Training Deep Ranking Model with Weak Relevance Labels

Cheng Luo, Yukun Zheng, Jiaxin Mao, Yiqun Liu, Min Zhang, and Shaoping Ma

Tsinghua National Laboratory for Information Science and Technology, Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China
yiqunliu@tsinghua.edu.cn
http://www.thuir.cn

Abstract. Deep neural networks have already achieved great success in a number of fields, for example, computer vision, natural language processing, speech recognition, and etc. However, such advances have not been observed in information retrieval (IR) tasks yet, such as ad-hoc retrieval. A potential explanation is that in a particular IR task, training a document ranker usually needs large amounts of relevance labels which describe the relationship between queries and documents. However, this kind of relevance judgments are usually very expensive to obtain. In this paper, we propose to train deep ranking models with weak relevance labels generated by click model based on real users' click behavior. We investigate the effectiveness of different weak relevance labels trained based on several major click models, such as DBN, RCM, PSCM, TCM, and UBM. The experimental results indicate that the ranking models trained with weak relevance labels are able to utilize large scale of behavior data and they can get similar performance compared to the ranking model trained based on relevance labels from external assessors, which are supposed to be more accurate. This preliminary finding encourages us to develop deep ranking models with weak supervised data.

Keywords: ranking model, click model, deep learning

1 Introduction

Deep neural networks have already delivered great improvements in many machine learning tasks, such as speech recognition, computer vision, natural language processing, and etc. This line of research is often referred to as *deep learning*, as these neural networks usually comprise multiple interconnected layers. A number of "deep models" have been proposed to address the challenges in IR tasks, in particular ad-hoc search. For example, Huang et al. proposed DSSM [1], which is a feed forward neural network to predict the click probability given a query string and a document title.

Learning such kind of deep models requires large amount of labeled data. In IR tasks, relevance judgments, i.e. query-document pairs, often provide supervision for the training process of ranking models. However, large scale of labeled data can be very expensive and time consuming to obtain. To circumvent the lack of labeled samples, researchers proposed to use unsupervised learning methods or weak relevance labels to train ranking models.

Unsupervised neural models aim to describe the implicit internal structure of the textual contents. Several methods for distributed text representations (for example, word2vec [3], GloVe [4], Paragraph2vec [5], and etc.) have been shown to be effective in various tasks such as text classification, recommendation, as well as Web search. The pre-trained distribution of text can be fed into document ranking algorithms to capture the relationship between query string and target documents.

Another line of research attempts to utilize weak labels for model training. The weak labels can be generated based on heuristic methods or users' behavior. Yin et al. proposed to use a 10-slot windows where the first document is treated as a positive sample while the remaining ones are treated as negative ones [7]. Dehghani et al. developed a neural network using the output of an unsupervised ranking model, BM25, as the weak supervision signal [6]. During a user's search session, the click-through data can be collected by the search engine, which is often treated as pseudo feedback from users. Huang et al. used the clicked result as a positive document and randomly selected unclicked results as negative documents. Compared to relevance judgments assessed by people, it is able to obtain much more weak labels by utilizing large scale of behavior data, i.e much more queries and documents. This proves to be vital to the success of a lot of deep models [6, 8].

In this paper, we try to train deep ranking models based on the relevance labels estimated by click model. Click model is widely used nowadays in commercial search engine to model user clicks on a search engine result page (SERP). Different click models are actually based on different user behavior assumptions. One of the key functions of click model is to predict the click probability of a result given the users' behavior on the corresponding query [9]. This probability is shown to be strongly related to the relevance score of the result. We first train several click models with query logs of a commercial search engine to generate weak relevance labels. Then we learn ranking models based on both weak relevance labels and actual relevance judgments assessed people. We adopt several major evaluation metrics to compare the ranking performance.

In summary, the main contributions of our study are as following:

- We investigate the effectiveness of weak relevance labels estimated by several major click models in training deep ranking models.
- We compare the performance of the ranking model trained based on weak relevance labels to that trained on relevance judgments made by assessors.

The remaining of this paper is organized as follows: we review related work in Section 2 and describe the weak relevance label generation procedure in Section 3. The training of deep ranking models is illustrated in Section 4 and the performances of different models are presented in Section 5. Finally, we conclude our research in Section 6.

2 Related Work

2.1 Ranking with deep models

Deep neural networks have achieved dramatic improvements in multiple fields of computer sciences. IR community has begun to apply neural methods to advance state of the art retrieval technology. The central IR task can be typically formalized as a matching problem [10]. Guo et al. suggested that most of recent neural models in IR application can be generally partitioned into two categories [10] according to the model architecture.

The first category is the representation-focused model, which tries to construct a representation for the text in both queries and candidate documents with deep neural networks. Then the similarity between a query and a candidate document can be measured between the two representations with a similarity function. This line of research includes DSSM [1], C-DSSM [12] and ARC-I [13]. These approaches are also referred to as “late combination methods” since the representations of queries and documents are learnt separately. Guo et al. argued that the shortcoming of representation-focused model is that the *semantic* matching is not necessarily appropriate for *relevance* matching in IR tasks.

The second category is the interaction-focused model including DeepMatch [14], ARC-II [13], DRMM [10], and MatchPyramid [15]. In these models, the interactions between queries and candidate documents are fed into neural networks. Thus, the neural networks get the opportunity to capture various matching patterns between pieces of text. In a typical ad-hoc search scenario, the query is usually very brief while the candidate documents can be much longer. The information of matched terms and matched positions is very valuable to learn a good ranker. Therefore, recently more effort has been spent on the interactions-focused models [16].

In this study, we conduct a preliminary study based on a deep ranking model, Duet, which is proposed by Mitra et al. [17]. In this approach, the local and distributional representations (early combination model and late combination model) are learnt simultaneously to take advantage of both relevance matching and semantic matching. More details about our experiment will be presented in Section 4.

2.2 Click model

Modern search engines exploit user’s interaction logs to improve search quality. However, although the click-through-rate (CTR) of a result can be regarded as an implicit relevance feedback from real users, it is systematically affected by some biases. For example, Joachims et al. [19] showed that the CTR could be affected by the *position bias* and top results could attract more user clicks than results in lower positions. Wang et al. [21] found that the presentation style of search results could influence their CTRs.

To distill accurate relevance labels from the noisy and biased query logs, a series of *click models* were proposed in previous studies. Most click models are probabilistic models that follow the *examination hypothesis* [26]: a search result will be clicked ($C_i = 1$) only if it is examined ($E_i = 1$) and it is relevant to the query ($R_i = 1$):

$$C_i = 1 \rightarrow E_i = 1 \wedge R_i = 1 \tag{1}$$

Under this hypothesis, the click probability is given by:

$$P(C_i = 1) = P(E_i = 1)P(R_i = 1) \tag{2}$$

Most click models assume that $P(R_i)$ only depends on the query and result (URL): $P(R_i = 1) = r_{qu}$ and incorporate the behavior biases in the estimation of $P(E_i)$. By inferring r_{qu} from the query log, we can estimate the relevance score between query q and result u.

Different click models make different assumptions of how users browse and interact with SERPs, and therefore, have different estimation of $P(E_i)$. For example, the cascade model proposed by Craswell et al. [26] assumes the user will examine the results sequentially until he or she finds and clicks a relevant result:

$$P(E_1 = 1) = 1 \tag{3}$$

$$P(E_{i+1} = 1|E_i = 1, C_i) = 1 - C_i \tag{4}$$

While the cascade model assumes that the user will always be satisfied with a single click, the Dynamic Bayesian Network model (DBN) model proposed by Chapelle and Zhang [23] uses a separate variable (S_i) to model whether the user will be satisfied after a click.

$$P(S_i = 1|C_i = 1) = s_{qu} \tag{5}$$

$$P(E_{i+1} = 1|E_i = 1, S_i = 0) = \lambda \tag{6}$$

$$P(E_{i+1} = 1|E_i = 1, S_i = 1) = 0 \tag{7}$$

The assumption that the user scans the results on the SERP one-by-one might be too strong. Therefore, Dupret and Piwowarski [24] proposed the User Browsing Model (UBM), which allows the user to skip some of the results. The examination probability of UBM depends on the position of last click (r_i) and the its distance to current result (d_i):

$$P(E_i = 1|C_{1\ldots i-1}) = \gamma_{r_i,d_i} \tag{8}$$

Recently, Wang et al. [21] further found that the user does not always browse the SERP in the top-to-bottom order and there is revisiting behavior in user's interaction with SERPs. So they incorporated these non-sequential behaviors into the Partially Sequential Click Model (PSCM), in which the examination probability is determined by the position of current result (i), the position of previous clicks (m), and the position of next click (n):

$$P(E_i = 1|C_{1\ldots N}) = \gamma_{i,m,n} \tag{9}$$

The CTR of the result can also be influenced by the current search context. Therefore, Zhang et al. [25] built a Task-Centric Click Model (TCM), which incorporates the *query bias* (i.e. whether the query actually matches user's information need) and *duplicate bias* (i.e. whether the result is examined before), to model the click probability at a session level.

In this study, we will use a series of click models, including DBN, UBM, PSCM, and TCM, along with the RCM in which the examination probability $P(E_i = 1)$ is always set to 1, to generate weak relevance labels deep ranking models training. The detailed process of weak relevance label generation is described in Section 3.

3 Weak Relevance Label Generation

Click-through behavior during Web search provides implicit feedback of users' click preferences [18]. Joachims et al. looked into the reliability of implicit feedback and found that the click-through information is "informative yet biased" [19]. User clicks are biased toward many aspects: *position bias*: users tend to prefer the documents higher in the ranking list [19]; *novelty bias*: previously unseen documents are more likely to be clicked [20]; *attention bias* states that the impact of visually salient documents [21].

The central of a click model is to predict the clicked probability (Click-through rate) of a search result. Although the click possibility is not defined as the document relevance, it is closely related to document relevance. It is intuitive that the more relevant a document is, the more likely that a user will click it. Therefore, we can infer the document relevance based on the click probability predicted by click models.

In this study, we adopt several popular click models including DBN, RCM, PSCM, TCM, and UBM. We use an open-source implementation of these models [22].

We trained these click models with a real-world dataset collected by a commercial search engine in China. We removed all the queries that appeared less than 10 times (i.e. less than 10 sessions) since it seem unlikely to train a reliable click model with insufficient behavior data. For each query, at most 500 search sessions are selected for click model training to keep a balance between model precision and the amount of calculation. The statistics of our behavior dataset is shown in Table 1.

Table 1. The statistics of user behavior dataset

Num of Queries	64,169
Num of Documents	747,792
Date	From Apr. 1st 2015 to Apr. 18 2015
Language	Chinese

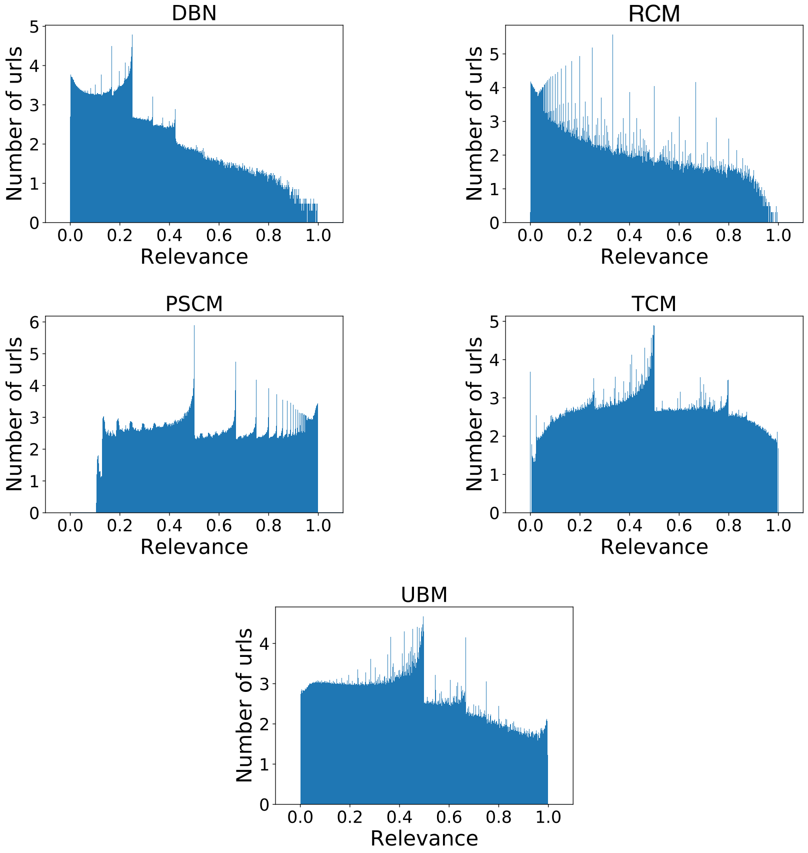

Fig. 1. Click probability distributions of different click models

The distributions of click probability for each click model is illustrated in Figure 1. The x-axis is the click probability ranged from 0 to 1 while the y-axis is the number of corresponding documents in logarithmic scale. We can see that the distributions of DBN, UBM and RCM are quite similar to each other, i.e. the documents with relatively low click probability are much more that that with high click probability. The distributions of PSCM and TCM are close to uniform distribution.

Though we have the predicted click probability for each URL, it is not obvious how to map the click probability to document relevance. We adopt two different strategies to organize document pairs. More detailed will be discussed in the Section 4.

In this section, we discuss how to generate weak relevance labels with click models and users' click-through data. In previous study by Huang et al. [1], they proposed to use a clicked document as the positive (relevant) sample and randomly select an unclicked document as the negative sample. The click-through action can also be treated as a kind of "weak supervision". Although their approach may generate more document pairs for training as each search session can be utilized to generate training pairs, we argue that their methods are more likely to be affected by the noise and bias from individual user's actions. For example, if we have a document pair $\langle D^{+}, D^{-}\rangle$ sampled from a search session, it is possible that D^{-} is as relevant as D^{+}, even more relevant than D^{-}. The reason that the user did not click D^{-} might be that the D^{-} is ranked at lower positions and the user got satisfied by D^{+}, i.e. "position bias" in Web search. Our weak relevance label generation method is able to utilize the behavior of a group of users to reduce the impact of position bias. Our method may need more behavior data to train click model compared to directly sampling from search sessions. We argue that it is possible to construct a dataset with millions of queries with public available query logs (Sogou or Yandex) [22] in lab environment.

4 Deep Ranking Models with Weak Relevance Label

In this section, we describe how we train the deep ranking model with weak relevance labels estimated by click model. The performances of ranking models based on the output of several click models are compared. We also investigate whether the ranking model based on weak labels can get similar performance compared to that based on strong labels which are assessed by human.

4.1 Ranking Model

In this study, we choose to train our ranking model based on one of the most recent approaches, Duet, which was proposed by Mitra et al. [17]. According to Guo et al. taxonomy [10], most neural ranking models can be classified into two categories: *representation-focused* methods which try to get a good representation for query and document and *interaction-focused* methods which put emphasis on capturing the textual matching pattern between query and document.

Duet model actually combines these two lines of research. It composed of two separate neural networks, a local one and a distributed one. The two networks are jointly learnt as part of a single network.

The local model estimates the document relevance based on the exact matches of query terms in the document. It uses a local term representation, i.e. the one-hoc vectors which are widely used in traditional retrieval models. The local model focuses on capturing the exact matches on term level and terms are considered to be distinct entities. As suggested by Guo et al., the exact matching between query and document is valuable to measure the document relevance [10]. The distributed model first learns low-dimensional vector representations for both query and document. Then it estimates the positional similarity between query and document. Instead of the higher-dimensional one-hot representations, distributed model projects n-graph vectors of query and document into an lower-dimensional embedding space. This would be helpful to solve the vocabulary mismatch problem. The Duet model linearly combines the local model and the distributed model, which are jointly trained on labeled query-document pairs:

$$f^{duet}\left(\mathbf{Q},\mathbf{D}\right) = f^{l}\left(\mathbf{Q},\mathbf{D}\right) + f^{d}\left(\mathbf{Q},\mathbf{D}\right) \tag{10}$$

where $\mathbf{Q}$ is the query and $\mathbf{D}$ is the document pair. f^l and f^d denotes the local model and the distributed model respectively.

In our experiment, we use the implementation of Duet model which was released by the authors[1]. The original Duet model was trained on an English corpus. In our experiment, we did some necessary data pre-processing to make the model appropriate for Chinese environment: First, all queries and documents are segmented into words. The original Duet model used 2000 most frequent n-grams for n-graph. We put 5000 most frequent Chinese n-grams into the vocabulary. We adopt the other parameters in the original Duet mode, including the dropout rate and the learning rate. The model were trained based on a single GPU.

4.2 Dataset

The dataset for model training includes the following parts:

1. Weak Relevance Label: as mentioned in Section 3. We have estimated click probability for query-document pairs.
2. Strong Relevance Label: we have 200 queries which are released for NTCIR WWW task [27]. For each query, there are some documents whose relevances are judged by professional assessors in a five level scale (from irrelevant to high relevant). The number of max/avg/min judged documents is 424/170/120 respectively.

We want to investigate the performance of ranking models trained with strong relevance label. Therefore, we randomly split the Strong Relevance Label dataset into two parts: Training Set contains 150 queries while the Test Set contains 50 queries.

In the remaining of this paper, we evaluate all the ranking models based on the Test Set. We use AP, ERR, nDCG@10, P@10, Q-measure and RBP, which is calculated with an open-source tool `NTCIREVAL`[2]. We also introduce a widely used baseline method BM25.

4.3 Comparison between models based on different weak labels

We first look into the performance of the rankers based on different click models. Recall that the Duet model was trained based on document pairs, e.g. $\langle D^{+}, D^{-}\rangle$. We design two methods to organize training samples.

The first method is called Absolute method (ABS): we can map the click probability to relevance score by using a map function $rel\left(p\right)$. In our approach,

[1] `https://github.com/bmitra-msft/NDRM/blob/master/notebooks/Duet.ipynb`

[2] `http://research.nii.ac.jp/ntcir/tools/ntcireval-en.html`

we simply split the probability into 4 segments and each segments represents a relevance level respectively, e.g. the relevance score is 1 if the click probability is between 0.0 to 0.25. Then we adopt the method in Mitra et al.'s study [17] to organize document pairs. For a document pair $\langle D^+, D^- \rangle$, the relevance scores of two documents can be 3 v.s. 1/0, or 2 v.s. 0.

The second method is called Relative methods (REL): Assume we have two documents, d_a and d_b, their relevance scores are s_a and s_b respectively. If $s_a - s_b$ is greater than a predefined threshold t. Then $\langle d_a, d_b \rangle$ can be viewed as a valid training sample. In our experiments, we use $t = 0.42$ to make sure that the number of training pairs are comparable to that with Absolute method.

Table 2. Comparison between ranking models based on different weak relevance labels (The ranker with best performance in FIX is marked in bold while that in REL is marked with underline.)

	Model	AP	ERR	nDCG@10	P@10	Q	RBP	#Pair
ABS	DBN	**0.6283**	0.5180	0.5374	0.6540	**0.6385**	**0.3925**	11,251
	RCM	0.5569	0.4924	0.4922	0.6080	0.5761	0.3446	18,554
	PSCM	0.6251	0.5151	0.5364	0.6640	0.6344	0.3829	59,296
	TCM	0.6264	0.5124	0.5412	0.6680	0.6359	0.3889	42,268
	UBM	0.6240	**0.5302**	**0.5542**	**0.6840**	0.6333	0.3855	34,537
REL	DBN	0.6271	0.5197	0.5387	0.6600	0.6379	0.3880	8,339
	RCM	0.5662	0.5083	0.5107	0.6200	0.5863	0.3528	20,719
	PSCM	0.6276	0.5251	0.5469	0.6720	0.6366	0.3866	70,682
	TCM	0.6265	0.5012	0.5454	0.6760	0.6348	0.3872	43,088
	UBM	0.6221	0.5097	0.5427	0.6660	0.6333	0.3871	32,919
Baseline	BM25	0.5591	0.4657	0.4405	0.5560	0.5772	0.3386	747,792

The performance ranking models based on different weak relevance labels is presented in Table 2. We can see that the methods based on the training samples which are generated by ABS method is slightly better than that generated by REL method. It is potentially due to ABS method is able to produce samples with higher quality. For example, the REL sample may generate a document pair whose click probability is $\langle 0.42, 0.0 \rangle$. This sample will not be accepted by ABS method. We find that the click models which are most helpful are different for ABS and REL. For ABS, DBN and UBM are more effective while for REL, PSCM, TCM and DBN are more beneficial for model training. The more complex models (DNB, PSCM, TCM and UBM) are more likely to generate training samples of high quality than naive model like RCM, since the more complex models can better estimate the click probability of documents.

4.4 Comparison between strong/weak relevance labels

We further investigate the performances of rankers based on strong and weak relevance labels.

For strong relevance labels, we adopt a similar approach like ABS method to differentiate positive documents and negative ones. The smaller the threshold (t) is, the more training samples we will get. The evaluation results are shown in Table 3.

Table 3. Comparison between models which are based on strong/weak relevance labels (The ranker with best performance trained on weak labels is marked in bold while that trained on strong labels is marked with underline.)

	Model	AP	ERR	nDCG@10	P@10	Q	RBP	#Pair
Strong label	Duet(t=1)	0.6416	0.5418	0.5578	0.6800	0.6466	0.3897	469,790
	Duet(t=2)	0.6293	0.5214	0.5403	0.6560	0.6395	0.3821	95,985
	Duet(t=3)	0.6203	0.5118	0.5084	0.6400	0.6263	0.3781	2,557
Weak label+ABS	DBN	**0.6283**	0.5180	0.5374	0.6540	**0.6385**	**0.3925**	11,251
	RCM	0.5569	0.4924	0.4922	0.6080	0.5761	0.3446	18,554
	PSCM	0.6251	0.5151	0.5364	0.6640	0.6344	0.3829	59,296
	TCM	0.6264	0.5124	0.5412	0.6680	0.6359	0.3889	42,268
	UBM	0.6240	**0.5302**	**0.5542**	**0.6840**	0.6333	0.3855	34,537

We can see that the models which are based on strong labels are slightly better than that are based on weak labels. This conclusion is consistent across different evaluation metrics. The reason may be due to that in our experiments, rankers with strong labels have the opportunity to utilize much more training samples. If we look into the rankers with different threshold in strong label group, we find that the Duet(t=1) performs much better that the remaining two models. The number of training document pair in Duet(t=1) is also much larger than that in the other two models. This observation suggest that it is necessary to feed large amounts of training samples, even they contains more noise, to train a good ranking sample. In this pilot study, the scale of data we used is relatively small due to the limit of calculation resource. We would like to leave exploration with much more data in our future work.

All the neural ranking models (except that for RCM) performs significantly better than BM25 (p¡0.01). This encourages us to continue applying neural network in IR tasks.

5 Conclusions and Future Work

In this study, we present a novel neural ranking model training method based on weak relevance labels. We propose to generate weak relevance labels for documents by training click models with users' click behavior. Experiments based on a real-world user behavior dataset demonstrate that the ranking models trained with weak labels can get similar performance compared to that with relevance judgments. We also find that the more data (even more noisy) fed into the neural model, the better performance the model can achieve.

Our work has a few limitations. First, deep learning for IR is developing rapidly and a number of neural methods have been proposed recently. We should validate the effectiveness training methods with various neural models. Second, compared to previous attempts [10, 12, 6] based on millions of queries, the dataset in our experiment is too small. We would like to explore if we will get better performance with a larger dataset in our future work.

References

1. Huang, Po-Sen, et al. "Learning deep structured semantic models for web search using clickthrough data." Proceedings of the 22nd ACM international conference on Conference on information & knowledge management. ACM, 2013.
2. Voorhees, Ellen M., and Donna K. Harman, eds. TREC: Experiment and evaluation in information retrieval. Vol. 1. Cambridge: MIT press, 2005.
3. Mikolov, Tomas, et al. "Distributed representations of words and phrases and their compositionality." Advances in neural information processing systems. 2013.
4. Pennington, Jeffrey, Richard Socher, and Christopher D. Manning. "Glove: Global Vectors for Word Representation." EMNLP. Vol. 14. 2014.
5. Le, Quoc V., and Tomas Mikolov. "Distributed Representations of Sentences and Documents." ICML. Vol. 14. 2014.
6. Dehghani, Mostafa, Hamed Zamani, Aliaksei Severyn, Jaap Kamps, and W. Bruce Croft. "Neural Ranking Models with Weak Supervision." arXiv preprint arXiv:1704.08803 (2017).
7. Yin, Dawei, Yuening Hu, Jiliang Tang, Tim Daly, Mianwei Zhou, Hua Ouyang, Jianhui Chen et al. "Ranking relevance in yahoo search." In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 323-332. ACM, 2016.
8. Salakhutdinov, Ruslan, and Geoffrey Hinton. "Semantic hashing." International Journal of Approximate Reasoning 50, no. 7 (2009): 969-978.
9. Chuklin, Aleksandr, Ilya Markov, and Maarten de Rijke. "Click models for web search." Synthesis Lectures on Information Concepts, Retrieval, and Services 7.3 (2015): 1-115.
10. Guo, Jiafeng, Yixing Fan, Qingyao Ai, and W. Bruce Croft. "A deep relevance matching model for ad-hoc retrieval." In Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, pp. 55-64. ACM, 2016.
11. Mitra, Bhaskar, and Nick Craswell. "Neural Models for Information Retrieval." arXiv preprint arXiv:1705.01509 (2017).
12. Shen, Yelong, Xiaodong He, Jianfeng Gao, Li Deng, and Grgoire Mesnil. "Learning semantic representations using convolutional neural networks for web search." In Proceedings of the 23rd International Conference on World Wide Web, pp. 373-374. ACM, 2014.
13. Hu, Baotian, Zhengdong Lu, Hang Li, and Qingcai Chen. "Convolutional neural network architectures for matching natural language sentences." In Advances in neural information processing systems, pp. 2042-2050. 2014.
14. Lu, Zhengdong, and Hang Li. "A deep architecture for matching short texts." In Advances in Neural Information Processing Systems, pp. 1367-1375. 2013.
15. Pang, Liang, Yanyan Lan, Jiafeng Guo, Jun Xu, Shengxian Wan, and Xueqi Cheng. "Text matching as image recognition." arXiv preprint arXiv:1602.06359 (2016).

16. Hui, Kai, Andrew Yates, Klaus Berberich, and Gerard de Melo. "A Position-Aware Deep Model for Relevance Matching in Information Retrieval." arXiv preprint arXiv:1704.03940 (2017).
17. Mitra, Bhaskar, Fernando Diaz, and Nick Craswell. "Learning to Match Using Local and Distributed Representations of Text for Web Search." arXiv preprint arXiv:1610.08136 (2016).
18. Agichtein, Eugene, Eric Brill, Susan Dumais, and Robert Ragno. "Learning user interaction models for predicting web search result preferences." In Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 3-10. ACM, 2006.
19. Joachims, Thorsten, Laura Granka, Bing Pan, Helene Hembrooke, and Geri Gay. "Accurately interpreting clickthrough data as implicit feedback." In Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 154-161. Acm, 2005.
20. Zhang, Yuchen, Weizhu Chen, Dong Wang, and Qiang Yang. "User-click modeling for understanding and predicting search-behavior." In Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 1388-1396. ACM, 2011.
21. Wang, C., Liu, Y., Zhang, M., Ma, S., Zheng, M., Qian, J. and Zhang, K., 2013, July. Incorporating vertical results into search click models. In Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval (pp. 503-512). ACM.
22. Wang, Chao, Yiqun Liu, Meng Wang, Ke Zhou, Jian-yun Nie, and Shaoping Ma. "Incorporating non-sequential behavior into click models." In Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 283-292. ACM, 2015.
23. Chapelle, Olivier, and Ya Zhang. "A dynamic bayesian network click model for web search ranking." In Proceedings of the 18th international conference on World wide web, pp. 1-10. ACM, 2009.
24. Dupret, Georges E., and Benjamin Piwowarski. "A user browsing model to predict search engine click data from past observations." In Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval, pp. 331-338. ACM, 2008.
25. Xu, Wanhong, Eren Manavoglu, and Erick Cantu-Paz. "Temporal click model for sponsored search." In Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval, pp. 106-113. ACM, 2010.
26. Nick Craswell, Onno Zoeter, Michael Taylor, and Bill Ramsey. 2008. An experimental comparison of click position-bias models. In WSDM?08. ACM, 87?94.
27. Cheng Luo, Yukun Zheng, Yiqun Liu, Xiaochuan Wang, Jingfang Xu, Min Zhang and Shaoping Ma. SogouT-16: A New Web Corpus to Embrace IR Research. The 40th ACM SIGIR International Conference on Research and Development in Information Retrieval. In SIGIR 2017.