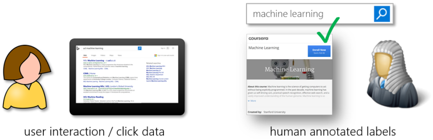
Outline

Supervised text matching

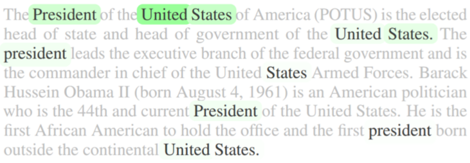
Traditional IR data consists of search queries and document collection

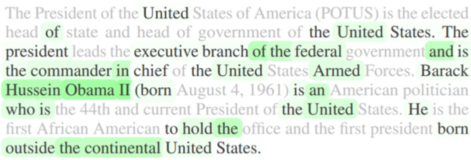
Ground truth can be based on explicit human judgments or implicit user behaviour data (e.g., clickthrough rate)



Lexical vs. Semantic matching

Query: united states president


The President of the United States of America (POTUS) is the elected head of state and head of government of the United States. The president leads the executive branch of the federal government and is the commander in chief of the United States Armed Forces. Barack Hussein Obama II (born August 4, 1961) is an American politician who is the 44th and current President of the United States. He is the first African American to hold the office and the first president born outside the continental United States.


The President of the United States of America (POTUS) is the elected head of state and head of government of the United States. The president leads the executive branch of the federal government and is the commander in chief of the United States Armed Forces. Barack Hussein Obama II (born August 4, 1961) is an American politician who is the 44th and current President of the United States. He is the first African American to hold the office and the first president born outside the continental United States.

Traditional IR models estimate relevance based on lexical matches of query terms in document

Representation learning based models garner evidence of relevance from all document terms based on semantic matches with query

Both lexical and semantic matching are important and can be modelled with **neural networks**

Outline

Semantic matching

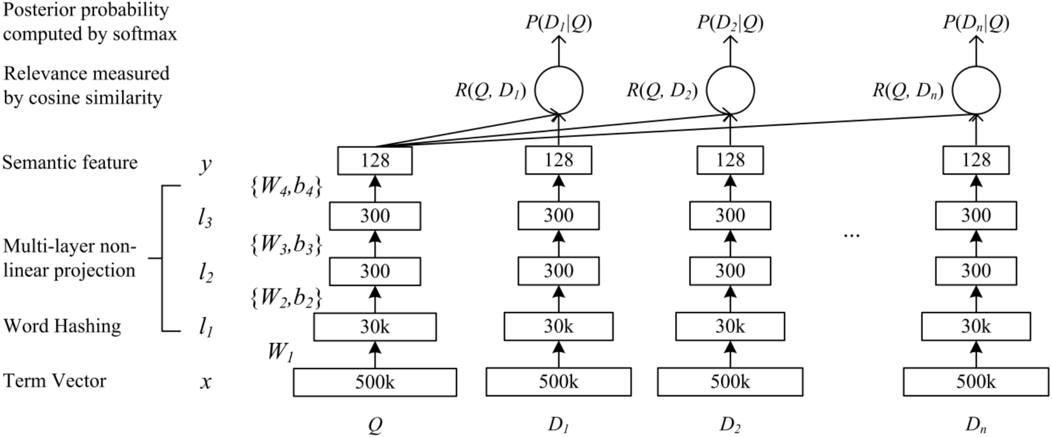
Pros

- Ability to match synonyms and related words
- Robustness to spelling variations
  ($\approx 10\%$ of search queries contain spelling errors)
- Helps in cases where lexical matching fails

Cons

- More computationally expensive than lexical matching

Deep Structured Semantic Model (DSSM) [Huang et al., 2013]

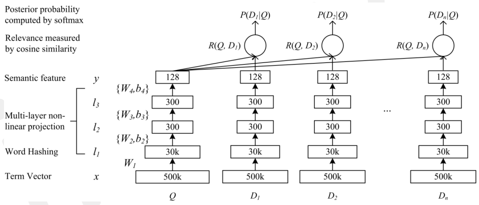


Deep Structured Semantic Model (DSSM) [Huang et al., 2013]

DSSM - Siamese Network

1. Represent query and document as vectors $\mathbf{q}$ and $\mathbf{d}$ in a latent vector space

2. Estimate the matching degree between $\mathbf{q}$ and $\mathbf{d}$ using cosine similarity



Deep Structured Semantic Model (DSSM) [Huang et al., 2013]

We learn to represent queries and documents in the latent vector space by forcing the vector representations (i) for relevant query-document pairs (q, d^+) to be close in the latent vector space (i.e., $\cos(\mathbf{q}, \mathbf{d}^+) \to \max$); and (ii) for irrelevant query-document pairs $(\mathbf{q}, \mathbf{d}^-)$ to be far in the latent vector space (i.e., $\cos(\mathbf{q}, \mathbf{d}^-) \to \min$)

DSSM - Word hashing

How to represent text (e.g., Shinjuku Gyoen)?

1. Bag of Words (BoW) [large vocabulary (500000 words)]

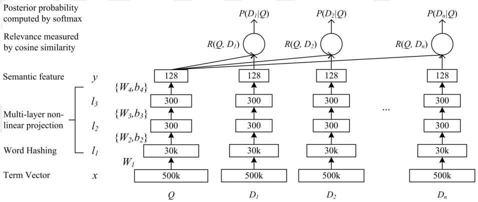
   { 0, . . . , 0 (apple), 0, . . . , 0, 1 (gyoen), 0, . . . , 0, 1 (shinjuku), 0, . . . , 0 }

2. Bag of Letter Trigrams (BoLT) [small vocabulary (30621 letter 3-grams)]

   { 0, . . . , 0 (abc), 0, . . . , 1 (_gy), 0, . . . , 0, 1 (_sh), 0, . . . , 0, 1 (en_), 0, . . . , 0, 1 (gyo), 0, . . . , 0, 1 (hin), 0, . . . , 0, 1 (inj), 0, . . . , 0, 1 (juk), 0, . . . , 0, 1 (ku_), 0, . . . , 0, 1 (oen), 0, . . . , 0, 1 (shi), 0, . . . , 0, 1 (uku), 0, . . . , 0, 1 (yoe), 0 }

DSSM - Architecture

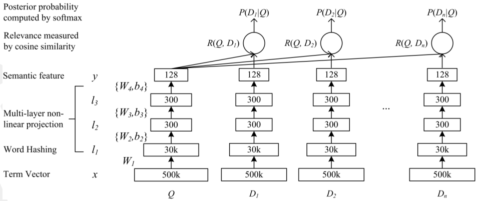
$$\mathbf{x} = \text{BoW}(\text{text})$$
$$\mathbf{l}_1 = \text{WordHashing}(\mathbf{x})$$
$$\mathbf{l}_2 = \tanh(W_2\mathbf{l}_1 + b_2)$$
$$\mathbf{l}_3 = \tanh(W_3\mathbf{l}_2 + b_3)$$
$$\mathbf{l}_4 = \tanh(W_4\mathbf{l}_3 + b_4)$$



DSSM - Training objective

Likelihood

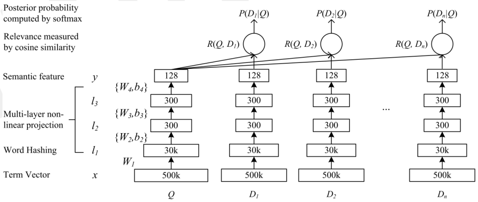
$$\prod_{(q,d^+)\in \text{DATA}} P(d^+ \mid q) \to \max$$



$$P(d^+ \mid q) = \frac{e^{\gamma \cos(\mathbf{q},\mathbf{d}^+)}}{\sum_{d\in D} e^{\gamma \cos(\mathbf{q},\mathbf{d})}} \approx \frac{e^{\gamma \cos(\mathbf{q},\mathbf{d}^+)}}{\sum_{d\in D^+\cup D^-} e^{\gamma \cos(\mathbf{q},\mathbf{d})}}$$

DSSM - Results

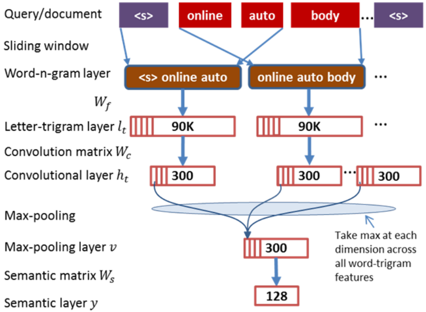
	NDCG		
Model	**@1**	**@3**	**@10**
TF-IDF	0.319	0.382	0.462
BM25	0.308	0.373	0.455
WTM	0.332	0.400	0.478
LSA	0.298	0.372	0.455
PLSA	0.295	0.371	0.456
DAE	0.310	0.377	0.459
BLTM	0.337	0.403	0.480
DPM	0.329	0.401	0.479
DSSM	0.362	0.425	0.498



CLSM

1. Embeds N-grams similar to DSSM
2. Aggregates phrase embeddings by max-pooling

	NDCG		
Model	**@1**	**@3**	**@10**
BM25	0.305	0.328	0.388
DSSM	0.320	0.355	0.431
CLSM	0.342	0.374	0.447

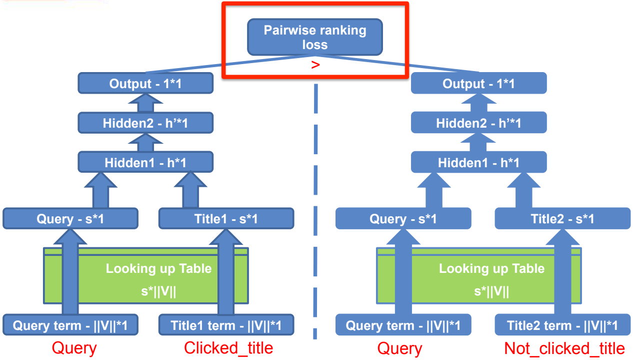


A Latent Semantic Model with Convolutional-Pooling Structure for Information Retrieval [Shen et al., 2014].

In industry

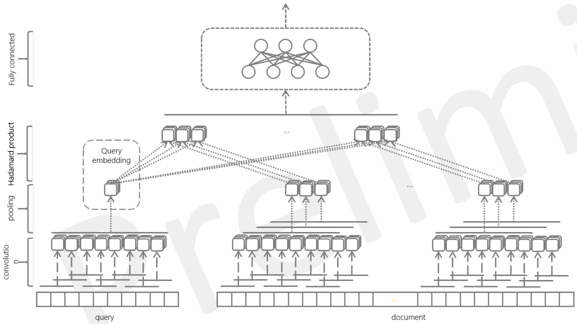
Baidu's DNN model

- Around 30% of total 2013, 2014 relevance improvement
- Use 10B clicks for training (more than 100M parameters)

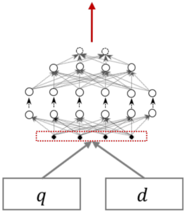


Semantic matching for long text

Semantic matching can also be applied to long text retrieval but requires large scale training data to learn meaningful representations of text



Mitra et al. [2017] train on large manually labelled data from Bing



Dehghani et al. [2017] train on pseudo labels (e.g., BM25)

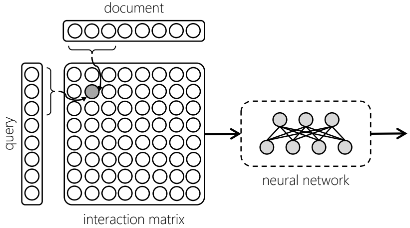
Interaction matrix based approaches

Alternative to Siamese networks

Interaction matrix X, where $x_{i,j}$ is obtained by comparing the the i^{th} word in source sentence with j^{th} word in target sentence

Comparisons can be both lexical or semantic

E.g., Hu et al. [2014], Mitra et al. [2017], Pang et al. [2016]



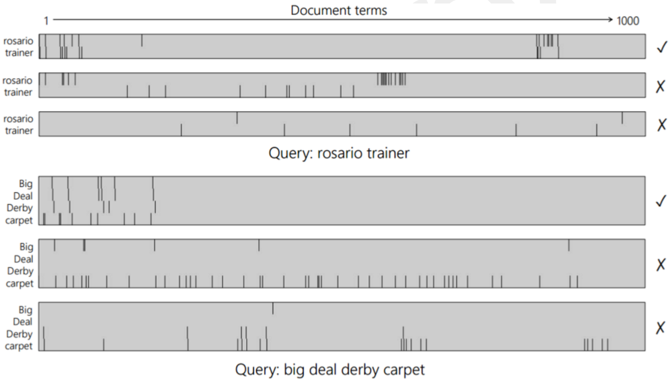
Outline

Lexical matching

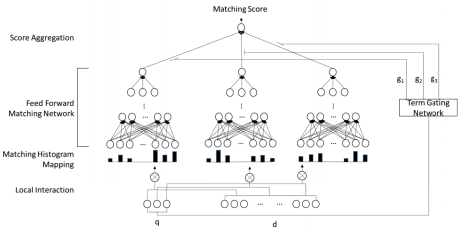
Query: “rosario trainer”

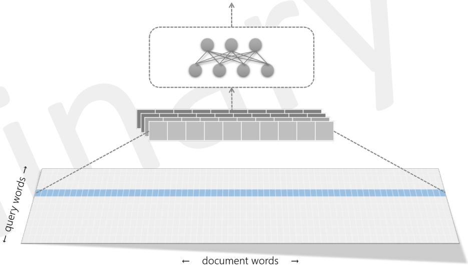
The rare term “rosario” may have never been seen during training and unlikely to have meaningful representation

But the patterns of lexical matches of rare terms in document may be very informative for estimating relevance



Lexical matching





Guo et al. [2016] train a DNN model using features derived from frequency histograms of query term matches in document

Mitra et al. [2017] convolve over the binary interaction matrix to learn interesting patterns of lexical term matches

Outline

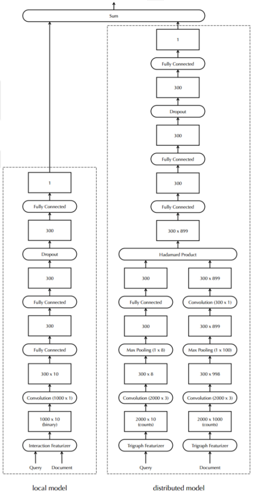
Duet

Jointly train two sub-networks focused on lexical and semantic matching [Mitra et al., 2017, Nanni et al., 2017]

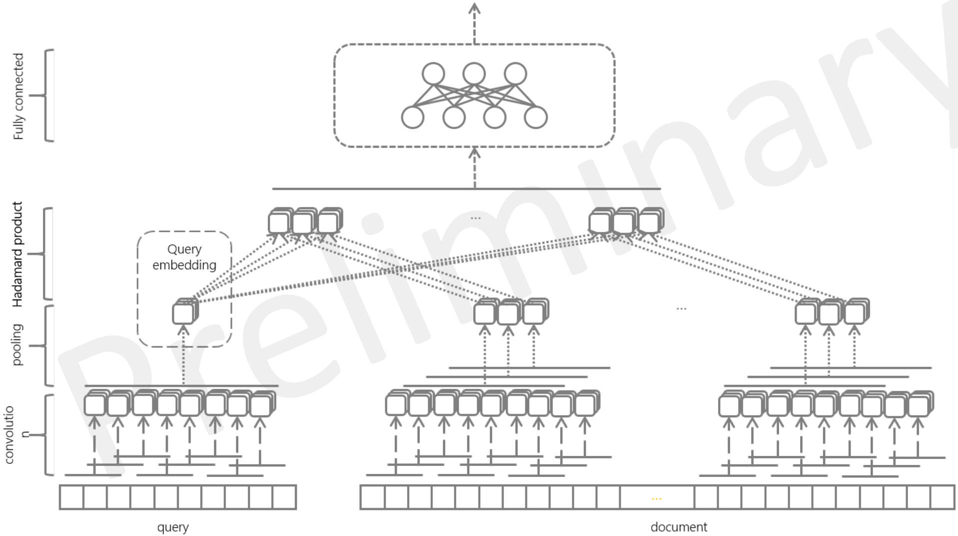
Training sample: q, d^+, d_1, d_2, d_3, d_4

$$p(d^+|q) = \frac{e^{\text{ndrm}(q,d^+)}}{\sum_{d \in D^-} e^{\text{ndrm}(q,d)}} \qquad (1)$$

Implementation on GitHub: https://github.com/bmitra-msft/NDRM/blob/master/notebooks/Duet.ipynb



Distributed model



Results

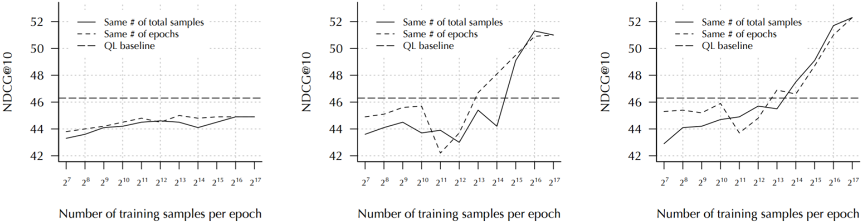
(a) weighted

	NDCG@1	NDCG@10
Non-neural baselines		
LSA	22.4	44.2
BM25	24.2	45.5
DM	24.7	46.2
QL	24.6	46.3
Neural baselines		
DRMM	24.3	45.2
DSSM	25.8	48.2
CDSSM	*27.3*	48.2
DESM	25.4	*48.3*
Our models		
Local model	24.6	45.1
Distributed model	28.6	50.5
Duet model	**32.2**	**53.0**

(b) unweighted

	NDCG@1	NDCG@10
Non-neural baselines		
LSA	31.9	62.7
BM25	34.9	63.3
DM	35.0	63.4
QL	34.9	63.4
Neural baselines		
DRMM	*35.6*	*65.1*
DSSM	34.3	64.4
CDSSM	34.3	64.0
DESM	35.0	64.7
Our models		
Local model	35.0	64.4
Distributed model	35.2	64.9
Duet model	**37.8**	**66.4**

Duet



(a) Local model (b) Distributed model (c) Duet model

The biggest impact of training data size is on the performance of the representation learning sub-model

Important: if you want to learn effective representations for semantic matching you need large scale training data!

Duet

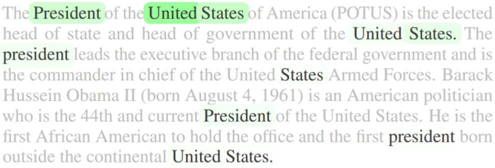
Term importance

Local model

Only query terms have an impact

Earlier occurrences have bigger impact

Query: united states president



Visualizing impact of dropping terms on model score

Duet

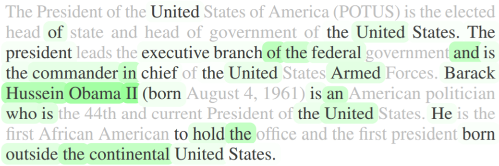
Term importance

Distributed model

Non-query terms (e.g., *Obama* and *federal*) has positive impact on score

Common words like ‘the’ and ‘of’ probably good indicators of well-formedness of content

Query: united states president



Visualizing impact of dropping terms on model score

Duet

If we classify models by query level performance there is a clear clustering of lexical and semantic matching models