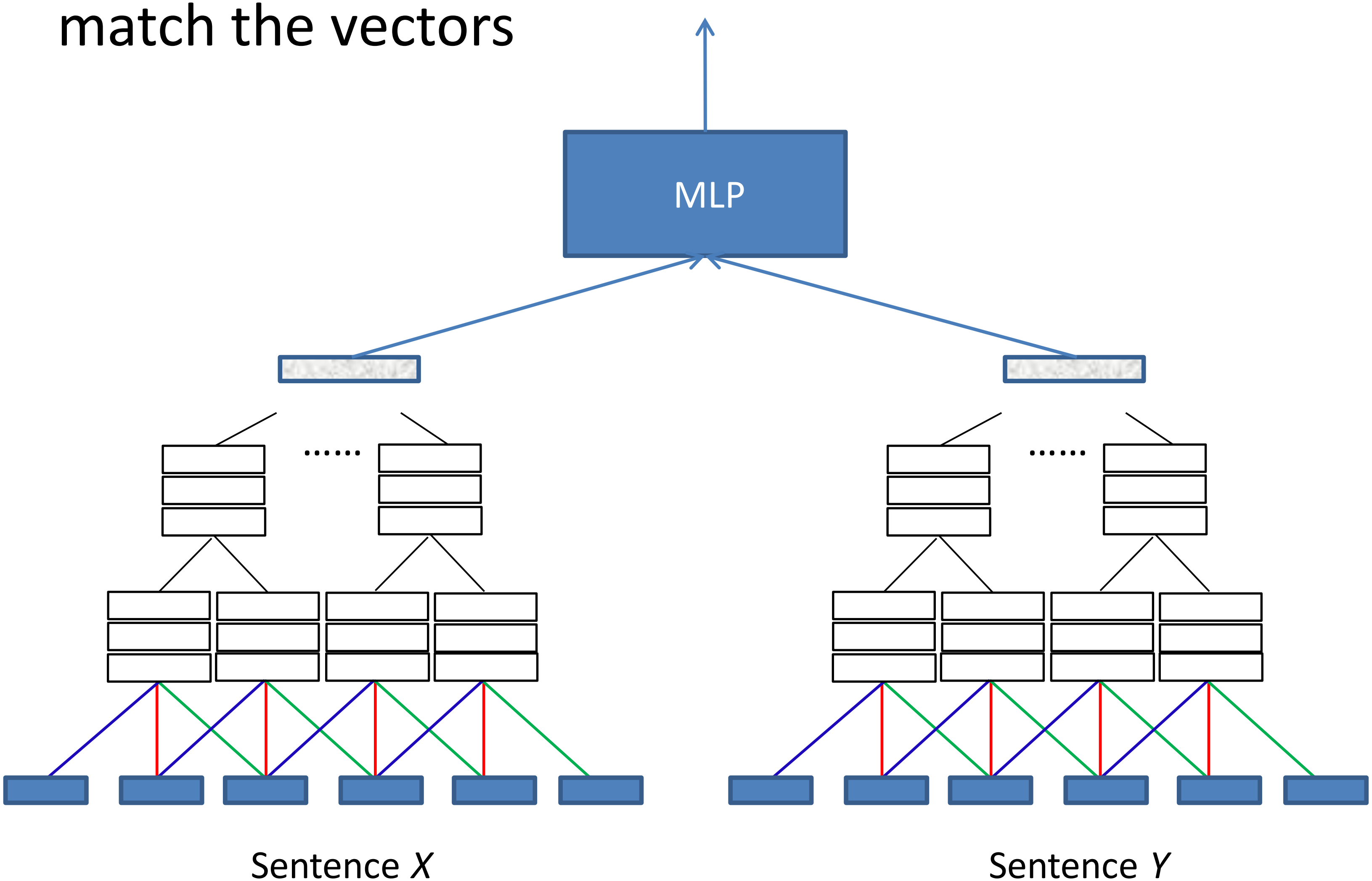


Retrieval based Question Answering System



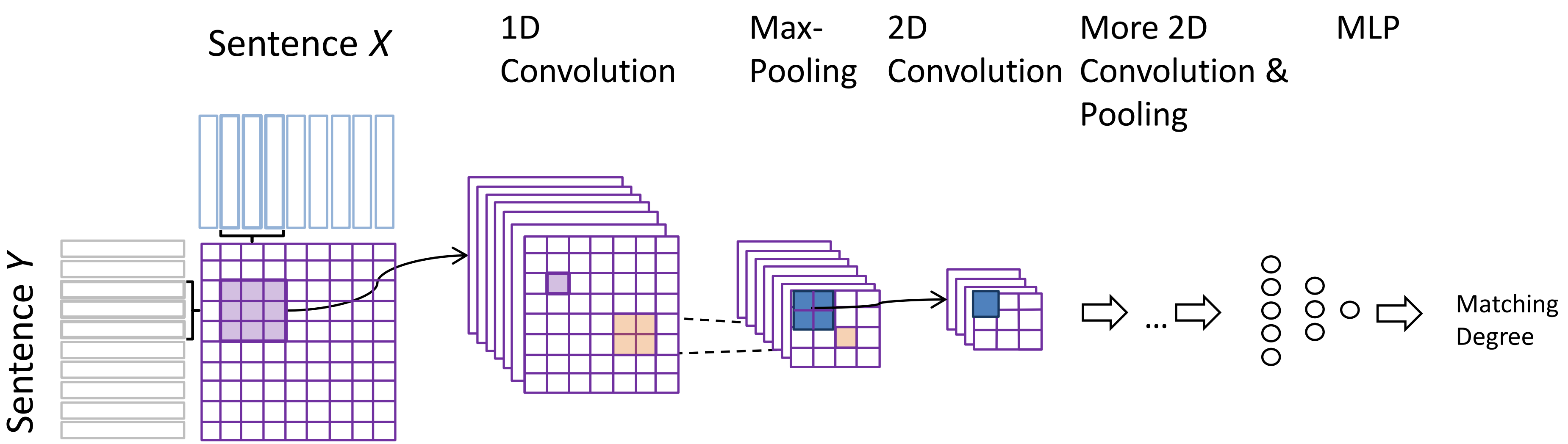
Deep Match CNN
- Architecture I

- First represent two sentences as vectors, and then match the vectors



Deep Match CNN
- Architecture II

- Represent and match two sentences simultaneously
- Two dimensional model



Experimental Results

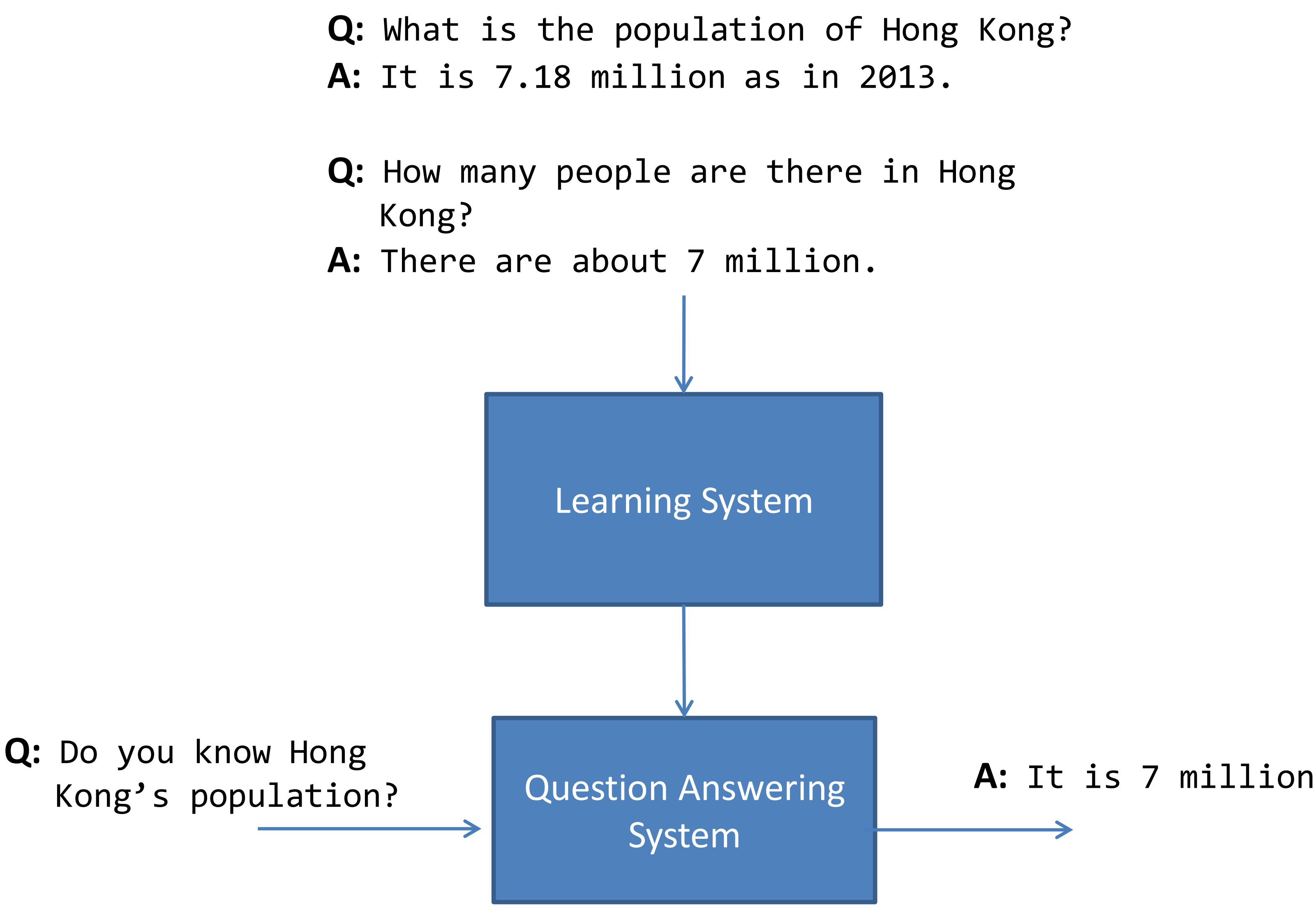
- Experiment
  - 4.4 million Weibo data (Chinese)
  - 70% of responses are appropriate as replies

	Accuracy
Word Embedding	54.3
SENNA + MLP	56.5
Deep Match CNN 1-dim	59.2
Deep Match CNN 2-dim	62.0
Whole System	70.0

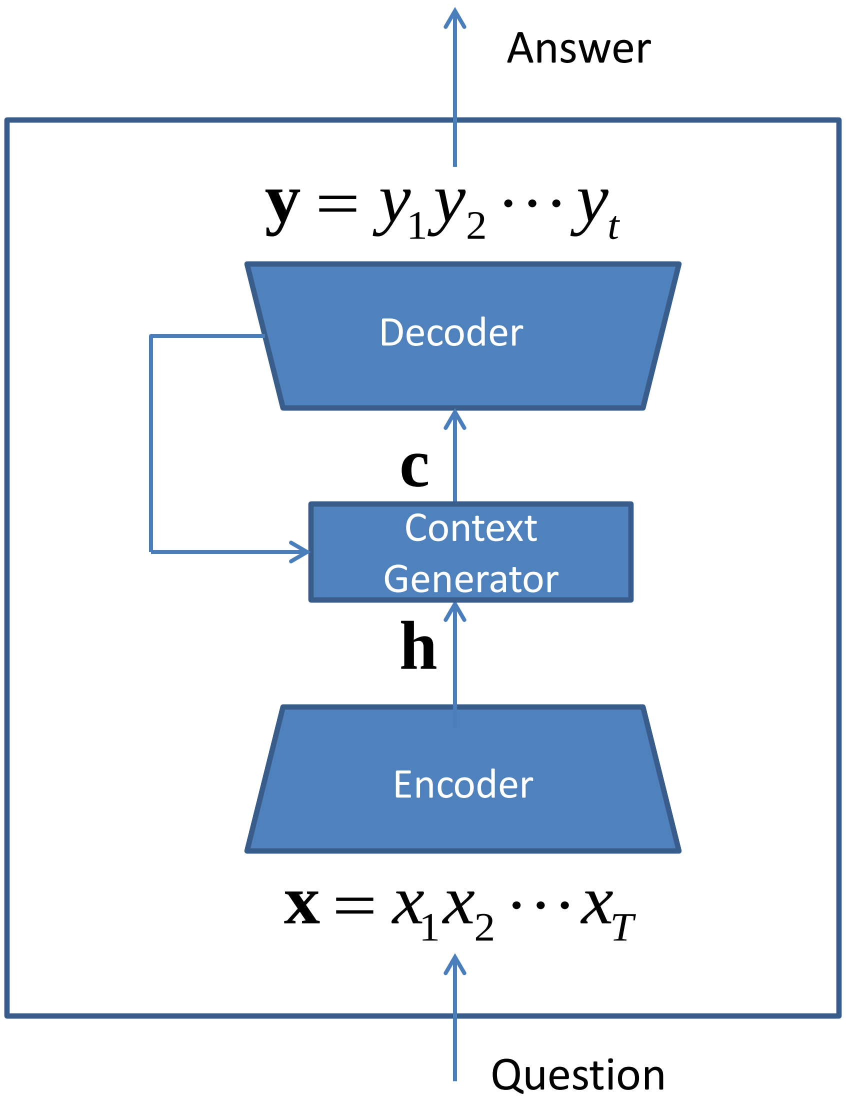
Generation based Question Answering

Shang et al. 2015

Generation-based Question Answering

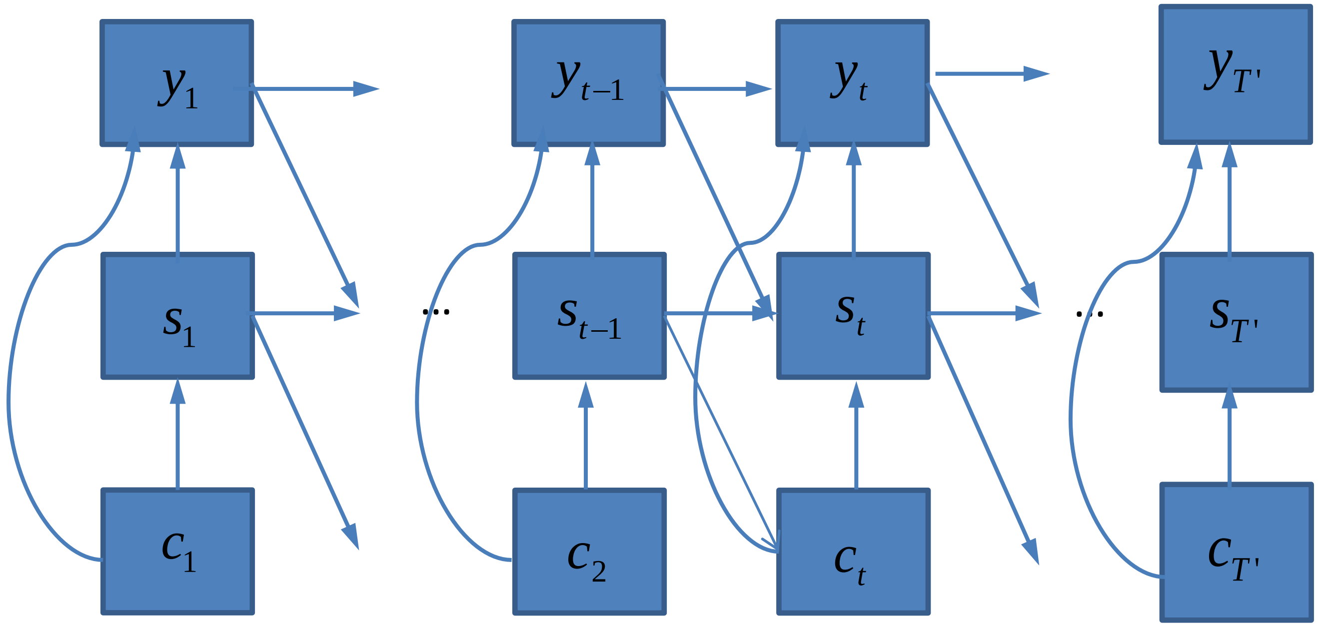


Neural Responding Machine



- Encoding questions to internal representations
- Decoding internal representations to answers
- Using GRU

Decoder



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

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

y_t is one-hot vector

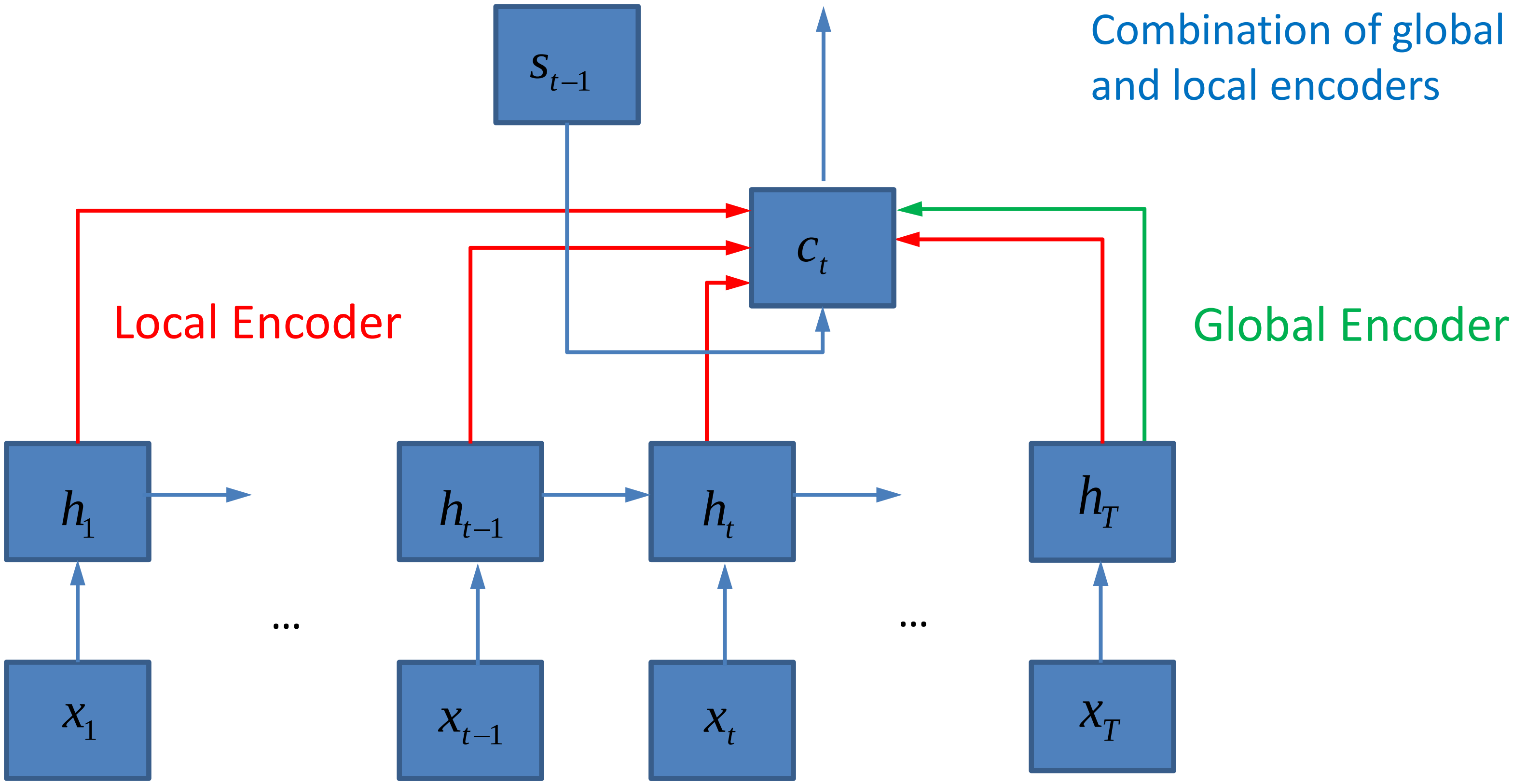
s_t is hidden state of decoder

c_t is context vector

$g()$ is softmax function, $f()$ is GRU

Similar to attention mechanism in RNN Encoder-Decoder

Encoder



$$c_t = \sum_{j=1}^{T} \alpha_{tj} \left[h_j^l : h_T^g\right], \alpha_{tj} = q(h_j, s_{t-1})$$

c_t is context vector, α_{tj} is weight

$\left[h_j^l : h_T^g\right]$ is concatenation of local and global hidden states

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

x_t is word embedding

h_t is hidden state of encoder

$f()$ is GRU

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.

Question Answering from Relational Database

Yin et al. 2016

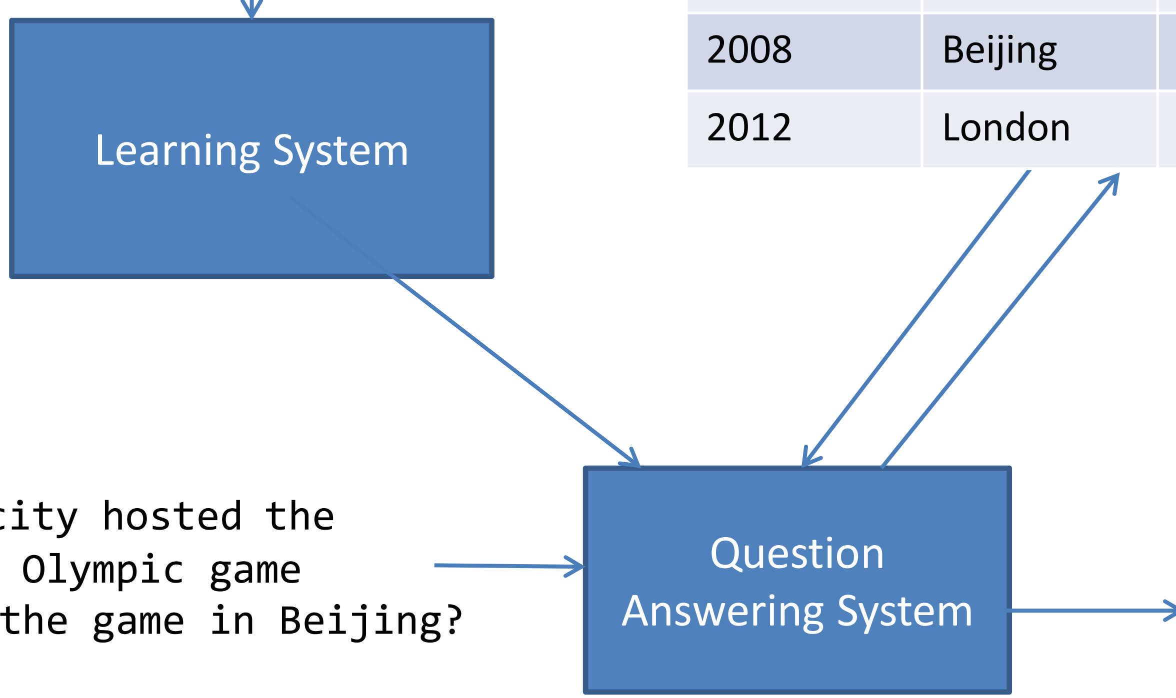
Question Answering from Relational Database

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



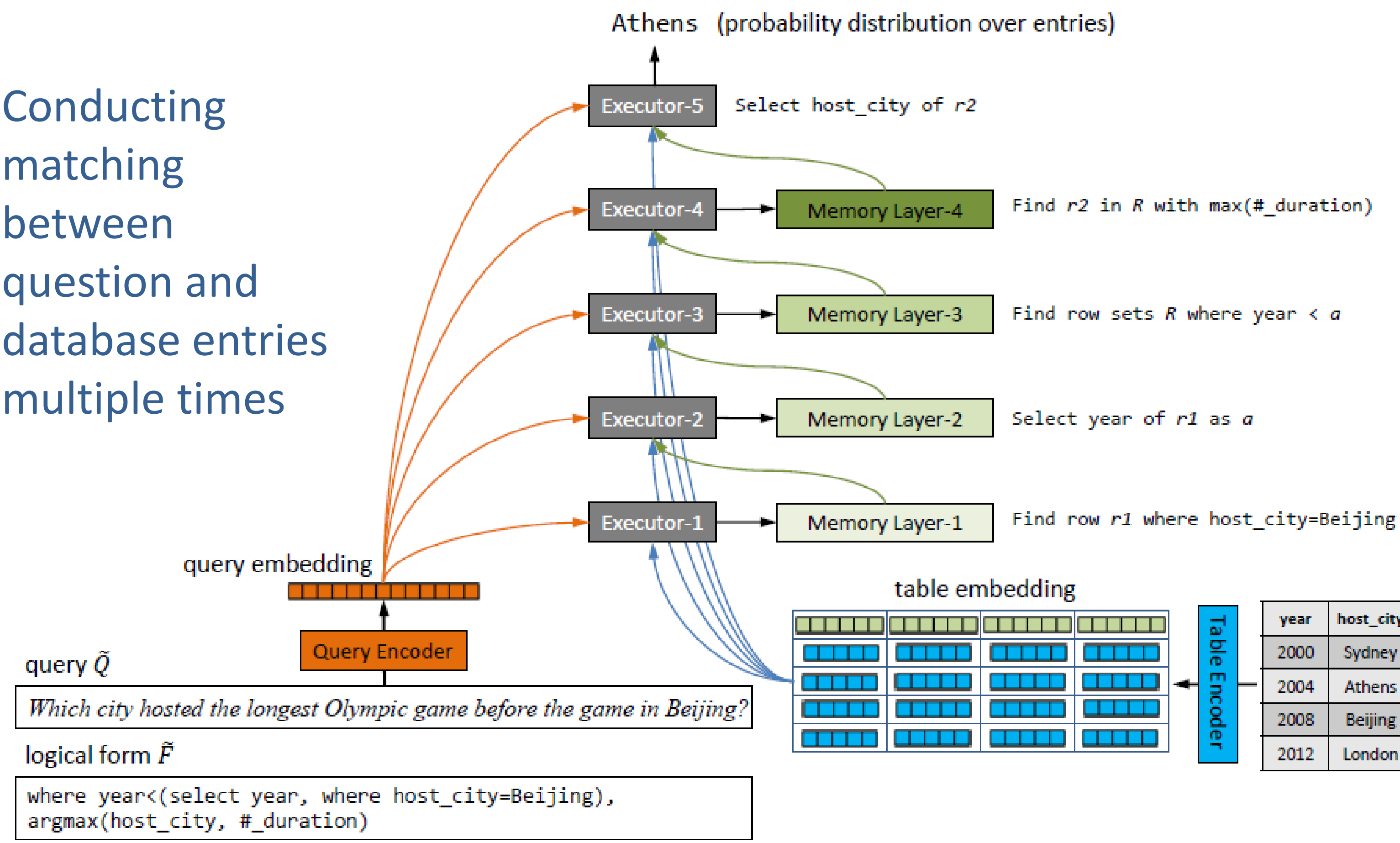
Q: `Which city hosted the longest Olympic game before the game in Beijing?`

A: `Athens`

Neural Enquirer

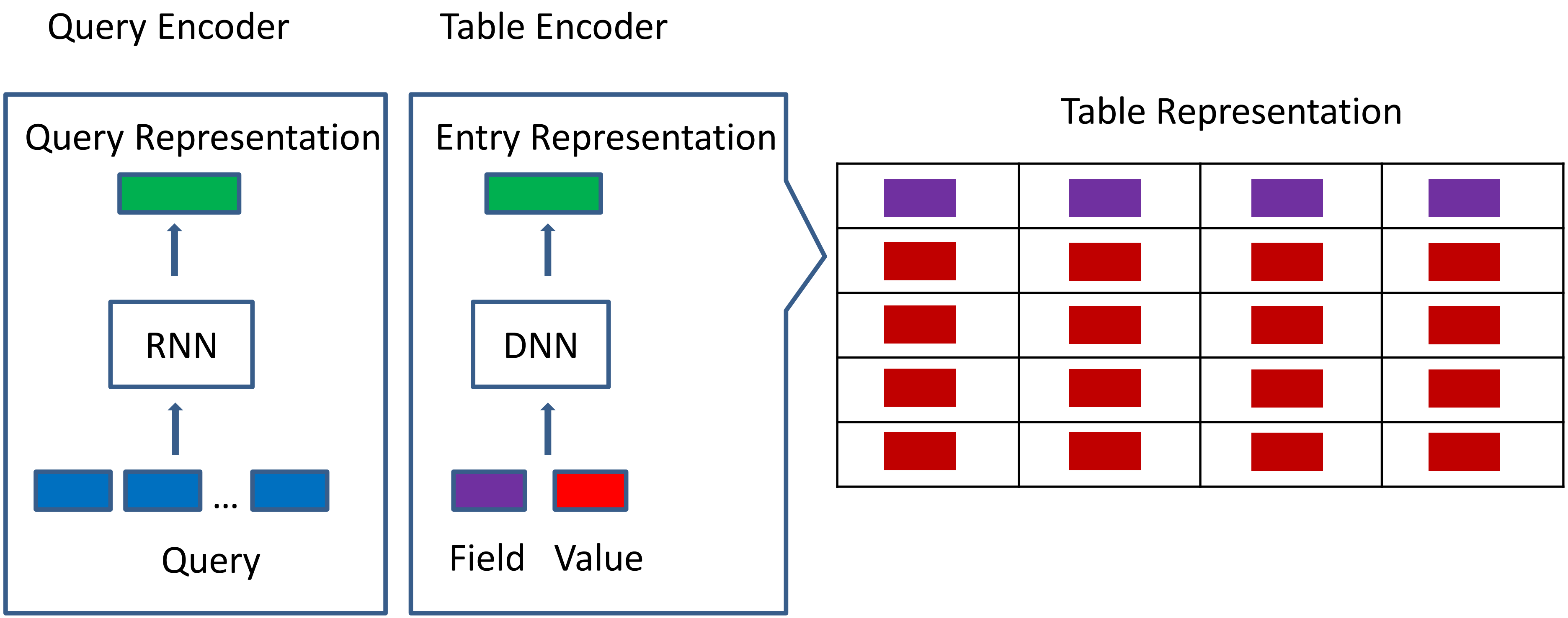
- Query Encoder: encoding query
- Table Encoder: encoding entries in table
- Five Executors: executing query against table

Conducting matching between question and database entries multiple times



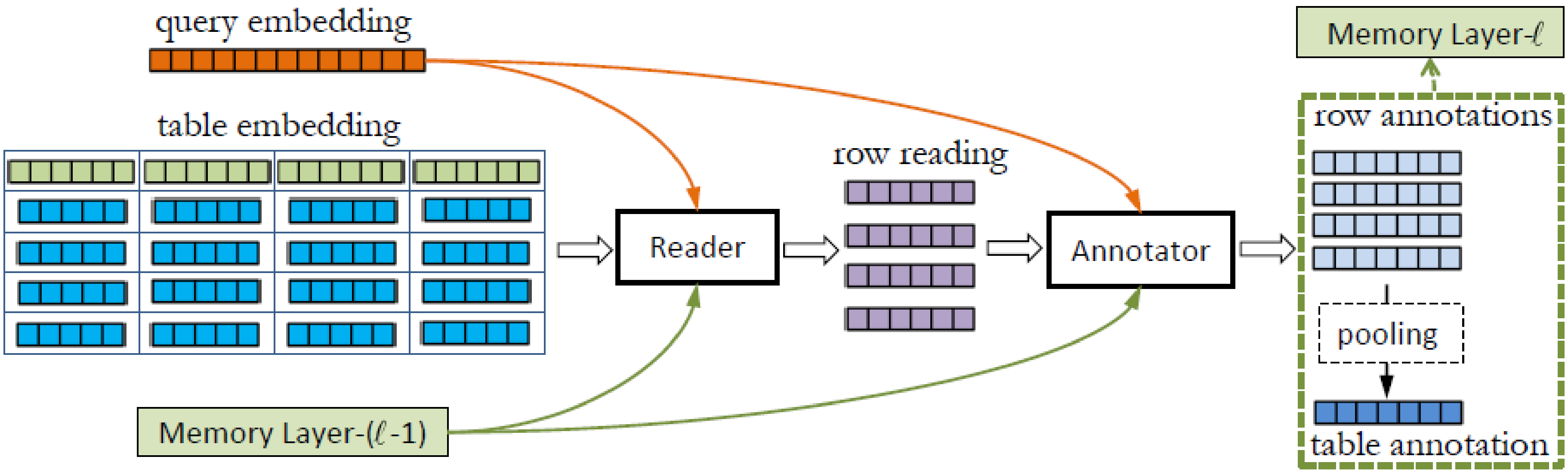
year	host_city	#_duration	#_medals
2000	Sydney	20	2,000
2004	Athens	35	1,500
2008	Beijing	30	2,500
2012	London	40	2,300

Query Encoder and Table Encoder



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

Executors



Select #_participants where city = beijing

- 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%

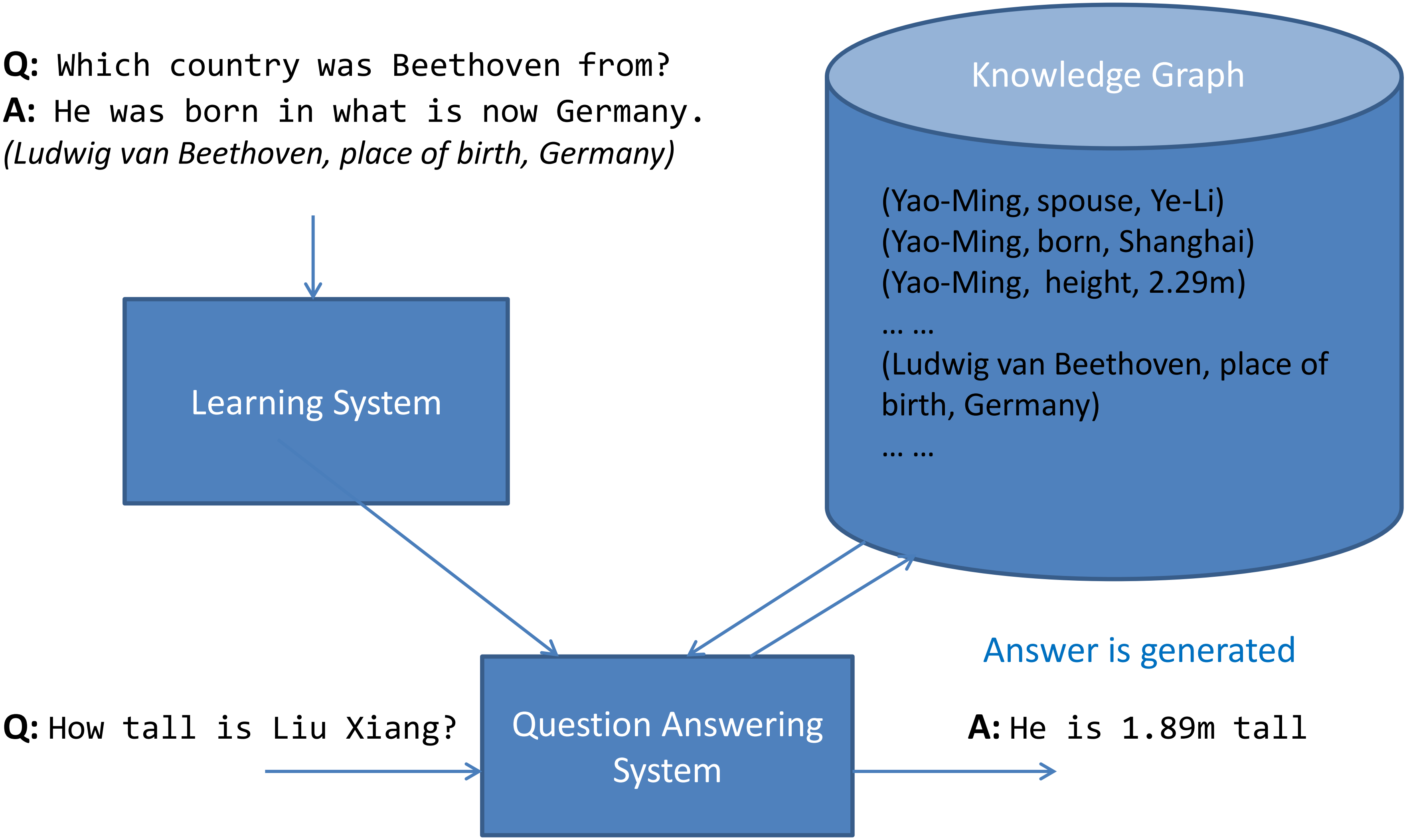
Question Answering from Knowledge Graph

Yin et al. 2016

Question Answering from Knowledge Graph

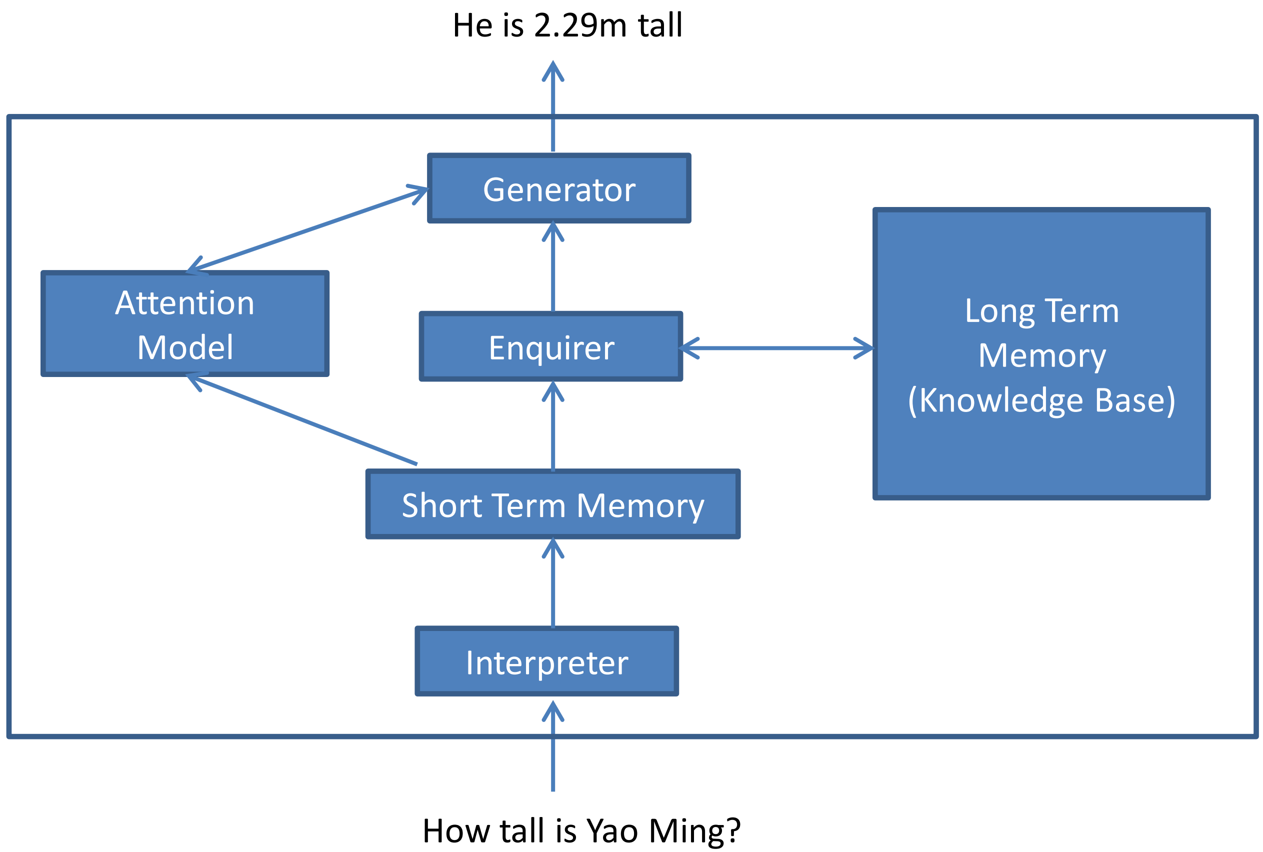
Q: `How tall is Yao Ming?`
A: `He is 2.29m tall and is visible from space.`
(Yao Ming, height, 2.29m)

Q: `Which country was Beethoven from?`
A: `He was born in what is now Germany.`
(Ludwig van Beethoven, place of birth, Germany)



GenQA

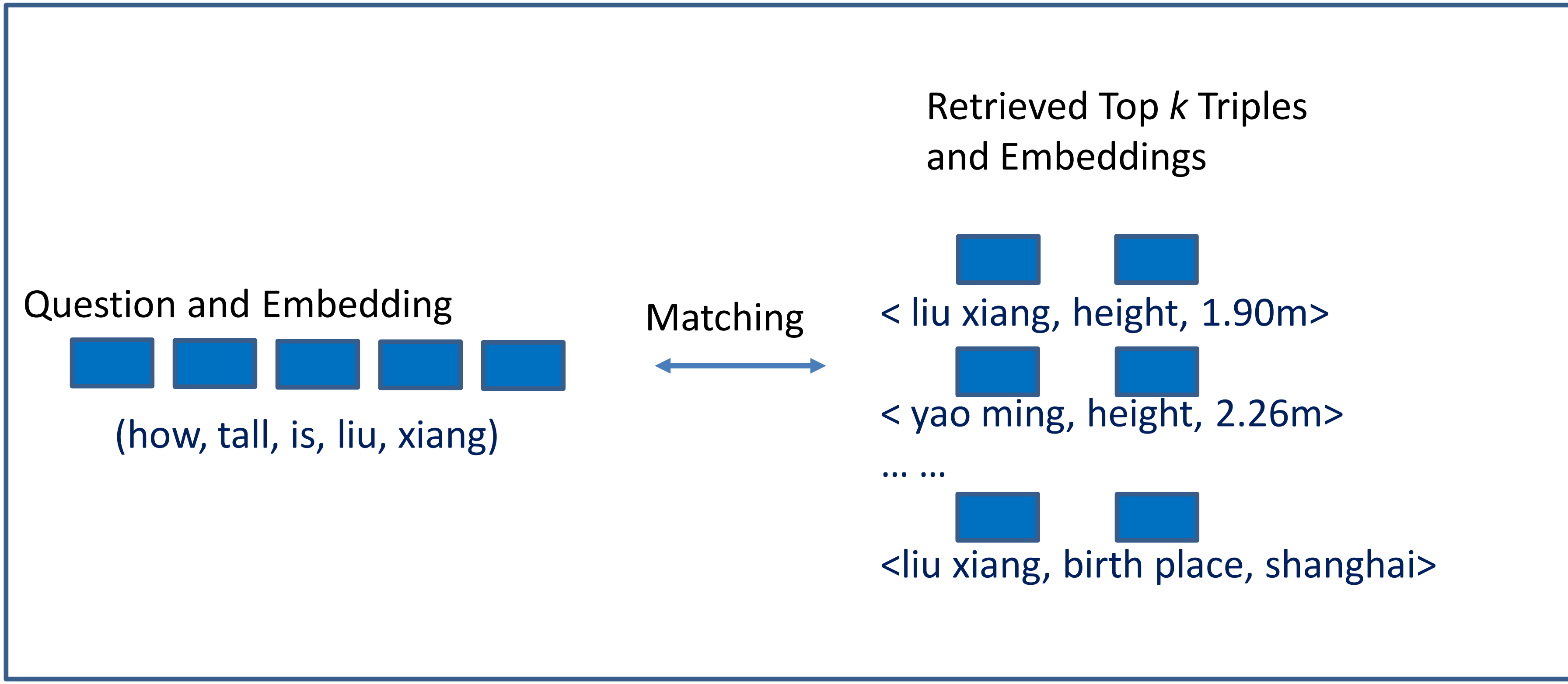
- **Interpreter:** creates representation of question using RNN
- **Enquirer:** retrieves top k triples with highest matching scores using CNN model
- **Generator:** generates answer based on question and retrieved triples using attention-based RNN
- **Attention model:** controls generation of answer



Key idea:

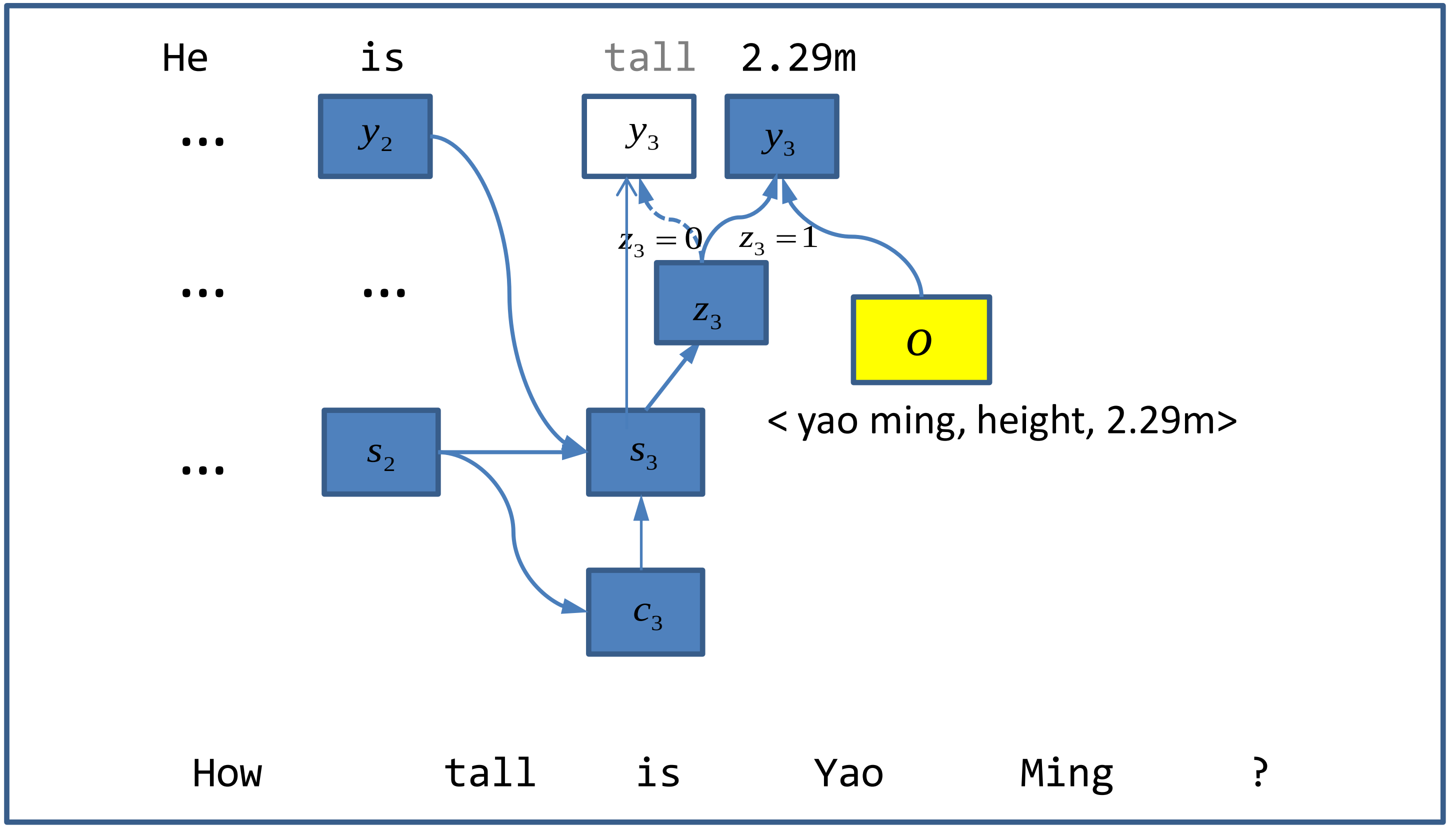
- Generation of answer based on question and retrieved result
- Combination of neural processing and symbolic processing

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

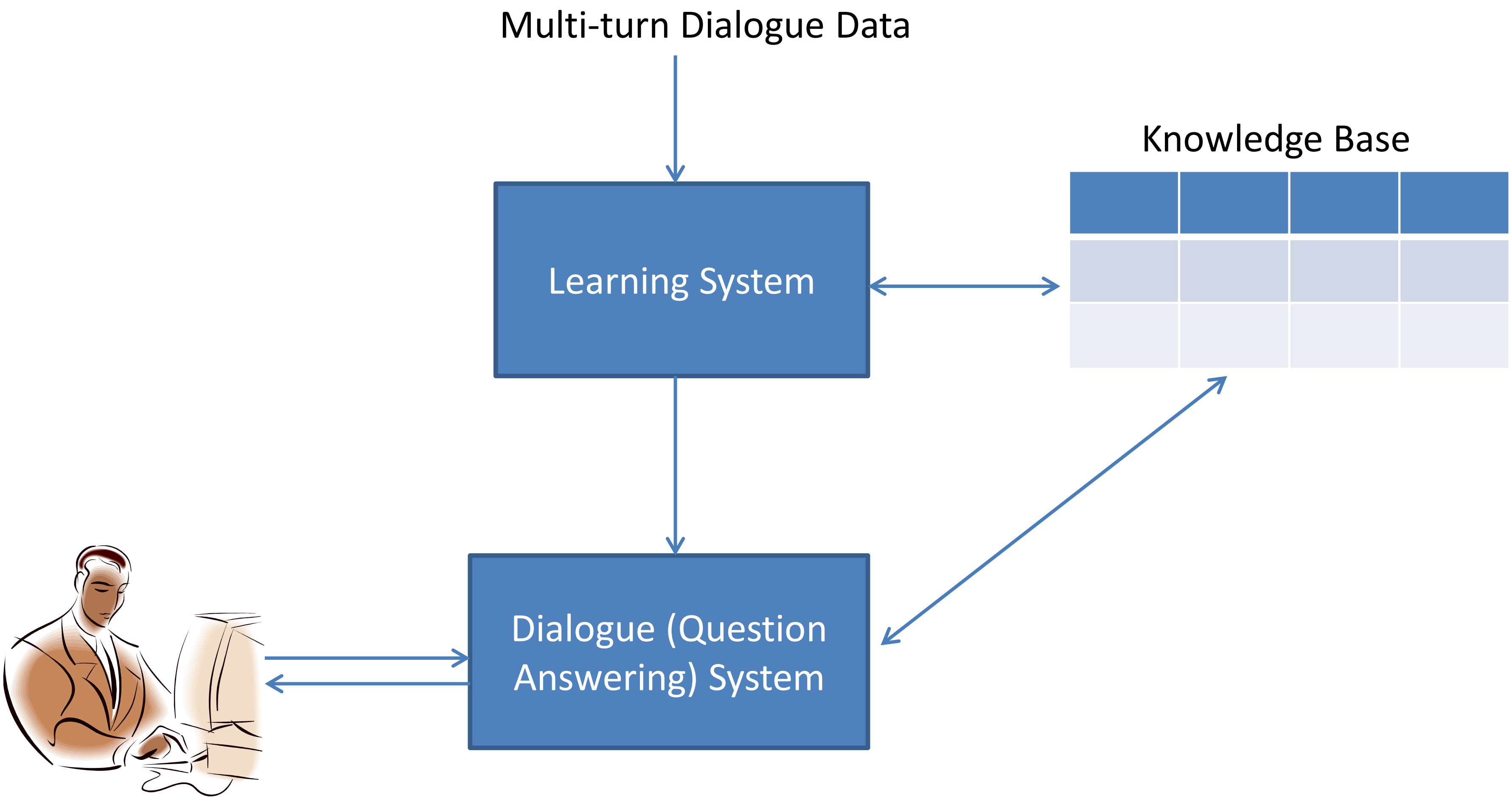
- Experiment
  - Trained with 720K question-answer pairs (Chinese) associated with 1.1M triples in knowledge-base, *data is noisy*
  - Accuracy = 52%
  - Data is still noisy

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

Multi-turn Dialogue

Wen et al. 2016

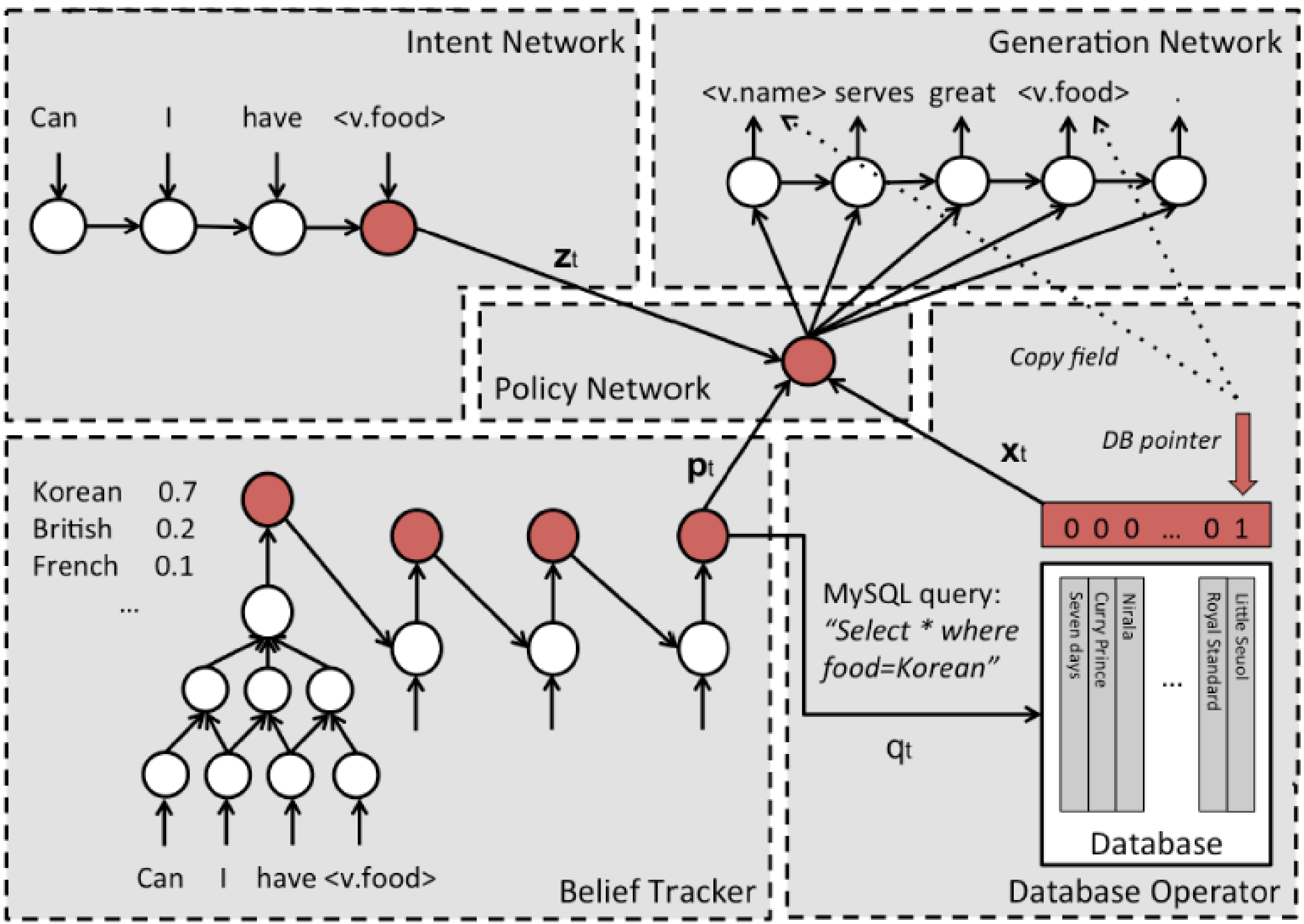
Multi-turn Dialogue (Question Answering) System



Neural Network based Multi-turn Dialogue System

- Neural network model for entire task
- Combination of symbolic processing and neural processing
- End-to-end trainable, but modularly connected
  - Intent Network: neural processing
  - Belief Trackers: symbolic and neural processing
  - Database Operator: symbolic processing
  - Policy Network: neural processing
  - Generation Network: neural and symbolic processing

System Architecture



Functions of Modules

- Intent Network
  - Creating *question representation* from user question, using CNN or LSTM
- Belief Trackers
  - Each database attribute has one tracker
  - Each tracker calculates *probabilities of values for the attribute*, based on dialog so far, using RNN
- Database Operator
  - Querying database using belief trackers' updates, and obtaining result
- Policy Network
  - Creating *action representation*, based on question representation, belief trackers' updates, database operator's update, using DNN
- Generation Network
  - Generating answer, based on action representation, using LSTM

Example of Dialogue

- Human : restaurant in any area that serves vietnamese food
- Machine : thanh binh serves vietnamese food type , is there anything else i can help you with ?
- Human : what is the phone number and post code
- Machine : the number is 01223 362456 and the area code is c.b 3, 0 a.f . is there anything else i can do for you ?
- Human : good bye
- Machine : thank you for using the cambridge restaurant system .

Experimental Results

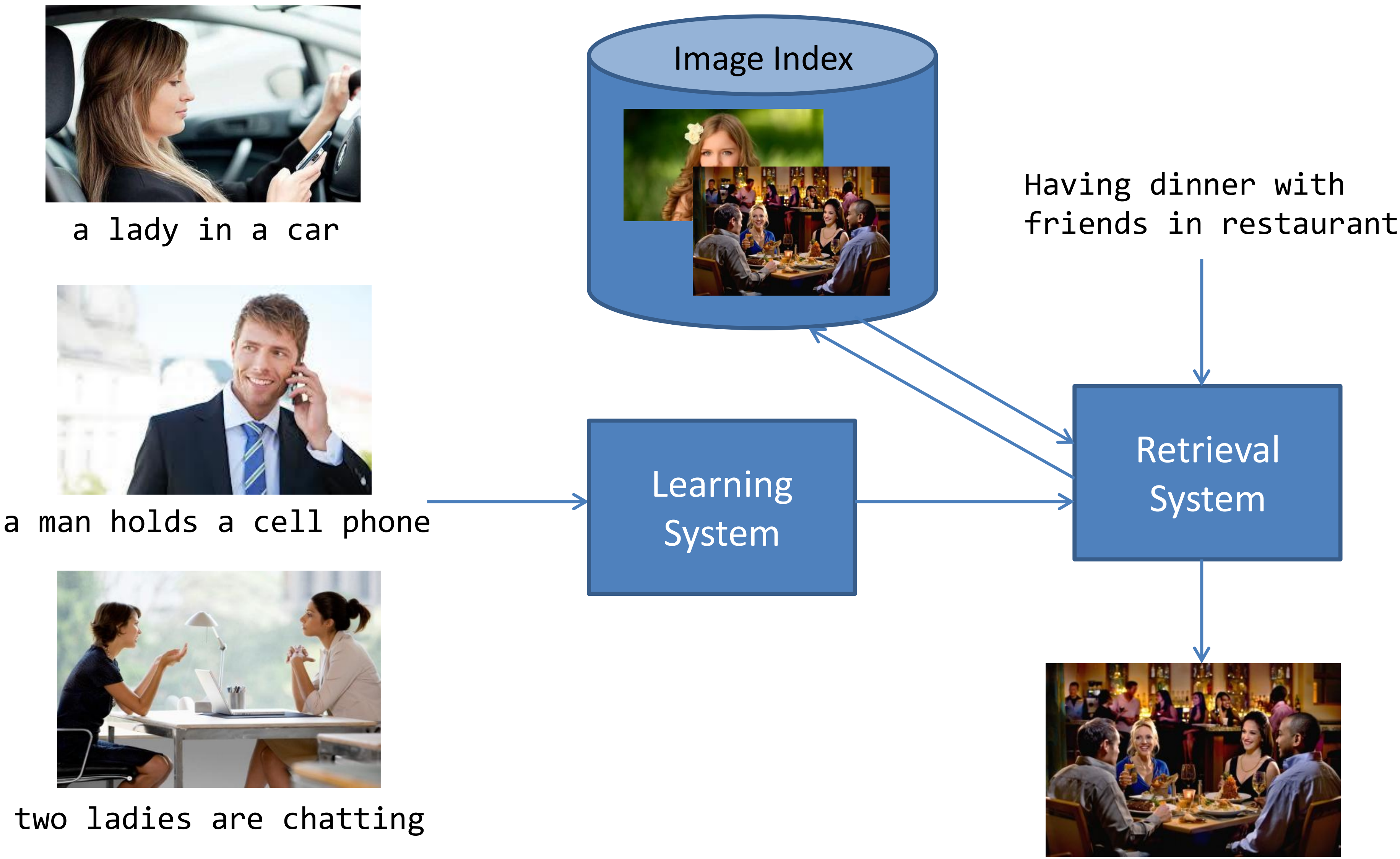
- Experiment
  - Database of 99 restaurants, 6 slots
  - 680 dialogues
  - Human evaluation with 245 dialogues

	Neural Network	Rule-based
Success rate	96.95	95.12
Avg. # of turns	3.95	4.54

Image Retrieval

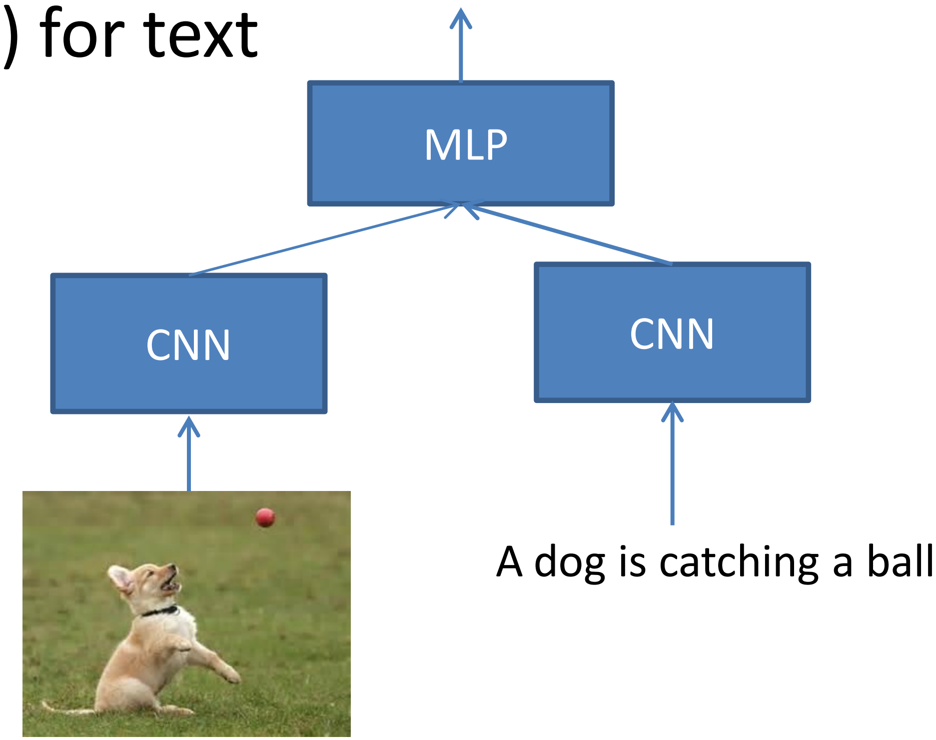
Ma et al. 2015

Image Retrieval

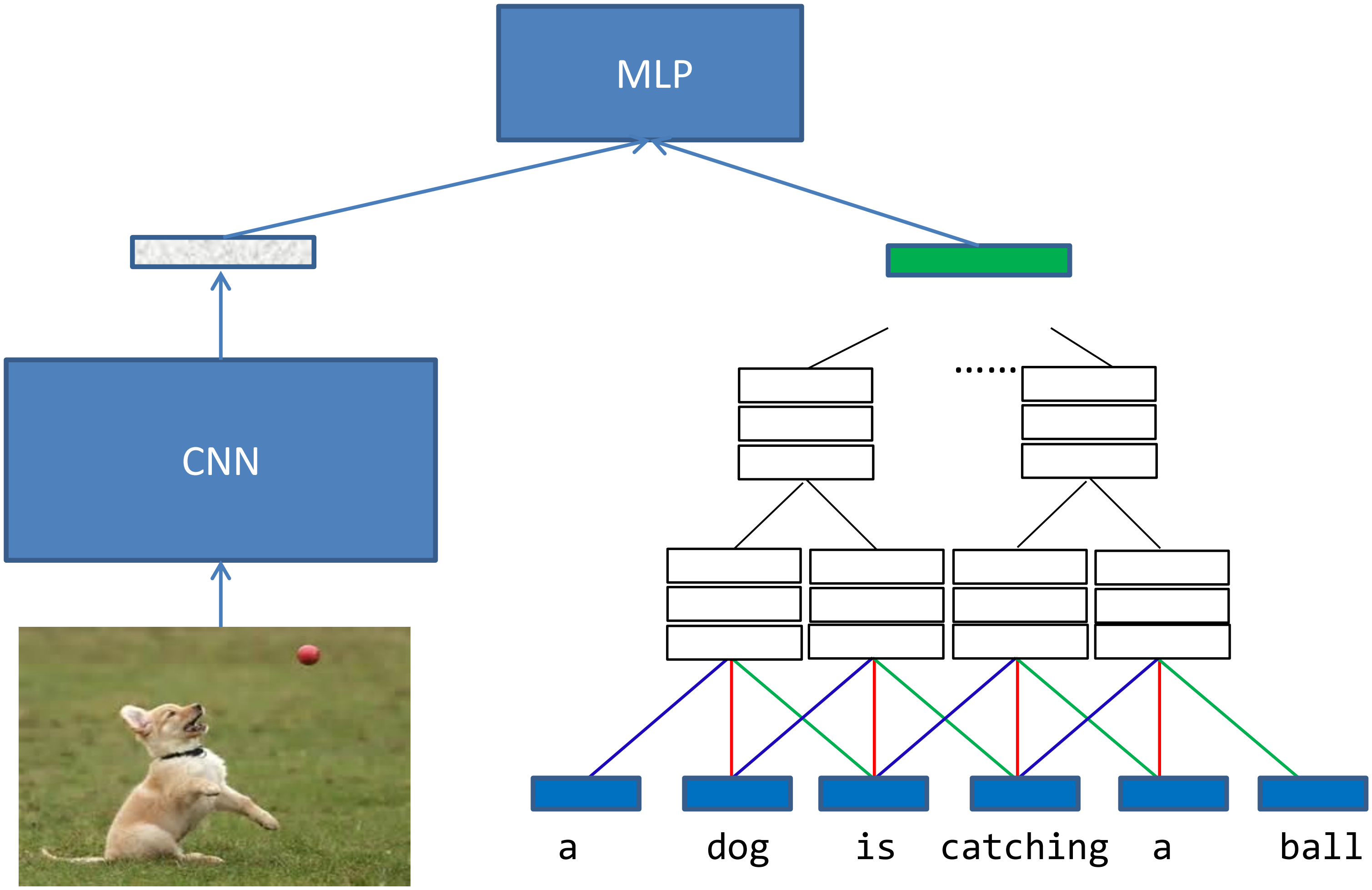


Multimodal CNN

- Represent text and image as vectors and then match the two vectors
- Word-level matching, phrase-level matching, sentence-level matching
- CNN model works better than RNN models (state of the art) for text

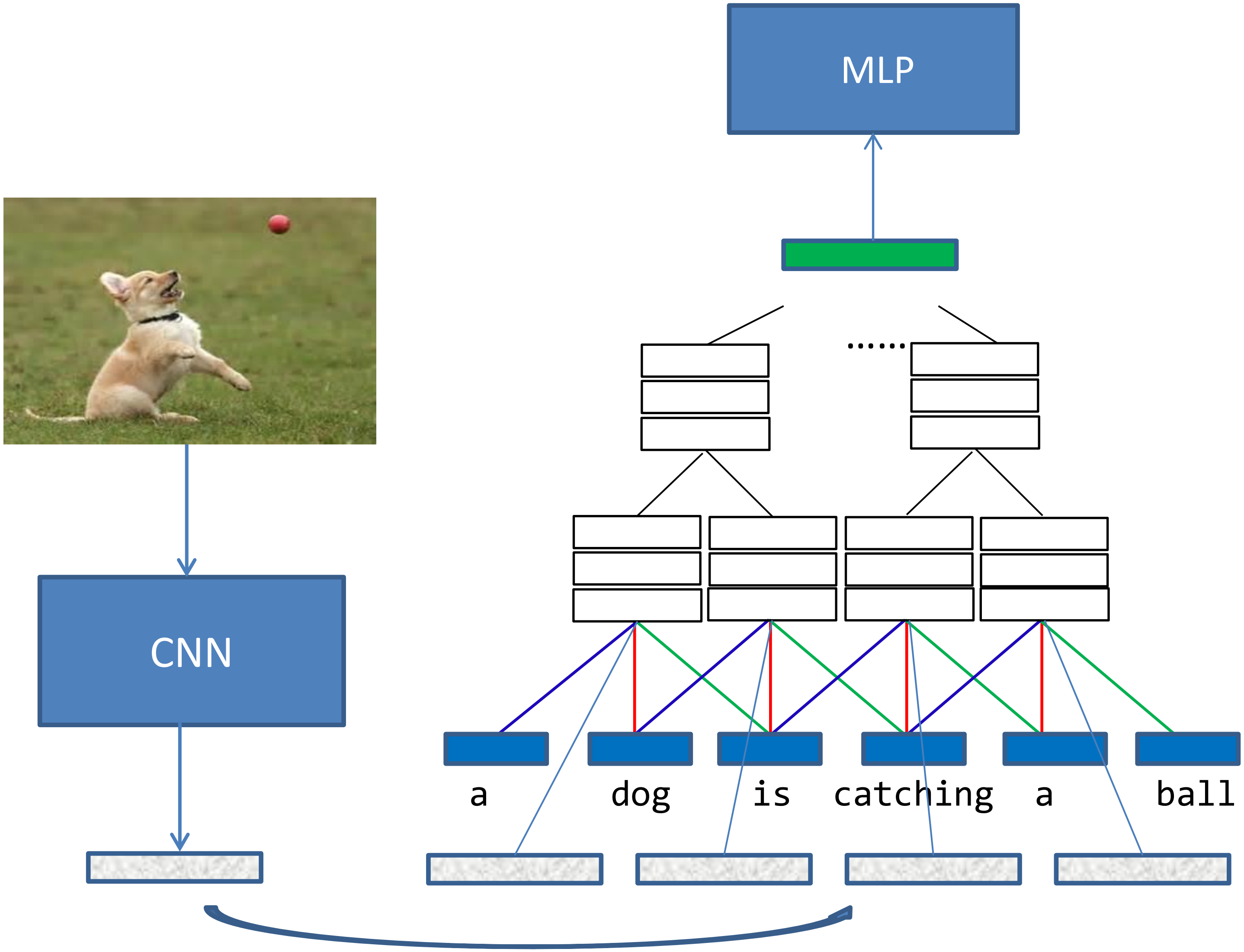


Sentence-level Matching



- Combing image vector and sentence vector

Word-level Matching Model



- Adding image vector to word vectors

Experimental Results

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

	R@1	R@5	R@10
MNLM-VGG	12.5	37.0	51.5
DVSA (BRNN)	15.2	37.7	50.5
NIC	17.0	NA	57.0
M-RNN-VGG	22.8	50.7	63.1
M-CNN	**26.2**	**56.3**	**69.6**

References

- P.-S. Huang, X. He, J. Gao, L. Deng, A. Acero, and L. Heck. Learning Deep Structured Semantic Models for Web Search Using Clickthrough Data. *CIKM* 2013.
- B. Hu, Z. Lu, H. Li, and Q. Chen. Convolutional Neural Network Architectures for Matching Natural Language Sentences. *NIPS* 2014.
- Z. Ji, Z. Lu, and H. Li. An Information Retrieval Approach to Short Text Conversation, *arXiv:*1408.6988, 2014.
- A. Severyn, and A. Moschitti. Learning to Rank Short Text Pairs with Convolutional Deep Neural Networks. *SIGIR* 2015..
- L. Shang, Z. Lu, H. Li. Neural Responding Machine for Short Text Conversation. *ACL* 2015.
- P. Yin, Z. Lu, H. Li, B. Kao. Neural Enquirer: Learning to Query Tables. *IJCAI* 2016.
- J. Yin, X. Jiang, Z. Lu, L. Shang, H. Li, X. Li. Neural Generative Question Answering. *IJCAI* 2016.
- L. Ma, Z. Lu, L. Shang, Hang Li . Multimodal Convolutional Neural Networks for Matching Image and Sentence. *ICCV* 2015.
- T.-H. Wen, M. Gasic, N. Mrksic, L. M. Rojas-Barahona, P.-H. Su, S. Ultes, D. Vandyke, and S. Young. A Network-based End-to-End Trainable Task-oriented Dialogue System. *arXiv:*1604.04562, 2016.

References (Question Answering)

- A. Bordes, J. Weston, and S. Chopra. Question Answering with Subgraph Embeddings. *EMNLP* 2014.
- M . Iyyer, J. L. Boyd-Graber, L. Max, B. Claudino, R. Socher, and H. Daumé III. A Neural Network for Factoid Question Answering over Paragraphs. *EMNLP* 2014.
- L. Dong, F. Wei, M. Zhou, and K. Xu. Question Answering over Freebase with Multi-Column Convolutional Neural Networks. *ACL* 2015.
- J. Weston, S. Chopra, and A. Bordes. Memory Networks. *ICLR* 2015.
- A. Bordes, N. Usunier, S. Chopra, and J. Weston. Large-Scale Simple Question Answering with Memory Networks. *arXiv:*1506.02075, 2015.
- T.-H. Wen, M. Gasic, N. Mrksic, P.-H. Su, S. Ultes, D. Vandyke, and S. Young. Semantically Conditioned lSTM-based Natural Language Generation for Spoken Dialogue Systems. *arXiv:*1508.01745, 2015.
- A. Neelakantan, Q. V. Le, and I. Sutskever. Neural Programmer: Inducing Latent Programs with Gradient Descent. *ICLR 2016*.
- L. V. Serban, A. Sordoni, Y. Bengio, A. Courville, and J. Pineau. Building End-to-End Dialogue Systems Using Generative Hierarchical Neural Network Models. *AAAI* 2016.
- J. Andreas, M. Rohrbach, T. Darrell, D. Klein. Learning to Compose Neural Networks for Question Answering. *NAACL*, 2016.
- Z. Dai, L. Li and W. Xu, CFO: Conditional Focused Neural Question Answering with Large-scale Kowledge Graphs. *ACL* 2016.

References (Image Retrieval)

- A. Frame, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, M. A. Ranzato, and T. Mikolov. Devise: A Deep Visual-Semantic Embedding Model. *NIPS* 2013.
- A. Karpathy, A. Joulin, and F. Li. Deep Fragment Embeddings for Bidirectional Image Sentence Mapping. *NIPS* 2014.
- R. Kiros, R. Salakhutdinov, and R. S. Zemel. Unifying Visual-Semantic Embeddings with Multimodal Neural Language Models. arXiv:1411.2539, 2014.
- J. Mao, W. Xu, Y. Yang, J. Wang, and A. L. Yuille. Deep Captioning with Multimodal Recurrent Neural Networks. arXiv:1412.6632, 2014.
- O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and Tell: a Neural Image Caption Generator. arXiv:1411.4555, 2014.
- R. Socher, Q. V. L. A. Karpathy, C. D. Manning, and A. Y. Ng. Grounded Compositional Semantics for Finding and Describing Images with Sentences. *TACL* 2014.
- A. Karpathy and F. Li. Deep Visual-Semantic Alignments for Generating Image Descriptions. *CVPR* 2015.

Summary

Summary

- Fundamental IR problems
  - Matching
  - Translation
  - Classification
  - Structured Prediction
- Matching is important issue for IR
- DL can learn better representations for matching and other problems
- Useful DL tools
  - Word Embedding
  - Recurrent Neural Networks
  - Convolutional Neural Networks

Summary (cont')

- Recent progress made in IR tasks
  - Document Retrieval
  - Retrieval-based Question Answering
  - Generation-based Question Answering
  - Question Answering from Knowledge Graph
  - Question Answering from Database
  - Multi-turn Dialogue
  - Image Retrieval
- DL is particularly effective for hard IR problems

Open Question for Future Research

- How to combine symbolic processing and neural processing
- Advantage of symbolic processing: direct, interpretable, and easy to control
- Advantage of neural processing: flexible, robust, and automatic
- Challenge: difficult to make the combination

Acknowledgement

- We thank Xin Jiang, Xi Zhang, Lin Ma , Jun Xu, Shengxian Wan, Liang Peng for providing references for this tutorial

Paper of This Tutorial: Hang Li, Zhengdong Lu, Deep Learning for Information Retrieval, in Proceedings of SIGIR 2016

hangli.hl@huawei.com
lu.zhengdong@huawei.com

Thank you!