SEMANTIC MODELLING WITH LONG-SHORT-TERM MEMORY FOR INFORMATION RETRIEVAL

H. Palangi[1], **L. Deng**[2], **Y. Shen**[2], **J. Gao**[2], **X. He**[2], **J. Chen**[2], **X. Song**[2], **R. Ward**[1]
[1]University of British Columbia, Vancouver, BC, Canada
[2]Microsoft Research, Redmond, WA, USA
[1]{hamidp,rababw}@ece.ubc.ca
[2]{deng,jfgao,xiaohe,yeshen,jianshuc,xinson}@microsoft.com

ABSTRACT

In this paper we address the following problem in web document and information retrieval (IR): How can we use long-term context information to gain better IR performance? Unlike common IR methods that use bag of words representation for queries and documents, we treat them as a sequence of words and use long short term memory (LSTM) to capture contextual dependencies. To the best of our knowledge, this is the first time that LSTM is applied to information retrieval tasks. Unlike training traditional LSTMs, the training strategy is different due to the special nature of information retrieval problem. Experimental evaluation on an IR task derived from the Bing web search demonstrates the ability of the proposed method in addressing both lexical mismatch and long-term context modelling issues, thereby, significantly outperforming existing state of the art methods for web document retrieval task.

1 INTRODUCTION

Two important issues to measure semantic similarities among different text strings include lexical mismatch and the difficulty of incorporating context information. Lexical mismatch means that one can use different vocabulary items and language styles to express the same concept. This problem is addressed using the translation models (Gao et al., 2010), the topic models (Deerwester et al., 1990; Gao et al., 2014), and the Deep Structured Semantic Model (DSSM) which makes use of the bag-of-words representation (Huang et al., 2013). Incorporation of context information for modelling semantic similarity, on the other hand, can be accomplished by language modelling (Platt et al., 2010; Gao et al., 2004; Metzler & Croft, 2005; 2007). There are a few recent models which intend to address both issues in a single framework, including the Convolutional DSSM (CLSM) proposed in (Shen et al., 2014) and Recurrent DSSM (R-DSSM) proposed in (Palangi et al., 2015). The main difference between the R-DSSM and CLSM is that while CLSM needs a fixed size sliding window to capture local context information, and a maxpooling layer to capture global context information, the R-DSSM captures both with a recurrent layer without the need for the maxpooling layer.

In this paper, we extend the R-DSSM to incorporate the structure called the Long Short Term Memory DSSM (LSTM-DSSM). The motivations of the extension are as follows. First, due to vanishing and exploding gradient problems, it is difficult for an R-DSSM to capture long term context information effectively. Second, training an R-DSSM is significantly slower than training its DSSM counterpart. Third, the LSTM-DSSM has the potential to significantly outperform the R-DSSM in practical tasks as evidenced in recent successful applications of the LSTM in large-scale tasks of speech recognition (Sak et al., 2014) and machine translation (Sutskever et al., 2014).

To the best of our knowledge, this is the first time that LSTM is applied to information retrieval tasks. Unlike training traditional LSTMs, the training strategy is significantly different due to the special nature of information retrieval problem. Specifically, the error signal is generated from the cosine distance between the two semantic embedding vectors of the two text strings (i.e., query and document title), and is then propagated towards the query-LSTM model and the document-LSTM model separately — see Fig. 1. From the figure, we also note that, the error signal is only generated from the end of the output sequence and is required to be back propagated to the beginning. This is different from traditional LSTM models, e.g., in speech recognition, where the error signals are generated at every output sample. For this reason, it is critical to use LSTM model in information

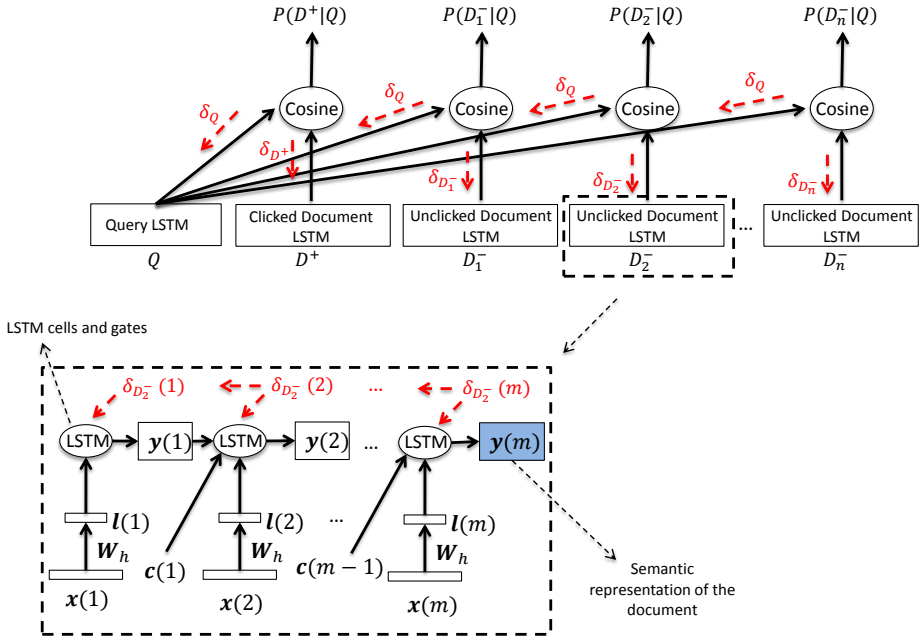

Figure 1: Architecture of the proposed method. n is the number of negative samples (unclicked documents)

retrieval problems in order to learn the long-term memory, which is known to be difficult in standard RNN with sigmoid/tanh neurons.

In addition to faster convergence and practical better performance than the R-DSSM, we argue that the LSTM-DSSM can potentially provide valuable information about correlations among different topics and about transitions from one topic to another in a long document.

2 The Model

The LSTM-DSSM model developed in this work is aimed to overcome the weakness of the R-DSSM in capturing long-term contextual information effectively. The solution provided by the new model is to incorporate LSTM memory cells, as proposed originally in Hochreiter & Schmidhuber (1997) and further developed in Gers et al. (1999) and Gers et al. (2003) by adding forget gate and peephole connections to the architecture. The architecture of the LSTM cell used in the LSTM-DSSM is illustrated in Fig. 2, where $\mathbf{i}(t)$, $\mathbf{f}(t)$, $\mathbf{o}(t)$ are the input gate, forget gate and output gate, respectively, $\mathbf{c}(t)$ is the cell state, $\mathbf{W}_{p1}$, $\mathbf{W}_{p2}$ and $\mathbf{W}_{p3}$ are peephole connections, $g(.)$ and $h(.)$ are $tanh(.)$ functions and $\sigma(.)$ is a sigmoid function.

We use this architecture to find $\mathbf{y}$ for each word and then use equation (7) to find the similarity between query and documents. Subsequently, the LSTM-DSSM is trained using truncated back-propagation-through-time.

Assuming that we have just one layer of the LSTM (for simplicity of presentation), the mathematical formulation of the LSTM cell according to Fig. 2 is as follows:

$$\mathbf{y}_g(t) = g(\mathbf{W}_4\mathbf{l}(t) + \mathbf{W}_{rec4}\mathbf{y}(t-1) + \mathbf{b}_4) \tag{1}$$

$$\mathbf{i}(t) = \sigma(\mathbf{W}_3\mathbf{l}(t) + \mathbf{W}_{rec3}\mathbf{y}(t-1) + \mathbf{W}_{p3}\mathbf{c}(t-1) + \mathbf{b}_3) \tag{2}$$

$$\mathbf{f}(t) = \sigma(\mathbf{W}_2\mathbf{l}(t) + \mathbf{W}_{rec2}\mathbf{y}(t-1) + \mathbf{W}_{p2}\mathbf{c}(t-1) + \mathbf{b}_2) \tag{3}$$

$$\mathbf{c}(t) = \mathbf{f}(t) \circ \mathbf{c}(t-1) + \mathbf{i}(t) \circ \mathbf{y}_g(t) \tag{4}$$

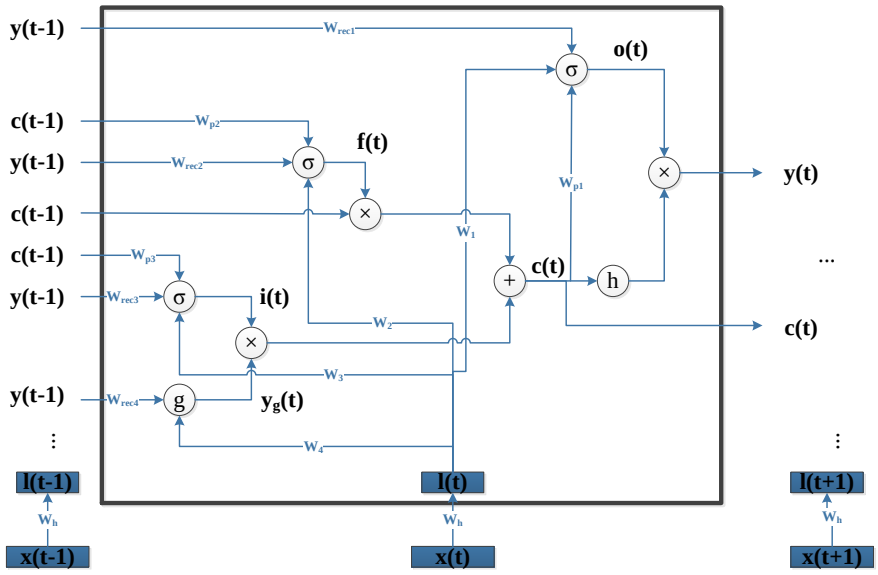

Figure 2: The architecture of an LSTM cell used in the LSTM-DSSM

$$\mathbf{o}(t) = \sigma(\mathbf{W}_1\mathbf{l}(t) + \mathbf{W}_{rec1}\mathbf{y}(t-1) + \mathbf{W}_{p1}\mathbf{c}(t) + \mathbf{b}_1) \tag{5}$$

$$\mathbf{y}(t) = \mathbf{o}(t) \circ h(\mathbf{c}(t)) \tag{6}$$

where $\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3, \mathbf{b}_4$ are bias vectors (not shown in the figure) and $\mathbf{l}(t)$ is the t-th word representation after hashing ($\mathbf{W}_h$).

For training the full LSTM-DSSM, we adopt the cosine similarity measure:

$$R(Q, D) = \frac{\mathbf{y}_Q(t = T_Q)^T \mathbf{y}_D(t = T_D)}{\|\mathbf{y}_Q(t = T_Q)\| \|\mathbf{y}_D(t = T_D)\|} \tag{7}$$

where T_Q and T_D are the indexes of the last word in query and document, respectively. The goal is to maximize the likelihood of the clicked document given a query. Therefore, the following optimization problem is to be solved:

$$L(\mathbf{\Lambda}) = \min_{\mathbf{\Lambda}} \; -log \prod_{r=1}^{R} P(D_r^+|Q_r) \tag{8}$$

where $\mathbf{\Lambda}$ is the parameter to be learned and $P(D_r^+|Q_r)$ is the probability of clicked document given the r-th query as a function of cosine similarity measure according to

$$P(D_r^+|Q_r) = \frac{e^{R(Q_r, D_r^+)}}{\sum_{D_{r,j} \in \mathbf{D}} e^{R(Q_r, D_{r,j})}} \tag{9}$$

In (8) and (9), Q_r is the r-th query out of R queries, D_r^+ is the clicked document for r-th query and $D_{r,j}$ is the j-th unclicked document for r-th query. In the learning algorithm, error signals are back propagated through time using following equations which we derived for the LSTM-DSSM:

$$\delta_Q^{rec1}(t-1) = [\mathbf{o}_Q(t-1) \circ (1 - \mathbf{o}_Q(t-1)) \circ h(\mathbf{c}_Q(t-1))] \circ \mathbf{W}_{rec1}^T . \delta_Q^{rec1}(t) \tag{10}$$

Table 1: Comparisons of NDCG performance measures (the higher the better) of proposed models and a series of baseline models, where *nhid* refers to the number of hidden units, *ncell* refers to number of cells. The RNN and LSTM-RNN models are chosen to have the same number model parameters as the DSSM and CLSM models: 14.4M, where $1\text{M} = 10^6$. The boldface numbers are the best results.

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%**

$$\delta_Q^{rec_i}(t-1) = [(1 - h(\mathbf{c}_Q(t-1))) \circ (1 + h(\mathbf{c}_Q(t-1))) \circ \mathbf{o}_Q(t-1)] \circ \mathbf{W}_{rec_i}^T . \delta_Q^{rec_i}(t) \quad (11)$$

where Q stands for query. And we have derived a similar set of equations for the "document" part of the full LSTM-DSSM network. In the gradient-descent training, we have one large update after folding back in time and adding gradients in each minibatch. We use Nesterov method to accelerate learning convergence.

3 Evaluation Results

For training and evaluating the LSTM-DSSM and comparing it with the state of the art IR baselines, we have used a real world dataset consisting of 200,000 click-through data collected from Bing search to carry out evaluation experiments. Experimental results are presented in Table 1 using the standard metric of mean Normalized Discounted Cumulative Gain (NDCG) (Järvelin & Kekäläinen, 2000) for evaluating ranking performance. For fair comparisons, we have designed the LSTM-DSSM so that it uses the same number of model parameters as the baseline R-DSSM and other well known baselines in IR. As is clear from these results, the LSTM-DSSM outperforms all existing baselines significantly in terms of the NDCG metric. Analysis of the results demonstrates the effectiveness of the LSTM cells in capturing long-term correlations in the input text strings, accounting for the success in the information retrieval task designed from Bing search.

REFERENCES

Deerwester, Scott, Dumais, Susan T., Furnas, George W., Landauer, Thomas K., and Harshman, Richard. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990.

Gao, Jianfeng, Nie, Jian-Yun, Wu, Guangyuan, and Cao, Guihong. Dependence language model for information retrieval. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '04, pp. 170–177. ACM, 2004.

Gao, Jianfeng, He, Xiaodong, and Nie, Jian-Yun. Clickthrough-based translation models for web search: From word models to phrase models. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, CIKM '10, pp. 1139–1148, New York, NY, USA, 2010. ACM.

Gao, Jianfeng, Pantel, Patrick, Gamon, Michael, He, Xiaodong, Deng, Li, and Shen, Yelong. Modeling interestingness with deep neural networks. *EMNLP*, October 2014.

Gers, Felix A., Schmidhuber, Jrgen, and Cummins, Fred. Learning to forget: Continual prediction with lstm. *Neural Computation*, 12:2451–2471, 1999.

Gers, Felix A., Schraudolph, Nicol N., and Schmidhuber, Jürgen. Learning precise timing with lstm recurrent networks. *J. Mach. Learn. Res.*, 3:115–143, March 2003.

Hochreiter, Sepp and Schmidhuber, Jürgen. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November 1997.

Huang, Po-Sen, He, Xiaodong, Gao, Jianfeng, Deng, Li, Acero, Alex, and Heck, Larry. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22Nd ACM International Conference on Conference on Information & Knowledge Management*, CIKM '13, pp. 2333–2338. ACM, 2013.

Järvelin, Kalervo and Kekäläinen, Jaana. Ir evaluation methods for retrieving highly relevant documents. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR, pp. 41–48. ACM, 2000.

Metzler, Donald and Croft, W. Bruce. A markov random field model for term dependencies. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '05, pp. 472–479. ACM, 2005.

Metzler, Donald and Croft, W. Bruce. Latent concept expansion using markov random fields. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '07, pp. 311–318. ACM, 2007.

Palangi, H., Deng, L., Shen, Y., Gao, J., He, X., Chen, J., Song, X., and Ward, R. Learning sequential semantic representations of natural language using recurrent neural networks. In *ICASSP*, 2015.

Platt, John C., Toutanova, Kristina, and Yih, Wen-tau. Translingual document representations from discriminative projections. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, EMNLP 2010, 9-11 October 2010, MIT Stata Center, Massachusetts, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pp. 251–261, 2010.

Sak, H., Senior, A., and Beaufays, F. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. *INTERSPEECH*, 2014.

Shen, Yelong, He, Xiaodong, Gao, Jianfeng, Deng, Li, and Mesnil, Gregoire. A latent semantic model with convolutional-pooling structure for information retrieval. In *CIKM*, November 2014.

Sutskever, Ilya, Vinyals, Oriol, and Le, Quoc V. Sequence to sequence learning with neural networks. *NIPS*, 2014.