Deep Sentence Embedding

刘荣

2013-03-04

Deep Sentence Embedding using Long Short Term Memory Networks

Basic RNN

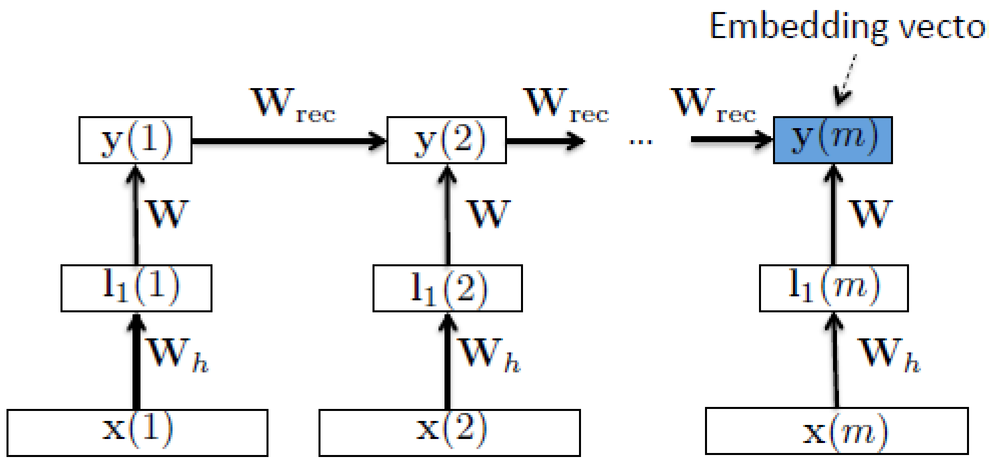


Figure 1. The basic architecture of the RNN for sentence embedding, where temporal recurrence is used to model the contextual information across words in the text string. The hidden activation vector corresponding to the last word is the sentence embedding vector (blue).

$$\mathbf{l}_1(t) = \mathbf{W}_h \mathbf{x}(t)$$
$$\mathbf{y}(t) = f(\mathbf{W}\mathbf{l}_1(t) + \mathbf{W}_{\text{rec}}\mathbf{y}(t-1)) \qquad (1)$$

Deep Sentence Embedding using Long Short Term Memory Networks

RNN with LSTM

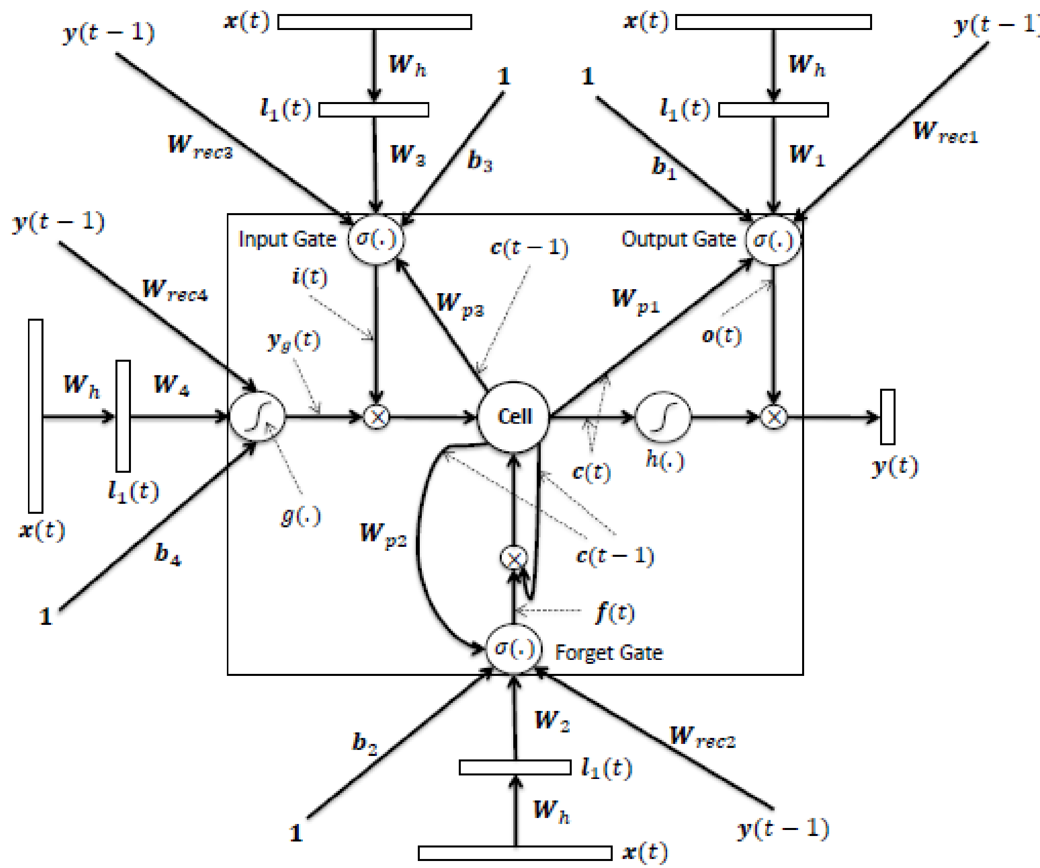


Figure 2. The basic LSTM architecture used for sentence embedding

Deep Sentence Embedding using Long Short Term Memory Networks

Object function

$$L(\mathbf{\Lambda}) = \min_{\mathbf{\Lambda}} \left\{ -\log \prod_{r=1}^{N} P(D_r^+|Q_r) \right\} = \min_{\mathbf{\Lambda}} \sum_{r=1}^{N} l_r(\mathbf{\Lambda}) \quad (4)$$

$$\begin{aligned} l_r(\mathbf{\Lambda}) &= -\log \left(\frac{e^{\gamma R(Q_r,D_r^+)}}{e^{\gamma R(Q_r,D_r^+)} + \sum_{i=j}^{n} e^{\gamma R(Q_r,D_{r,j}^-)}} \right) \\ &= \log \left(1 + \sum_{j=1}^{n} e^{-\gamma \cdot \Delta_{r,j}} \right) \qquad (5) \end{aligned}$$

where $\Delta_{r,j} = R(Q_r, D_r^+) - R(Q_r, D_{r,j}^-)$, $R(\cdot,\cdot)$ was de-

$$R(Q,D) = \frac{\mathbf{y}_Q(T_Q)^T \mathbf{y}_D(T_D)}{\|\mathbf{y}_Q(T_Q)\| \cdot \|\mathbf{y}_D(T_D)\|} \qquad (3)$$

Deep Sentence Embedding using Long Short Term Memory Networks

Object function

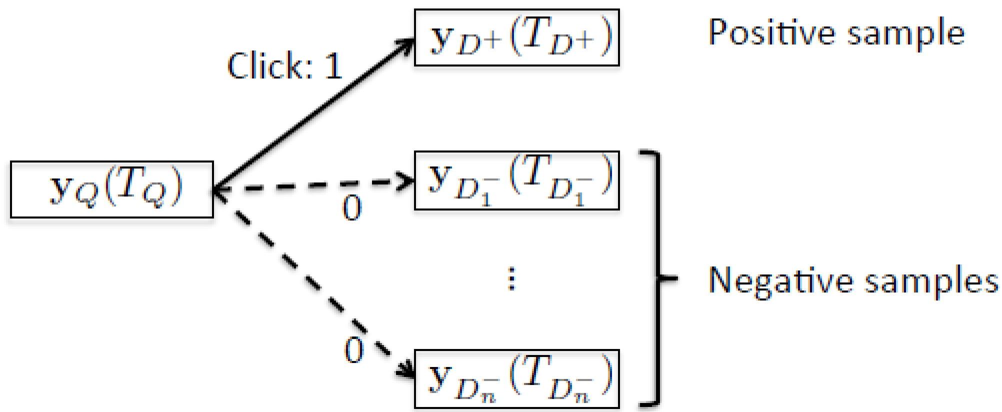


Figure 3. The click-through signal can be used as a (binary) indication of the semantic similarity between the sentence on the query side and the sentence on the document side. The negative samples are randomly sampled from the training data.

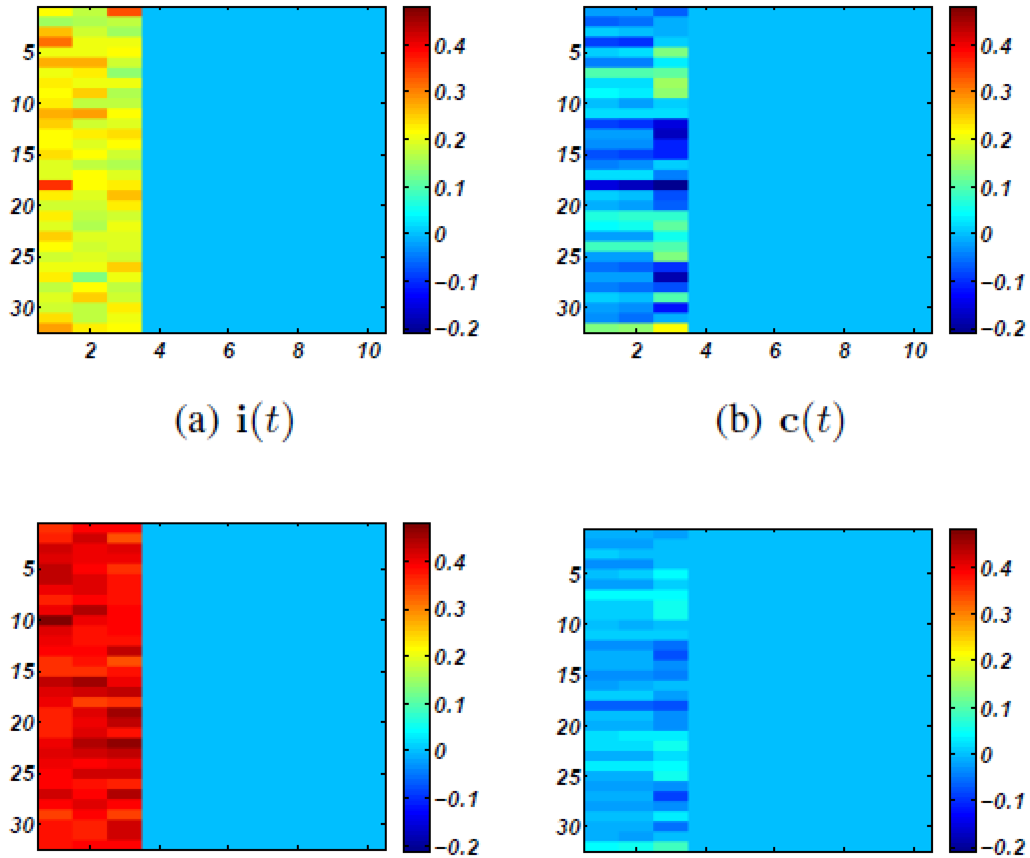
$$R(Q, D) = \frac{\mathbf{y}_Q(T_Q)^T \mathbf{y}_D(T_D)}{\|\mathbf{y}_Q(T_Q)\| \cdot \|\mathbf{y}_D(T_D)\|} \qquad (3)$$

Deep Sentence Embedding using Long Short Term Memory Networks

Analysis

- Query: "*hotels in shanghai*"
- Document: "*shanghai hotels accommodation hotel in shanghai discount and reservation*"

ATTENUATING UNIMPORTANT INFORMATION

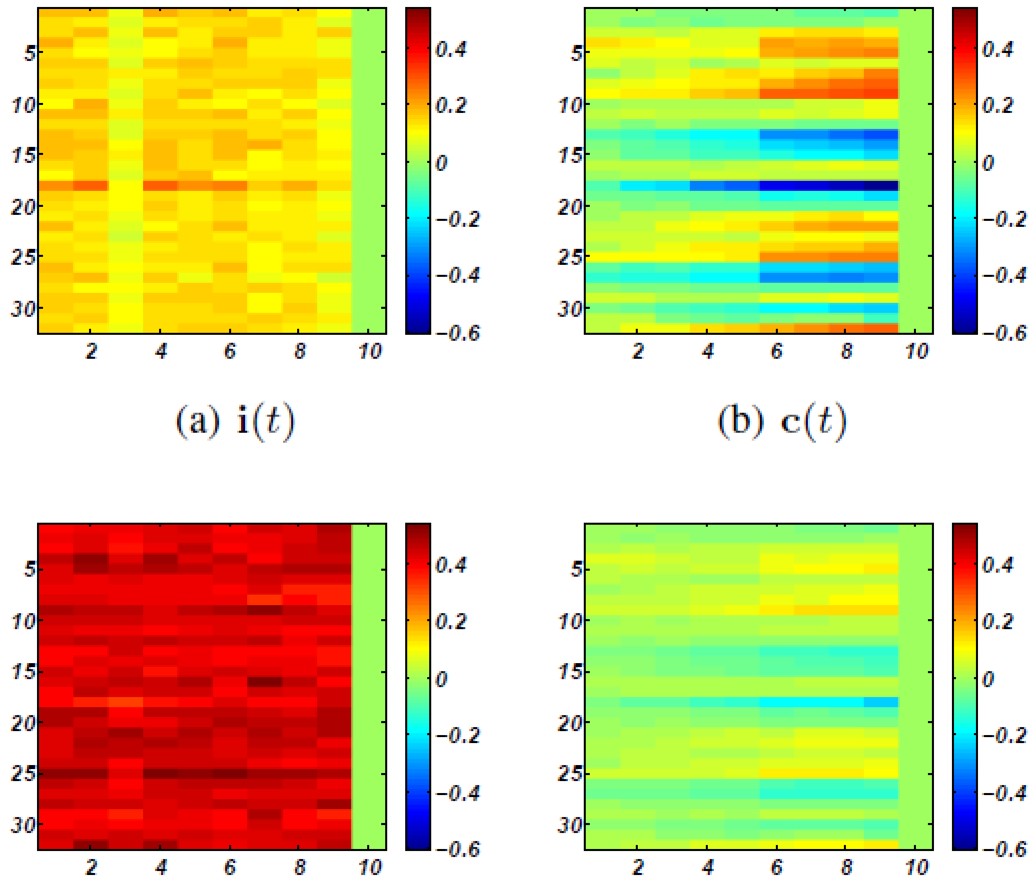


(a) $\mathbf{i}(t)$ (b) $\mathbf{c}(t)$

(c) $\mathbf{o}(t)$ (d) $\mathbf{y}(t)$

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

ATTENUATING UNIMPORTANT INFORMATION



(a) $\mathbf{i}(t)$ (b) $\mathbf{c}(t)$

(c) $\mathbf{o}(t)$ (d) $\mathbf{y}(t)$

Figure 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).

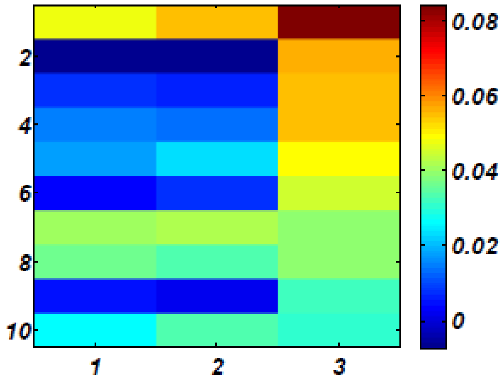


Figure 6. Activation values, $\mathbf{y}(t)$, of 10 most active cells for Query: "*hotels in shanghai*"

Table 1. Key words for query: "*hotels in shanghai*"

	hotels	*in*	*shanghai*
Number of assigned cells out of 10	-	0	7

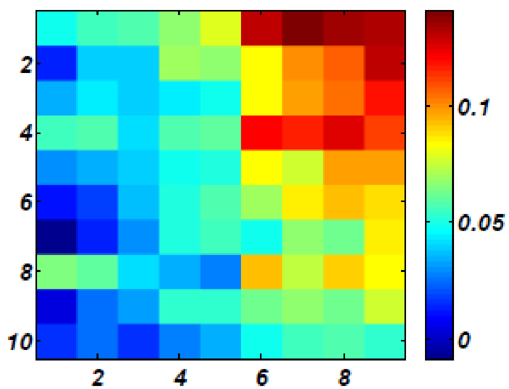


Figure 7. Activation values, $\mathbf{y}(t)$, of 10 most active cells for Document: "*shanghai hotels accommodation hotel in shanghai discount and reservation*"

Table 2. Key words for document: "*shanghai hotels accommodation hotel in shanghai discount and reservation*"

	shanghai	*hotels*	*accommodation*	*hotel*	*in*	*shanghai*	*discount*	*and*	*reservation*
Number of assigned cells out of 10	-	4	3	8	1	8	5	3	4

Deep Sentence Embedding using Long Short Term Memory Networks

Result

Model	NDCG@1	NDCG@3	NDCG@10
BM25	30.5%	32.8%	38.8%
PLSA (T=500)	30.8%	33.7%	40.2%
DSSM (nhid = 288/96), 2 Layers	31.0%	34.4%	41.7%
CLSM (nhid = 288/96), 2 Layers	31.8%	35.1%	42.6%
RNN (nhid = 288), 1 Layer	31.7%	35.0%	42.3%
LSTM-RNN (ncell = 96), 1 Layer	**33.1%**	**36.5%**	**43.6%**

DSSM

Learning Deep Structured Semantic Models for Web Search using Clickthrough Data(13)

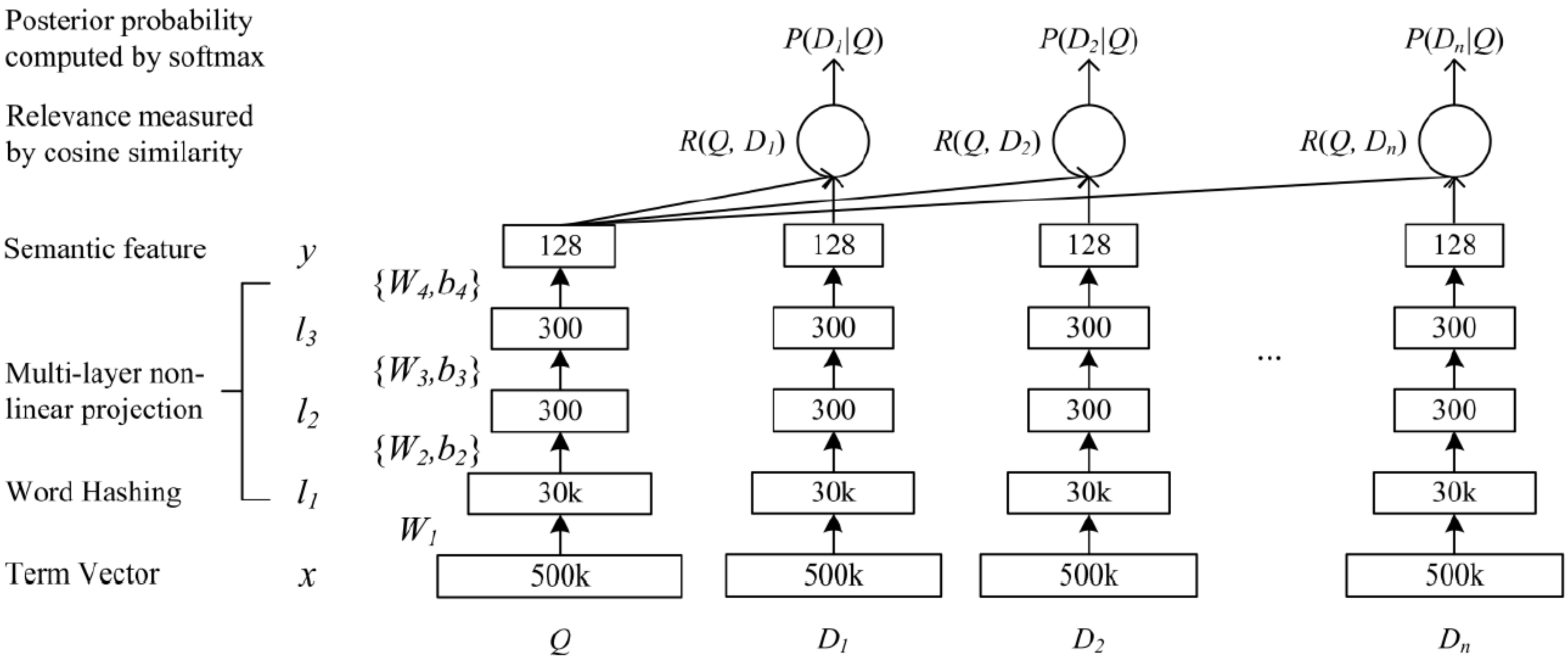


Figure 1: Illustration of the DSSM. It uses a DNN to map high-dimensional sparse text features into low-dimensional dense features in a semantic space. The first hidden layer, with 30k units, accomplishes word hashing. The word-hashed features are then projected through multiple layers of non-linear projections. The final layer's neural activities in this DNN form the feature in the semantic space.

$$
\begin{aligned}
l_1 &= W_1 x \\
l_i &= f(W_i l_{i-1} + b_i), i = 2, \dots, N-1 \\
y &= f(W_N l_{N-1} + b_N)
\end{aligned}
\quad (3)
$$

$$R(Q, D) = \text{cosine}(y_Q, y_D) = \frac{{y_Q}^T y_D}{\|y_Q\| \|y_D\|}$$

$$f(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}}$$

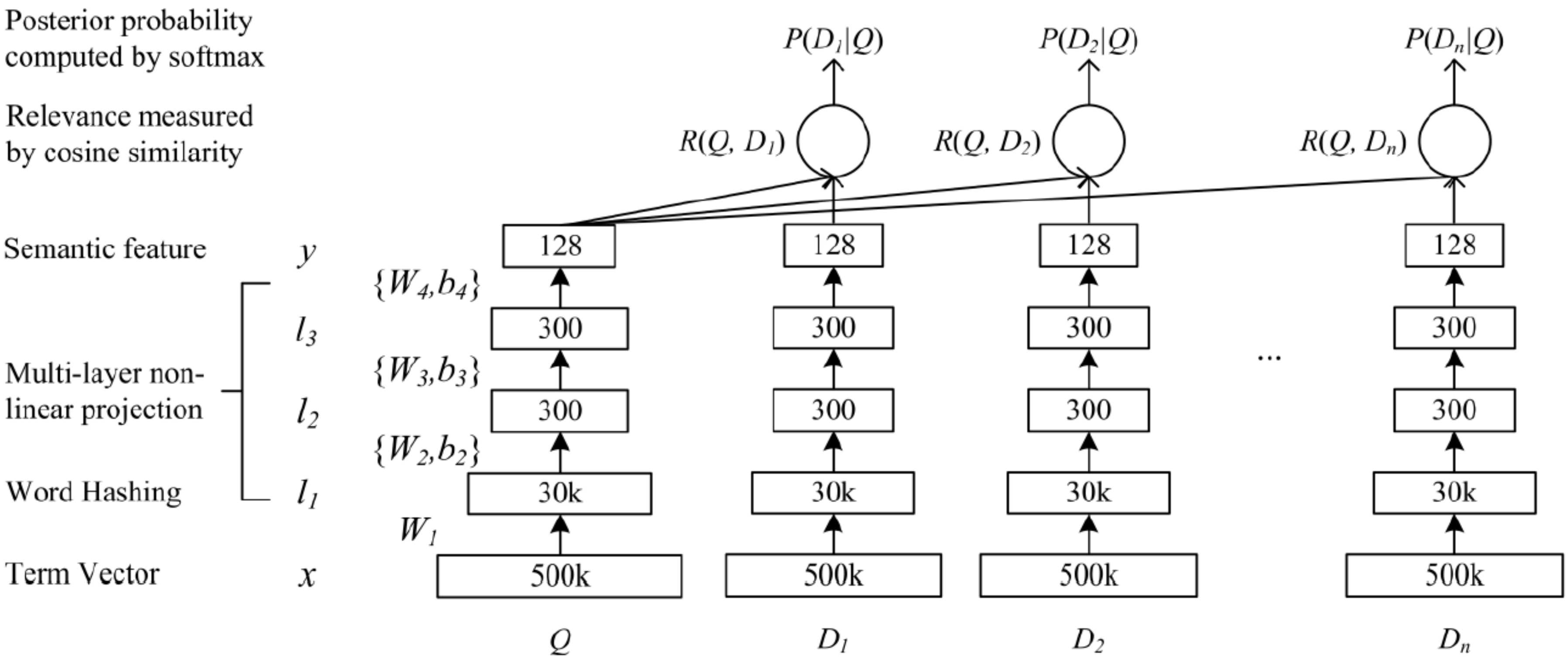
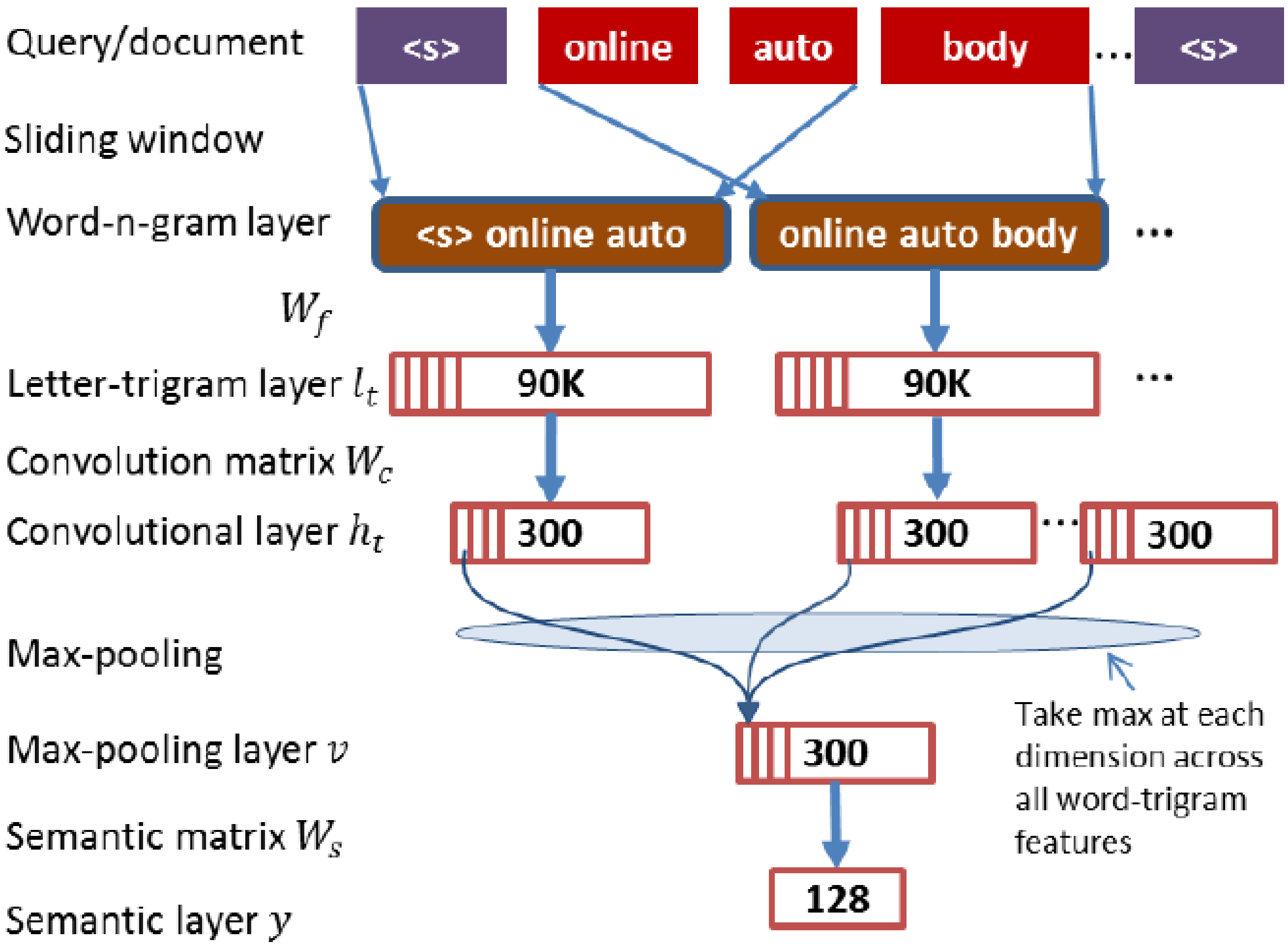


Figure 1: Illustration of the DSSM. It uses a DNN to map high-dimensional sparse text features into low-dimensional dense features in a semantic space. The first hidden layer, with 30k units, accomplishes word hashing. The word-hashed features are then projected through multiple layers of non-linear projections. The final layer's neural activities in this DNN form the feature in the semantic space.

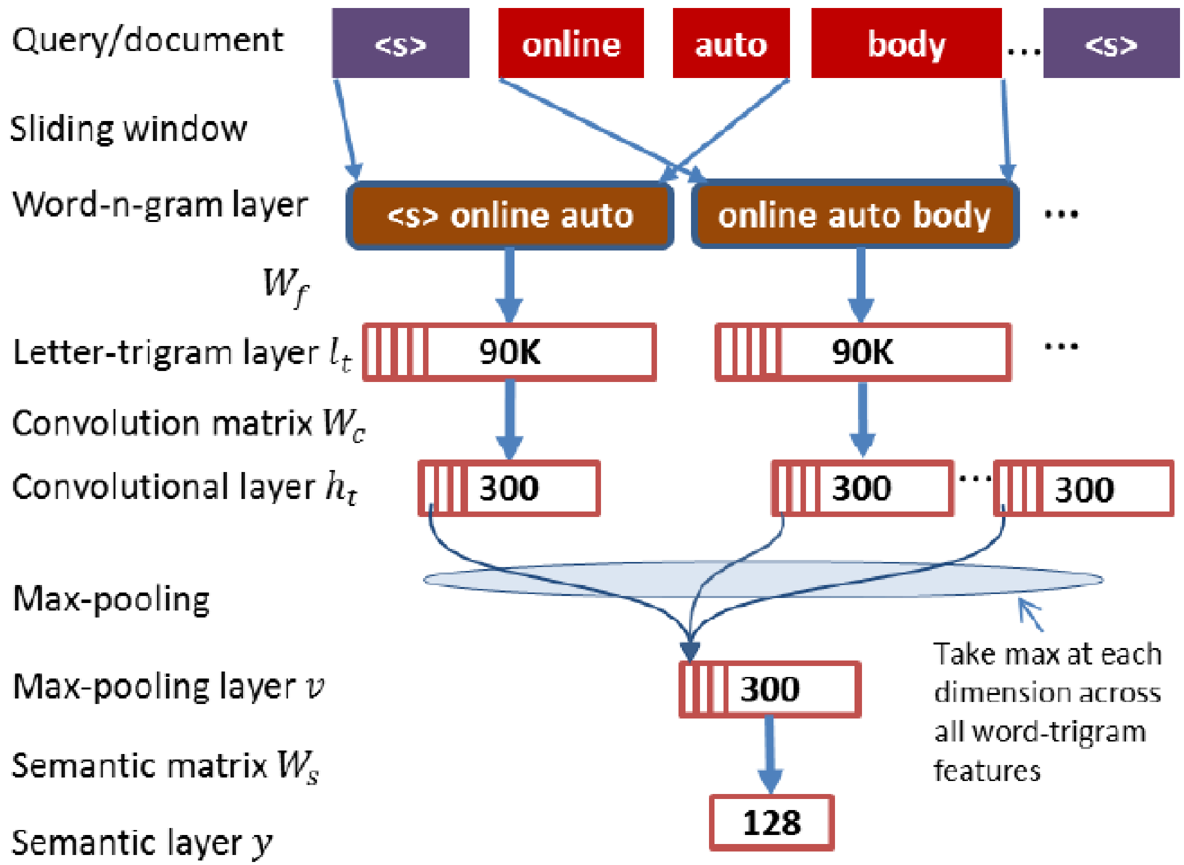
$$P(D|Q) = \frac{\exp\left(\gamma R(Q, D)\right)}{\sum_{D' \in \mathbf{D}} \exp\left(\gamma R(Q, D')\right)}$$

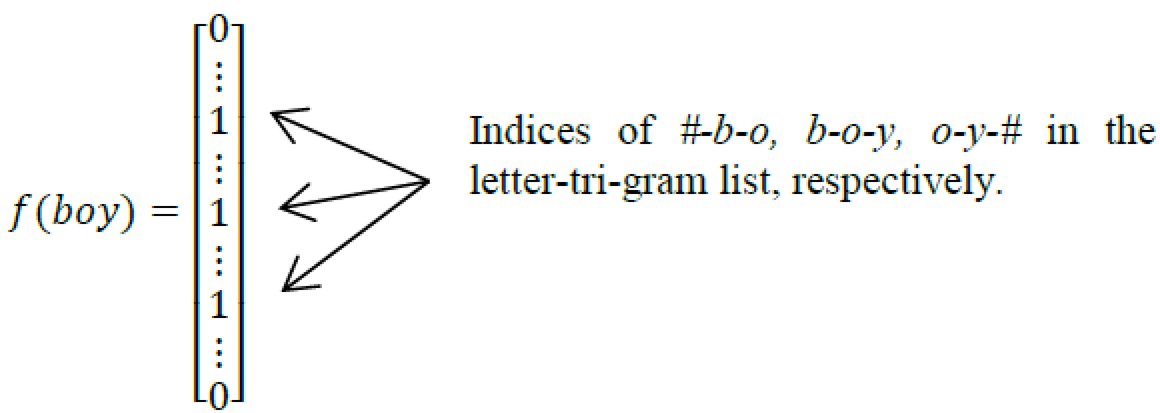
$$L(\Lambda) = -\log \prod_{(Q, D^+)} P(D^+|Q)$$

CLSM



A Latent Semantic Model with Convolutional-Pooling Structure for Information Retrieval


Query/document
<s>
online
auto
body
...
<s>
Sliding window
Word-n-gram layer
<s> online auto
online auto body
...
W_f
Letter-trigram layer l_t
90K
90K
...
Convolution matrix W_c
Convolutional layer h_t
300
300
...
300
Max-pooling
Take max at each dimension across all word-trigram features
Max-pooling layer v
300
Semantic matrix W_s
128
Semantic layer y


$f(boy) =$
0
1
1
1
0
Indices of #-b-o, b-o-y, o-y-# in the letter-tri-gram list, respectively.

IDEA

1. 利用这种区分性的信息去训练word/sentence embedding. 如训练PV时加入加入文章的分类信息等。