Ad Recommendation for Sponsored Search Engine via Composite Long-Short Term Memory

Dejiang Kong Fei Wu Siliang Tang Yueting Zhuang
College of Computer Science, Zhejiang University, China
{kdjysss, siliang.tang}@gmail.com wufei@cs.zju.edu.cn yzhuang@zju.edu.cn

ABSTRACT

Search engine logs contain a large amount of users' click-through data that can be leveraged as implicit indicators of relevance. In this paper we address ad recommendation problem that finding and ranking the most relevant ads with respect to users' search queries. Due to the click sparsity, the conventional methods can hardly model the both inter- and intra-relations among users, queries and ads. We utilize the long-short term memory(LSTM) network to effectively encode two kinds of sequences: the (user, query) sequence and the query word sequence to represent users' query intention in a continuous vector space and decode them as distributions over ads respectively. Further more, we combine these two LSTM networks in an appropriate way to build up a more robust model referred as composite LSTM model(cLSTM) for ad recommendation. We evaluate the proposed cLSTM on real world click-through data set comparing with two baseline methods, the results demonstrate that our proposed model outperforms two baselines and mitigate the click sparsity problem to a certain degree.

Keywords

Ad recommendation; Long-short term memory; Click-through data; Web logs

1. INTRODUCTION

Search advertising has been one of the major revenue sources of the Internet industry for years. A key technology behind search advertising is to predict the click-through rate (pCTR) of ads, since the economic model behind search advertising utilizes pCTR values to rank ads and to price clicks.

Recently, click prediction(ad recommendation) has received much attention from both industry and academia[9] and many works[1, 16, 14, 18] have been done to improve the prediction accuracy. For example, [1] uses their fine-designed query-ad click graph to better discover similarities between queries and leverage the Collaborative Filtering(CF) approach to make a better ad recommendation. [14] concentrates on modeling user psychological desire using special textual patterns and condenses them into powerful features to improve the prediction performance. Unlike [14] and [1], [16] opens another way to the problem, they pay attention to the relations between ads which are proved an important factor in predicting click probability.

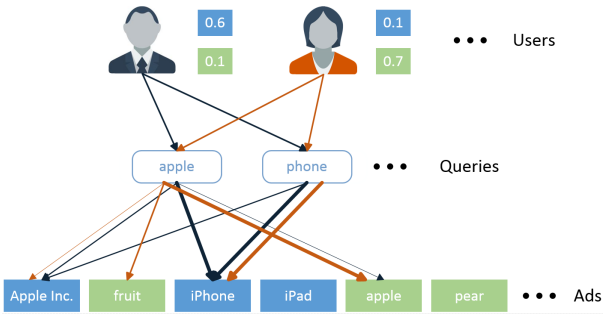

Figure 1: One example of interactions among users, queries and ads. The colored rectangles represent the categories of ads(e.g. blue for tech ad and green for food ad), and colored arrows stand for different users' query intentions(e.g., one user clicks his/her favorite ads after submitting a query).

All of aforementioned works merely modeled one or two kinds of relations among users, queries and ads, while we find that the inter- and intra-relations among them are complicated as Figure 1 illustrated. (1) Different users with the same query may have their favorite desires, i.e. the man in Figure 1 prefers tech more than food, when he issues a query "apple", it is more likely that he will click the "iPhone" ad rather than fruit apple ad, while the woman with another preference is more likely to click the latter one. (2) Different users with different queries may have the same intention. Two different queries "apple" and "phone" are individually issued but both of them result in strong intention to click the "iPhone" ad. (3) Same user with different queries may have the same intention, i.e., the man issues different queries "apple" and "phone" but both have a high probability to click the "iPhone" ad as he desires to buy a phone at that time.

The click tuple (user, query, ad) can be regarded as a three-dimensional data, which is then natural to use Tensor Factorization(TF) based methods[13, 10] to model them for prediction, but these methods are intrinsically prone to perform badly when data is sparse. Long-Short Term Memory(LSTM) network[8] has shown its ability in modeling se-

MM '16, October 15-19, 2016, Amsterdam, Netherlands

DOI: http://dx.doi.org/10.1145/2964284.2967254

quential data in many fields recently[5, 6, 12]. To our problem, we have two kinds of sequences: the global sequences (user, query, ad) and the local query word sequences(e.g. "apple latest released products"). In this paper, we leverage LSTM networks to model these two sequences in order to reveal the inter- and intra-relations among users, queries and ads. All of our contributions in this paper can be concluded as follows:

1.We implement two kinds of LSTM models, one emphasizes on local query contents and another focuses on global relations among users, queries and ads. By this way we can encode users' query intentions and decode them as distributions over ads respectively. To the best of our knowledge, we are the first one to employ LSTM networks in ad recommendation problem for sponsored search engine.

2.We combine aforementioned two LSTM models to build up a more powerful composite LSTM network(cLSTM) to solve the ad recommendation problem.

3.Finally, we show empirically that our proposed model outperforms two baselines on a real world click data set.

2. AD RECOMMENDATION VIA CLSTM

2.1 Problem Introduction

Sponsored search engine can be seen as interaction between three involved factors: user, query and ad. In general a user issues a query to a search engine to seek information, search engine returns highly ranked items and the corresponding ads according to user's query intention. One user may click the returned ad if it is highly relevant to the user's preference. Therefore, lots of these click-through (user, query, ad) tuples are accumulated in search engine. The underlying mission of this paper can be stated as follows: *Given a user and his/her search query, we want to model user's query intention and then recommend the most likely ads to the user.* In this problem, we have two kinds of sequences – query word sequences and (user, query) sequences, it is effective to use Recurrent Neural Networks[15] to encode them. LSTM[8] is an optimized RNN model that avoids the vanishing and exploding gradients problem[2, 7] by introducing *input gate*, *forget gate*, *output gate* and *memory cell*. These multiple gates allow the memory cell in LSTM to keep, update or forget information over time. LSTM model has shown a great ability in modeling sequential data[5, 6, 12] recently and we will discuss how to utilize LSTM model to effectively encode aforementioned two kinds of sequences in following subsections.

2.2 Local LSTM Model

Our local LSTM(Figure 2) model emphasizes on local query contents and encodes the relations between queries and ads. Given the i^{th} query $\boldsymbol{q}^i = \{v_1^i, v_2^i, ..., v_k^i\}$ by a user with k words, $k = |\boldsymbol{q}^i|$ and $v \in V$ where V is a vocabulary consisting of all of individual query words. We first transform each query word v in terms of one-hot representation into a latent vector $\boldsymbol{v}$ via the transformation word embedding matrix $\boldsymbol{W}_v$, then our local LSTM sequentially takes these query word vectors as input and learns the hidden output vectors as follows:

$$\begin{aligned} \boldsymbol{h}_{l,j}^i &= LSTM(\boldsymbol{h}_{l,j-1}^i, \boldsymbol{v}_j^i) \\ \boldsymbol{h}_{l,0}^i &= \boldsymbol{0}; j = 1, ..., k \end{aligned} \tag{1}$$

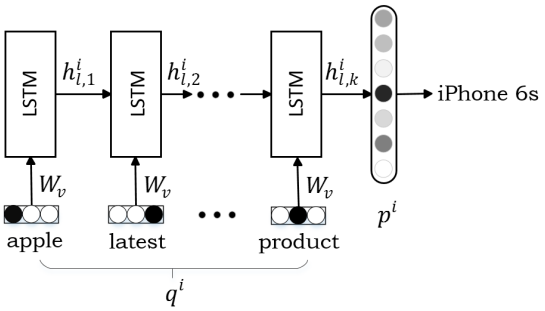

Figure 2: The encoding of query sequence via local LSTM. For simplicity, we only give out one query sequence with k words $\boldsymbol{q}^i = \{apple, latest, released, products\}$ as well as its clicked ads "iPhone 6s". Here $\boldsymbol{h}_{l,j}^i (j = 1, ..., k)$ is the output after encoding first j query words, $\boldsymbol{W}_v$ is the learned word embedding matrix which used to transform one-hot representation words into latent vectors and $\boldsymbol{p}^i$ is the probability distribution of ads. We can see the last hidden layer $\boldsymbol{h}_{l,k}^i$ of $\boldsymbol{q}^i$ is employed to classify the query sequence as "iPhone 6s".

$$\boldsymbol{h}_l^i \equiv \boldsymbol{h}_{l,k}^i \tag{2}$$

where $LSTM(\cdot)$ denotes one step forward pass of the encoding and $\boldsymbol{v}_j^i$ is the j^{th} word of $\boldsymbol{q}^i$. We assume that the output $\boldsymbol{h}_l^i$ of the local LSTM describes the intention of query $\boldsymbol{q}^i$ and can be decoded as a distribution over the clicked ads:

$$\boldsymbol{p}^i = softmax(\boldsymbol{W}_l \cdot \boldsymbol{h}_l^i + \boldsymbol{b}_l) \tag{3}$$

where $\boldsymbol{p}^i$ is the probability distribution with respect to query $\boldsymbol{q}^i$, $\boldsymbol{W}_l$ is the transformation matrix and $\boldsymbol{b}_l$ is the corresponding bias. The probability of the most likely clicked ad for $\boldsymbol{q}^i$ is predicted as follows:

$$p^i = max(\boldsymbol{p}^i) \tag{4}$$

2.3 Global LSTM Model

Our global LSTM model(Figure 3) is proposed to learn the interactions among users, queries and users. After given user u^i and his/her corresponding query q^i, we first transform u^i and q^i into latent vectors $\boldsymbol{u}^i$ and $\boldsymbol{q}^i$ via user embedding matrix $\boldsymbol{W}_u$ and query embedding matrix $\boldsymbol{W}_q$ respectively. Then the global LSTM encodes (user, query) sequence as follows:

$$\begin{aligned} \boldsymbol{h}_{g,u}^i &= LSTM(\boldsymbol{h}_{g,0}^i, \boldsymbol{u}^i) \\ \boldsymbol{h}_{g,0}^i &= \boldsymbol{0} \end{aligned} \tag{5}$$

$$\boldsymbol{h}_{g,q}^i = LSTM(\boldsymbol{h}_{g,u}^i, \boldsymbol{q}^i) \tag{6}$$

$$\boldsymbol{h}_g^i \equiv \boldsymbol{h}_{g,q}^i \tag{7}$$

where $LSTM(\cdot)$ denotes one step forward pass of the encoding. $\boldsymbol{h}_g^i$ is the encoding output of (u^i, q^i) via global LSTM and can been decoded into a distribution over ads as follows:

$$\boldsymbol{p}^i = softmax(\boldsymbol{W}_g \cdot \boldsymbol{h}_g^i + \boldsymbol{b}_g) \tag{8}$$

where $\boldsymbol{W}_g$ and $\boldsymbol{b}_g$ are the transformation matrix and the corresponding bias. The probability of the most likely clicking ad for u^i q^i is calculated as local LSTM does.

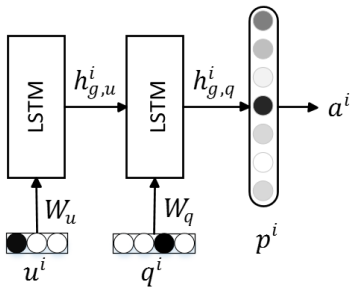

Figure 3: The encoding of *user-query* sequence via global LSTM. For simplicity, we only give out one user-query sequence $\{u^i, q^i\}$ as well as the clicked ad a^i here. $\boldsymbol{W}_u$ is the user embedding matrix, $\boldsymbol{W}_q$ is the query embedding matrix.

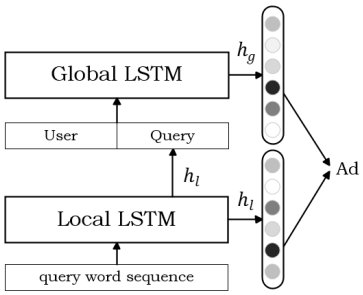

Figure 4: The composite LSTM model for ad recommendation. $\boldsymbol{h}_l$ and $\boldsymbol{h}_g$ are the outputs of local LSTM and global LSTM respectively. The local LSTM receives query word sequence as input, while the global one takes two kinds of inputs, one is the user and another is the encoding of query sequence $\boldsymbol{h}_l$ (learned via local LSTM).

2.4 The Composite LSTM Model

The local LSTM encodes query word sequences and decodes them as distributions over ads. By this way, local LSTM can learn the intra-relations among queries (i.e., how is searching mission is formulated) and the inter-relations between queries and ads (i.e., users' searching intention). The global LSTM encodes (user, query) sequences and decodes them as local LSTM does. In this way, global LSTM can disentangle the inter-relations among users, queries and ads.

In order to fully utilize the interactions between users, queries and ads, we propose composite LSTM here. Generally speaking, we send the last hidden layer output h_l of the local LSTM (i.e., the encoding of query sentences) to the global LSTM via a linear transformation(Figure 4):

$$\boldsymbol{q'} = \boldsymbol{W}_{lg} \cdot \boldsymbol{h}_l + \boldsymbol{b}_{lg} \tag{9}$$

where $\boldsymbol{W}_{lg}$ is the transformation matrix and the new generated $\boldsymbol{q'}$ can be taken as a stronger representation of each query than original $\boldsymbol{q}$. In composite LSTM, the local LSTM keeps predicting ads and at the same time is used as one auxiliary network encoding queries for the master one – global LSTM. During the back-propagation of training process, $\boldsymbol{h}_l$ receives two gradients from both the gobal and local LSTM networks, we define the composite gradient as follows:

$$\triangle \boldsymbol{h}_l = c \cdot \triangle \boldsymbol{h}_m + (1-c) \cdot \triangle \boldsymbol{h}_a \tag{10}$$

where $\triangle \boldsymbol{h}_m$ and $\triangle \boldsymbol{h}_a$ are the gradients from the master and auxiliary network respectively, c is a weight parameter to balance these two gradients.

2.5 Learning

Given all of training examples $T = \cup_{(u,q,a)}$, the objective function of composite LSTM can be defined as follows:

$$J(\theta) = \sum \log p(y|x, \theta) \tag{11}$$

where x represents input sequence(query word sequence for local LSTM and (user, query) sequence for global LSTM) and y is the ground truth clicked ads. θ represents all of the model parameters which are learned by maximizing the log-likelihood of $J(\theta)$ and the gradients of the objective function are computed using the back-propagation through time (BPTT) algorithm[11].

3. EXPERIMENTS

3.1 Experiments Settings

Dataset. The data set 2012 KDD Cup[1] is used in this paper. The original data set is derived from session logs of the Tencent proprietary search engine - soso.com, which contains 149,639,105 records with 23,669,283 distinct users, 26,243,606 distinct queries and 22,238,277 ads. We remove some trivial users, queries and ads. Finally, we obtain our experimental data set with total 57,540 click-through tuples consist of 9,791 distinct users, 7,228 distinct queries and 8,435 distinct ads. The obtained data set is still sparse that over fifty percent of users, queries and ads appear only one click-through tuple. The experiment data are divided into three parts: 42,540 for training, 7,500 for validation and 7,500 for testing. The 7,228 distinct query word sequences have a vocabulary with size of 8,122.

Parameter Settings. For all original input u, q and v we take the one-hot representation and then translate them to the same input dimension d_i using transformation embedding matrix $\boldsymbol{W}_u, \boldsymbol{W}_q, \boldsymbol{W}_v$ respectively. All of embedding matrices are learned with a random initialization. The LSTM cell dimension is denoted as d_c. Parameter optimization is done using mini-batch RMSPROP[4] and the training is terminated when the likelihood of the validation set does not improve for 5 consecutive iterations. The best results of our model reported in section 3.2 has following parameter settings: $d_i = 300$, $d_c = 600$ and the balance weight c is set to 0.7.

Baseline Methods. We implement two baseline methods: (1) Collaborative filtering(CF). We divide our click-through tuple (u, q, a) into two 2-dimension pairs (u, a) and (q, a) and apply collaborative filtering method to discover user-ad and query-ad interrelation respectively. (2) Tensor factorization(TF). As our click-through tuple (u, q, a) is a 3-dimensional data, it is natural to represent the tuple by a 3-order tensor and then factorize and reconstruct the tensor to discover the missing part for prediction. Our TF experiments use HOSVD proposed in [3].

Evaluation Criteria. We evaluate the performance of the model and the baselines in terms of two kinds of criteria.

[1]http://www.kddcup2012.org/c/kddcup2012-track2/data

Table 1: The overall performance for ad recommendation with $d_i = 300$, $d_c = 600$ and $c = 0.7$

Methods	Accuarcy@1	Accuarcy@3	Accuarcy@5	Accuarcy@10	Accuarcy@15	Accuarcy@20	MRR
CF - (q, a)	0.002	0.006	0.009	0.014	0.018	0.022	0.006
CF - (u, a)	0.003	0.010	0.021	0.049	0.074	0.097	0.021
HOSVD	0.198	0.355	0.403	0.474	0.519	0.541	0.296
local LSTM	0.164	0.379	0.514	0.663	0.742	0.788	0.320
global LSTM	0.369	0.607	0.700	0.801	0.842	0.867	0.516
cLSTM	**0.451**	**0.684**	**0.765**	**0.849**	**0.888**	**0.909**	**0.591**

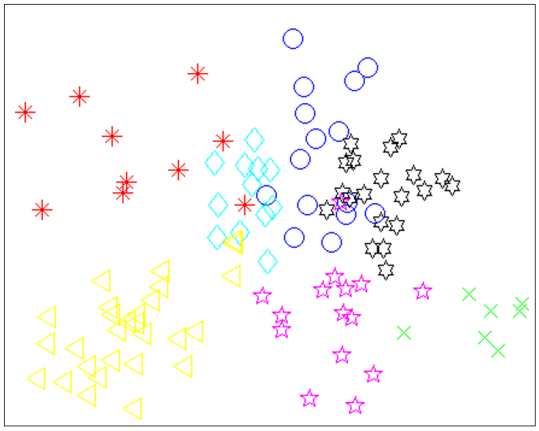

Figure 5: The projection of encoding of query sentences via cLSTM in the 2-dimensional space. The data points with the same colors and shapes indicate that they come from a same category (i.e., different query sentences click a same ads). Here we can see that the encoding exhibits a discriminative embedding representation. Blue circle queries have some overlapping regions in the vicinity with dark hexagon queries since we find in data set that two kinds of clicked ads share many of describing words.

The first one is Accuracy@k proposed in [17] which can be defined as:

$$Accuracy@k = \frac{\#hit@k}{\#tests}$$

where $\#hit@k$ means the number of predicted ads ranking at top k and $\#test$ stands for the number of total testing tuples. Another evaluation criteria is mean reciprocal rank(MRR), which is common for tasks with one ground truth instance.

3.2 Experiments Results

We give out the overall results in Table 1 and analyze them from the following two aspects.

LSTMs *vs.* **CF & TF.** The LSTM models outperform the CF & TF methods much more. For both user-based and query-based CF methods, there are over 8 thousands ads to be predicted while only around 40 thousands training instances used (i.e., data sparisty). The CF-based models even can not accomplish the task due to the data sparsity. TF-based method HOSVD solves the sparsity problem better than CF-based models as it simultaneously utilizes 3-dimensional information. HOSVD gets about **0.275** MRR gain over best performed CF-bases methods. LSTM models show a great superiority to handle data sparsity, the MRR value of the cLSTM comes up to **0.591** and the *Accuracy*@20 of it is around **91%**. The LSTM models learn the interactions between users, queries and clicked ads, and therefore are attractive to perform ads prediction. The worst performed LSTM model – local LSTM that only models the interactions between queries and clicked ads, is still better than conventional CF and TF-based models as the results demonstrated.

cLSTM *vs.* **local & global LSTMs.** The experiment results indicate that the cLSTM model perform better than both the local LSTM and global LSTM in terms of all evaluation criteria. More specifically, the composite model achieves **14.5%** and **84.7%** gains over global and local models in MRR respectively, and in *Accuracy*@20 the performance improvement are **4.8%** and **15.4%**. Comparing the local and global models we find that the latter solves the recommendation problem better, which proves that users are an essential role in recommendation (e.g., different users have similar query intentions). In cLSTM, the local LSTM works like a regularization network by sending the appropriate query encoding to the global LSTM and overcomes the over fitting problem. The cLSTM can learn the encodings of query sentences.

In Figure 5, we embed the encoding learned into a 2D space. In total 107 query sentences belonging to 7 clicked ads are chosen. From Figure 5, we observe that the encodings of query sentences has implicit margins according to their belonging clicked ads(i.e.,categories), which means that the query sentences from the same category are grouped together.

4. CONCLUSIONS

This paper proposes a composite LSTM(cLSTM) model to learn the inter and intra-relations among users, queries and ads. The composite model consists of two sub LSTM networks: global LSTM and local LSTM. The global LSTM works as a master network to encode user and query information while the local LSTM works as an auxiliary network to give a richer representation of query sentences for the global LSTM. The experiment results show the effectiveness of our proposed model and outperform much of the conventional CF and TF-based methods.

5. ACKNOWLEDGMENTS

This work was supported in part by the National Basic Research Program of China under Grant 2015CB352300, NSFC(61402401, U1509206), and the Zhejiang Provincial NSFC(LQ14F010004), in part by the China Knowledge Centre for Engineering Sciences and Technology, and in part by the Fundamental Research Funds for the Central Universities, Qianjiang Talents Program of Zhejiang Province.

6. REFERENCES

[1] T. Anastasakos, D. Hillard, S. Kshetramade, and H. Raghavan. A collaborative filtering approach to ad recommendation using the query-ad click graph. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 1927–1930. ACM, 2009.

[2] Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *Neural Networks, IEEE Transactions on*, 5(2):157–166, 1994.

[3] L. De Lathauwer, B. De Moor, and J. Vandewalle. A multilinear singular value decomposition. *SIAM journal on Matrix Analysis and Applications*, 21(4):1253–1278, 2000.

[4] A. Graves. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*, 2013.

[5] A. Graves, A.-r. Mohamed, and G. Hinton. Speech recognition with deep recurrent neural networks. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 6645–6649. IEEE, 2013.

[6] A. Graves and J. Schmidhuber. Offline handwriting recognition with multidimensional recurrent neural networks. In *Advances in neural information processing systems*, pages 545–552, 2009.

[7] S. Hochreiter, Y. Bengio, P. Frasconi, and J. Schmidhuber. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies, 2001.

[8] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[9] B. J. Jansen and T. Mullen. Sponsored search: an overview of the concept, history, and technology. *International Journal of Electronic Business*, 6(2):114–131, 2008.

[10] S. Rendle, L. Balby Marinho, A. Nanopoulos, and L. Schmidt-Thieme. Learning optimal ranking with tensor factorization for tag recommendation. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 727–736. ACM, 2009.

[11] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Cognitive modeling*, 5(3):1, 1988.

[12] M. Sundermeyer, R. Schlüter, and H. Ney. Lstm neural networks for language modeling. In *INTERSPEECH*, pages 194–197, 2012.

[13] P. Symeonidis, A. Nanopoulos, and Y. Manolopoulos. Tag recommendations based on tensor dimensionality reduction. In *Proceedings of the 2008 ACM conference on Recommender systems*, pages 43–50. ACM, 2008.

[14] T. Wang, J. Bian, S. Liu, Y. Zhang, and T.-Y. Liu. Psychological advertising: exploring user psychology for click prediction in sponsored search. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 563–571. ACM, 2013.

[15] R. J. Williams and D. Zipser. A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1(2):270–280, 1989.

[16] C. Xiong, T. Wang, W. Ding, Y. Shen, and T.-Y. Liu. Relational click prediction for sponsored search. In *Proceedings of the fifth ACM international conference on Web search and data mining*, pages 493–502. ACM, 2012.

[17] H. Yin, B. Cui, Y. Sun, Z. Hu, and L. Chen. Lcars: A spatial item recommender system. *ACM Transactions on Information Systems (TOIS)*, 32(3):11, 2014.

[18] Y. Zhang, H. Dai, C. Xu, J. Feng, T. Wang, J. Bian, B. Wang, and T.-Y. Liu. Sequential click prediction for sponsored search with recurrent neural networks. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014.