中国科学院计算技术研究所
INSTITUTE OF COMPUTING TECHNOLOGY, CHINESE ACADEMY OF SCIENCES

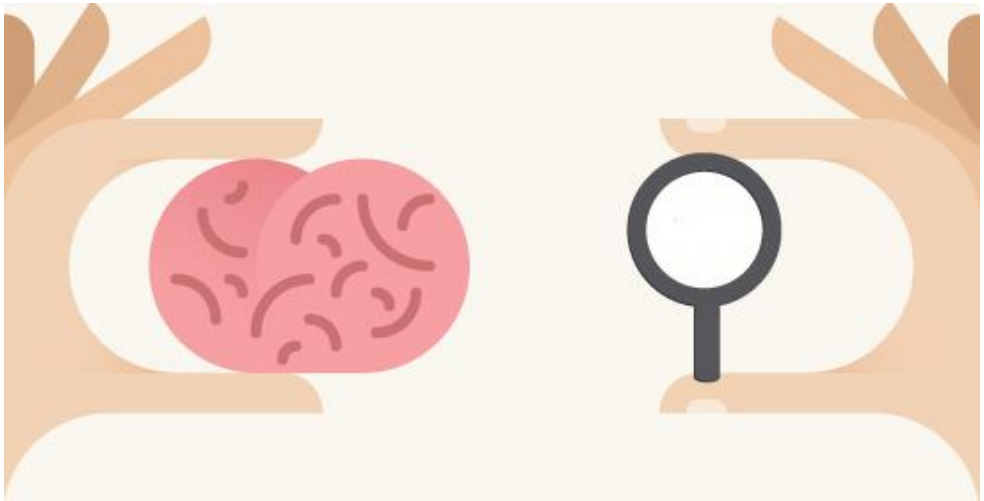
中国科学院网络数据科学与技术重点实验室
Key Laboratory of Network Data Science & Technology ,CAS

Neural Models for Information Retrieval Part II

Jiafeng Guo (郭嘉丰), Researcher

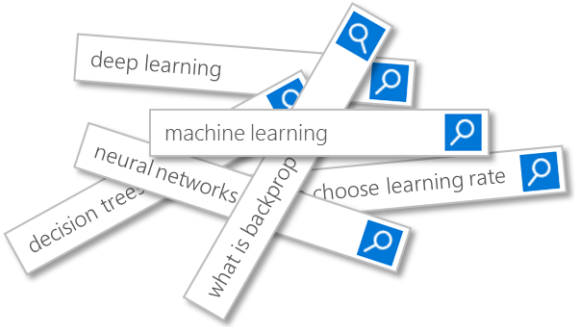
Institute of Computing Technology, Chinese Academy of Sciences

Homepage: www.bigdatalab.ac.cn/~gjf



What does ad-hoc IR data look like?

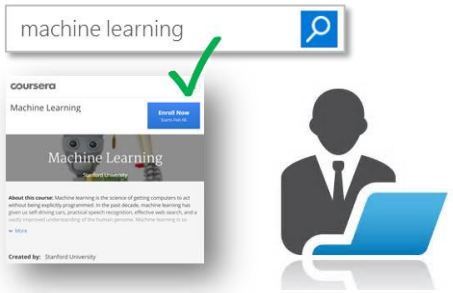
- Traditional IR uses human labels as ground truth for evaluation
- So ideally we want to train our ranking models on human labels
- User interaction data is rich but may contain different biases compared to human annotated labels



search queries

document corpus

user interaction / click data

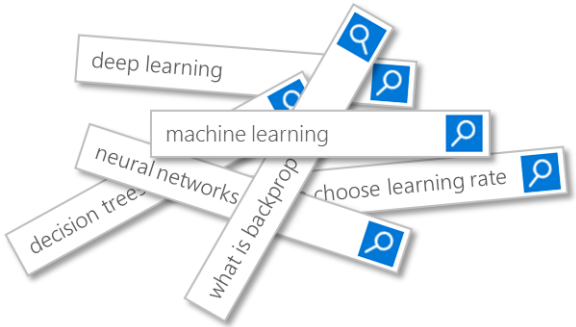


human annotated labels

What does ad-hoc IR data look like?

In industry:

- Document corpus: billions?
- Query corpus: many billions?
- Labelled data w/ raw text: hundreds of thousands of queries
- Labelled data w/ learning-to-rank style features: same as above
- User interaction data: billions?



search queries

document corpus

user interaction / click data

human annotated labels

Success stories in industry

DSSM

Established: January 30, 2015

The goal of this project is to develop a class of deep representation learning models. DSSM stands for Deep Structured Semantic Model, or more general, Deep Semantic Similarity Model. DSSM, developed by the MSR Deep Learning Technology Center(DLTC), is a deep neural network (DNN) modeling technique for representing text strings (sentences, queries, predicates, entity mentions, etc.) in a continuous semantic space and modeling semantic similarity between two text strings (e.g., Sent2Vec). DSSM has wide applications including information retrieval and web search ranking (Huang et al. 2013; Shen et al. 2014a,2014b), ad selection/relevance, contextual entity search and interestingness tasks (Gao et al. 2014a), question answering (Yih et al., 2014), knowledge inference (Yang et al., 2014), image captioning (Fang et al., 2014), and machine translation (Gao et al., 2014b) etc. DSSM can be used to develop latent semantic models that project entities of different types (e.g., queries and documents) into a common low-dimensional semantic space for a variety of machine learning tasks such as ranking and classification. For example, in web search ranking, the relevance of a document given a query can be readily computed as the distance between them in that space. With the latest GPUs from Nvidia, we are able to train our models on billions of words. Readers that are interested in deep learning for text processing may refer to our recent tutorial (He et al., 2014).

We released the predictors and trained model files of the DSSM (also a.k.a. Sent2Vec).

People


Po-Sen Huang
Researcher


Hamid Palangi
Associate Researcher II


Xiaodong He
Principal Researcher


Jianfeng Gao
Partner Research Manager

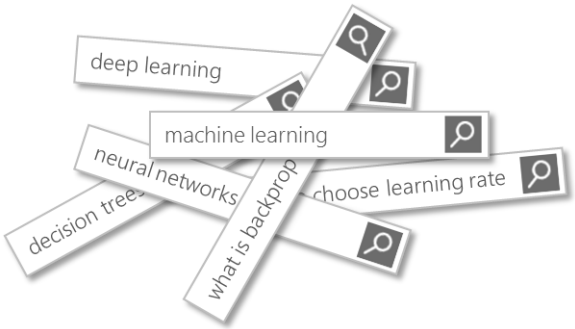
Yelong Shen
Senior RSDE


Scott Wen-tau Yih
Senior Researcher

What does ad-hoc IR data look like?

In academia:

- Document corpus: few billion
- Query corpus: few million but other sources (e.g., wiki titles) can be used
- Labelled data w/ raw text: few hundred to few thousand queries (TREC, LETOR)
- Labelled data w/ learning-to-rank style features: tens of thousands of queries (Yahoo! Challenge)



search queries

document corpus

human annotated labels

As a result…

- Most published neural models for IR are not as deep as those for images or speeches.
- There have not been many significant improvements in Neural IR as compared with traditional LTR.

Levels of Supervision

Unsupervised	Semi-supervised	Fully supervised
• Train embeddings on unlabeled corpus and use in traditional IR models • E.g., GLM, NTLM, DESM	• DNN models using pre-trained embeddings for input text representation • E.g., DRMM, MatchPyramid	• DNNs w/ raw text input (one-hot word vectors or n-graph vectors) trained on labels or click • E.g., DSSM, Duet

We have covered most of these

We will talk about these

Today's Agenda

Part I

- Fundamentals of IR
- Word Representations
- Word Representations for IR

Part II

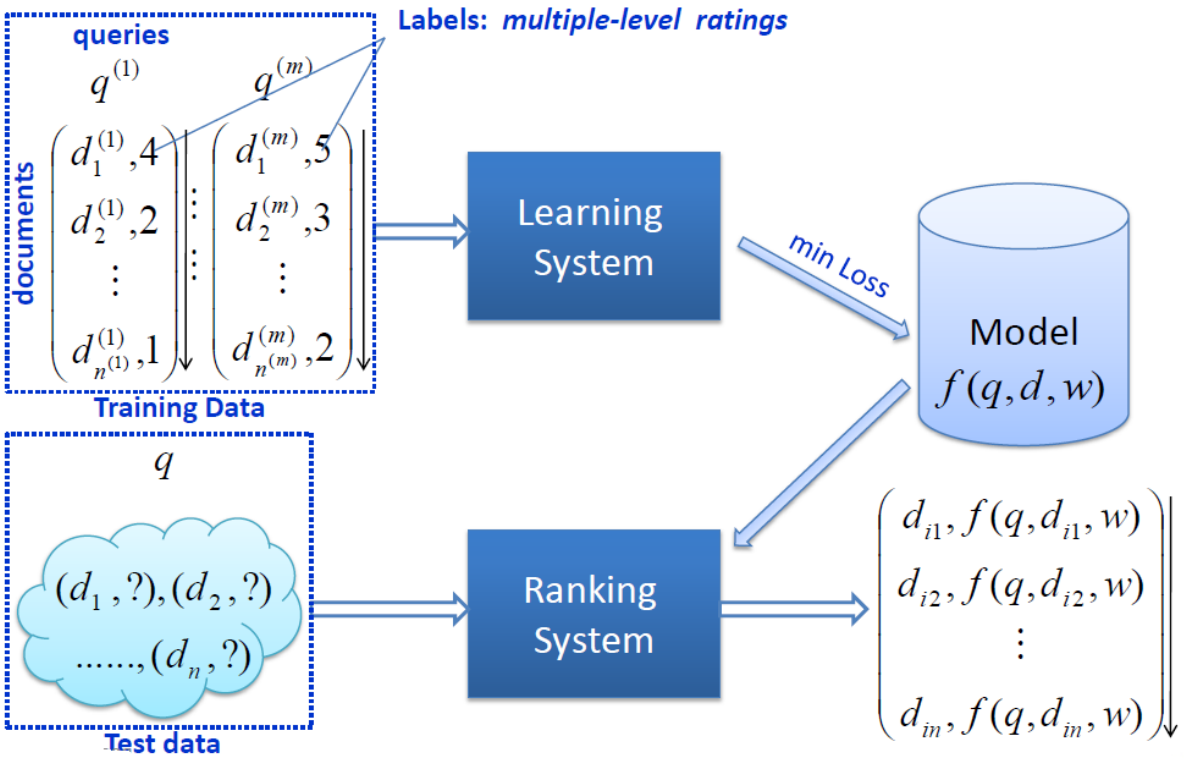
- Supervised learning for rank
- Deep neural nets
- Deep neural nets for IR

Chapter 4

Supervised Learning to Rank

Machine Learning to Rank

- Major Steps
  - Extract matching features from <query, document> pairs
  - Labeling documents according to relevance to the query
  - Learning a ranking function by minimizing a loss function

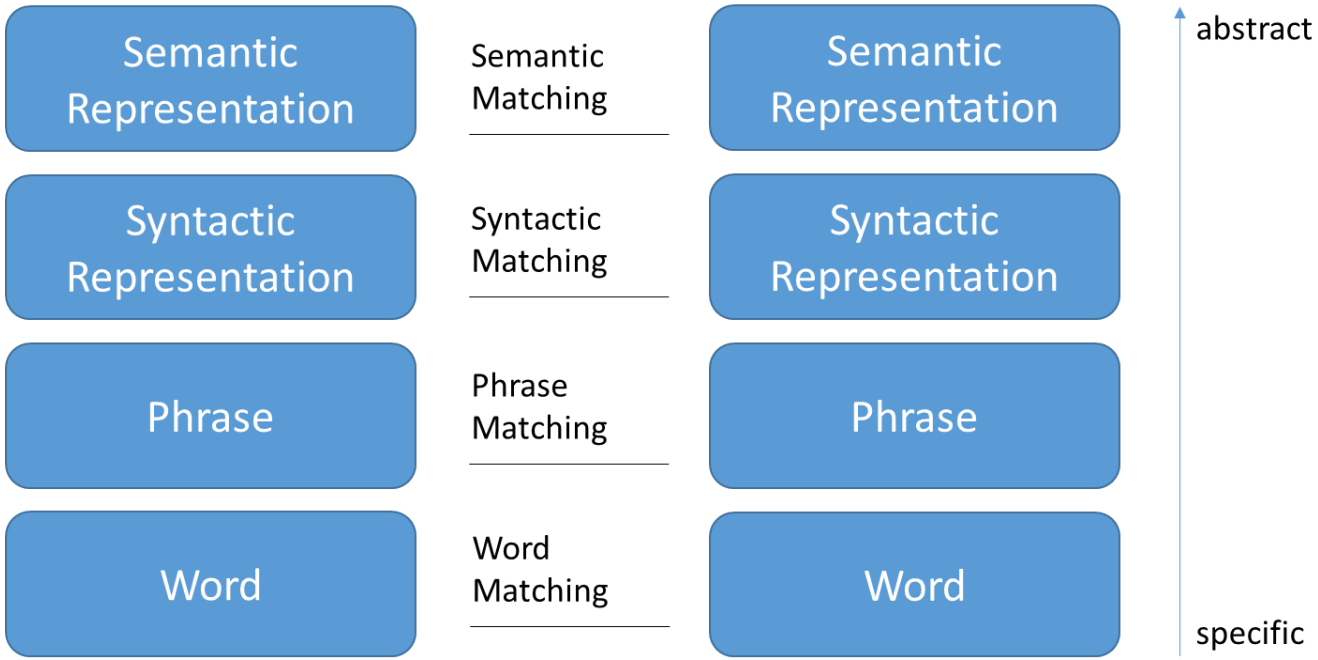


Input Features

- Hand-crafted features for representing query-document pairs
  - *Query-independent* or *static* features
    - e.g., incoming link count and document length
  - *Query-dependent* or *dynamic* features
    - e.g., BM25
  - *Query-level* features
    - e.g., query length

Input Features

1. Handcrafting matching features is **time-consuming**
   - Feature design often requires expertise knowledge
   - The work has to be done again for each task/domain/…
2. Human defined features are often **incomplete**
3. Human defined features are often **over-specified**



Taxonomy of LTR Approaches

$x \rightarrow y$

Pointwise Approach

- Regression, classification or ordinal classification for each query-document pair
- OC SVM, McRank

$(x_1, x_2) \rightarrow y$

Pairwise Approach

- Preference classification between pairs of documents with respect to individual queries
- RankSVM, RankBoost, RankNet, GBRank

$(x_1, x_2, \cdots, x_n) \rightarrow \vec{y}$

Listwise Approach

- Directly optimizing for a rank-based metric for a list
- ListMLE, ListNet, RankCosine, StructureSVM, SoftRank, AdaRank

Pointwise Approach

- Regression Model $L(f; x_j, y_j) = \big(y_j - f(x_j)\big)^2.$
- Classification
  - Support Vector Machine

$$\min \frac{1}{2}\|w\|^2 + \lambda \sum_{i=1}^{n} \sum_{j=1}^{m^{(i)}} \xi_j^{(i)}$$

$$\text{s.t.} \quad w^T x_j^{(i)} \le -1 + \xi_j^{(i)}, \quad \text{if } y_j^{(i)} = 0.$$

$$w^T x_j^{(i)} \ge 1 - \xi_j^{(i)}, \quad \text{if } y_j^{(i)} = 1.$$

$$\xi_j^{(i)} \ge 0, \quad j = 1, \ldots, m^{(i)}, i = 1, \ldots, n,$$

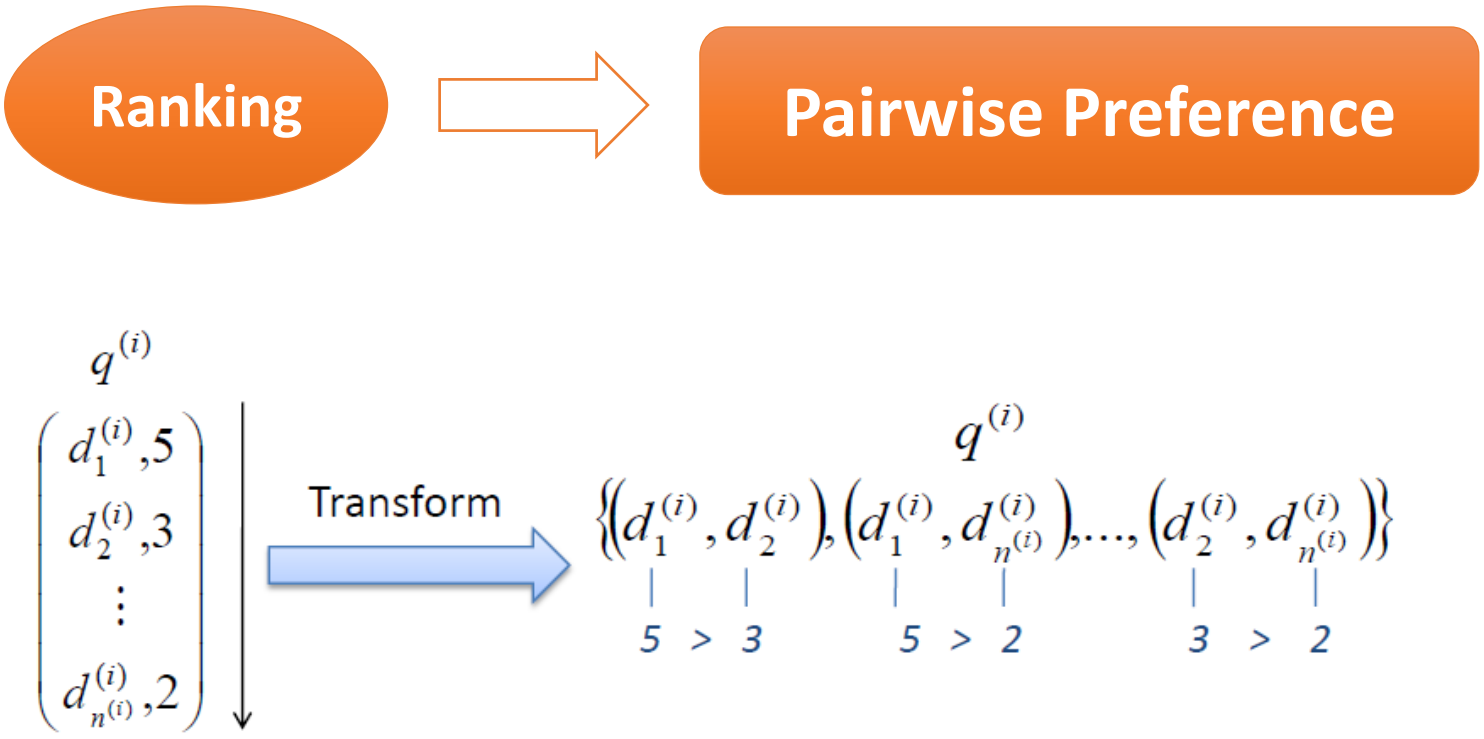
  - Logistic Regression

$$\log\left(\frac{P(R|x_j)}{1 - P(R|x_j)}\right) = c + \sum_{t=1}^{T} w_t x_{j,t} \qquad P(R|x_j) = \frac{1}{1 + e^{-c - \sum_{t=1}^{T} w_t x_{j,t}}}.$$

Cons & Pros of Pointwise Approach

- Pros:
  - Simple and straightforward
  - Direct apply existing algorithms to solve the ranking task
- Cons:
  - Consider each document independently, no relative order is taken into account
  - Learning objective deviates from the evaluation metrics
    - Position
    - Multi-grade labels

Pairwise Approach

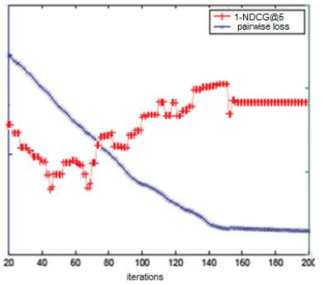


Cons & Pros of Pairwise Approach

- Pros:
  - ➢ Simple and intuitive
  - ➢ Modeling relative order, better capture the inherent property of ranking than pointwise approach
  - ➢ Strong performance, widely adopted by modern search engines
- Cons:
  - ➢ Learning objective deviates from the evaluation metrics
  - ➢ A large number of pairs to train

p: *perfect*, g: *good*, b: *bad*
Ideal: p g g b b b b

ranking 1: g p g b b b b one wrong pair **Worse**
ranking 2: p g b g b b b one wrong pair **Better**



Pairwise loss vs. (1-NDCG@5)
TREC Dataset

Listwise Approach

- Optimize Surrogate Loss
  - ListMLE，ListNet，StrctRank，BoltzRank
- Direct Optimize Evaluation Metrics
  - Optimize an approximate function of the evaluation metric (continuous & differentiable)
    - SoftRank, AppRank, SmoothRank
  - Optimize an upper bound of the evaluation matric (continuous & differentiable)
    - SVMmap，SVMNDCG，PermuRank
  - Direct optimization techniques on evaluation metrics
    - AdaRank，RankGP，LambdaMart

Cons & Pros of Listwise Approach

- Pros:
  - Learning objective is consistent with the evaluation metrics
  - Strong performance on different datasets
- Cons:
  - Models are usually complicated
  - Training is not efficient

Typical Loss Functions for NeuIR

- Many judged query-document pairs
  - Preference probability (sigmoid function)

$$p_{ij} \stackrel{\text{def}}{=} p(d_i \succ d_j) \stackrel{\text{def}}{=} \frac{1}{1 + e^{-\delta(s_i - s_j)}}$$

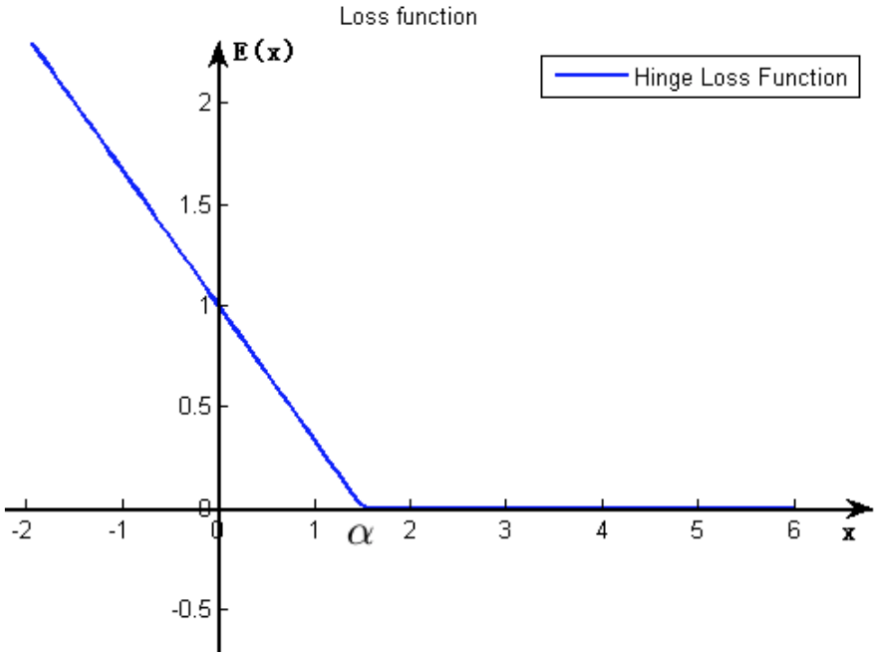
  - Cross-entropy loss

$$\begin{aligned}\mathcal{L} &= -\overline{p_{ij}} \log(p_{ij}) - (1 - \overline{p_{ij}}) \log(1 - \overline{p_{ij}}) \\ &= \frac{1}{2}(1 - S_{ij}) + \log\left(1 + e^{-\delta(s_i - s_j)}\right) \\ &= \log\left(1 + e^{-\delta(s_i - s_j)}\right) \text{ if, } d_i \succ d_j (S_{ij} = 1)\end{aligned}$$

Typical Loss Functions for NeuIR

- Many judged query-document pairs
  - Hinge loss

$$\mathcal{L} = \max(0, \alpha - s_i^+ + s_i^-)$$



Typical Loss Functions for NeuIR

- A single relevant document for a given query
  - Preference probability (softmax function)

$$p(d^+|q) = \frac{e^{-\gamma \cdot s(q,d^+)}}{\sum_{d \in D} e^{-\gamma \cdot s(q,d)}}$$

  - Cross-entropy loss

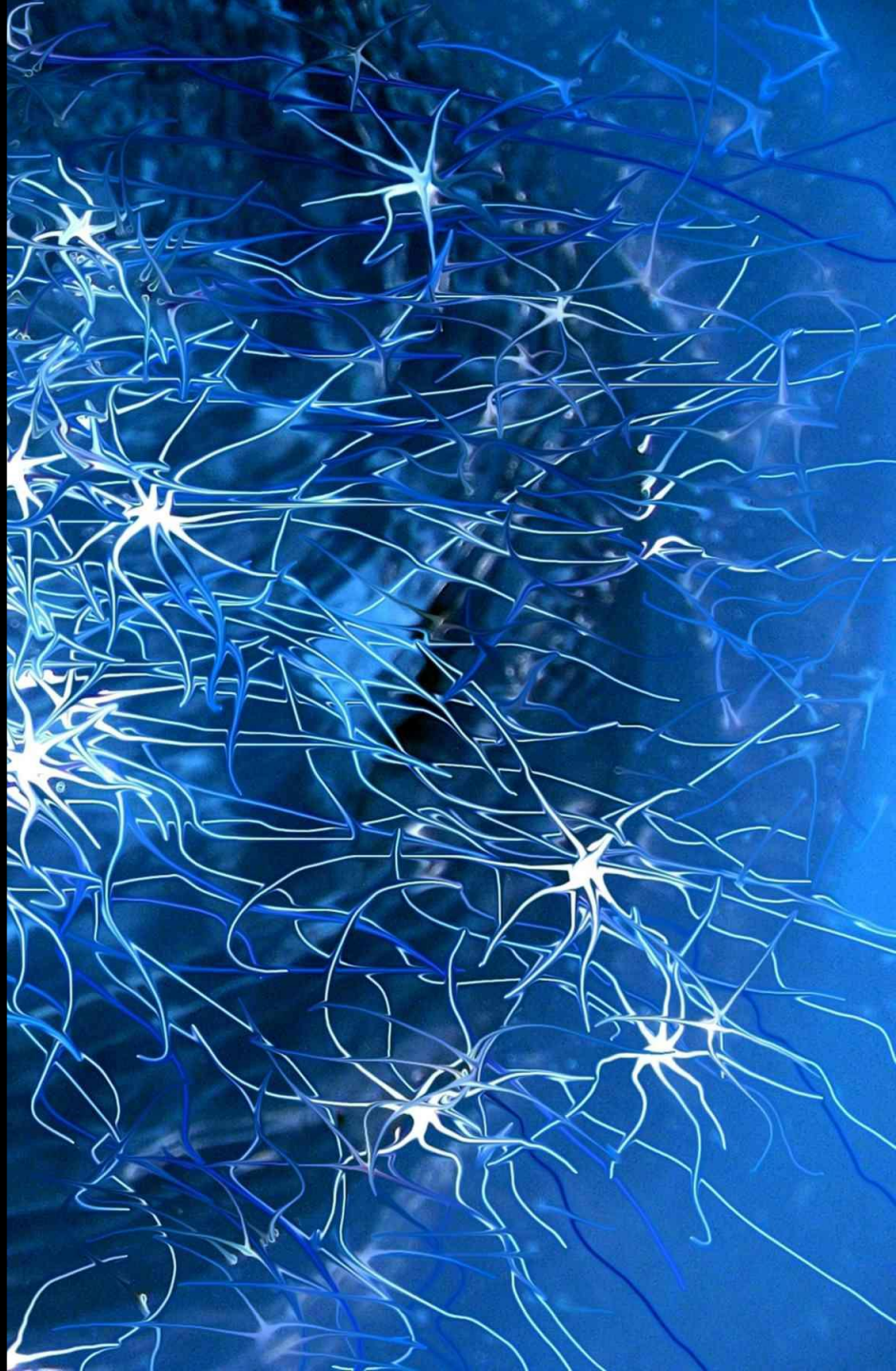
$$\mathcal{L}_{CE}(q, d^+, D) = -\log\big(p(d^+|q)\big) = -\log\left(\frac{e^{-\gamma \cdot s(q,d^+)}}{\sum_{d \in D} e^{-\gamma \cdot s(q,d)}}\right)$$

    - Hierarchical softmax
    - Importance sampling
    - Noise contrastive estimation
    - Negative sampling

$$\mathcal{L}_{NEG}(q, d^+, D) = -\sum_{<x,d^+>} \log\left(\frac{1}{1 + e^{-\gamma \cdot s(q,d^+)}} + \sum_{i=1}^{k} \log \frac{1}{1 + e^{-\gamma \cdot s(q,d_i^-)}}\right)$$

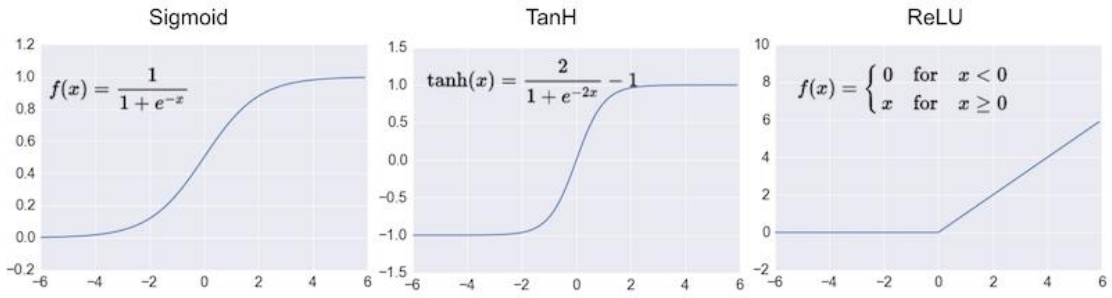
Chapter 5

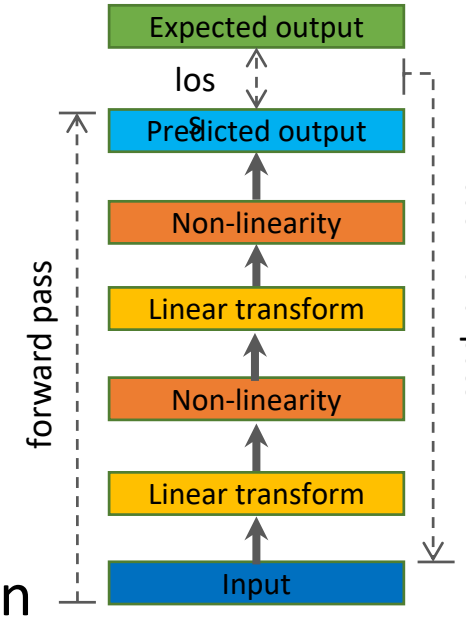
Deep Neural Nets



Background of NN

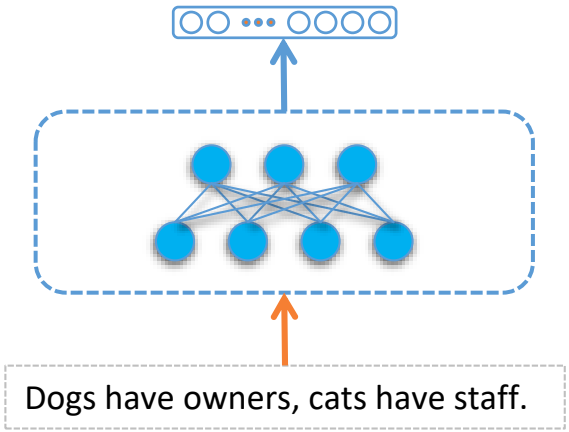
- Chains of parameterized linear transforms followed by non-linear functions:
  - Linear transforms: $y = W * x + b$
  - Popular non-linear function





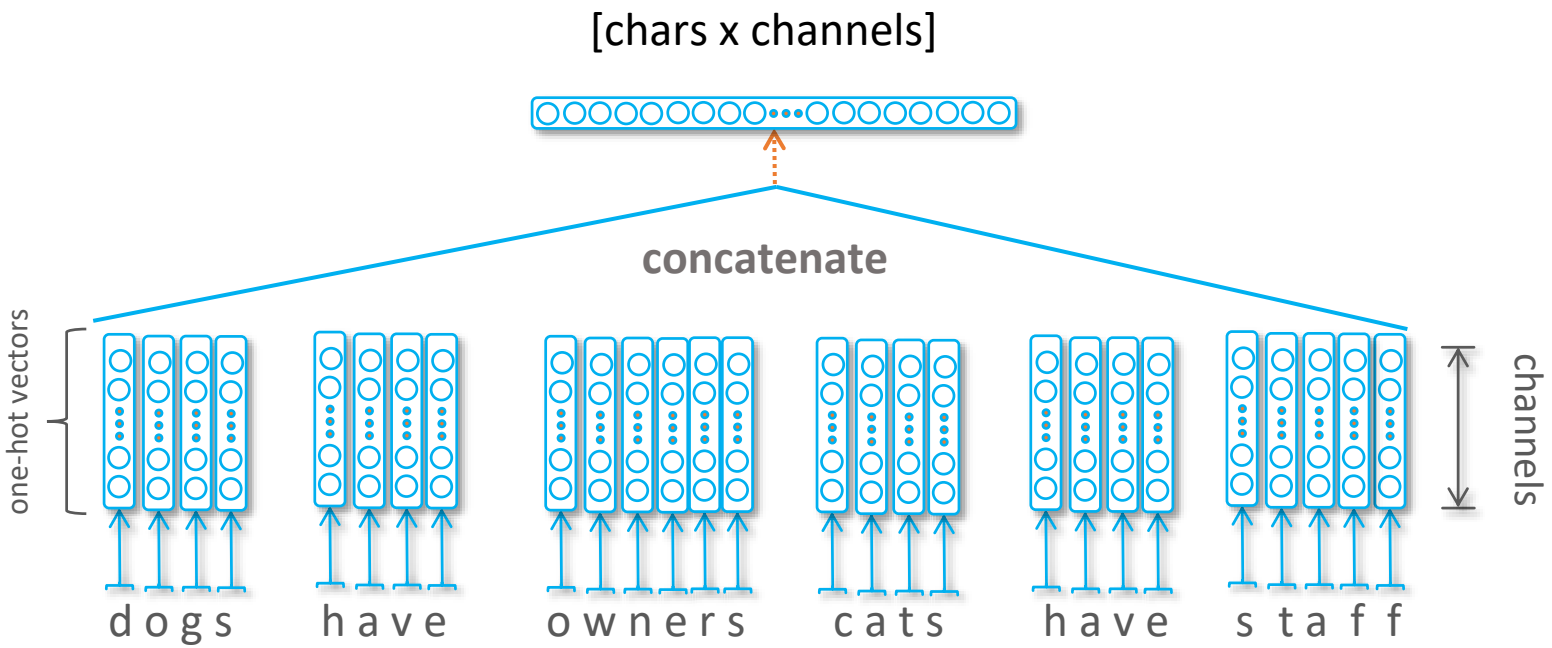
- Parameters are trained with backpropagation
- E2E training over millions of samples in batched mode

Neural models for text



How do you feed text to a neural network?

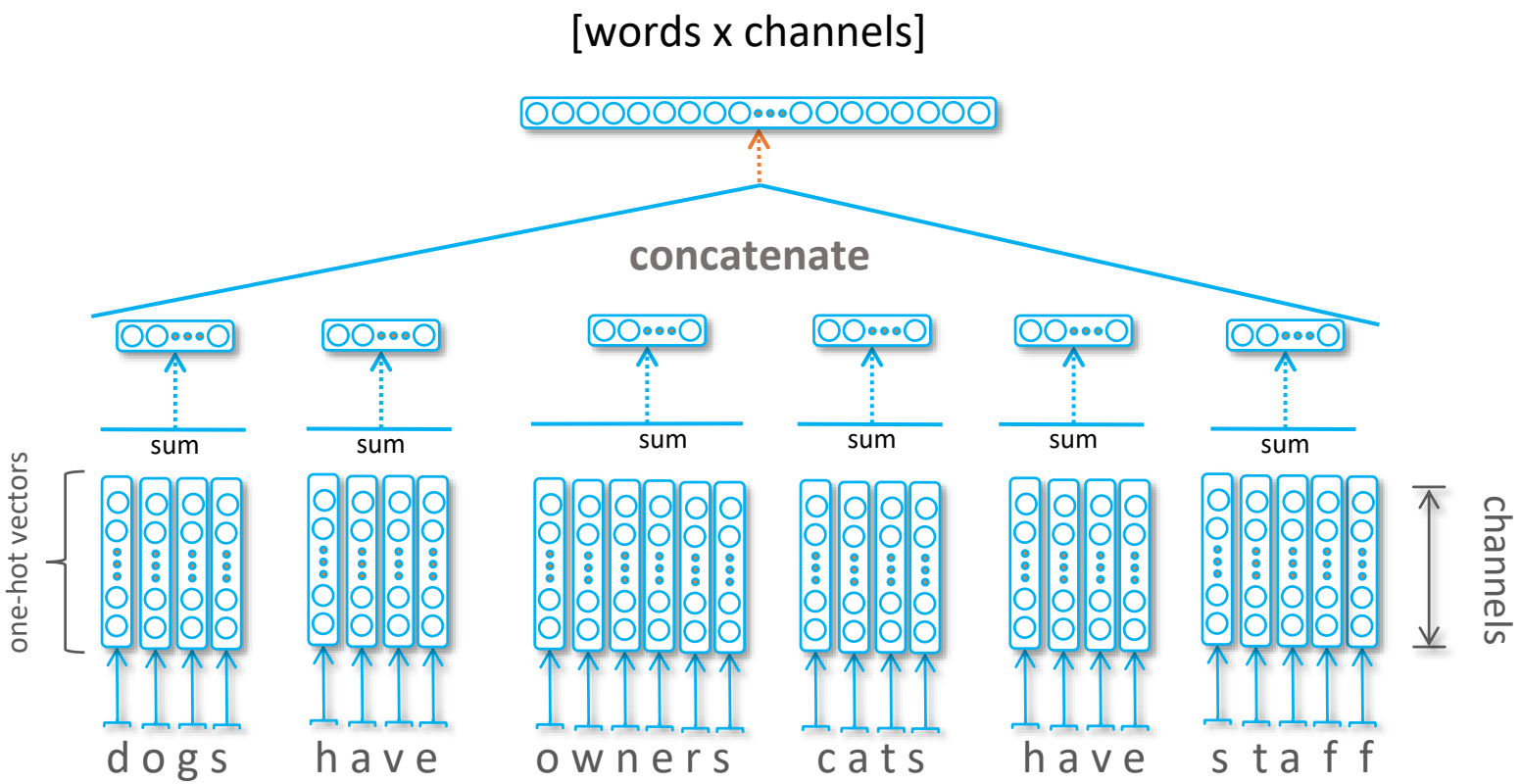
Local representations of input text



Char-level models

Figure from Mitra & Craswell Tutorial @WSDM 2017

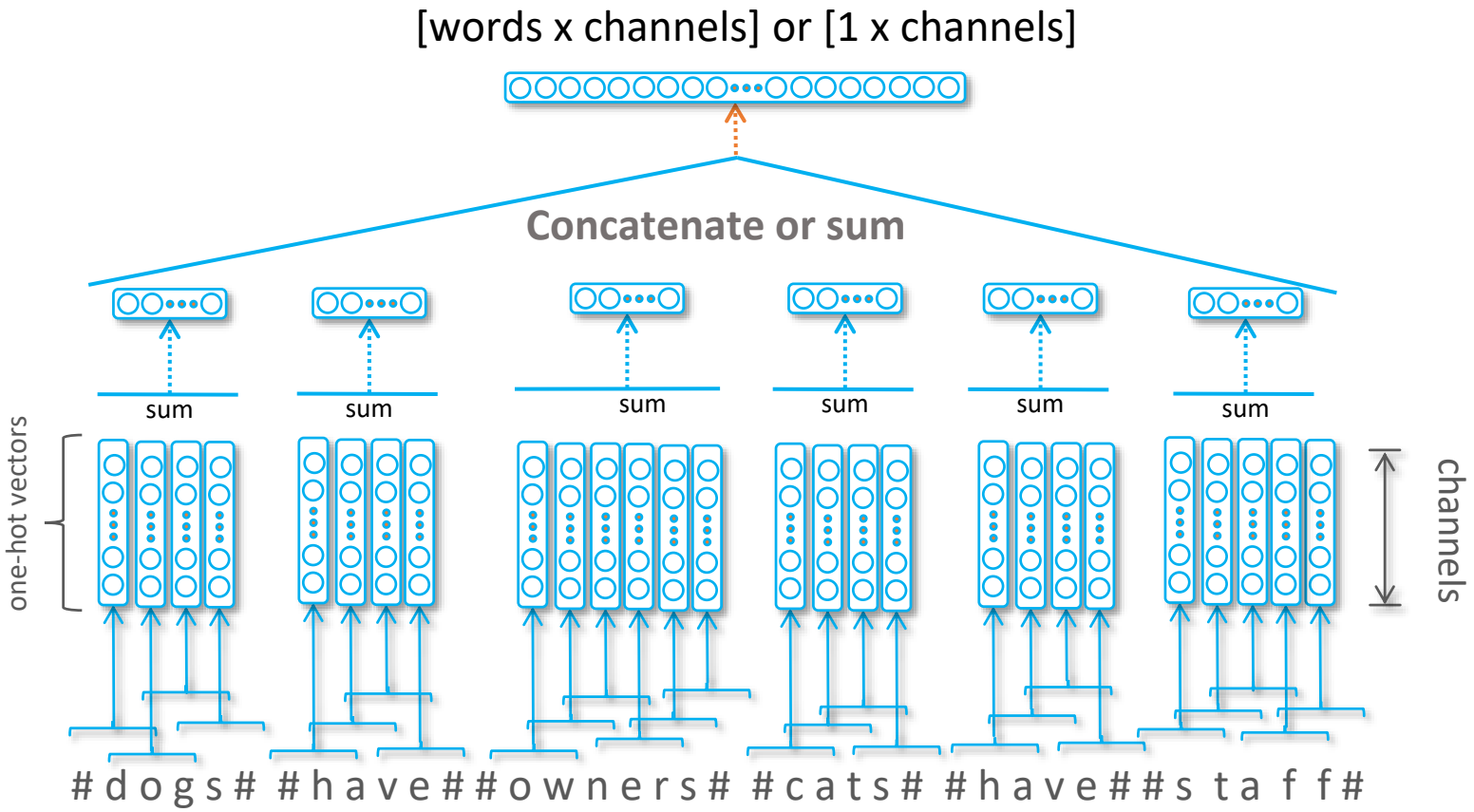
Local representations of input text



Word-level models w/ bag-of-chars per word

Figure from Mitra & Craswell Tutorial @WSDM 2017

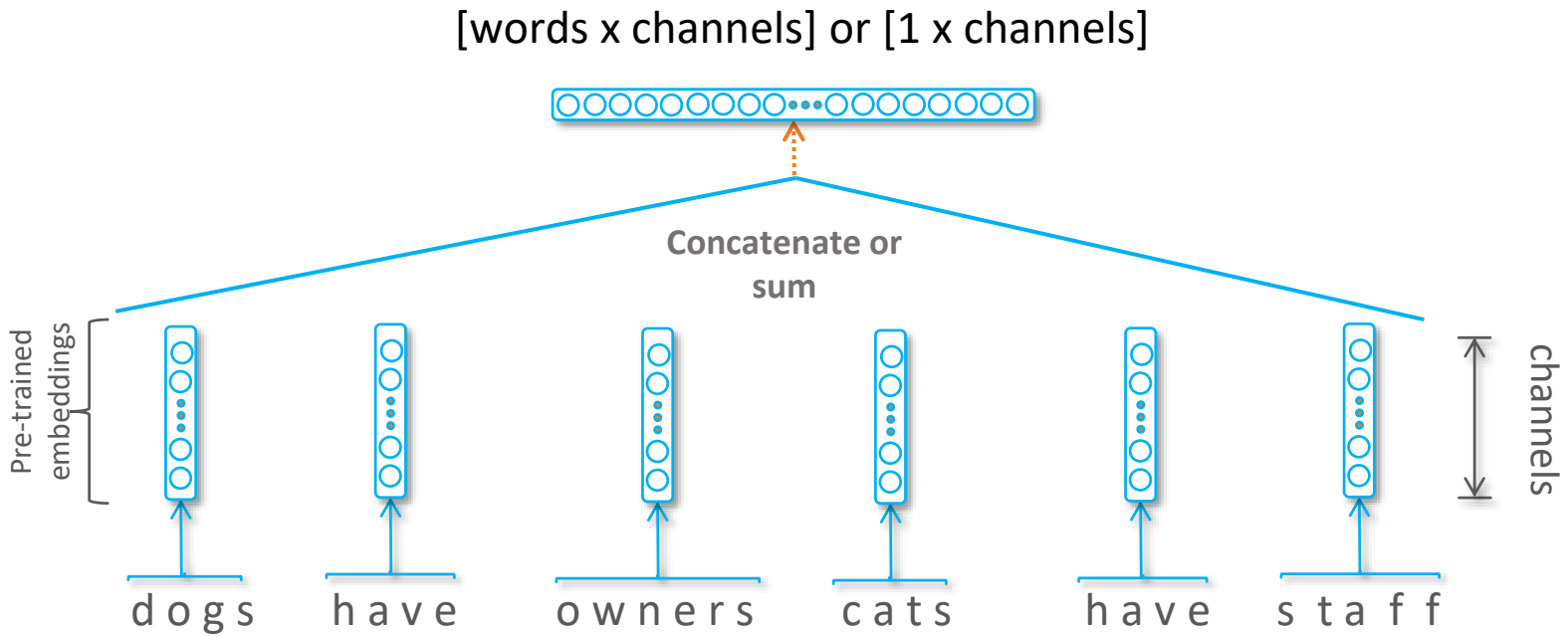
Local representations of input text



Word-level models w/ bag-of-trigrams per word

Figure from Mitra & Craswell Tutorial @WSDM 2017

Distributed representations of input text

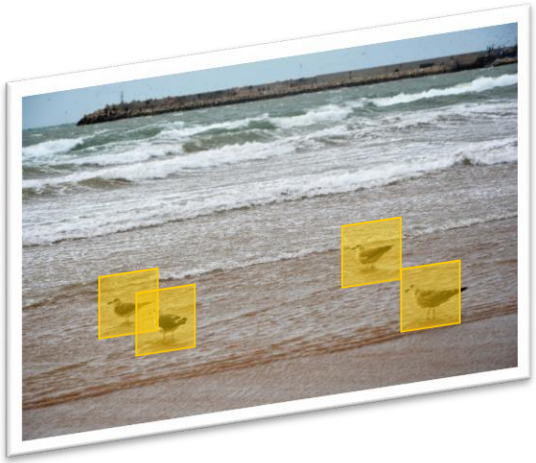


Word-level models w/ pre-trained embeddings

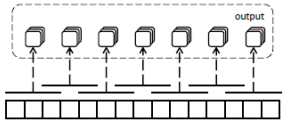
Figure from Mitra & Craswell Tutorial @WSDM 2017

Popular Architectures in IR

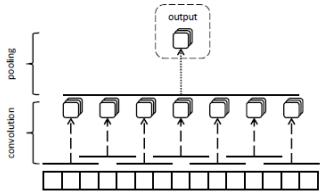
Shift-invariant neural operations



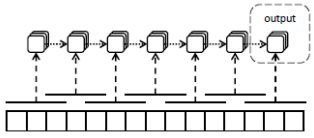
- Detecting a pattern in one part of input space is same as detecting it in another
  - (also applies to sequential inputs, and inputs with dims >2)
- Leverage redundancy by operating on a moving window over whole input space and then aggregate
- The repeatable operation is called a kernel, filter, or cell
- Aggregation strategies leads to different architectures



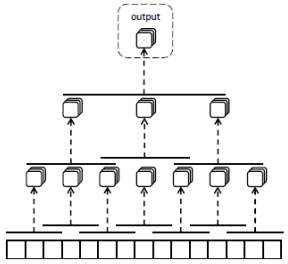
(a) Convolution or pooling



(b) Convolution w/ global pooling



(c) Recurrent

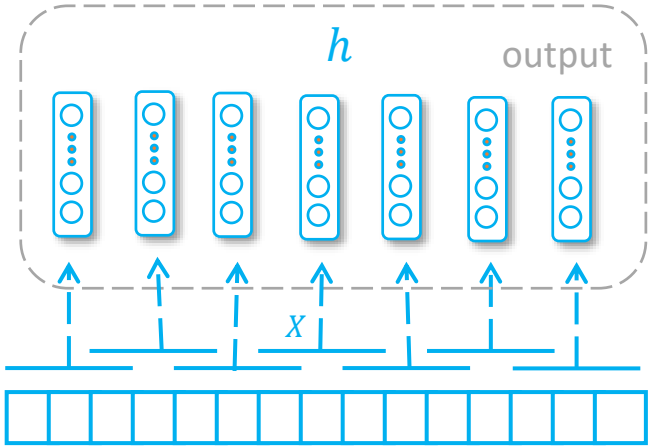


(d) Recursive or tree

Shift-invariant neural operations - Convolution

- Move the window over the input space each time applying the same cell over the window
- A typical cell operation can be,

$$h = \sigma(WX + b)$$



Full Input	[words x in_channels]
Cell Input	[window x in_channels]
Cell Output	[1 x out_channels]
Full Output	[1 + (words – window) / stride x out_channels]

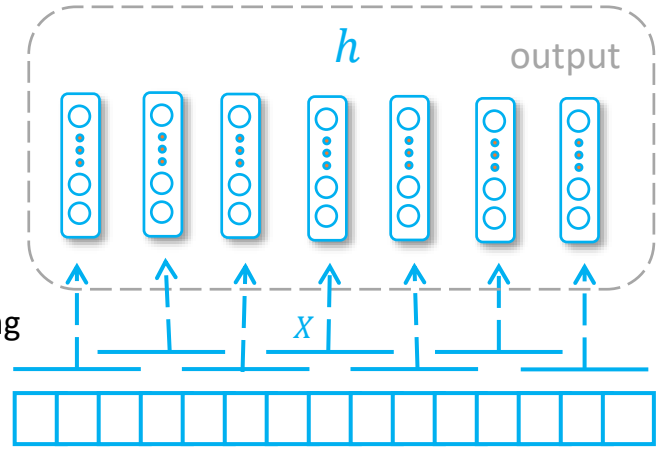
Shift-invariant neural operations - Pooling

- Move the window over the input space each time applying an aggregate function over each dimension in within the window

$$h_j = max_{i \in win}(X_{i,j}) \quad or \quad h_j = avg_{i \in win}(X_{i,j})$$

max -pooling

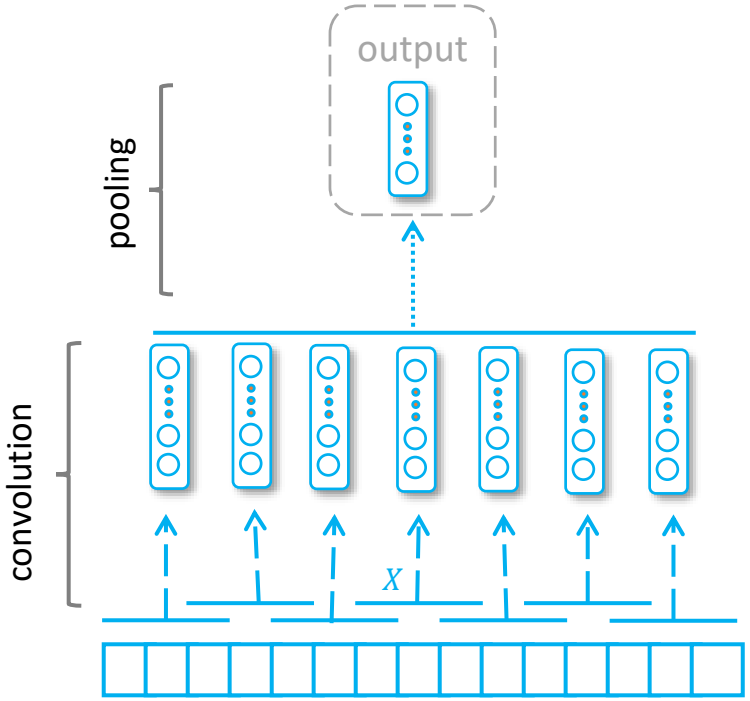
average -pooling



Full Input	[words x channels]
Cell Input	[window x channels]
Cell Output	[1 x channels]
Full Output	[1 + (words – window) / stride x channels]

Shift-invariant neural operations - Convolution w/ Global Pooling

- Stacking a global pooling layer on top of a convolutional layer is a common strategy for generating a fixed length embedding for a variable length text



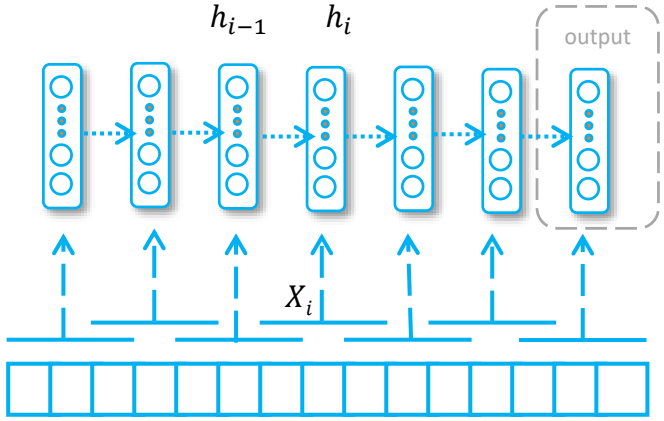
Full Input [words x in_channels]

Full Output [1 x out_channels]

Shift-invariant neural operations - RNN

- Similar to a convolution layer but additional dependency on previous hidden state
- A simple cell operation shown below but others like LSTM and GRUs are more popular in practice,

$$h_i = \sigma(WX_i + Uh_{i-1} + b)$$

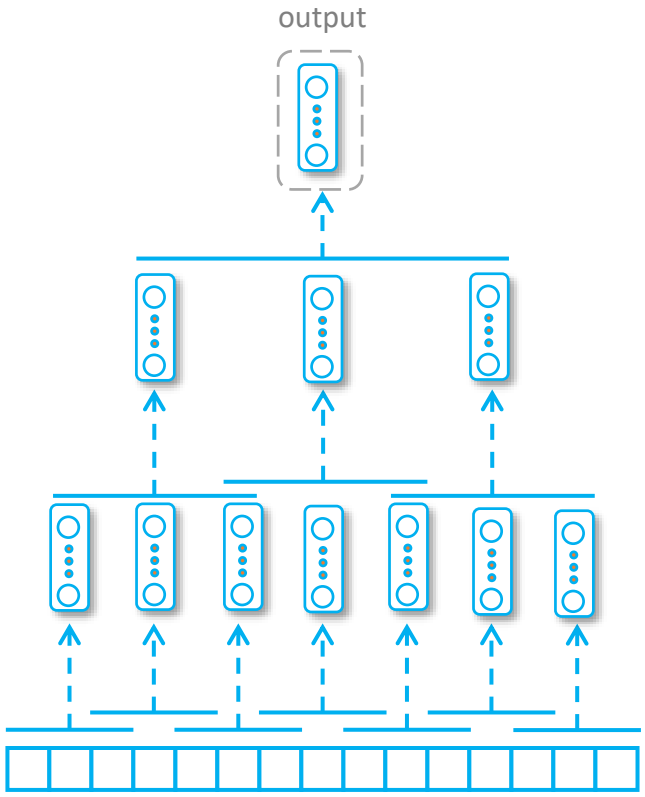


Full Input	[words x in_channels]
Cell Input	[window x in_channels] + [1 x out_channels]
Cell Output	[1 x out_channels]
Full Output	[1 x out_channels]

Shift-invariant neural operations - Recursive NN

- Shared weights among all the levels of the tree
- Cell can be an LSTM or as simple as

$$h = \sigma(WX + b)$$



Full Input	[words x channels]
Cell Input	[window x channels]
Cell Output	[1 x channels]
Full Output	[1 x channels]

Autoencoders

- Unsupervised models trained to minimize reconstruction errors

$$\mathcal{L}(\mathbf{x}, \mathbf{x}') = \|\mathbf{x} - \mathbf{x}'\|^2$$

- Information Bottleneck method (Tishby et al., 1999)
- The bottleneck layer captures "minimal sufficient statistics" of X and is a compressed representation of the input

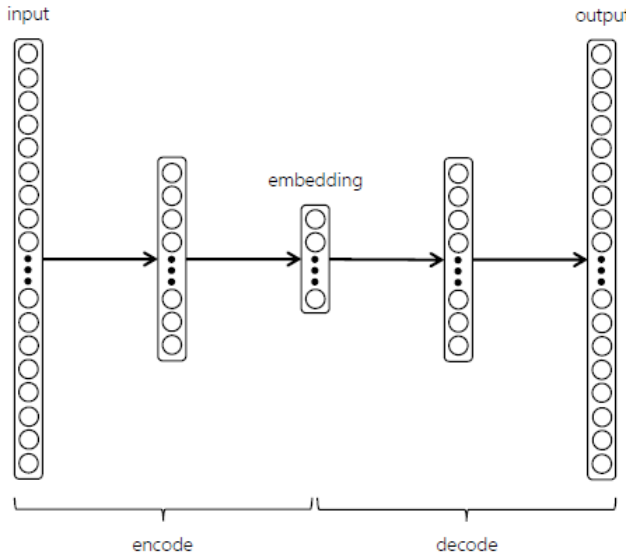


Image source: Mitra & Craswell, An Introduction to Neural Information Retrieval

Siamese networks

- Originally proposed for comparing fingerprints and signatures
- Consists of two models that project two inputs into a common embedding space
- A predefined metric (e.g., cosine similarity) is then used to compute the similarity

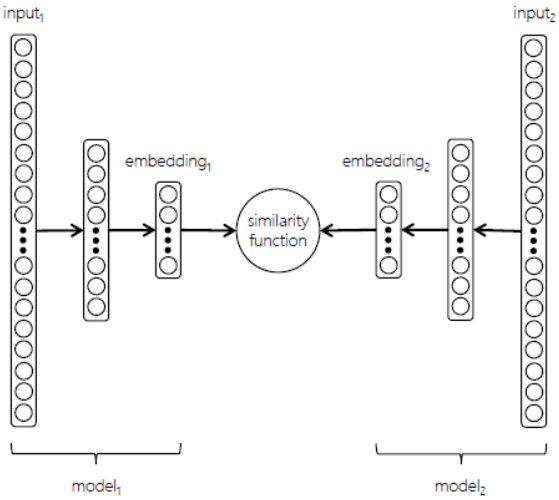
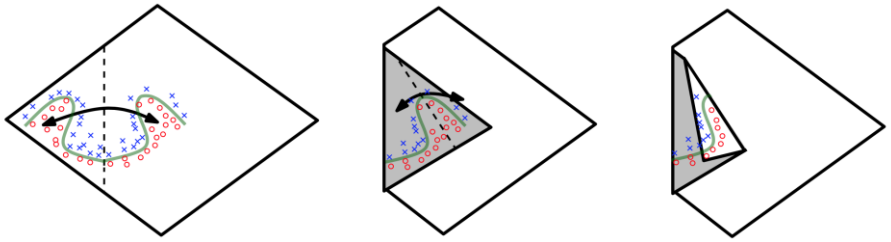


Image source: Mitra & Craswell, An Introduction to Neural Information Retrieval

$$\mathcal{L}_{siamese}(\overrightarrow{v_q}, \overrightarrow{v_{d1}}, \overrightarrow{v_{d2}}) = \log\left(1 + e^{-\gamma(sim(\overrightarrow{v_q},\overrightarrow{v_{d1}}) - sim(\overrightarrow{v_q},\overrightarrow{v_{d2}}))}\right)$$

Why add depth helps

Deeper networks can split the input space in many (non-independent) linear regions than shallow networks



Each layer **folds** its input space onto itself.

Theorem 1.

- *A* **shallow** *rectifier neural network with* m *units can compute functions with at most this many linear regions:*

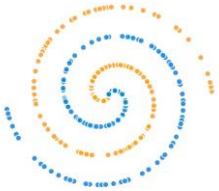
$$O(m^{n_0}) \qquad (\textbf{polynomial} \textit{ in } m).$$

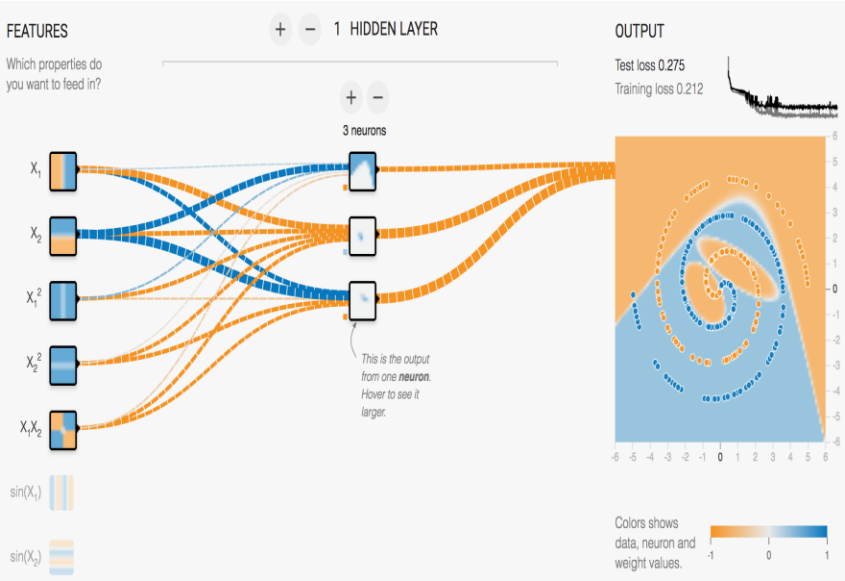
- *A* **deep** *rectifier network with* L *layers of* n *units each can compute functions with this many linear regions:*

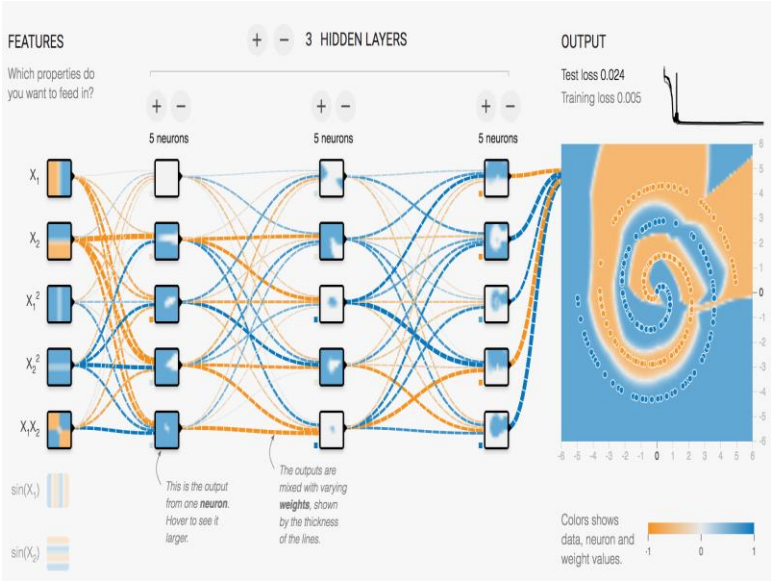
$$O\left(\left(\frac{n}{n_0}\right)^{n_0(L-1)} n^{n_0}\right) \qquad (\textbf{exponential} \textit{ in the depth } L).$$

Montúfar, Pascanu, Cho and Bengio. On the number of linear regions of deep neural networks NIPS 2014

Why add depth helps

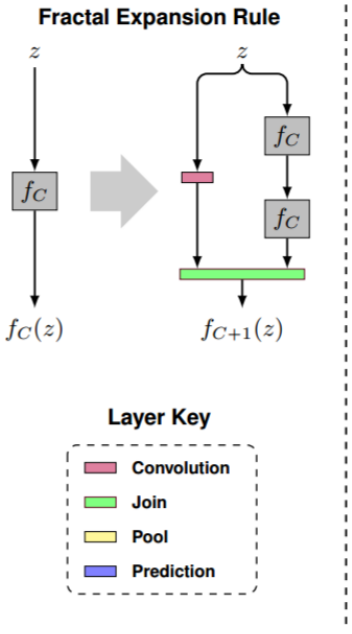


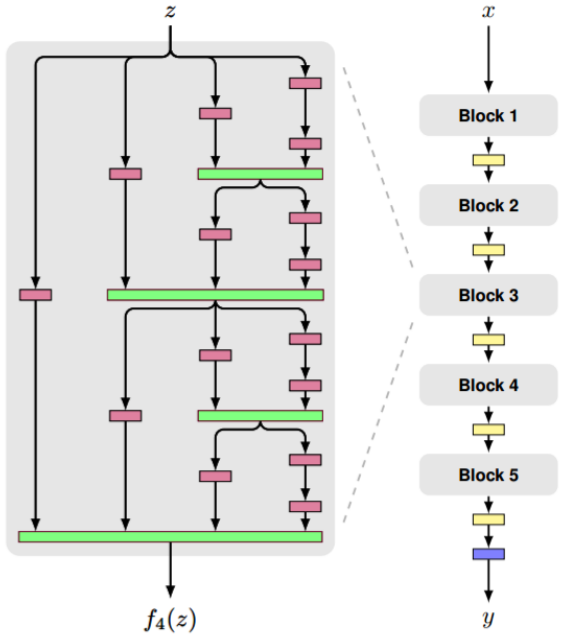




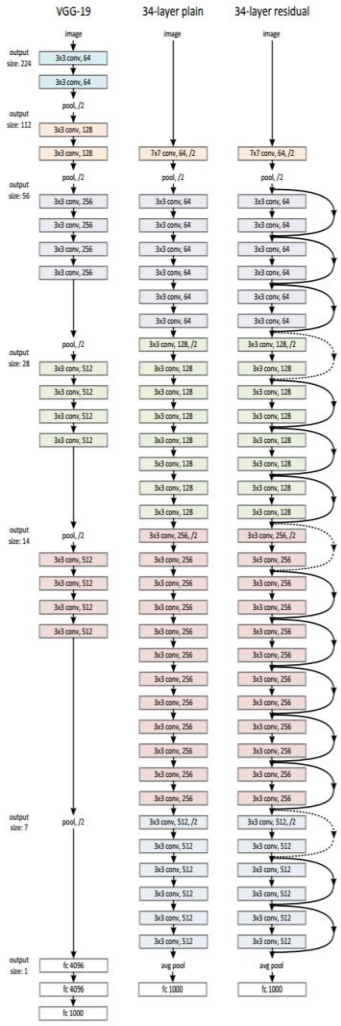
From website: http://playground.tensorflow.org/

Really Deep Neural Models

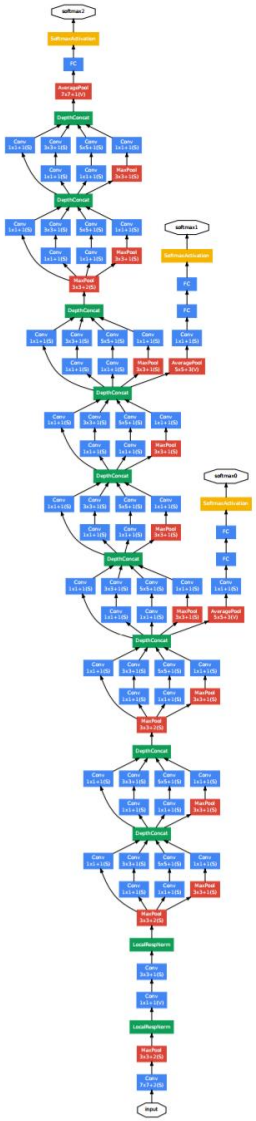




(Larsson et al., 2016)



(He et al., 2015)



(Szegedy et al., 2014)

Chapter 6

Deep Neural Nets for IR

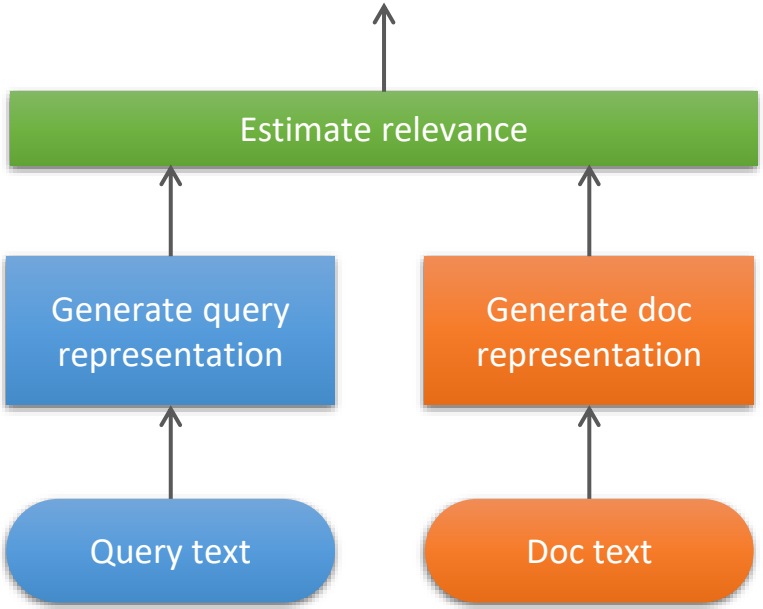
Problem Formulation

- IR as a learning to match problem

$$match(T_1, T_2) = F(\emptyset(T_1), \emptyset(T_2))$$

Scoring function based on the interaction between texts

Map each text to a representation vector

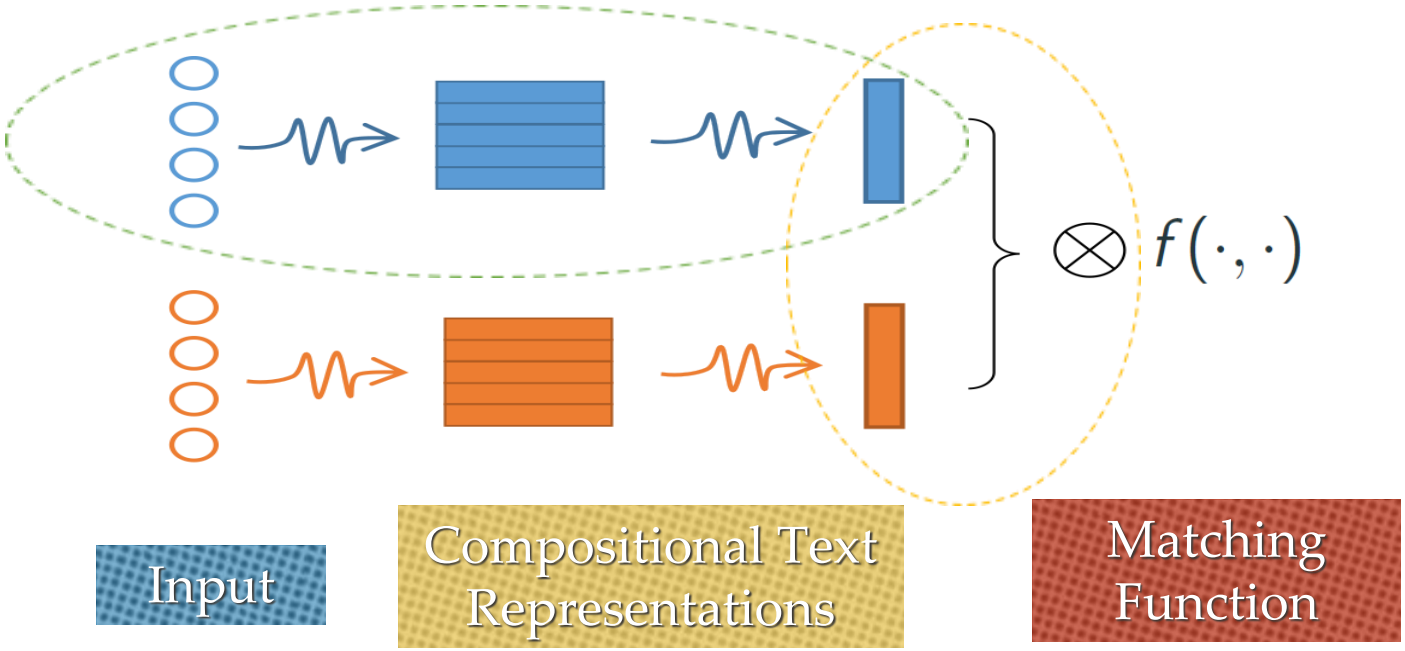


Compositional-focused models

- Focus on learning better representation of text

Step 1: Compositional Text Representation $\phi(x)\ \psi(y)$

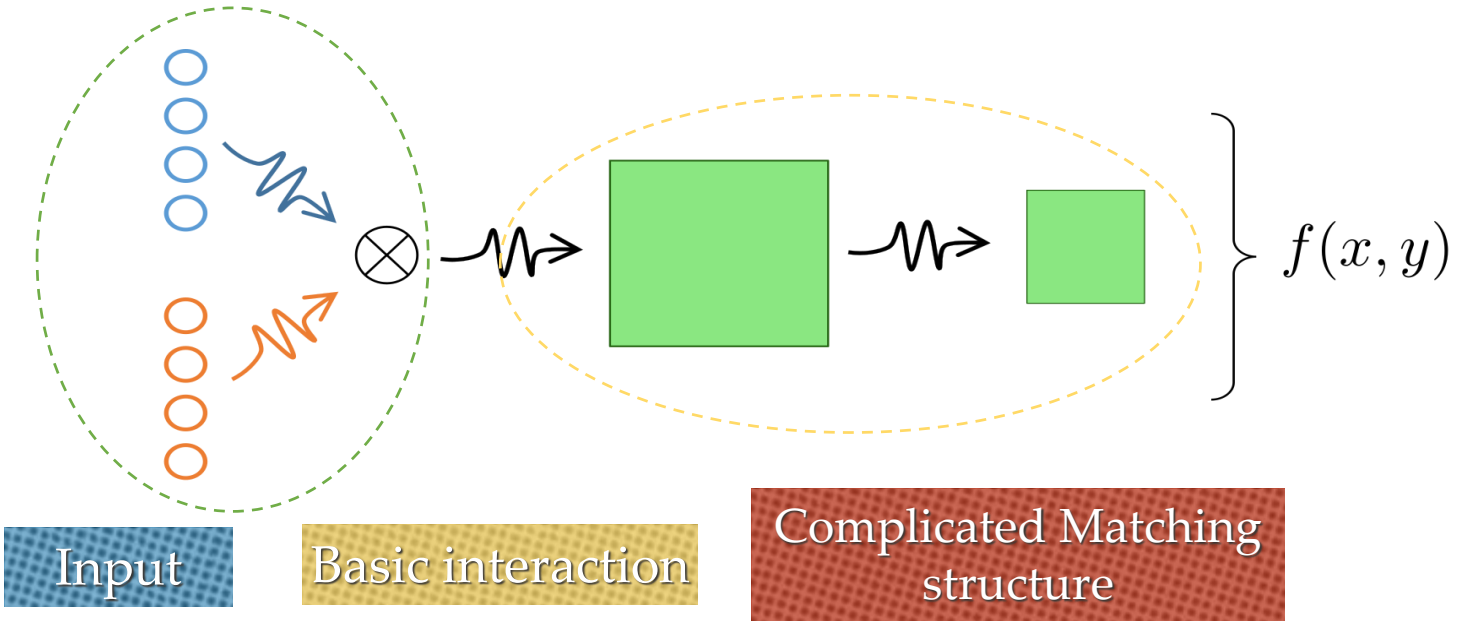
Step 2: Matching Function $f(\cdot,\cdot)$



Interaction-focused models

- Focus on learning better interaction between texts

Step 1: Two sentences meet before their own high-level representations mature
Step 2: Capture complex matching patterns



Typical Composition-Focused Deep Matching Models

- DSSM: Learning Deep Structured Semantic Models for Web Search using Click-through Data (Huang et al., CIKM'13)
- CDSSM: A latent semantic model with convolutional-pooling structure for information retrieval (Shen et al. CIKM'14)
- LSTM-DSSM: Deep Sentence Embedding Using LSTM Analysis and application to Information Retrieval (Palangi et al. ADCS'16)
- MV-LSTM: Match-SRNN: Modeling the Recursive Matching Structure with Spatial RNN (Wan et al. IJCAI'16)
- …

Deep Semantic Structure Model (DSSM)

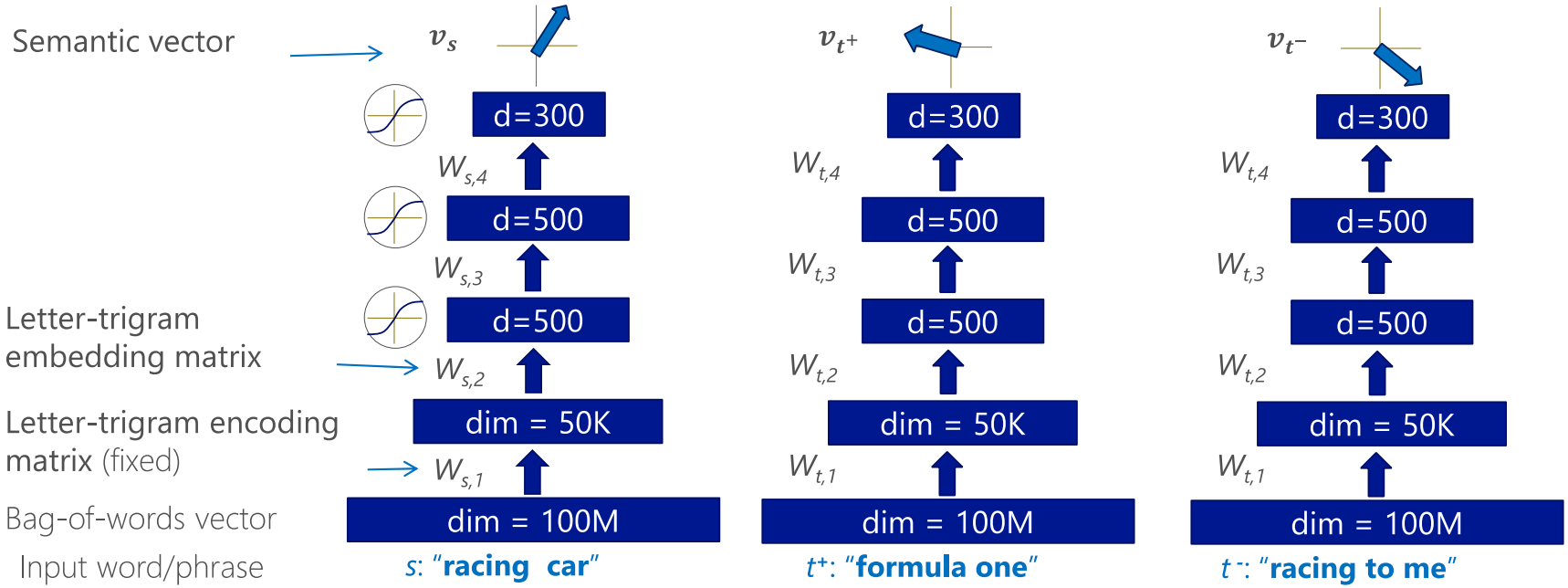


Figure from He et al., CIKM '14 tutorial

- Input: letter tri-gram counts (BoW assumption)
- Relevance is estimated by cosine similarity between Q and D
- Minimize cross-entropy loss against randomly sampled negative documents

Huang et al. Learning deep structured semantic models for web search using clickthrough data, 2013 CIKM.

DSSM – Word Hashing

- **Word hashing** : use sub-word unit (e.g., letter n-gram) as raw input to handle very large vocabulary,
- Letter-trigram Representation:

  deep -> #deep# -> #-d-e, d-e-e, e-e-p, e-p-#
- Only around 50K letter-trigram in English
- **Advantages**
  - Capture sub-word semantics
  - Control the dimensionality of the input space
  - Words with small typos have similar raw representations

Huang et al. Learning deep structured semantic models for web search using clickthrough data, 2013 CIKM.

DSSM

- Evaluated on a document ranking task
  - Docs are ranked by the cosine similarity between embedding vectors of the query and the doc
  - Training data: 100 million query-title pairs from search log
  - Test set: 16,510 English queries sampled from 1-y log

#	Models	Input dimension	NDCG@1
1	BM25 baseline	--	30.8
2	Probabilistic LSA (PLSA)		29.5
3	Auto-Encoder (Word)	40k	31.0 (+0.2)
4	DSSM (Word)	40k	34.2 (+3.4)
5	DSSM (Random projection)	30k	35.1 (+4.3)
6	**DSSM (Letter-trigram)**	**30k**	**36.2 (+5.4)**

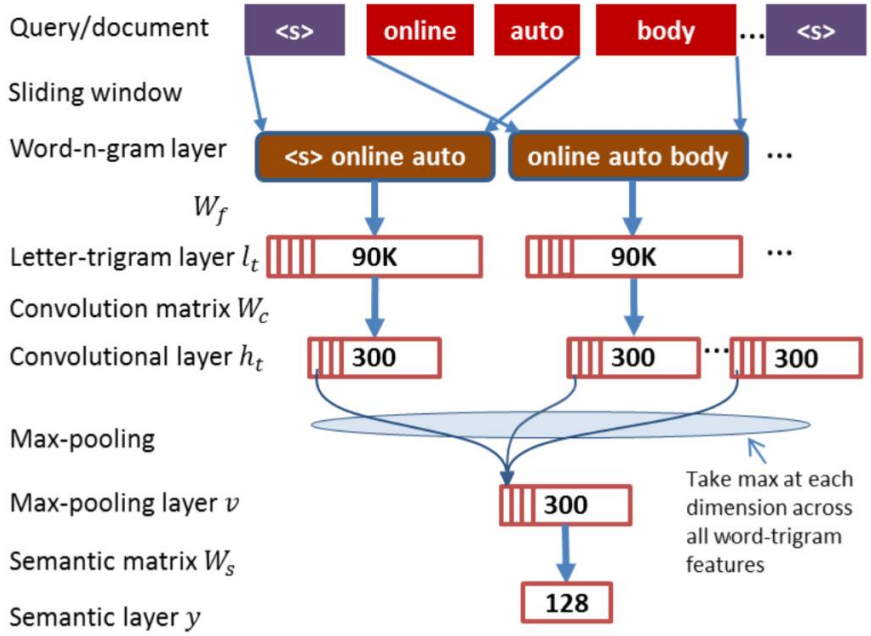
DSSM-based embedding improves 5~7 pt NDCG over shallow models

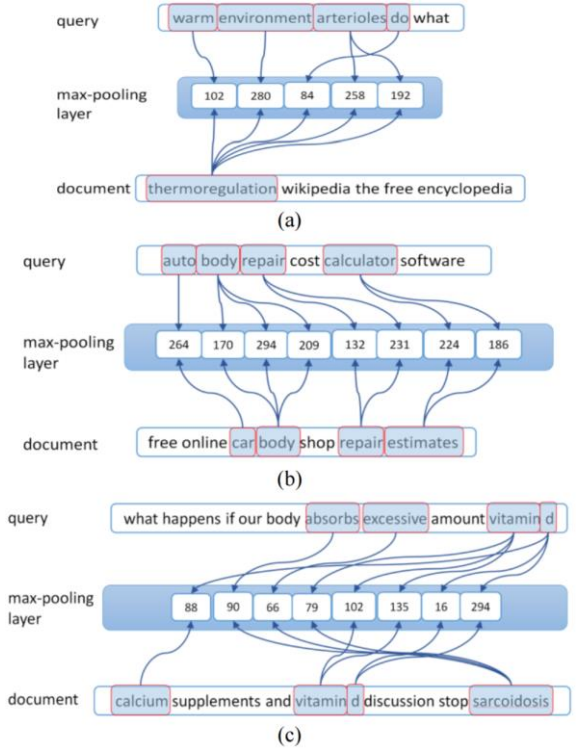
The higher the NDCG score the better, 1% NDCG different is statistically significant.

  - The DSSM learns superior semantic embedding
  - Letter-trigram + DSSM gives better results

Huang et al. Learning deep structured semantic models for web search using clickthrough data, 2013 CIKM.

Convolutional – DSSM



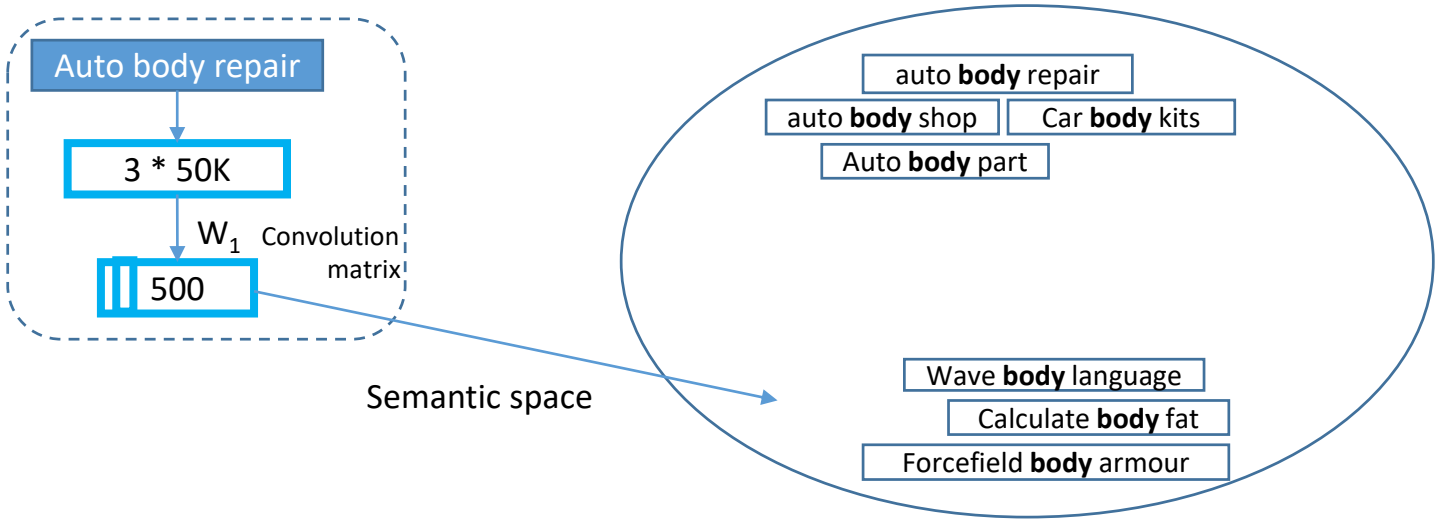


- Input: replace BoW assumption by concatenating term vectors in a sequence, bag-of n-grams (window).
- Convolution followed by global max-pooling.
- Performance improves further by including more negative samples

Shen et al. A latent semantic model with convolutional-pooling structure for information retrieval. 2014 CIKM

Convolutional - DSSM

- What does the model learn at the convolutional layer?
  - Capture the local context dependent word sense
  - Learn one embedding vector for each local context-dependent word



Shen et al. A latent semantic model with convolutional-pooling structure for information retrieval. 2014 CIKM

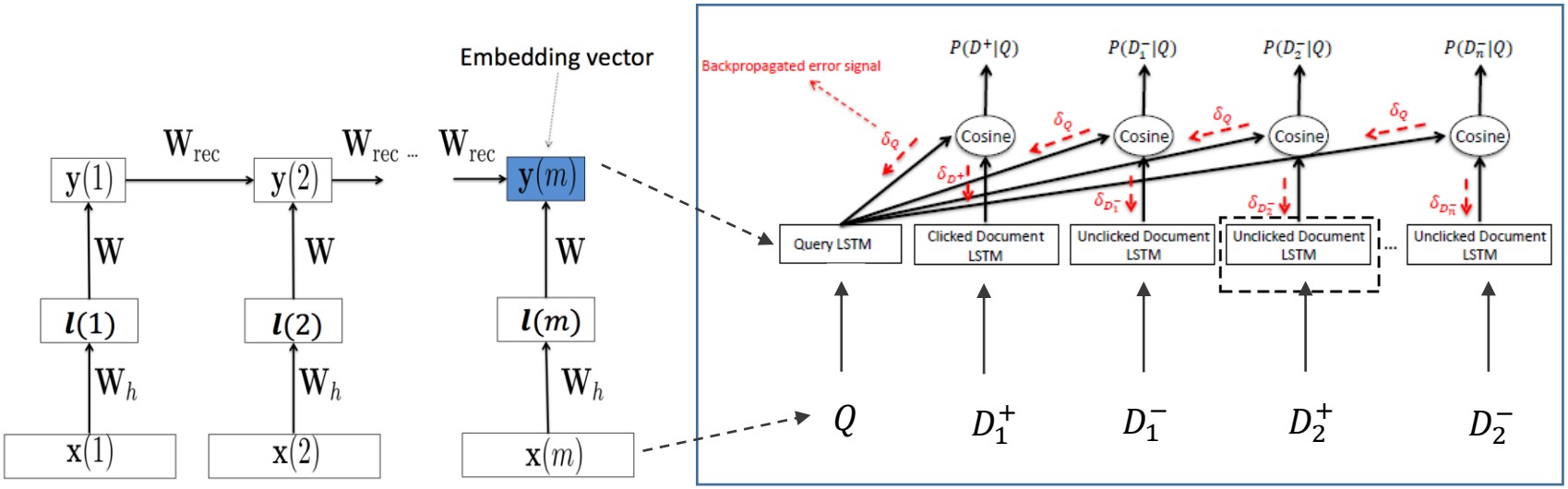
Convolutional - DSSM

- Dataset
  - Training data: 82,834,648 query-title pairs from search log
  - Test set: 12,071 English queries sampled from 1-y log

#	Models	NDCG@1	NDCG@3
	Lexical Matching Models		
1	BM25	30.5	32.8
2	Unigram LM	30.4(-0.1)	32.7(-0.1)
	Topic Models		
3	PLSA [Hofmann 1999]	30.5(+0.0)	33.5(+0.7)
4	BLTM [Gao et al. 2011]	31.6 (+1.1)	34.4(+1.6)
	Clickthrough-based Translation Models		
5	WTM [Gao et al. 2010]	31.5 (+1.0)	34.2(+1.4)
6	PRM [Gao et al. 2010]	31.9 (+1.4)	34.7(+1.9)
	Deep Structure Semantic Model		
7	DSSM [Huang et al. 2013]	32.0 (+1.5)	35.5 (+2.7)
8	**C-DSSM [Shen et al. 2014]**	**34.2 (+3.7)**	**37.4 (+4.6)**

Shen et al. A latent semantic model with convolutional-pooling structure for information retrieval. 2014 CIKM

LSTM- DSSM



- Input: a sequence of letter tri-gram vectors.
- Capture long term dependencies.
- Automatic detect salient keywords in the sentence.

Palangi, et al. Deep sentence embedding using long short-term memory networks *TASLP 2016*

LSTM- DSSM

1. The LSTM-DSSM is robust to noise, i.e., it mainly embeds **keywords** in the final semantic vector representing the whole sentence.
2. In LSTM-DSSM, each cell is usually **allocated** to keywords from a specific topic.
3. The ability to embed the **contextual** and semantic information of the sentences into a finite dimension vector.

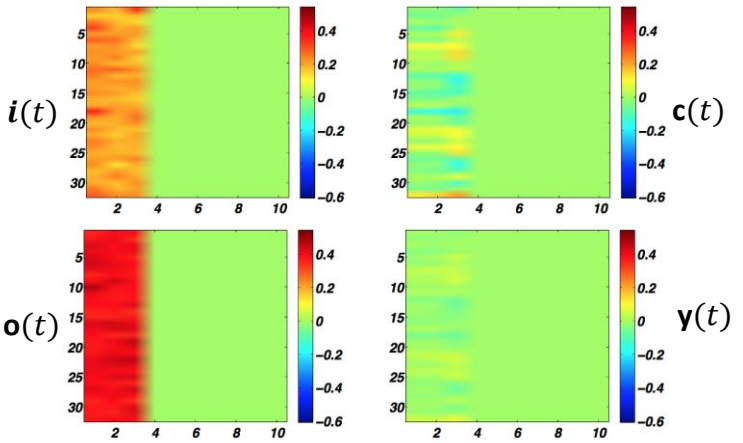


Fig. 4. Query: "*hotels in shanghai*". Since the sentence ends at the third word, all the values to the right of it are zero (green color).

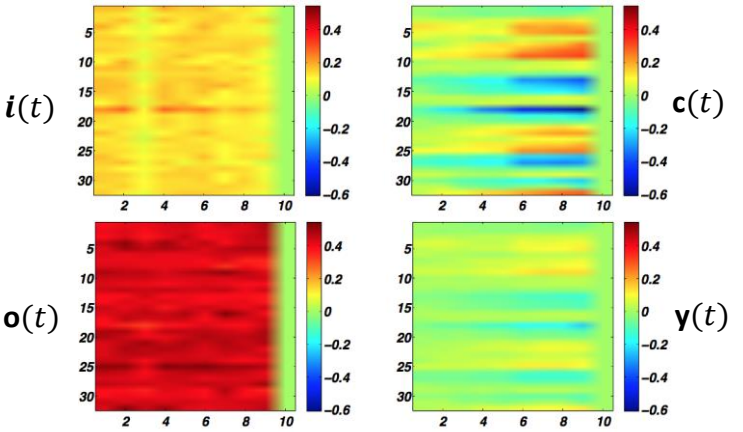


Fig. 5. Document: "*shanghai hotels accommodation hotel in shanghai discount and reservation*". Since the sentence ends at the ninth word, all the values to the right of it are zero (green color).

Palangi, et al. Deep sentence embedding using long short-term memory networks *TASLP 2016*

LSTM- DSSM

Dataset:

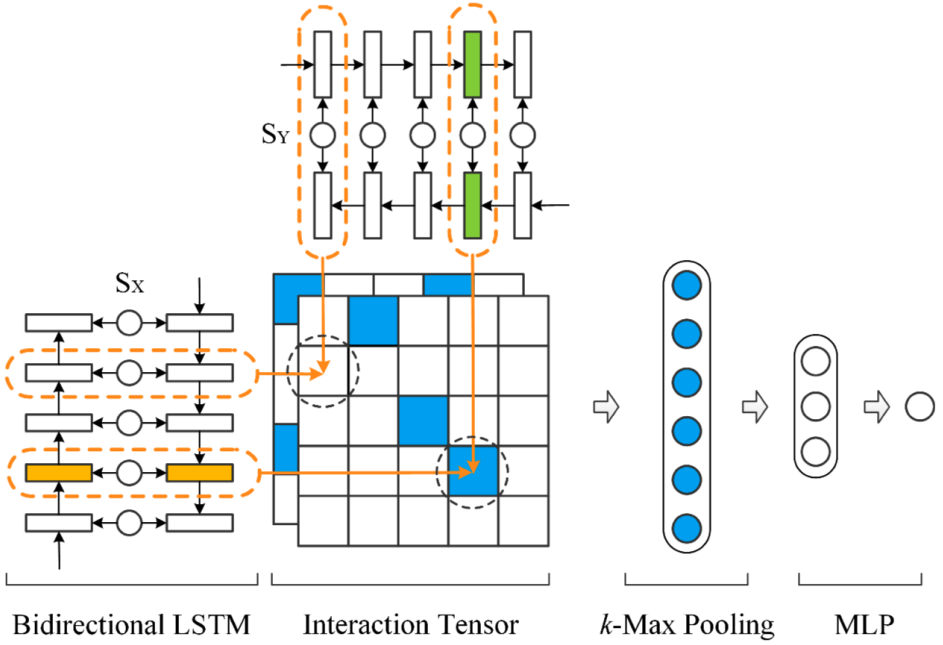
- **Click-through** dataset by a commercial web search engine:
- 200,000 positive query-doc pairs.

#	Models	NDCG@1	NDCG@3	NDCG@10
1	Skip-Thought off-the-shelf	26.9%	29.7%	36.2%
2	Doc2Vec	29.1%	31.8%	38.4%
3	ULM	30.4%	32.7%	38.5%
4	BM25	30.5%	32.8%	38.8%
5	PLSA (T=500)	30.8%	33.7%	40.2%
6	CLSM (nhid=288/96, win=1) 2 layers, 14.4M parameters	31.8%	35.1%	42.6%
7	CLSM (nhid=288/96, win=3) 2 layers, 43.2M parameters	32.1%	35.2%	42.7%
8	CLSM (nhid=288/96, win=5) 2 layers, 72M parameters	32.0%	35.2%	42.6%
9	RNN (nhid=288) 1 Layer	31.7%	35.0%	42.3%
10	LSTM-RNN (ncell=32) 1 Layer, 4.8M parameters	31.9%	35.5%	42.7%
11	LSTM-RNN (ncell=96) 1 Layer, n=2	32.6%	36.0%	43.4%
12	LSTM-RNN (ncell=96) 1 Layer, n=8	33.1%	36.4%	43.7%
13	Bidirectional LSTM-RNN (ncell=96) 1 Layer	**33.2%**	**36.6%**	**43.6%**

Palangi, et al. Deep sentence embedding using long short-term memory networks *TASLP 2016*

MV-LSTM

1. Scan each sentences by Bi-LSTM.
2. Use intermedia representations to construct interaction Tensor.
3. Well capture contextualized local information in the matching process.



Wan S, Lan Y, Guo J, et al. A deep architecture for semantic matching with multiple positional sentence representations//Proceedings of the 30th AAAI Conference on Artificial Intelligence . Phoenix, USA, 2016: 2835-2841.

MV-LSTM

- Contextualized local information
  - ➢ Treat one sentence in multiple views.
  - ➢ Concentrate on ‘She’ or concentrate on ‘shopping’.
  - ➢ A soft window convolution.

Wan S, Lan Y, Guo J, et al. A deep architecture for semantic matching with multiple positional sentence representations//Proceedings of the 30th AAAI Conference on Artificial Intelligence . Phoenix, USA, 2016: 2835-2841.

MV-LSTM

Table 1: Examples of QA dataset.

S_X	*How to get rid of **memory stick error** of my sony cyber shot?*
S_Y^+	*You might want to try to format the **memory stick** but what is the **error** message you are receiving.*
S_Y^-	*Never heard of stack underflow **error**, overflow yes, overflow is due to running out of virtual **memory**.*

Table 2: The effect of pooling parameter k on QA.

	P@1	MRR
LSTM-RNN	0.690	0.822
Bi-LSTM-RNN	0.702	0.830
MV-LSTM (k=1)	0.726	0.843
MV-LSTM (k=3)	0.736	0.849
MV-LSTM (k=5)	0.739	**0.852**
MV-LSTM (k=10)	**0.740**	**0.852**

Table 3: Experimental results on QA.

Model	P@1	MRR
Random Guess	0.200	0.457
BM25	0.579	0.726
ARC-I	0.581	0.756
CNTN	0.626	0.781
LSTM-RNN	0.690	0.822
RAE	0.398	0.652
DeepMatch	0.452	0.679
ARC-II	0.591	0.765
MultiGranCNN	0.725	0.840
MV-LSTM-Cosine	0.739	0.852
MV-LSTM-Bilinear	0.751	0.860
MV-LSTM-Tensor	**0.766**	**0.869**

1. The multiple positional sentence representations is useful to capture detailed local information with context.
2. The matching degree is usually determined by the combination of matchings at different positions.

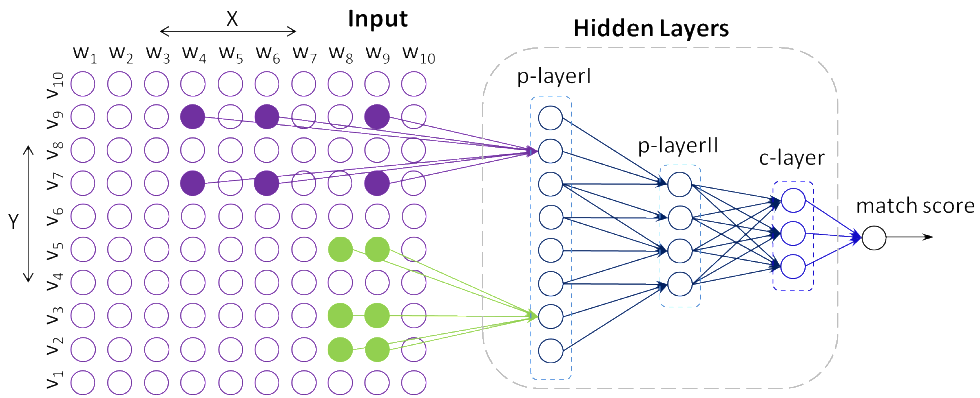
Wan S, Lan Y, Guo J, et al. A deep architecture for semantic matching with multiple positional sentence representations//Proceedings of the 30th AAAI Conference on Artificial Intelligence . Phoenix, USA, 2016: 2835-2841.

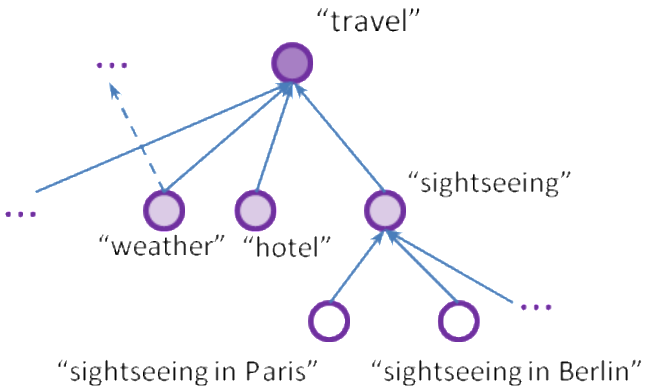
Typical Interaction Focused Methods

- DeepMatch: A Deep Architecture for Matching Short Texts (Lu and Li, NIPS'13)
- ARC II: Convolutional Neural Network Architectures for Matching Natural Language Sentences (Hu et al., NIPS'14)
- MatchPyramid: Text Matching as Image Recognition. (Pang et al. AAAI'16)
- Match-SRNN: Modeling the Recursive Matching Structure with Spatial RNN. (Wan et al. IJCAI'16)
- DRMM: A Deep Relevance Matching Model for Ad-hoc Retrieval. (Guo et al. CIKM'16)
- …

Interaction-focused model: DeepMatch

- Motivation: A good matching function should capture
  - ➢ Localness: a salient local structure in the semantic space of parallel text objects to be matched
  - ➢ Hierarchy: the decision making for matching has different levels of abstractions

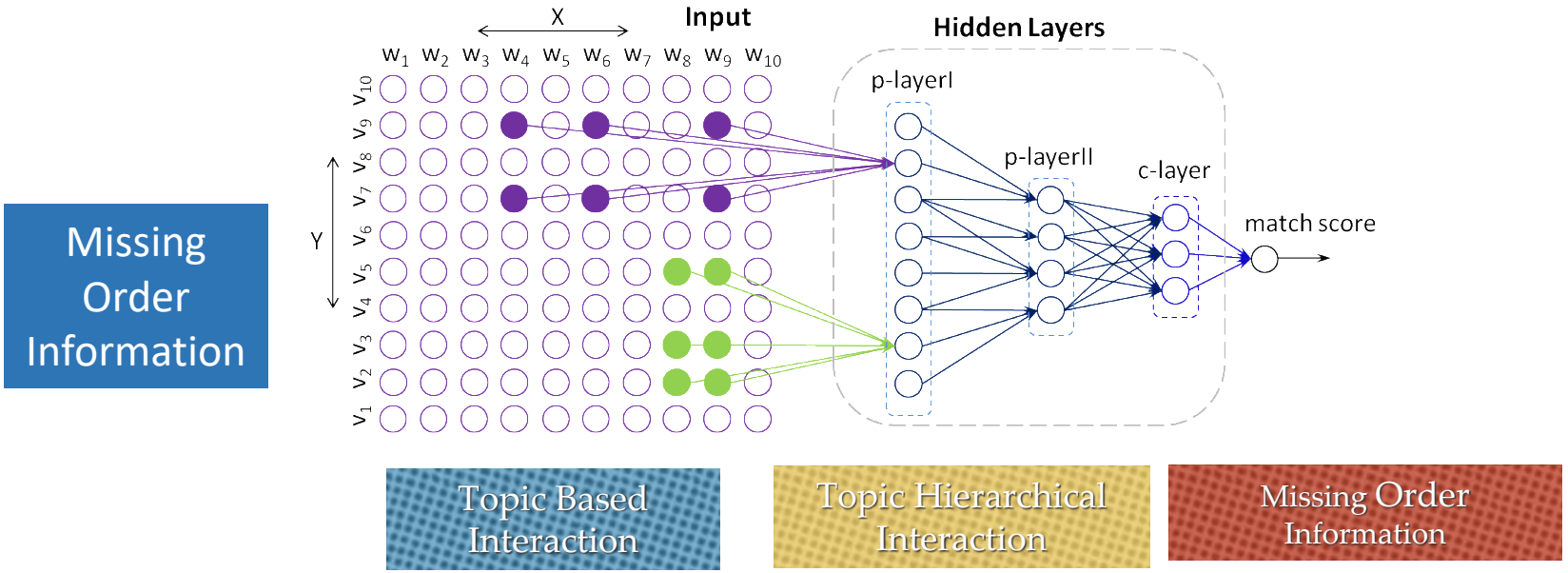




Lu Z, Li H. A deep architecture for matching short texts //NIPS 2013: 1367-1375

Interaction-focused model: DeepMatch

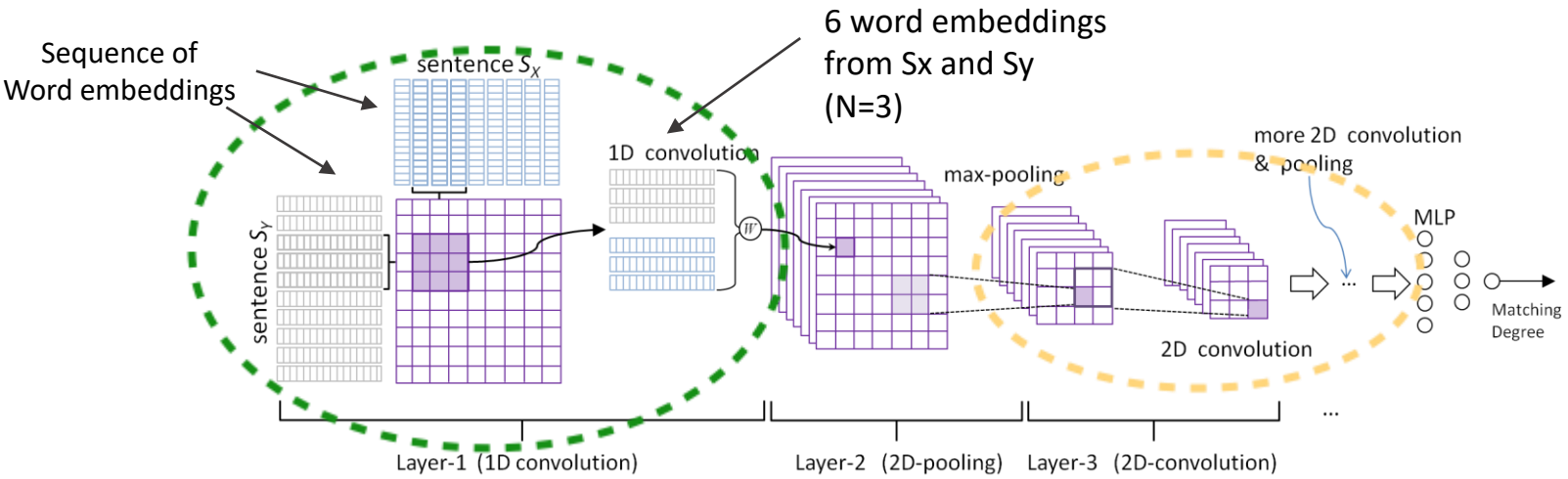
- Basic Interaction: topic model based interaction
  - ➢ Compositional Interaction Structure: topical hierarchies to capture local matching structures
  - ➢ Aggregation Function: MLP to generate the final matching score



Lu Z, Li H. A deep architecture for matching short texts //NIPS 2013: 1367-1375

Interaction-focused model: ARC-II

- Let two sentences meet before their own high-level representations mature.
- Basic Interaction: Phrase sum interaction matrix
- Compositional Interaction Structure: CNN to capture the local interaction structure
- Aggregation Function: MLP



Hu B, Lu Z, Li H, et al. Convolutional neural network architectures for matching natural language sentences//NIPS 2014: 2042-2050

Interaction-focused model: ARC-II

- Order Preservation
- Both the convolution and pooling have this order preserving property.

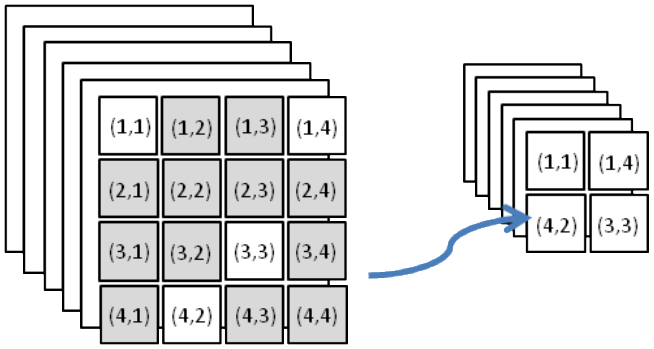


Figure 5: Order preserving in 2D-pooling.

- However, the word level matching signals are lost
- 2-D matching matrix is construct based on the embedding of the words in two N-grams

Hu B, Lu Z, Li H, et al. Convolutional neural network architectures for matching natural language sentences//NIPS 2014: 2042-2050

Interaction-focused model: MatchPyramid

- Inspired by image recognition task
  - ➢ Part 1: Construct Matching Matrix
  - ➢ Part 2: Hierarchical Convolution

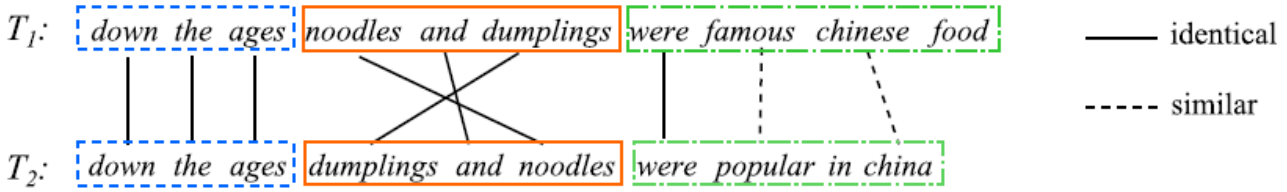
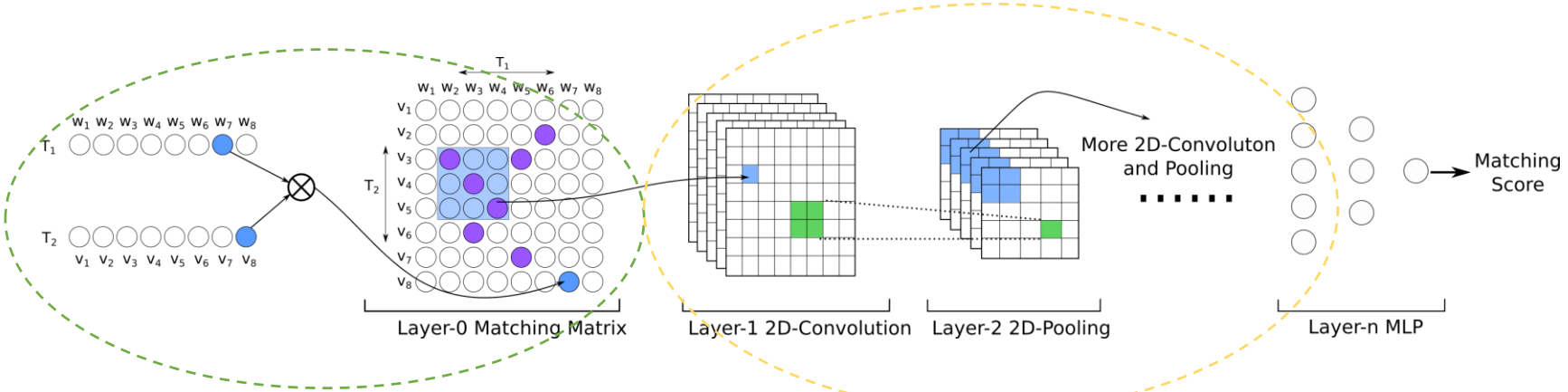


Figure 1: An example of interaction structures in paraphrase identification.

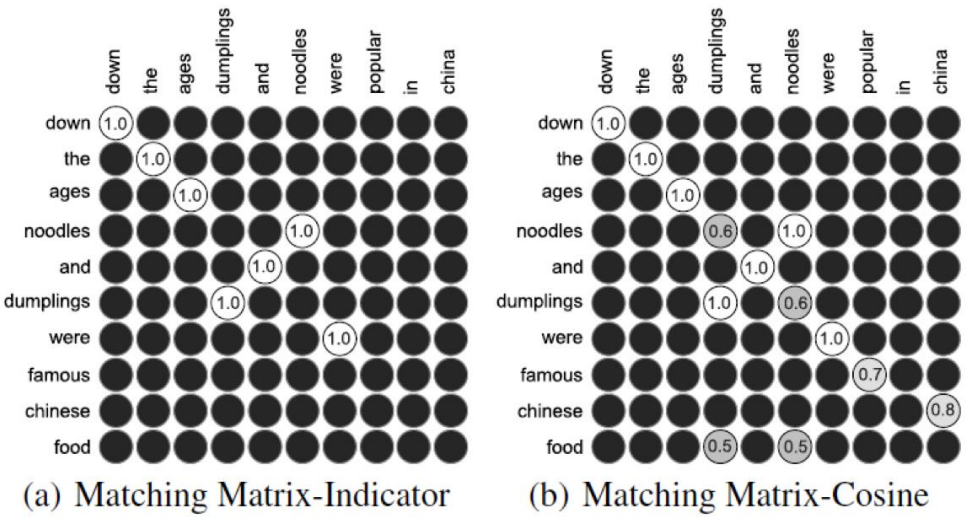


Liang P, Yanyan L, Jiafeng G et al. Text Matching as Image Recognition//AAAI 2016: 2793-2799

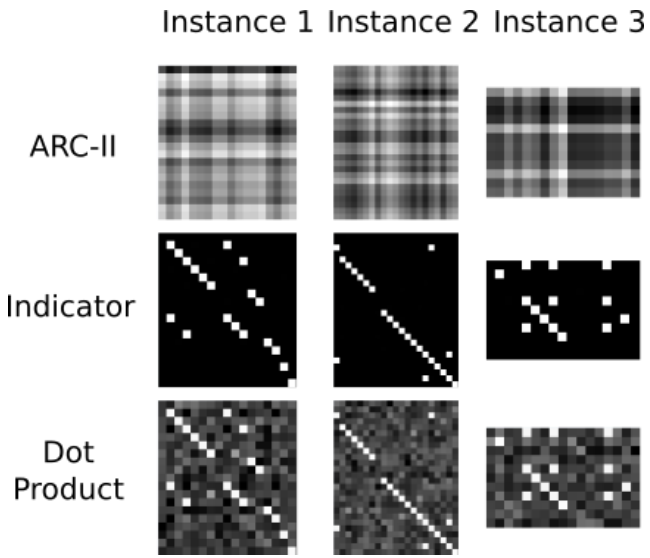
Interaction-focused model: MatchPyramid

- Matching Matrix: Bridging the Gap between Text Matching and Image Recognition.

$$\mathbf{M}_{ij} = w_i \otimes v_j$$



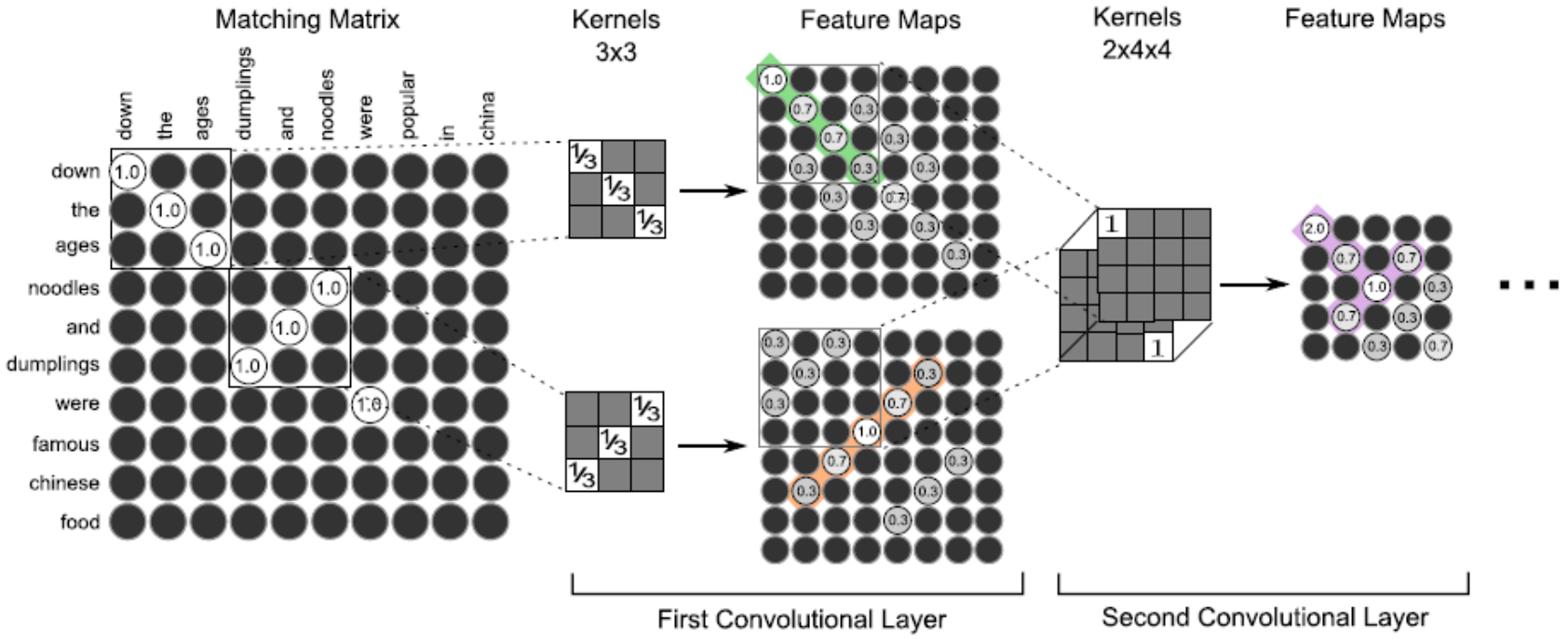
(a) Matching Matrix-Indicator (b) Matching Matrix-Cosine



Liang P, Yanyan L, Jiafeng G et al. Text Matching as Image Recognition//AAAI 2016: 2793-2799

Interaction-focused model: MatchPyramid

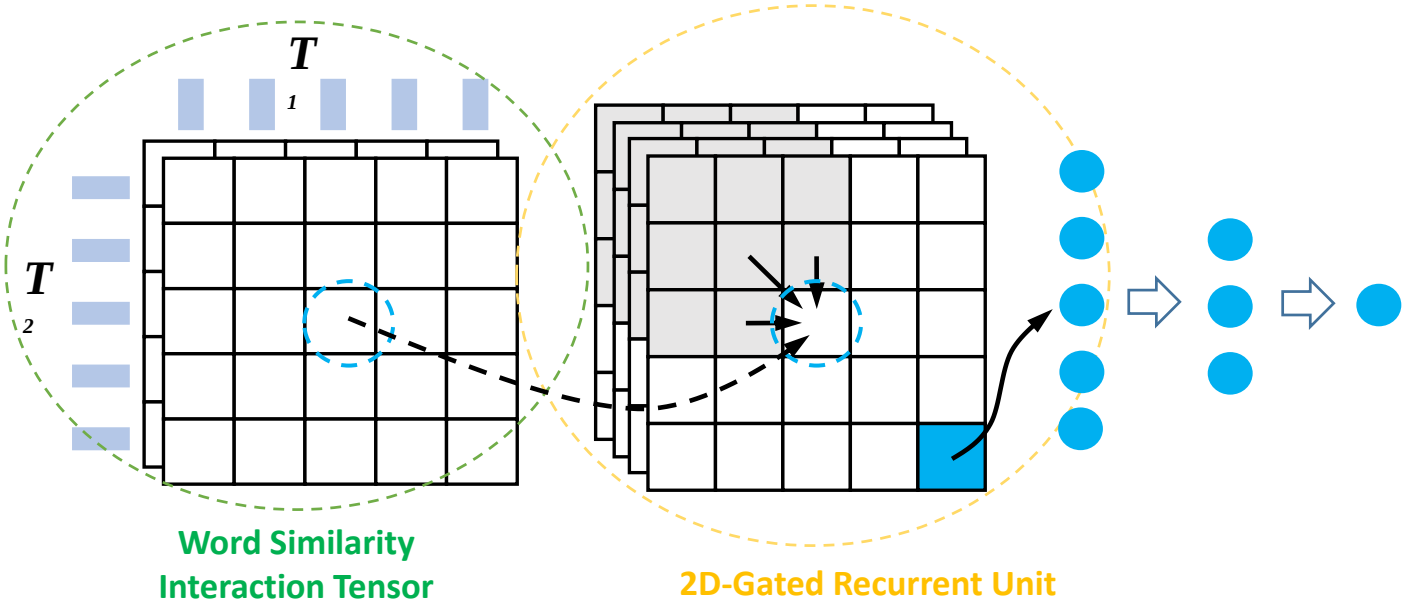
- Hierarchical Convolution: A way to capture rich matching patterns



Liang P, Yanyan L, Jiafeng G et al. Text Matching as Image Recognition//AAAI 2016: 2793-2799

Interaction-focused model: Match-SRNN

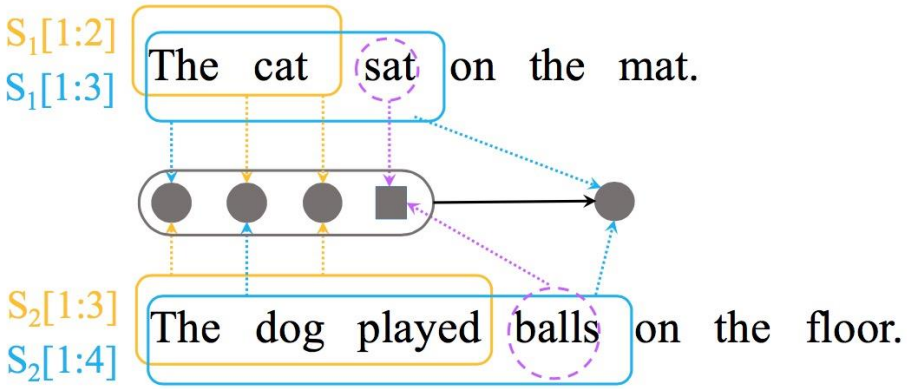
- Spatial recurrent neural network for text matching
- Basic Interaction: word similarity tensor
- Compositional Interaction Structure
- Recursive Matching Structure
- Aggregation Function: MLP

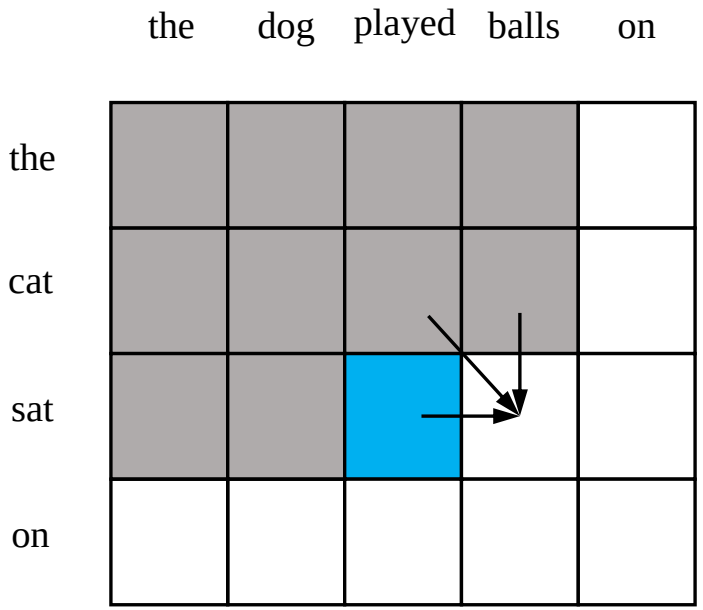


Wan S, Lan Y, Xu J, et al. Match-SRNN: Modeling the recursive matching structure with spatial rnn[J]. IJCAI 2016

Interaction-focused model: Match-SRNN

Recursive Matching Structure





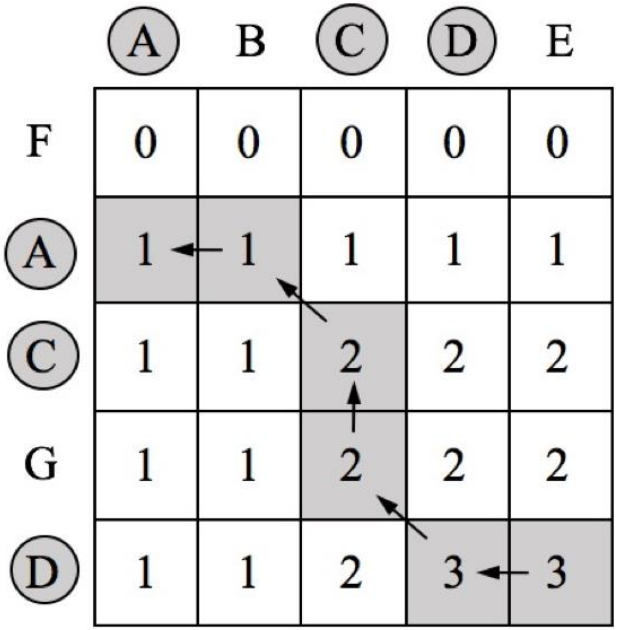
- We can see all matching between sub sentences have been utilized
  - The can sat ←→ The Dog played balls
  - The can sat ←→ The Dog played

Wan S, Lan Y, Xu J, et al. Match-SRNN: Modeling the recursive matching structure with spatial rnn[J]. IJCAI 2016

Interaction-focused model: Match-SRNN

Connection with LCS

- Theoretical Analysis
  - Longest Common Sub-Sequence
    - S1: A B C D E
    - S2: F A C G D
    - LCS: A C D



  - Dynamic Programming (DP)
    - Step function:

$$c[i,j] = \max(c[i,j-1], c[i-1,j], c[i-1,j-1] + \mathbb{I}_{\{x_i = y_j\}})$$

    - Backtrace: Depends on the selection of *"max"* operation

Wan S, Lan Y, Xu J, et al. Match-SRNN: Modeling the recursive matching structure with spatial rnn[J]. IJCAI 2016

Interaction-focused model

QA Task (Yahoo Data)

	Model	P@1	MRR
Statistic Traditional	Random	0.200	0.457
	BM25	0.579	0.726
Comosition Focused	ARC-I	0.581	0.756
	CNTN	0.626	0.781
	LSTM-RNN	0.690	0.822
	uRAE	0.398	0.652
	MultiGranCN	0.725	0.840
	MV-LSTM	0.766	0.869
Interaction Focused	DeepMatch	0.452	0.679
	ARC-II	0.591	0.765
	MatchPyramid	0.764	0.867
	Match-SRNN	0.790	0.882

No.2 No.1 No.3

PI Task (MSRP Data)

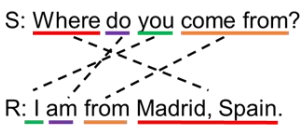
	Model	Accuracy(%)	F1(%)
Statistic Traditional	All positive	66.50	79.87
	TF-IDF	70.31	77.62
Composition Focused	DSSM	70.09	80.96
	CDSSM	69.80	80.42
	ARC-I	69.60	80.27
	uRAE	76.80	83.60
	MultiGranCN	78.10	84.40
Interaction Focused	MV-LSTM	75.40	82.80
	ARC-II	69.90	80.91
	MatchPyramid	75.94	83.01
	Match-SRNN	74.50	81.70

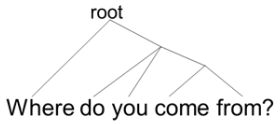
- LSTM is powerful than CNN?
  - Long term dependencies are important for the semantic understanding
- Interaction focused methods performs better
  - Interaction is more important than sentence representations

- CNN is powerful than LSTM?
  - Locally representation/interaction is enough for PI task
- Composition-focused methods performs better
  - sentence representations is more important than interaction

Interaction-focused model: DRMM

- There are large differences between relevance matching in ad-hoc retrieval and semantic matching in NLP tasks
  - ➢ Affect the design of deep model architectures
  - ➢ No "one-fit-all" matching models





S: Where do you come from?
R: I am from Madrid, Spain. ✔
R: I am a student. ✖

- **Similarity Matching Signals**
  - Important to capture the semantic similarity/relatedness;
- **Compositional meanings**
  - Natural language sentences;
  - Compositional meaning based on their grammatical structures;
- **Global matching requirement**
  - Limited lengths and concentrated topic scope;
  - Treat the text as a whole to infer the semantic relations

Question Answering
Automatic Conversation

- **Exact matching signals**
  - The exact matching of terms is still the most important signal in ad-hoc retrieval;
  - Indexing and search paradigm in modern search engines;
- **Query term importance**
  - Query: mainly short and keyword based without complex grammar
  - Critical to take into account term importance
- **Diverse matching requirement**
  - Long document: Different hypotheses concerning document length in ad-hoc retrieval

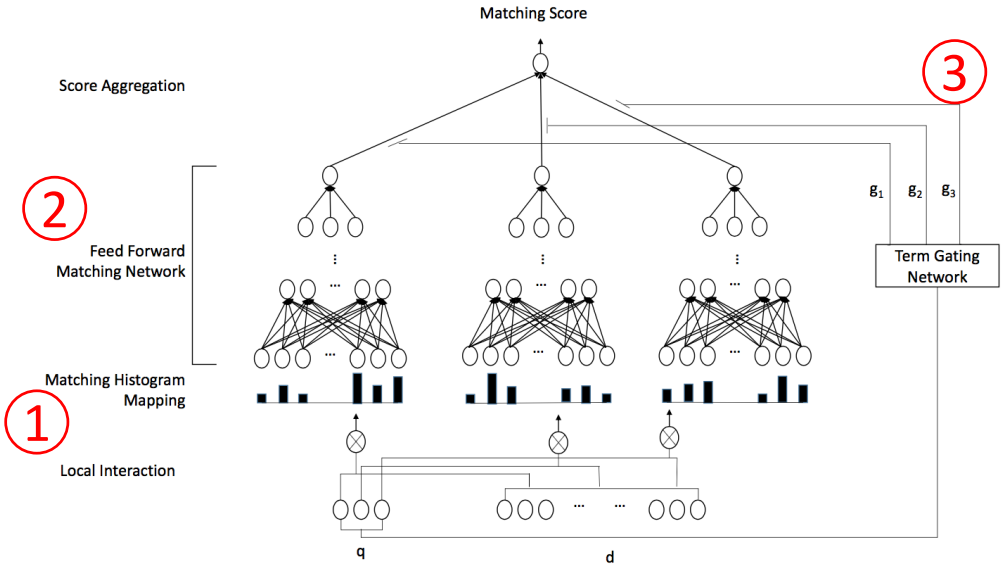
1. ***Verbosity Hypothesis***
   - Global relevance
2. ***Scope Hypothesis***
   - Relevance matching could happen in any part of a document

Ad-hoc Retrieval

Jiafeng G, Yixing F, Qingyao A et al. A Deep Relevance Matching Model for ad-hoc retrieval//CIKM 2016: 55-64

Interaction-focused model: DRMM

1. Matching histogram:
   - map the varied-size interactions into a fixed-length representation
   - Position-free but strength-focused
2. Feed forward Matching Network:
   - Extract hierarchical matching patterns from different levels of interaction signals.
3. Term Gating Network:
   - Control how much relevance score on each query term contribute to the final relevance score.



A joint deep architecture designed for relevance matching

Jiafeng G, Yixing F, Qingyao A et al. A Deep Relevance Matching Model for ad-hoc retrieval//CIKM 2016: 55-64

Interaction-focused model: DRMM

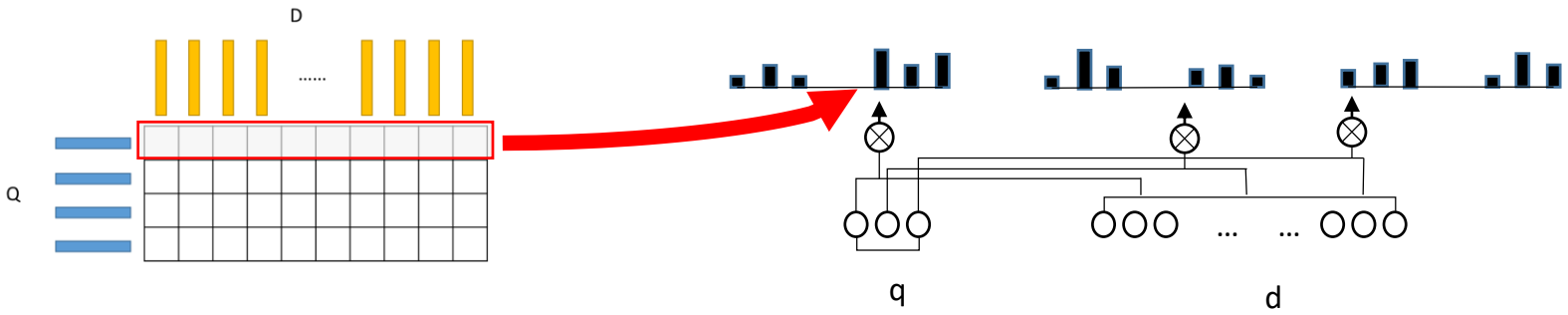
Different Input Representations

Existing matching models

Matching matrix
- position preserving
- zero-padding
- signals are equal

Our Model

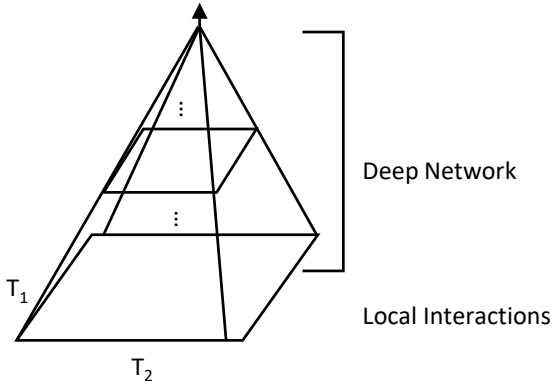
Matching histogram
- strength preserving
- no need for padding
- distinguish exact/similarity signals

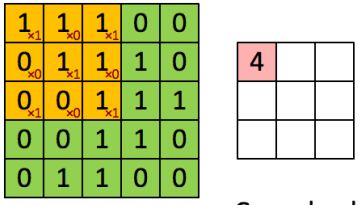


Interaction-focused model: DRMM

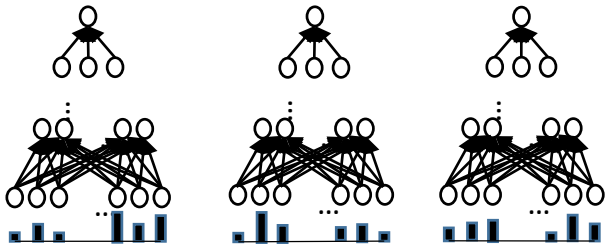
Different Model Architectures

- Existing matching models: CNNs base on matching matrix
  - Learn positional regularities in matching patterns
    - Suitable for image recognition and global matching requirement (i.e., all the positions are important)
    - Not suitable for diverse matching requirement (i.e., no positional regularity)





- Our method: DNN on matching histogram
  - Learn position-free but strength-focused patterns
  - Explicitly model term importance



Interaction-focused model: DRMM

Robust-04 collection(TREC data)

Significant improvement or degradation with respect to QL is indicated (+/-)

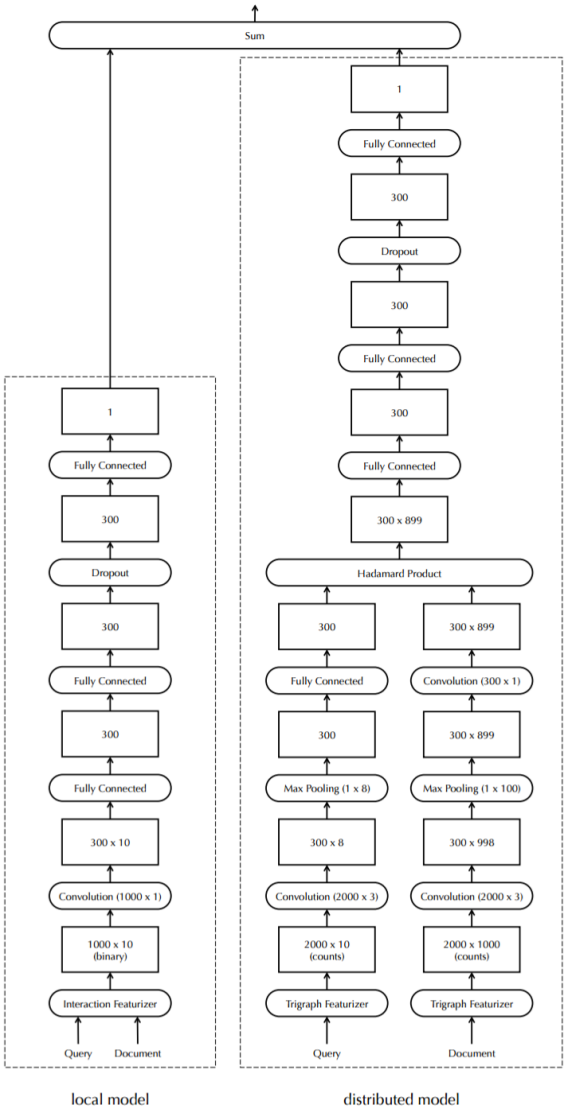
Model Type	Model Name	Topic Titles			Topic Descriptions		
		MAP	nDCG@20	P@20	MAP	nDCG@20	P@20
Traditional Retrieval Baselines	QL	0.253	0.415	0.369	0.246	0.391	0.334
	BM25	0.255	0.418	0.370	0.241	0.399	0.337
Deep Learning Baselines	$DSSM_T$	0.095^-	0.201^-	0.171^-	0.078^-	0.169^-	0.145^-
	$CDSSM_T$	0.067^-	0.146^-	0.125^-	0.050^-	0.113^-	0.093^-
	ARC-I	0.041^-	0.066^-	0.065^-	0.030^-	0.047^-	0.045^-
	ARC-II	0.067^-	0.147^-	0.128^-	0.042^-	0.086^-	0.074^-
	MP_{COS}	0.189^-	0.330^-	0.290^-	0.094^-	0.190^-	0.162^-
Our Approach	$DRMM_{LCH\times IDF}$	**0.279**$^+$	**0.431**$^+$	**0.382**$^+$	**0.275**$^+$	**0.437**$^+$	**0.371**$^+$

1. All the deep learning baselines perform significantly worse than the traditional retrieval models
2. The performance on topic descriptions can be comparable to that on topic titles

Jiafeng G, Yixing F, Qingyao A et al. A Deep Relevance Matching Model for ad-hoc retrieval//CIKM 2016: 55-64

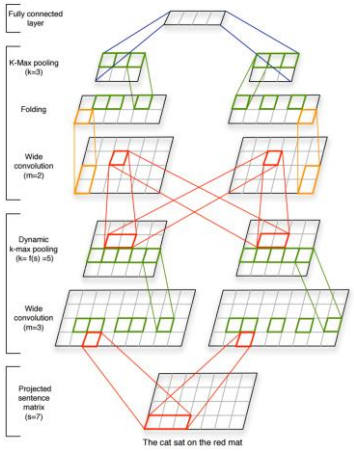
Joint modeling: Ad-hoc retrieval using local and distributed representation

- Argues both “lexical” and “semantic” matching is important for document ranking
- Duet model is a linear combination of two DNNs using local and distributed representations of query/document as inputs, and jointly trained on labelled data
- Local model operates on lexical interaction matrix
- Distributed model operates on *n*-graph representation of query and document text

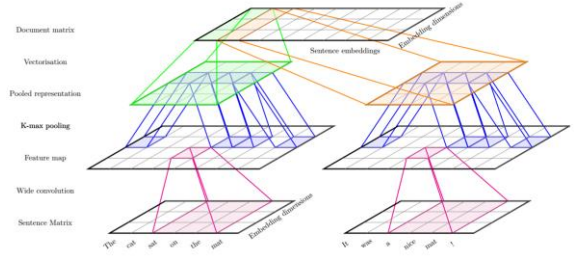


Mitra B, Diaz F, Craswell N. Learning to Match using Local and Distributed Representations of Text for Web Search//WWW 2017: 1291-1299

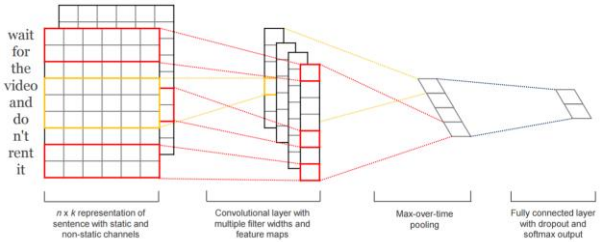
Related Neural Models on Non-IR tasks



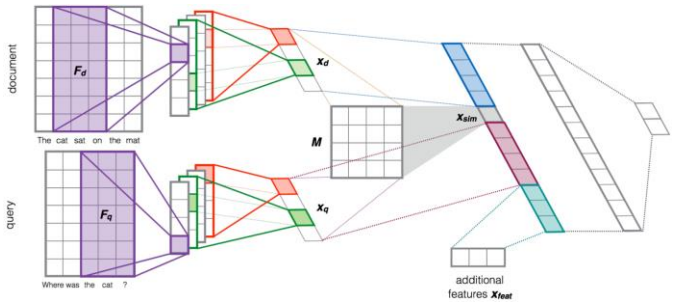
(Kalchbrenner et al., 2014)



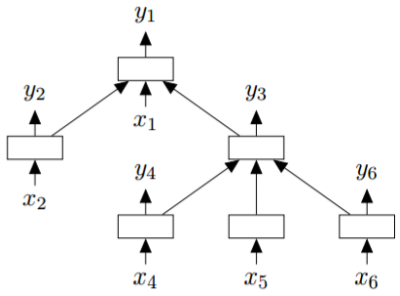
(Denil et al., 2014)



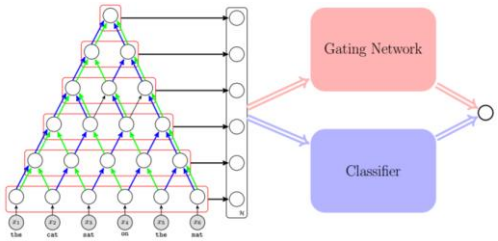
(Kim, 2014)



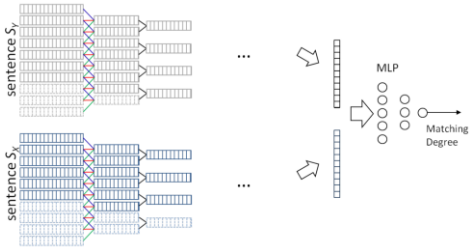
(Severyn and Moschitti, 2015)



(Tai et al., 2015)



(Zhao et al., 2015)



(Hu et al., 2014)

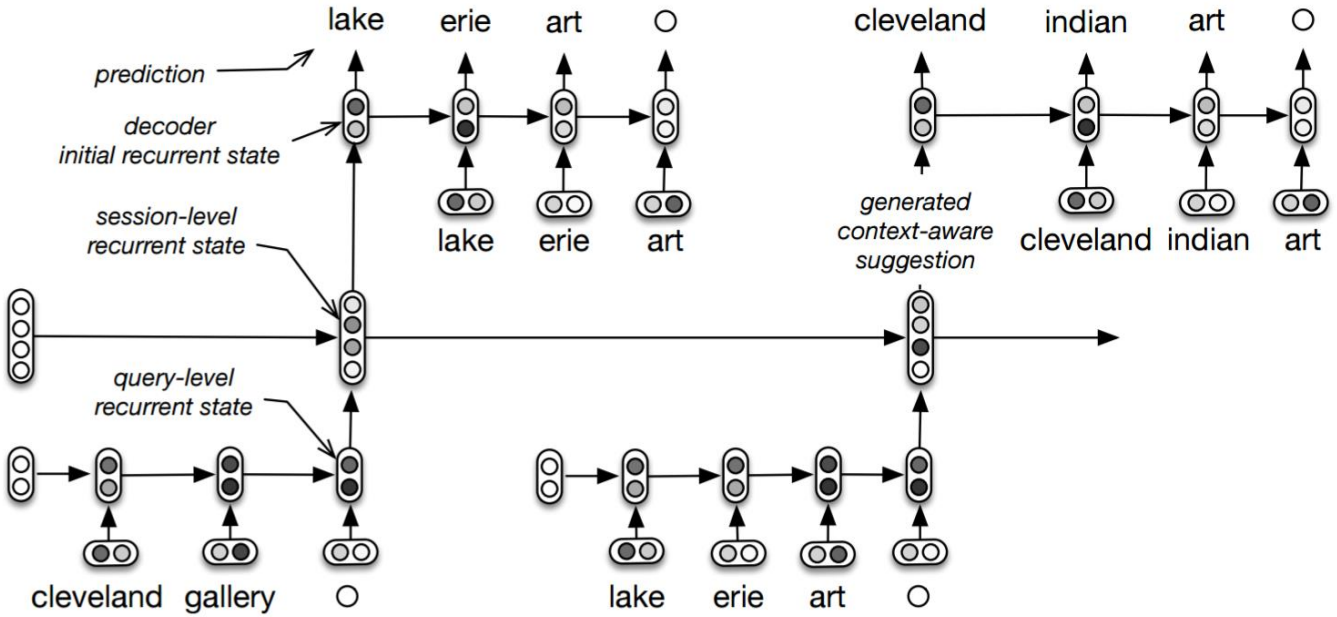
Neural Network Approaches to IR

Task	Related Work
Ad-hoc Retrieval	BP-ANN (Yang et al. (2016a)), CDNN (Severyn and Moschitti (2015)), CDSSM (Shen et al. (2014b)), CLSM ((Shen et al., 2014a)), DSSM (Huang et al. (2013)), DRMM (Guo et al. (2016)), GDSSM (Ye et al. (2015)), Gupta et al. (2014), Li et al. (2014), Nguyen et al. (2016), QEM (Sordoni et al. (2014))
Conversational Agents	DL2R (Yan et al. (2016))
Proactive Search	Luukkonen et al. (2016)
Query Autocompletion	Mitra (2015); Mitra and Craswell (2015)
Query Suggestion	Sordoni et al. (2015)
Question Answering	BLSTM (Wang and Nyberg (2015)), CDNN (Severyn and Moschitti (2015)), DFFN (Suggu et al. (2016)), DL2R (Yan et al. (2016)), Yu et al. (2014)
Recommendation	Gao et al. (2014), Song et al. (2016)
Related Document Search	Salakhutdinov and Hinton (2009)
Result Diversification	Xia et al. (2016)
Sponsored Search	Zhang et al. (2016a)
Summarizing Retrieved Documents	Lioma et al. (2016)
Temporal IR	Kanhabua et al. (2016)

Updated table based on Ye Zhang et al. Neural Information Retrieval: A literature Review

Other tasks - Query Recommendation

- Hierarchical sequence-to-sequence model for term-by-term query generation
- Similar to ad-hoc ranking the DNN feature alone perform poorly but shows significant improvements over a model with lexical contextual features.



Alessandro S, Yoshua B, Hossein V. et al. A hierarchical recurrent encoder-decoder for generative context-aware query suggestion[c]//CIKM 2015: 553-562

Other tasks - Query auto-completion

- Given a (rare) query prefix retrieve relevant suffixes from a fixed set
- CDSSM model trained on query prefix-suffix pairs can be used for suffix ranking ("breaking bad cast" → "breaking", "bad cast")
- Training on prefix-suffix produces a leads to a more "Typical" embedding space

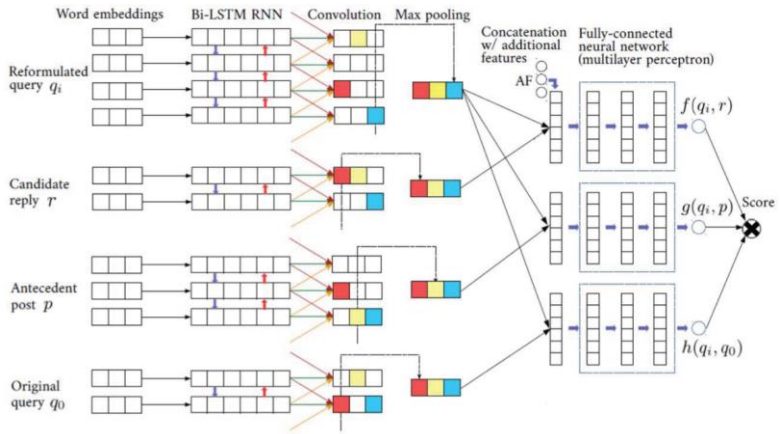
what to cook with chicken and broccoli and
what to cook with chicken and broccoli *and bacon*
what to cook with chicken and broccoli *and noodles*
what to cook with chicken and broccoli *and brown sugar*
what to cook with chicken and broccoli *and garlic*
what to cook with chicken and broccoli *and orange juice*
what to cook with chicken and broccoli *and beans*
what to cook with chicken and broccoli *and onions*
what to cook with chicken and broccoli *and ham soup*

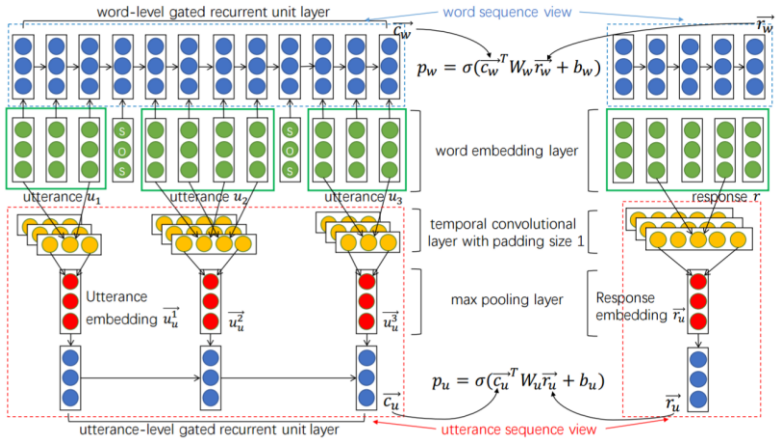
cheapest flights from seattle to
cheapest flights from seattle *to dc*
cheapest flights from seattle *to washington dc*
cheapest flights from seattle *to bermuda*
cheapest flights from seattle *to bahamas*
cheapest flights from seattle *to aruba*
cheapest flights from seattle *to punta cana*
cheapest flights from seattle *to airport*
cheapest flights from seattle *to miami*

Mitra B, Craswell N. Query auto-completion for rare prefixes[c]//CIKM 2015: 1755-1758

Other tasks - Conversational response retrieval

- Ranking responses for conversational systems
- Interesting challenges in modelling utterance level discourse structure and context
- Can multi-turn conversational tasks become a key playground for neural retrieval models?



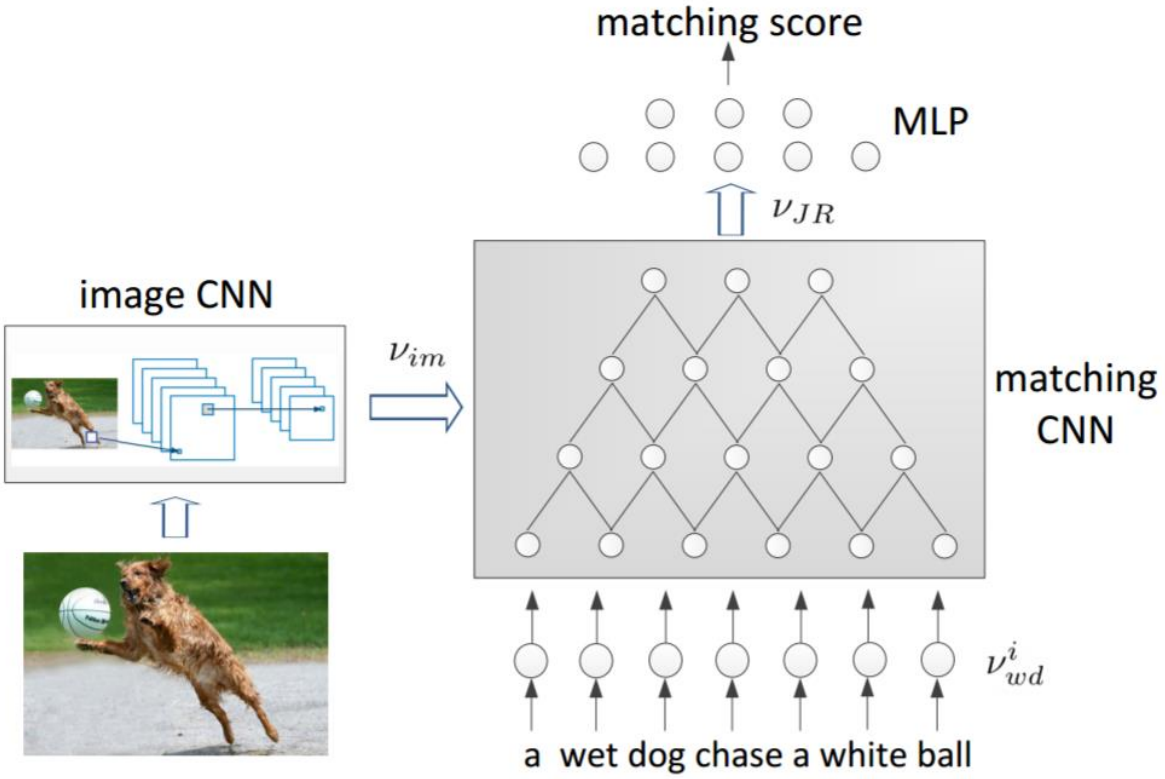


Yan R, Song Y, Wu Learning to respond with deep neural networks for retrieval-based human-computer conversation system//SIGIR 2016:55-64

Zhou X, Dong D, Wu H, et al. Multi-view Reponse Selection for Human-Computer Conversation [C] //EMNLP 2016: 372-381.

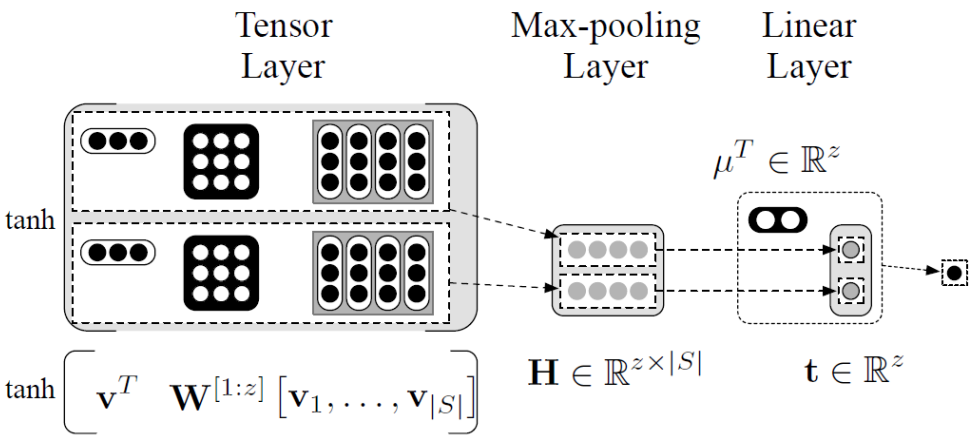
Other tasks - Multimodal retrieval

- Neural representation learning is also leading towards breakthroughs in multimodal retrieval



Lin M, Zhengdong L, Lifeng S et al. Multimodal Convolutional Neural Network for matching Image and sentence//CVPR 2015 2623-2631

Other tasks – Diverse Ranking



Neural Tensor Network for Search Result Diversification

- Diverse ranking as sequential document selection
- Ranking model f(d, s) selects a document per iteration

$$f(d,S) = g_r(\mathbf{x}) + g_n(\mathbf{v},S)$$
$$= \omega^T \mathbf{x} + \mu^T \max\left\{\tanh\left(\mathbf{v}^T \mathbf{W}^{[1:z]}\left[\mathbf{v}_1,\ldots,\mathbf{v}_{|S|}\right]\right)\right\}$$

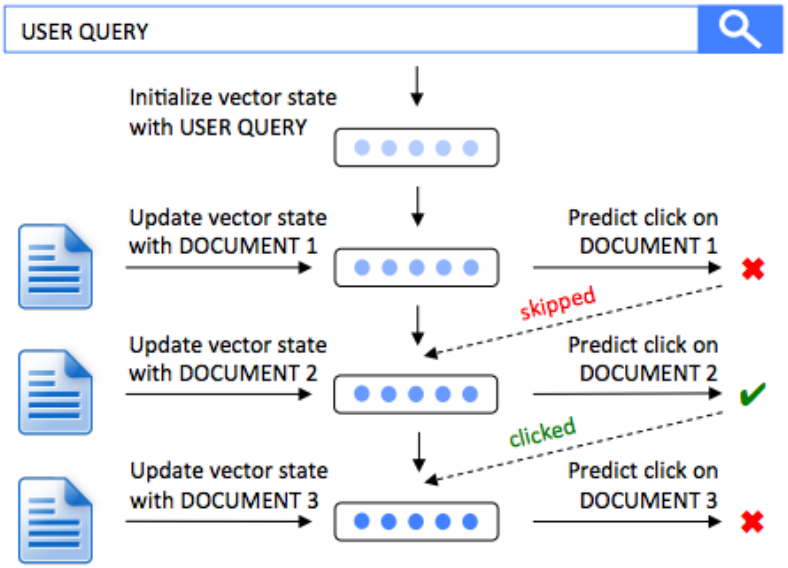
- Relevance: linear combination of relevance features
- Novelty: calculated with modified NTN

Long Xia, Jun Xu, Yanyan Lan, Jiafeng Guo, Xueqi Cheng, Modeling Document Novelty with Neural Tensor Network for Search Result Diversification, the 39th Annual ACM SIGIR Conference, Pisa, Italy (SIGIR 2016)

Other tasks – User Behavior Modeling

q – user query
d_r – document at rank r

i_r – user interaction with document at rank r



$$\mathbf{s}_0 = \mathcal{I}(q)$$
$$\mathbf{s}_{r+1} = \mathcal{U}(\mathbf{s}_r, i_r, d_{r+1})$$

$$P(C_{r+1} = 1 \mid q, i_1, \ldots, i_r, d_1, \ldots, d_{r+1}) = \mathcal{F}(\mathbf{s}_{r+1})$$

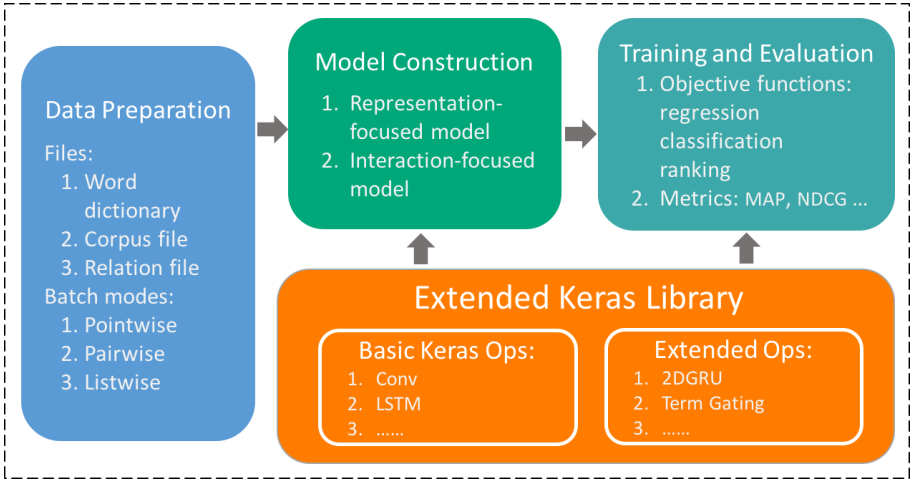
$\mathcal{I}$: Feed-forward neural network
$\mathcal{U}$: Recurrent neural network (RNN, LSTM)
$\mathcal{F}$: Feed-forward neural network (with one output unit and sigmoid activation function)

Alexey Borisov, Ilya Markov Marteen de Rijke et al, A Neural Click Model for Web Search, WWW 2016

NeuIR Toolkit - MatchZoo

MatchZoo is a toolkit that aims to facilitate the designing, comparing and sharing of deep text matching models.

1. A **unified** data preparation module for different text matching problems.
2. A **flexible** layer-based model construction process
3. Implemented two schools of representative deep text matching models: representation-focused and interaction-focused models.

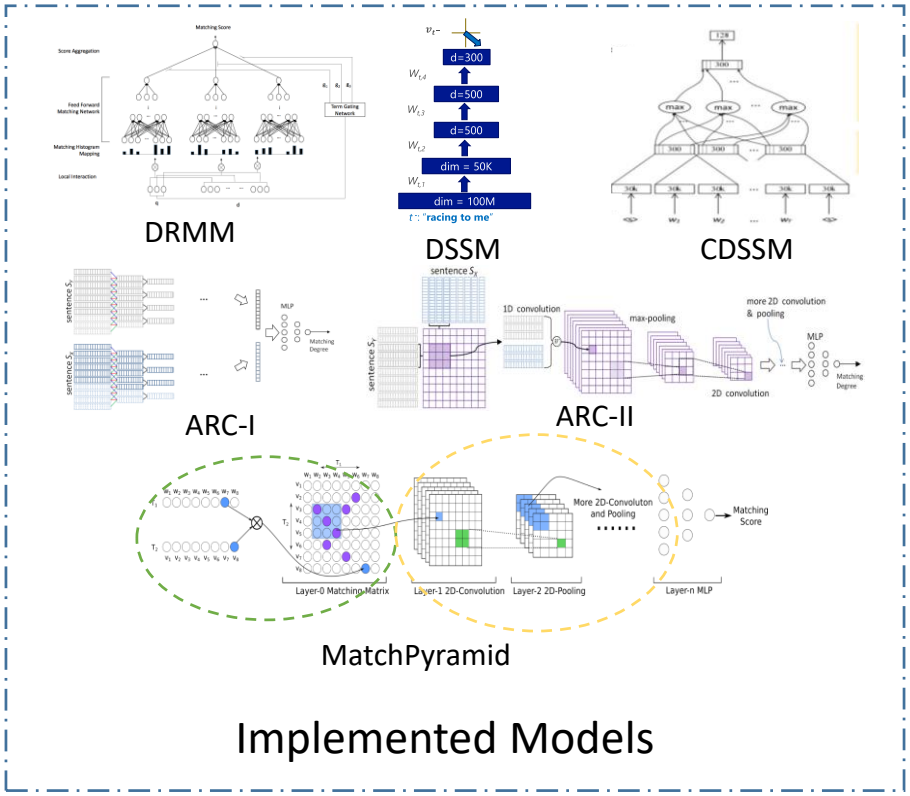


Fan, Yixing, Liang Pang, JianPeng Hou, Jiafeng Guo, et al. MatchZoo: A Toolkit for Deep Text Matching. SIGIR Workshop 2017

NeuIR Toolkit - MatchZoo

MatchZoo is a toolkit that aims to facilitate the designing, comparing and sharing of deep text matching models.

1. Data preparation: convert dataset of different text matching tasks into a unified format.
2. Model construction: build the deep matching model layer by layer based on keras library.
3. Training and Evaluation: provide a variety of objective functions for regression, classification, and ranking.

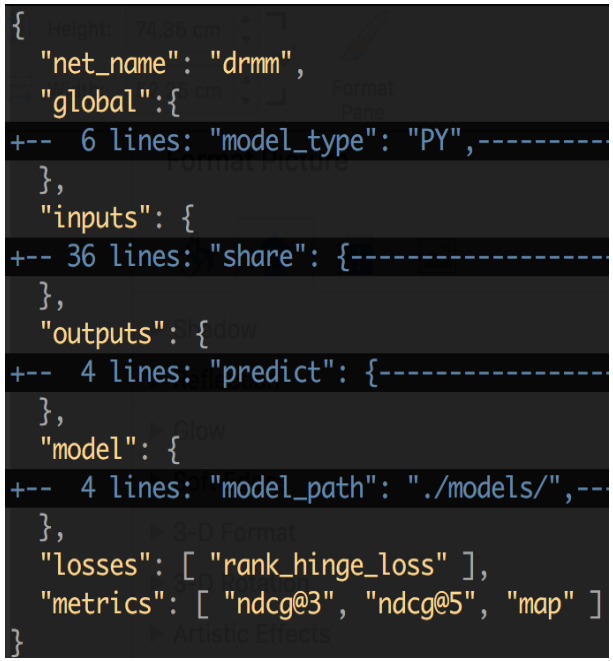


Fan, Yixing, Liang Pang, JianPeng Hou, Jiafeng Guo, et al. MatchZoo: A Toolkit for Deep Text Matching. SIGIR Workshop 2017

NeuIR Toolkit - MatchZoo

MatchZoo is a toolkit that aims to facilitate the designing, comparing and sharing of deep text matching models.

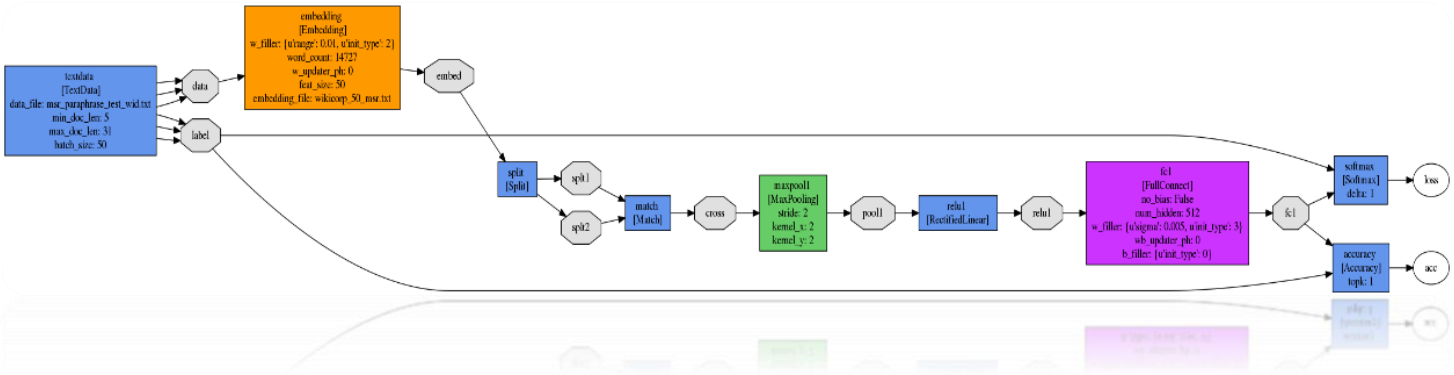
- Install:
  - git clone https://github.com/fanechion/MatchZoo.git
  - cd MatchZoo
  - python setup.py install
- Configure:
  1. global: global parameters such as learning rate, epochs, and batch_size.
  2. inputs: datasets for training, validation, and prediction.
  3. outputs: the predicted results of the inputs dataset.
  4. model: the core matching model and its hyper-parameters.
  5. losses and metrics: the objective and evaluation of the model.
- Run:
  - Train: python –phase train –model_file models/drmm.config
  - Predict: python –phase predict –model_file models/drmm.config



Fan, Yixing, Liang Pang, JianPeng Hou, Jiafeng Guo, et al. MatchZoo: A Toolkit for Deep Text Matching. SIGIR Workshop 2017

Other Useful Resources

- TextNet
  - Focus on text data, **Sparsity** and **Variance Length**.
  - Support **JSON** config file to construct **DAG** networks.



Pyndri for Python: https://github.com/cvangysel/pyndri

Luandri for LUA / Torch: https://github.com/bmitra-msft/Luandri

NeuIR Workshop – Join Us!

- Hottest Workshop in SIGIR 2016-2017

# of registrations
121
NeuIR 2016

# of registrations
178
NeuIR 2017

- Topics: fundamental challenges, best practice, novel applications, shared repository, large scale benchmarks, …


Nick Craswell


Bruce Croft


Maarten de Rijke


Jiafeng Guo


Bhaskar Mitra

Send me your questions and feedback during or after the tutorial

Jiafeng Guo(郭嘉丰)

 guojiafeng@ict.ac.cn

References

- [Singhal et al. 1996] Pivoted document length normalizaLon. A. Singhal, C. Buckley and M. Mitra. SIGIR 1996.
- [Montúfar et al. 2014] Montúfar, Pascanu, Cho and Bengio. On the number of linear regions of deep neural networks NIPS 2014
- [Larrson et al. 2016] Larsson G, Maire M, Shakhnarovich G. Fractalnet: Ultra-deep neural networks without residuals [J]// axXiv preprint: 1605.07648, 2016.
- [He et al. 2015] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition [C] //CVPR 2016: 770-778.
- [Szegedy et al. 2016] Szegedy C, Liu W, Jia Y, et al. Going deeper with convolutions [C] //CVPR 2015: 1-9.
- [Huang et al. 2013] Huang et al. Learning deep structured semantic models for web search using clickthrough data, 2013 CIKM.
- [Shen et al. 2014] Shen et al. A latent semantic model with convolutional-pooling structure for information retrieval. 2014 CIKM
- [Palangi et al. 2016] Palangi, et al. Deep sentence embedding using long short-term memory networks TASLP 2016
- [Wan et al. 2016] Wan S, Lan Y, Guo J, et al. A deep architecture for semantic matching with multiple positional sentence representations//Proceedings of the 30th AAAI Conference on Artificial Intelligence . Phoenix, USA, 2016: 2835-2841.
- [Lin et al. 2015] Lin M, Zhengdong L, Lifeng S et al. Multimodal Convolutional Neural Network for matching Image and sentence//CVPR 2015 2623-2631

References

- [Guo et al. 2016] Jiafeng G, Yixing F, Qingyao A et al. A Deep Relevance Matching Model for ad-hoc retrieval//CIKM 2016: 55-64
- [Alessandro et al. 2015] Alessandro S, Yoshua B, Hossein V. et al. A hierarchical recurrent encoder-decoder for generative context-aware query suggestion[c]//CIKM 2015: 553-562
- [Mitra et al. 2015] Mitra B, Craswell N. Query auto-completion for rare prefixes[c]//CIKM 2015: 1755-1758
- [Yan et al. 2016] Yan R, Song Y, Wu Learning to respond with deep neural networks for retrieval-based human-computer conversation system//SIGIR 2016:55-64
- [Zhou et al. 2016] Zhou X, Dong D, Wu H, et al. Multi-view Reponse Selection for Human-Computer Conversation [C] //EMNLP 2016: 372-381.
- [Denil et al. 2014] Denil M, Demiraj A, Kalchbrenner N, et al. Modeling, visualising and summarising documents with a single convolutional neural network[J]. arXiv preprint arXiv: 1406.3830, 2014.
- [Mitra et al. 2017] Mitra B, Diaz F, Craswell N. Learning to Match using Local and Distributed Representations of Text for Web Search//WWW 2017: 1291-1299
- [Severyn and Moschitti 2015] Severyn A, Moschitti A. Learning to rank short text pairs with convolutional deep neural networks [C]// SIGIR 2015: 373-382.
- [Kalchbrenner et al. 2014] Kalchbrenner N, Grefenstette E, Blunsom P. A convolutional neural network for modeling sentences[J]. arXiv preprint: 1404.2188, 2014.
- [Kim et al. 2014] Kim Y. Convolutional neural networks for sentence classification [J]. arXiv preprint: 1408.5882, 2014.

References

- [Lu et al. 2013] Lu Z, Li H. A deep architecture for matching short texts //NIPS 2013: 1367-1375
- [Hu et al. 2014] Hu B, Lu Z, Li H, et al. Convolutional neural network architectures for matching natural language sentences//NIPS 2014: 2042-2050
- [Wan et al. 2016] Wan S, Lan Y, Xu J, et al. Match-SRNN: Modeling the recursive matching structure with spatial rnn[J]. IJCAI 2016
- [Tai et al. 2015] Tai K S, Socher R, Manning C D. Improved semantic representations from tree-structured long short-term memory networks[J]. arXiv preprint arXiv: 1503.00075, 2015.
- [Liang et al. 2016] Liang P, Yanyan L, Jiafeng G et al. Text Matching as Image Recognition//AAAI 2016: 2793-2799
- [Zhou et al. 2015] Zhao H, Lu Z. Poupart P. Self-adaptive Hierarchical Sentence Model[C] ///IJCAI. 2015: 4069-4076
- [Long et al. 2016] Long Xia, Jun Xu, Yanyan Lan, Jiafeng Guo, Xueqi Cheng, Modeling Document Novelty with Neural Tensor Network for Search Result Diversification, the 39th Annual ACM SIGIR Conference, Pisa, Italy (SIGIR 2016)