Towards Better Ad Experience: Click Prediction Leveraging Sequential Networks Derived Specifically From User Search Behaviors

Shengzhe Li †, **Tomoko Izumi** §, **Yu Kuratake** §, **Jiali Yao** §, **Jerry Turner** §,
Daisuke Kawahara † and **Sadao Kurohashi** †
†Graduate School of Informatics, Kyoto University, Sakyo-ku, Kyoto 606-8501, Japan
§Microsoft Development Co., Ltd., Minato-ku, Tokyo 108-0075, Japan
{lisz, dk, kuro}@nlp.ist.kyoto-u.ac.jp
{tomitsui, yu.kuratake, jiayao, jeromet}@microsoft.com

Abstract

We propose a sequential modeling approach to improve click prediction for search engine advertising. Unlike previous studies leveraging advertisement content and their relevance-to-query information, we employ only users' search behavioral features such as users' query texts and actual click records of both organic search results and advertisements. By leveraging long short-term memory (LSTM) networks, we successfully modeled users' sequential search behaviors and fully utilized them in click predictions. Through experiments conducted with large-scale search log data obtained from an actual commercial search engine, we demonstrated that our method combining users' current and previous search behaviors reaches better prediction performance than baseline methods.

1 Introduction

Search advertisement (ad) is the fundamental source of revenue for internet search services. It plays a major role in digital advertising, which is estimated to account for 43.5% of the whole advertising market.[1,2] In order to gain more ad clicks, a search engine may simply show more ads to the user. Although this may help to increase revenue in a short time, it could hurt long term revenue as this increases users' ad blindness (Hohnhold et al., 2015), meaning they learn to simply ignore ads. Also, showing many unsatisfactory ads may eventually result in losing users, letting them switch to another search engine. For example, when a user searches about "Amazon CEO", the user's intent is obviously to look for information. But a search engine may understand as a search for item "CEO" at Amazon. This kind of errors may dissatisfy users, because mismatched ads occupy the best position in the page, where is supposed to be the answer of the information lookup. In order to satisfy both users and advertisers while not deteriorating the search service's revenue opportunities, it is essential to show ads at the right time when the user's search intent matches with the ad. This will lead the user to conduct a desired action, such as purchasing a product, at the advertiser's site. Therefore, when a search arrives, it is necessary to determine whether it is appropriate to display ads. In other words, if ads are displayed, we predict whether user would click at them. If the probability of click is high, we induce that it is suitable to present the ads. Otherwise, it would be better not to display ads. Therefore, the problem is converted to click prediction, and we determine the appropriateness of the ad display according to the click prediction results.

Click prediction is a widely used technology to improve the ad-related user experience by increasing click through rate (CTR). Click prediction anticipates the probability of ads to be clicked by leveraging various information such as ad contents, ad position, relevance scores between ads and queries, and detailed user's intent signals such as dwell time of each ad click.

[1] https://www.emarketer.com/content/emarketer-total-media-ad-spending-worldwide-will-rise-7-4-in-2018
[2] https://www.statista.com/outlook/216/100/digital-advertising/worldwide

Pacific Asia Conference on Language, Information and Computation (PACLIC 33), pages 461-470, Hakodate, Japan, September 13-15, 2019

However, if search ads are provided through a 3rd party ad platform, the search service may suffer poor ad performance. Because ads are often served through a simple API request between the search service and the ad platform, it is difficult to communicate complex signals such as user's search histories and relate those signals with ad features such as ad contents to maximize ad performance. Due to the limitations on available features, the ad performance of the search engine could not be optimized for the user and it may simply result in showing lots of ads (ad over-triggering) and thus hurt user experience. As of today, there are many internet search services that provide ads through a 3rd party ad platform, and so this is not a trivial issue.

In order to overcome this constraint and provide a better ad experience, we propose a click prediction approach leveraging sequential networks derived specifically from user search behavioral signals available for the search service. Our method utilizes long short-term memory (LSTM) networks to capture the user's one-shot search intent and overall personal preference over ads, and leverages this information to estimate the probability of ad clicks. In this study, we focus solely on user behavioral features, and thus the prediction model is designed without any ad-related information. Our experiments that used a large-scale search log validated the effectiveness of our proposed method.

The remaining of this paper is organized as follows. Section 2 summarizes previous studies on click prediction, Section 3 introduces our methodology of sequential ad click prediction model using users' search behavioral features. Section 4 details our experiments and analysis. Section 5 concludes the paper.

2 Related Work

Studies on ad click prediction have a long history. Using logistic regression with statistical features is the most common method in ad click prediction (Richardson et al., 2007; McMahan et al., 2013; He et al., 2014) because of its small computation complexity but relatively good performance. In recent years, factorization machines (FMs) (Juan et al., 2016; Chen et al., 2009; Ta, 2015), gradient boosting decision trees (GBDTs) (Trofimov et al., 2012), conditional random fields (CRF) (Xiong et al., 2012), and deep neural networks (DNNs) (Zhang et al., 2016) have also been utilized for the ad click prediction task within a single ad impression, and have achieved impressive results. Ling et al. (2017) make an ensemble of these models and apply the ensemble model to a real world search engine.

Recently, sequential ad click prediction based on user behavior has become a hot topic. For example, recurrent neural networks (RNNs) (Mikolov et al., 2010) are used to model users' click and behavioral sequences (Zhang et al., 2014; Liu et al., 2017a) for its good ability at capturing sequential information. In some other related tasks like query suggestion, RNN-based approaches also show their superiority (Chen et al., 2018; Sordoni et al., 2015). Recent studies also use long short-term memory (LSTM) networks (Hochreiter and Schmidhuber, 1997) to model query sequences thanks to its better ability on handling long sequences than RNNs (Deng et al., 2018; Zhu et al., 2017). Moreover, it is observed in Zhang et al. (2014) that the longer the time span between different two searches is, the less impact the former search brings to the latter search. Therefore, hierarchical architectures are proposed to model the difference impact by query level, session level and user level (Sordoni et al., 2015; Chen et al., 2018). All these pieces of evidence indicate that in addition to the information obtained from the current query, short-term and long-term historical features also play an important role in predicting user's ad click behaviors.

In regards to features that reflect users' behaviors, many studies indicate a variety of solutions. First, for the utilization of query texts, employing statistical language models (Salton and Buckley, 1988; Salton et al., 1975; Murdock et al., 2007; Raghavan and Iyer, 2008; Shaparenko et al., 2009; Liu et al., 2015) is common for ad prediction tasks. With the progress of neural networks, deeper utilization of semantics in texts have appeared. CNNs are used to capture semantic information in texts to conduct ad click prediction (Edizel et al., 2017), while RNNs are employed to encode query texts for next query prediction (Sordoni et al., 2015). Zhai et al. (2016) use RNN/LSTM networks to extract intents behind the query texts. In fact, in other areas such as machine translation, sequence-to-sequence modeling with an RNN or LSTM text encoder has become a de facto standard (Bahdanau et al., 2014; Neubig, 2017). Besides, in other tasks, we also find evidence of the availability of text encoding. Smith et al. (2018) employ RNNs to encode event texts, which are quite similar to query texts. These references are solid for us to try encoding query texts with sequential neural models. In the meantime, time interval is proved to be an important indicator on user intent (Liu et al., 2017b; Zhu et al., 2017). These related studies provide a solid base for our feature construction.

3 Methodology

First, we define three terminologies that represent how we partition a search log. An **impression** refers to the point when a search/ad result is shown to a user given a query. In this study, we only use ad impressions, which are logged when ads are shown to users. A **session** is a higher level unit which consists of a sequence of impressions within a short period of time (e.g., 30 minutes (Boldi et al., 2008)) by a specific user. Sessions expire due to various reasons such as task completion, timeout and unexpected cutoff. A **user**'s search history consists of a sequence of sessions by the same user.

As stated in Section 1, our model utilizes users' sequential behavioral information to enhance ad click predictions. To make the sequence better imply the user's preference on ads, we need to make this span as long as possible (Zhang et al., 2014). In regards to sequential modeling, we treat every single query as an estimation unit, and the whole history of the user as a sequence, and employ LSTM networks to process the sequence. In each query, we employ a series of features to represent user's search behaviors such as query texts and click histories on both sponsored and organic search results.

3.1 Observation and Principles

Let us consider the flow of a query to observe what information will render a user's search behavior. First, having a specific intent in her mind, the user issues a query from some entrance to the search engine, which we call **entry point**. Given the **query text**, the search service returns both organic and sponsored search results. The user may give **clicks** to both kinds of results. After a **time interval**, the user may search for another query with a different intent. This is a basic search cycle of a single query. Without any ad content information, it is obvious that one single query is not sufficient to predict ad click probabilities. In order to overcome this constraint, we want to learn the user's behavioral features not only from the current query but also from all the previous queries by this user. Especially, queries within the same session could indicate the transition of the user intent, and all the previous queries may indicate the user's personal preference to ads. Therefore, we propose a sequential framework to better capture the user's sequential behavioral information. For each query, we propose a set of behavioral features to represent the user behavior.

3.2 Framework

As shown in Figure 1, we employ an LSTM network to adapt the inter-impression task. Each state of LSTM in Figure 1 is corresponding with an ad impression returned for one query. Note that the queries corresponding with these ad impressions are not necessarily continuous, because queries that do not trigger ads are merged into features. We use the hidden state of cell t in the LSTM as the distributed representation of the t-th query, and use a multi-layer perception (MLP) to decode the hidden state to determine the probability of the ads being clicked. Note that the probability of a click refers to the likelihood that at least one of the ads presented in the result page is clicked. The probability of no click represents that no ads in the page are clicked. Unlike the hierarchical methods proposed in Sordoni et al. (2015) and Chen et al. (2018), we expand all sessions in the user level with the following two main reasons. First, the separation of sessions does not necessarily mean the gap of intent, manual boundaries in the session level may introduce noise. Second, according to the results in Zhang et al. (2014), the prediction for longer histories works better. Therefore, we connect all the sessions, that is, all the ad impressions in a row as shown in Figure 1.

In our work, we use a weighted softmax cross en-

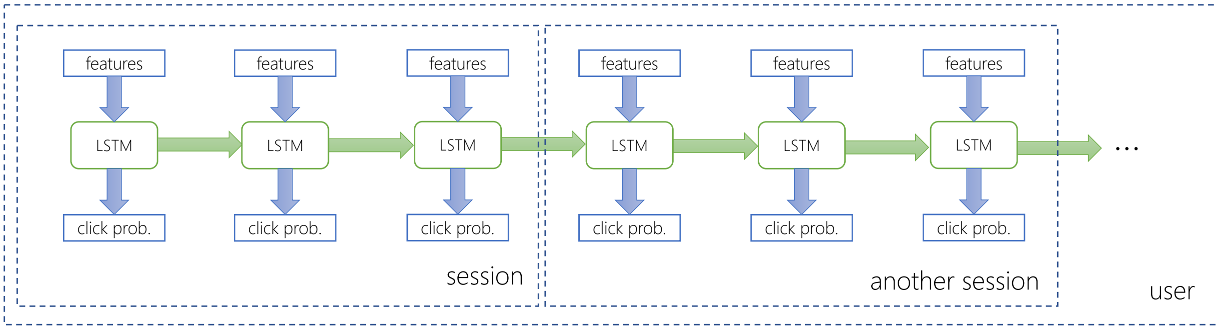

Figure 1: The architecture of our model. We expand all ad impressions of a user in a row, and apply LSTMs to the sequence. In each state, one-shot features are used as the input of the LSTM, and the hidden state is used to predict the ad click probability through an MLP.

tropy as our loss function,

$$L = -\frac{1}{N} \sum_{N} \left[w_0 y_{\text{true}} \log p_0 + w_1 (1 - y_{\text{true}}) \log p_1 \right], \quad (1)$$

where N is the number of training instances, w_1 and w_0 are the weights of clicked and not clicked ad impressions respectively. p_1, p_0 are the predicted probabilities of ad being clicked respectively, and $y_{\text{true}} \in \{0, 1\}$ is the true label indicating whether the ads are clicked.

3.3 Behavioral Feature Construction

As stated above, we define each ad impression as a minimum unit. To reflect user behavioral features as much as possible, we select a series of features that are listed below.

1. **Entry point**. We use signals that indicate where the user issues the query, such as "from the top page of serach service" and "from a browser's address bar".

2. **Query text**. Query text contains most of the information about user's intent for this query. Assume that each query is composed of several words $\{w_1, w_2, \cdots, w_k\}$. Then we obtain their embeddings $Q = \{\boldsymbol{e}_1, \boldsymbol{e}_2, \cdots, \boldsymbol{e}_k\}$. Through an encoder, the query is compressed into a D_h-dimensional embedding, as

$$\boldsymbol{h} = \text{Enc}(\boldsymbol{e}_1 \boldsymbol{e}_2 \cdots \boldsymbol{e}_k), \ \ \text{Enc} : \mathbb{R}^{k \times D_e} \to \mathbb{R}^{D_h}, \quad (2)$$

where D_e is the dimensionality of word embedding vectors. Since RNN and LSTM networks have shown incredible ability to capture sequential and semantic information in texts, as Smith et al. (2018) have done, we employ bidirectional LSTMs to encode each query as shown in Figure 2, and use the concatenation of the final states of the forward and backward cells as the encoding of the query text as $\boldsymbol{h}_Q = \text{concat}(\overrightarrow{\boldsymbol{h}_k}, \overleftarrow{\boldsymbol{h}_1})$. We compare the results of handling the query texts with mean-pooling, character-level CNNs, bidirectional RNNs, bidirectional LSTMs and CNN-BiLSTMs, and find that bidirectional LSTM is the optimal method.

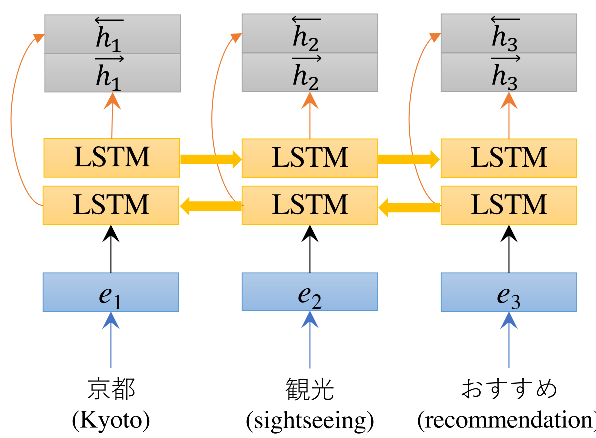

Figure 2: A BiLSTM query encoder to utilize query texts.

3. **Clicks to organic search results**. We count clicks that are made to organic search results between two ad impressions and incorporate this information into our model. We believe

users' clicks to organic search results are also a good indicator to capture the user's search intent. For example, getting more clicks to organic search results is more likely to indicate that the user is looking for pure informational contents, where ads are not useful.

4. **Ad click history**. We straightforwardly use the actual ad click history of the user to predict a personal preference for clicking ads.

5. **Time interval**. We use the time interval between two adjacent ad impressions. We believe that this time interval reflects behavioral features of the user (Liu et al., 2017b; Zhu et al., 2017).

6. **Authentication**. We use the boolean flag that indicates whether the user logged in.

7. **Day of week**. According to our statistics, we observed the likelihood of ads being clicked differs depending of the day of the week, which may be due to the nature of search intent difference between weekends and weekdays. Hence, we believe that it is useful for representing user behavioral features.

As stated in Section 1, our approach is under the limitation that the ad-related information is not available for a search engine employing a 3rd part ad platform. Therefore, no ads related features such as ad contents and relevance scores between ads and queries are used in this study, which are utilized in Zhang et al. (2014).

4 Experiments

We conduct two experiments to validate our proposed approach over a real-world search log dataset. First, we validate the model architecture by comparing it with existing baselines. Then, we explore the impact and relative importance of each feature used in the model. At last, we present two real cases to give an intuitive analysis on the model performance.

4.1 Dataset

We sampled a one-week search log that consists of 18 million impressions from one of the commercial search engines in Japan. From the log, we created a training set with 5.87M ad impressions derived from 508K users, and a validation set with 835K ad impressions from 72.3K users. Because we observed different ad click tendencies between weekdays and weekends, we created two types of test sets, one created from Wednesday's data which consists of 142K ad impressions from 15.4K users, and the other created from Saturday's data which consists of 136K ad impressions from 14.5K users. The Wednesday and the Saturday are in the following week of the week from which we retrieve the training data.

4.2 Evaluation Metrics

We use **AUC** and **F3** scores to evaluate a model from different viewpoints. The AUC score indicates the classification performance of the model, while the F3 score indicates the balance of prediction accuracy and business impact. It is worth mentioning that instead of the F1 score, which indicates the same weights for precision and recall, we set $\beta = 3$ to value recall more. The reason is that recall directly relates to revenue earned by the search engine provider; if a user is highly likely to click ads but the model erroneously predicts that the user would not click ads and does not show ads, the search service would miss its revenue opportunities. We believe that considering the impact to the real business, recall is far more sensitive in this study. Besides, we present a reference measurement, **reduction rate**, to directly measure how much a model is capable to reduce ad displays. A higher reduction rate means more ads removed, but at the same time it increases the risk of losing true clicks. Therefore, we hope to control the reduction rate within an acceptable range instead of making it as high as possible.

4.3 Sequential Framework Effectiveness

In order to prove the effectiveness of our framework, we compare our model with a non-sequential deep neural network method (DNN), as well as logistic regression (LR), which is the most commonly used simple classifier. We use the two test sets to evaluate the performances of these models. The results are listed in Table 1.

The results on the AUC and F3 scores indicate significant superiority of our model to the DNN and LR baselines. Comparing the results of our method and DNN, our utilization of a sequential model like

Table 1: Our model versus two baselines. Our approach is equipped with sequential architecture more than the DNN baseline, and the LR baseline represents with the common industrial practice which is simple and fast. Both ours and DNN are deep neural models, while LR is not.

Model	AUC score (%)			F3 score (%)			Reduction rate (%)		
	Dev.	Test (weekday)	Test (weekend)	Dev.	Test (weekday)	Test (weekend)	Dev.	Test (weekday)	Test (weekend)
Ours	**86.18**	**82.22**	**83.00**	**72.02**	**64.58**	**67.09**	67.63	67.59	65.24
DNN	84.52	80.28	80.97	70.10	63.00	65.73	67.54	65.30	63.19
LR	79.40	76.54	77.01	62.29	56.84	57.98	68.62	73.90	72.91

Table 2: The impacts of each feature tested over the validation set. We remove each feature while keeping other features inboard to see how much the scores deteriorate.

Model	AUC score (%)	F3 score (%)	Reduction rate
Our model	86.18	72.02	67.63
w/o entry point	85.66	69.62	71.84
w/o query text	84.11	70.52	65.19
w/o #click organic	**74.61**	**58.20**	63.59
w/o ad click history	84.40	68.26	68.87
w/o time interval	85.84	71.19	68.89
w/o authentication	86.14	69.11	73.13
w/o day of week	86.10	69.46	72.65

LSTMs indeed improves the performance of ad click prediction. Meanwhile, the improvement of DNN over LR indicates that deeper utilization of information makes sense in this task. Moreover, the comparison between our approach and LR indicates that our method would bring huge improvement over the industrial practice.

Moreover, the results on the two test sets indicate significant differences. This implies the fact that users have higher interest on ads on weekends than on weekdays. This conclusion is in accordance with our intuition that the user search intent is more likely to be informational on weekdays while it is more likely to be transactional, such as shopping, on weekends.

4.4 Feature Impact

In order to discover how much impact each feature brings to the model performance, we remove each feature while keeping other features inboard and observe how much the scores deteriorate on the validation set. A larger deterioration indicates a higher importance of the feature. The results are listed in Table 2.

The user behavioral features include the first five: entry point, query text, organic click count, actual ad click history and time interval, as described in Section 3.3. From the results, it is obvious that the count of organic result clicks from the previous ad impression brings a huge improvement to the model. Without this feature, the AUC score drastically deteriorates by 11.6%, and the F3 score drops by a shocking 13.8%. We explain the reason why it is so powerful as follows. (1) Clicks on organic search results reflect the actual intent of the user's query. From our observation, more clicks to organic results indicate stronger intent to look for information. (2) Only this feature contains sequential information in non-ad impressions. This information would be discarded for non-sequential ad click prediction tasks as no prediction task would be conducted if there is no ad. We also verify that the utilization of query texts is critical, as it could directly reflect the user's behavior and intent. When we stop using query texts, the AUC score drops by over 2%. The ad click history also proves its power with both AUC and F3 results. The drops of 1.78% on AUC and 3.76% on F3 suggest that it is important and useful historical information.

User	Ans.	Pred.	Query	Interval	Auth	Entry	ORC	Ad Click History
User X	0	0	ハヤシライス トマト缶 市販ルー (hashed beef tomato can source)	20	0	8	1	0, 0, 0
	0	0	**子供旅行 おすすめ (good place for traveling with kids)**	18	0	2	1	0, 0, 0, 0
	1	1	**格安航空券 国内 (domestic cheap flight ticket)**	**12**	0	2	**0**	0, 0, 0, 0, 0

Figure 3: A successful case.

User	Ans.	Pred.	Query	Interval	Auth	Entry	ORC	Ad Click History
User A	0	1	パワーポイント (PowerPoint)	<begin>	1	2	0	None
	0	1	パワーポイント 使い方 (how to use PowerPoint)	1	1	8	0	0
	0	0	パワーポイント 結合 (PowerPoint merge)	1	1	8	1	0, 0
User B	1	1	パワーポイント (PowerPoint)	<begin>	1	2	0	None
	0	1	パワーポイント (PowerPoint)	5	1	2	0	0
	0	0	パワーポイント 2018 (PowerPoint 2018)	2	1	8	1	0, 0

Figure 4: A failed case.

Supplementary features including authentication and day of week are also proved to contribute to accurate prediction. Although the differences on AUC are quite small, they have shown considerable impacts on F3. Our interpretation is that they mainly influence the trade-off between precision and recall. In this case, they help the model achieve higher precision, and thus generate differences on F3.

4.5 Case Analysis

We present two actual instances in Figures 3 and 4 to intuitively explain the advantages and shortcomings of our proposed method. Each case contains several continuous ad impressions, which are featurized with query texts, time intervals (larger number means longer interval), authentication (boolean), entry points (category), organic result clicks (ORC; count) and actual ad click histories (boolean sequence).

A successful case is shown in Figure 3, where the second query "good place for travelling with kids" does not yield an ad click while the third query "domestic cheap flight ticket" yields an ad click. In this case, there exists a transition of search intent between the second and the third queries: the second "good place for travelling with kids" being informational while the third "domestic cheap flight ticket" being transactional. To our interpretation, not only the query texts, but also the moderate time interval and the record of zero organic result click contribute to the success of predicting the intent transition, winning over the zero ad click histories.

Meanwhile, a failed case is shown in Figure 4. Two users with exactly the same features searching for the same word "PowerPoint" acted differently. According to their follow-ups, User A has an informational intent, while User B has a transactional one. However, in this case, our proposed model is unable to distinguish the informational intent, as no history is given. This case indicates that our model is highly context-dependent. In the same manner as common models, it cannot handle the randomness of queries well at the beginning of a user's search log.

5 Conclusion

In this paper, we presented an ad prediction method relying solely on user's sequential search behavioral information. Given the constraint that ad-related information is not available, we only used user's search behavioral features. Through the experiments using real data, we proved that our approach reaches better prediction performance than the baselines, and verified the effectiveness of our sequential framework and behavioral feature construction.

In the future, we will apply our model to an actual commercial search engine to validate the effectiveness of our proposed method for better ad experience. Furthermore, we would like to combine our proposed model with the existing ad click prediction models that leverage ad-related information. Because our proposed method and the existing ad click prediction method are complementary, we believe that our proposed method will contribute to further improving the existing click prediction tasks.

Recently, transfer learning methods have shown a strong ability for improving various NLP tasks. BERT (Devlin et al., 2018) based models refreshed almost all the records of open NLP tasks in one night. Therefore, we believe that it is worth trying to handle the query texts with BERT to further improve the performance of the model. Moreover, we plan to conduct visualization on user's behavior in order to better observe the transition of user intent among queries.

Acknowledgement

This research was supported by Rinna Co-operative Research Project, Microsoft Japan Co., Ltd.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

Paolo Boldi, Francesco Bonchi, Carlos Castillo, Debora Donato, Aristides Gionis, and Sebastiano Vigna. The query-flow graph: model and applications. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 609–618. ACM, 2008.

Wanyu Chen, Fei Cai, Honghui Chen, and Maarten de Rijke. Attention-based hierarchical neural query suggestion. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, SIGIR '18, pages 1093–1096, New York, NY, USA, 2018. ACM. ISBN 978-1-4503-5657-2. doi: 10.1145/3209978.3210079. URL `http://doi.acm.org/10.1145/3209978.3210079`.

Ye Chen, Michael Kapralov, John Canny, and Dmitry Y Pavlov. Factor modeling for advertisement targeting. In *Advances in Neural Information Processing Systems*, pages 324–332, 2009.

Weiwei Deng, Xiaoliang Ling, Yang Qi, Tunzi Tan, Eren Manavoglu, and Qi Zhang. Ad click prediction in sequence with long short-term memory networks: an externality-aware model. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 1065–1068. ACM, 2018.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Bora Edizel, Amin Mantrach, and Xiao Bai. Deep character-level click-through rate prediction for sponsored search. *arXiv preprint arXiv:1707.02158*, 2017.

Xinran He, Junfeng Pan, Ou Jin, Tianbing Xu, Bo Liu, Tao Xu, Yanxin Shi, Antoine Atallah, Ralf Herbrich, Stuart Bowers, et al. Practical lessons from predicting clicks on ads at facebook. In *Proceedings of the Eighth International Workshop on Data Mining for Online Advertising*, pages 1–9. ACM, 2014.

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

Henning Hohnhold, Deirdre O'Brien, and Diane Tang. Focusing on the long-term: It's good for users and business. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1849–1858. ACM, 2015.

Yuchin Juan, Yong Zhuang, Wei-Sheng Chin, and Chih-Jen Lin. Field-aware factorization machines for ctr prediction. In *Proceedings of the 10th ACM Conference on Recommender Systems*, pages 43–50. ACM, 2016.

Xiaoliang Ling, Weiwei Deng, Chen Gu, Hucheng Zhou, Cui Li, and Feng Sun. Model ensemble for click prediction in bing search ads. In *Proceedings of the 26th International Conference on World Wide Web Companion*, pages 689–698. International World Wide Web Conferences Steering Committee, 2017.

Pengqi Liu, Javad Azimi, and Ruofei Zhang. Contextual query intent extraction for paid search selection. In *Proceedings of the 24th International Conference on World Wide Web*, pages 71–72. ACM, 2015.

Qiang Liu, Shu Wu, and Liang Wang. Multi-behavioral sequential prediction with recurrent log-bilinear model. *IEEE Transactions on Knowledge and Data Engineering*, 29(6):1254–1267, 2017a.

Yiqun Liu, Xiaohui Xie, Chao Wang, Jian-Yun Nie, Min Zhang, and Shaoping Ma. Time-aware click model. *ACM Transactions on Information Systems (TOIS)*, 35(3):16, 2017b.

H Brendan McMahan, Gary Holt, David Sculley, Michael Young, Dietmar Ebner, Julian Grady, Lan Nie, Todd Phillips, Eugene Davydov, Daniel Golovin, et al. Ad click prediction: a view from the trenches. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1222–1230. ACM, 2013.

Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. Recurrent neural network based language model. In *Eleventh Annual Conference of the International Speech Communication Association*, 2010.

Vanessa Murdock, Massimiliano Ciaramita, and Vassilis Plachouras. A noisy-channel approach to contextual advertising. In *Proceedings of the 1st international workshop on Data mining and audience intelligence for advertising*, pages 21–27. ACM, 2007.

Graham Neubig. Neural machine translation and sequence-to-sequence models: A tutorial. *arXiv preprint arXiv:1703.01619*, 2017.

Hema Raghavan and Rukmini Iyer. Evaluating vector-space and probabilistic models for query to ad matching. In *SIGIR Workshop on Information Retrieval in Advertising (IRA)*, 2008.

Matthew Richardson, Ewa Dominowska, and Robert Ragno. Predicting clicks: estimating the click-through rate for new ads. In *Proceedings of the 16th international conference on World Wide Web*, pages 521–530. ACM, 2007.

Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5): 513–523, 1988.

Gerard Salton, Anita Wong, and Chung-Shu Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975.

Benyah Shaparenko, Özgür Çetin, and Rukmini Iyer. Data-driven text features for sponsored search click prediction. In *Proceedings of the Third International Workshop on Data Mining and Audience Intelligence for Advertising*, pages 46–54. ACM, 2009.

Noah A Smith, Yejin Choi, Maarten Sap, Hannah Rashkin, and Emily Allaway. Event2mind: Commonsense inference on events, intents, and reactions. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 463–473, 2018.

Alessandro Sordoni, Yoshua Bengio, Hossein Vahabi, Christina Lioma, Jakob Grue Simonsen, and Jian-Yun Nie. A hierarchical recurrent encoder-decoder for generative context-aware query suggestion. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 553–562. ACM, 2015.

Anh-Phuong Ta. Factorization machines with follow-the-regularized-leader for ctr prediction in display advertising. In *Big Data (Big Data), 2015 IEEE International Conference on*, pages 2889–2891. IEEE, 2015.

Ilya Trofimov, Anna Kornetova, and Valery Topinskiy. Using boosted trees for click-through rate prediction for sponsored search. In *Proceedings of the Sixth International Workshop on Data Mining for Online Advertising and Internet Economy*, page 2. ACM, 2012.

Chenyan Xiong, Taifeng Wang, Wenkui Ding, Yidong Shen, and Tie-Yan 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.

Shuangfei Zhai, Keng-hao Chang, Ruofei Zhang, and Zhongfei Mark Zhang. Deepintent: Learning attentions for online advertising with recurrent neural networks. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1295–1304. ACM, 2016.

Weinan Zhang, Tianming Du, and Jun Wang. Deep learning over multi-field categorical data. In *European conference on information retrieval*, pages 45–57. Springer, 2016.

Yuyu Zhang, Hanjun Dai, Chang Xu, Jun Feng, Taifeng Wang, Jiang Bian, Bin Wang, and Tie-Yan Liu. Sequential click prediction for sponsored search with recurrent neural networks. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014.

Yu Zhu, Hao Li, Yikang Liao, Beidou Wang, Ziyu Guan, Haifeng Liu, and Deng Cai. What to do next: Modeling user behaviors by time-lstm. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 3602–3608, 2017.