Soft Computing
https://doi.org/10.1007/s00500-022-07433-w

APPLICATION OF SOFT COMPUTING

Context-aware ranking refinement with attentive semi-supervised autoencoders

Bo Xu[1] · **Hongfei Lin**[1] · **Yuan Lin**[2] · **Kan Xu**[1]

Accepted: 2 August 2022

Abstract
Learning to rank methods aim to learn a refined ranking model from labeled data for desired ranking performance. However, the learned model may not improve the performance on each individual query because the distributions of relevant documents among queries are diversified in document feature space. The performance of learned ranking models may be largely affected by the usefulness of document features. To generate high-quality document ranking features, we capture the local context information of individual queries from the top-ranked documents of an initial retrieval using pseudo-relevance feedback. Based on the top-ranked feedback documents, we propose an attentive semi-supervised autoencoder to refine the ranked results using an optimized ranking-oriented reconstruction loss. Furthermore, we devise the hybrid listwise query constraints to capture the characteristics of relevant documents for different queries. We evaluate the proposed ranking model on LETOR collections including OHSUMED, MQ2007 and MQ2008. Our model produces better experimental results and consistent improvements of ranking performance over baseline methods.

Keywords Ranking refinement · Learning to rank · Pseudo-relevance feedback · Information retrieval · Machine learning

1 Introduction

Ranking performance of search engines affects user satisfaction in Web information acquisition. To improve the ranking performance, various information retrieval (IR) approaches have been devised to produce more accurate ranking lists of items for certain information needs. The information needs are always characterized by keyword-based queries. How to quantify the relevance between queries and documents remains a crucial topic for optimizing ranking approaches.

To achieve better performance in webpage ranking, supervised machine learning methods have been integrated in ranking process and exhibited satisfactory performance. These ranking methods are named as learning to rank (Liu 2009; Qin et al. 2010). Learning to rank methods incorporate the pointwise, pairwise or listwise ranking constraints into the original loss functions of supervised models for ranking-oriented loss. Fix-length document feature vectors are treated as inputs of learning to rank for constructing ranking models. The models are then iteratively optimized by reducing the pre-defined ranking loss. The outputted models are used to predict the document ranking list of unseen queries.

The quality of the learned ranking models can be affected by several factors, particularly by the usefulness of the document features. Document features of learning to rank are used to characterize the relevance between a certain query and its corresponding documents. Widely used document features include the scores yielded by existing unsupervised ranking models, such as the vector space model and the BM25 model. Although ranking performance has been much improved using various relevance-based statistical ranking features, the performance can be further enhanced by enriching the feature space with latently useful ranking information, which remains an open research question in the field of IR.

An effective way to generate useful document features is to take full consideration of the query-specific feature distributions. Ideal ranking performance would be achieved by optimizing the ranking models toward the characteristics of each individual query. However, since the number of queries tends to be infinite in real IR applications, the ideal performance can hardly be achieved due to high cost and low

✉ Bo Xu
xubo@dlut.edu.cn

[1] School of Computer Science and Technology, Dalian University of Technology, Dalian, China

[2] WISE Lab, Faculty of Humanities and Social Sciences, Dalian University of Technology, Dalian, China

Published online: 25 August 2022

generalization ability. Therefore, we can learn a feedback-enhanced ranking model in ranking refinement. For example, the queries can be represented with the top ranked documents from pseudo-relevance feedback as local ranking context. Pseudo-relevance feedback, as an effective method in IR optimization, assumes the top ranked documents are highly relevant to given queries, which can be used to refine the ranking results (Lavrenko and Bruce 2001; Robertson Stephen and Sparck Jones 1976; Salton and Buckley 1997; Zhai and Lafferty 2004). Learning to rank methods can benefit from the local context obtained by pseudo-relevance feedback (Lavrenko and Bruce 2001; Zhai and Lafferty 2004). However, the context information has not been effectively investigated because it is challenging to combine the global and local ranking information in an united framework.

In this work, we adopt autoencoder-based neural networks to generate highly effective and compact query-specific document features via pseudo-relevance feedback. Autoencoders (Bengio 2009) can be used for automatically generating learning representations of features in different tasks. An autoencoder works by reconstructing the inputs of neural networks from its outputs. Certain loss functions are usually used to measure the reconstruction capability. The loss functions are defined to retain the most useful information and remove the useless information of the inputs for high-quality hidden representations. The learned hidden representations are taken as the features for task-specific learning process. For example, Zhai and Zhang (2016) modified the loss function of autoencoders for supervised text-based sentiment analysis, which motivates other researchers to adapt modified autoencoders to other related tasks. Based on this idea, our work seeks to modify the learning process of autoencoders for learning to rank, particularly addressing the ranking context and query-level information.

Our previous work (Xu et al. 2017) has incorporated ranking information into autoencoders to improve the ranking performance by considering two important factors: feature importance and query constraints. Feature importance is integrated into the loss function while reconstructing different dimensions of original feature representations, and query constraints are used to measure the query-level difference in ranking performance. Three query constraints (Xu et al. 2019) were investigated to promote the ranking performance, respectively. This paper is based on our previous work to carry out further research for ranking refinement.

In general, compared with the previous works, this paper has made improvements and optimizations in three aspects. First, to consider the ranking context for accurate ranking, we adopt pseudo-relevance feedback for a context-aware ranking refinement. Pseudo-relevance feedback seeks to improve the ranking performance by generating effective query-specific document features. In our autoencoder-enhanced ranking framework, pseudo-relevance feedback guides the learned model to highlight the local feedback information from initial ranking, which has been largely ignored in previous research. Second, we propose an attention mechanism to accurately measure feature importance in reconstructing the inputs of autoencoders. Unlike measuring feature importance using an pre-trained ranker (Xu et al. 2017, 2019), in this work, attention mechanism is introduced to precisely evaluate the feature importance by dynamically adjusting the feature importance in an autoencoder-enhanced re-ranking process. Third, beyond the previously investigated pairwise and listwise query constraints (Xu et al. 2019), we propose the hybrid listwise constraints to effectively encode more comprehensive query-level ranking information. The hybrid listwise constraints not only consider the cross-entropy ranking loss of directly optimizing evaluation measures, but also make the best use of the change of ranking performance. Therefore, the hybrid constraints contribute more query-level ranking information to the overall ranking performance. Our experiments were done on three benchmark rank-based collections in LETOR. Evaluation results show that the proposed models can generate more useful document features, which significantly enhance the ranking performance. We summarize the main contributions of this work as follows.

(1) We use pseudo-relevance feedback to capture more context information for autoencoder enhanced ranking. Autoencoders are used to learn effective ranking features in consideration of ranking contexts. The learned features are more context-aware and highly useful for improving the ranking performance.
(2) We adopt the attention mechanism to measure the utility of ranking features. Compared with previous work, the attentive feature weighting dynamically computes the importance of features during model training, and helps refine the feature space for better ranking results.
(3) We propose the hybrid listwise query constraints to capture fine-grained query-level information. The hybrid constraints not only consider the change of ranking performance in query level, but also aim to directly optimize the listwise ranking loss for learning more powerful ranking features.

2 Related work

Ranking refinement, as a fundamental yet important task in IR, has been studied in recent years. To refine the ranked results, learning to rank methods have been proposed (Liu 2009; Qin et al. 2010) to boost retrieval effectiveness using supervised models. In related work, different learning to rank approaches are proposed (Friedman 2001; Burges et al. 2005; Cao et al. 2007; Freund et al. 1998; Burges 2010; Burges et al. 2007; Wang et al. 2018; Lucchese et al. 2018; Tax et al.

2015; Busolin et al. 2021; Chen et al. 2021; Tran and Yang 2021; Al-Asadi and Tasdemir 2021, 2022). For example, Feng et al. (2018) promoted the diversity of retrieved results using a Monte Carlo tree search enhanced decision process. Chen et al. (2021) developed an interaction observation-based model (IOBM) to estimate the observation probability in counterfactual learning to rank. In general, learning to rank approaches use different kinds of ranking loss functions to optimize the ranking models for refined ranking performance. Many previous works have been proposed to optimize learning to rank for better ranking performance (Schuth et al. 2016; Mehrotra and Yilmaz 2015; Niu et al. 2014).

The training data for ranking involve a query set. Each query is related to a document set, each document is represented as a feature vector, and each feature is used to indicate the relevance between the query and the document. Since the features are the basic learning units in training ranking models, the quality of features directly affect the usefulness of the learned model. To learn more useful ranking features, feature importance and query-level semantics need to be critically considered in model optimization. Deep neural networks can be used to produce effective ranking features, particularly using autoencoder-based building blocks (Bengio 2009). To learn useful data embedding, autoencoder seeks to model core information of data by reconstructing between inputs and outputs. The reconstruction capability is guided using certain loss functions in different tasks. Some recent works have used autoencoder to learn task-specific representations of data. For example, Wang et al. (2015) proposed to use stacked denoising autoencoders for tag recommendation. Sedhain et al. (2015) used autoencoder to improve the performance of collaborative filtering in recommendation systems. Zhuang et al. (2015) addressed the semi-supervised multi-task learning based on feature representation learning process. Li et al. (2015) embedded paragraphs using LSTM integrated autoencoders. These studies have demonstrated the powerful reconstruction capability of autoencoders in learning effective feature representations through effective task-oriented optimizations.

In the field of learning to rank, different neural ranking models have been proposed in recent years (Wang et al. 2017; Joachims et al. 2017; Zhuang et al. 2017; Wang and Klabjan 2017; Wu et al. 2018; He et al. 2018; Yin et al. 2016; Shao et al. 2019; Ahmad 2019; Rosset et al. 2019; Macavaney et al. 2019; Hansen et al. 2019; Formal et al. 2021; Kim et al. 2021, ?; Ai et al. 2018; Yoon et al. 2018). Choi et al. (2021) proposed to adopt multi-teacher distillation with a cross-encoder and a bi-encoder rankers for BERT-based neural ranking models. Zhu et al. (2021) proposed a contrastive learning based user behavior modeling method for context-aware document ranking. Lee et al. (2021) presented a dual correction strategy for distilling the ranking information from the teacher model to the student model. Huybrechts (Goeric 2016) enriched the feature space of learning to rank using both shallow networks and deep networks for effective ranking of documents. Fan et al. (2018) incorporated relevance in diversified granularity to gain an enhanced performance of ad-hoc retrieval. Yang et al. (2021) presented an overview of the people search system and discussed how to build deep neural network models for real scenarios. Different from these studies, this work mainly focuses on the enrichment of feature space of learning to rank using pseudo-relevance feedback and autoencoder.

3 Our method

In this section, we introduce our framework for context-aware ranking refinement. Our framework mainly contains two stages, which is shown in Fig. 1. The first stage uses pseudo-relevance feedback to capture the context information in three steps. The first step conducts a basic retrieval to obtain an initial ranking list. This retrieval can be based on any ranking models so that the outputted ranking list sorts the documents with higher relevance at the top of the list. Based on the ranking list, the second step adopts pseudo-relevance feedback (PRF) to obtain a set of feedback documents with rich context information. The last step uses relevance feedback to obtain the context information from the feedback documents to capture query-specific context-aware information. PRF, as a classical query expansion strategy in IR, assumes that top-ranked documents of the initial retrieval closely correlate to the query, and can be used to enrich the query. Therefore, we use the top-ranked documents, as context information, to refine the initial ranking list. The second stage incorporates context information into autoencoders to learn ranking model for a refined ranking list of documents. To construct the ranking refinement model, we adopt an attention mechanism to discriminate feature importance, and propose a hybrid method to incorporate query-level ranking constraints. Since the refined list considers more context information based on the initial list, we believe that the ranking performance can be further enhanced to fulfill the information needs of the queries.

3.1 Context information acquisition based on pseudo-relevance feedback

The initial retrieval can be performed based on traditional retrieval models or learning to rank based models. Since learning to rank-based models have exhibited better performance in relevant tasks, we adopt learning to rank-based ranking models to execute the initial retrieval and produce more relevant feedback documents. We then use the top-ranked documents from pseudo-relevance feedback as context to improve the performance.

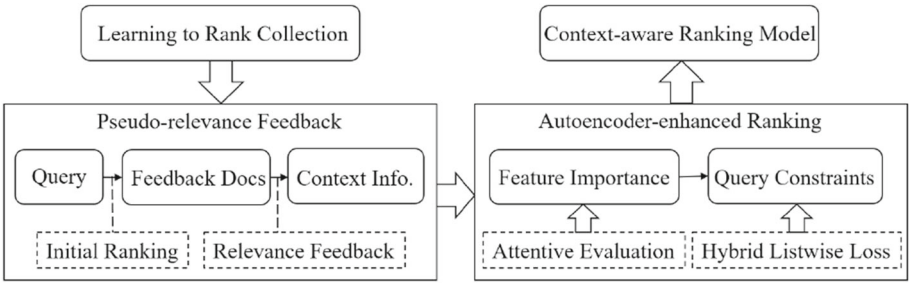

Fig. 1 Pipeline of the proposed framework

In our implementation, we adopt the listwise ListNet method. which uses neural networks as the scoring function, and the probabilistic listwise loss function as the objective of optimization. The training phase of ListNet is guided by a cross-entropy-based permutation probability. The permutation probability is obtained using the following equation.

$$loss(y, z(f(x))) = \sum_{j=1}^{n(i)} p_y(x_j) log(p_{z(f(x))}(x_j)) \tag{1}$$

where y denotes an ideal ranking list obtained using document labels. $z(f(x))$ denotes the predicted ranking list of documents by our model. $P_y(x_j)$ is the scoring function based on permutation probability. $P_{z(f_w)}(x_j)$ denotes the permutation probability on $z(f(x))$. $n(i)$ represents the number of documents of a certain query i^{th}. Based on the pre-trained ListNet-based model, we obtain an initial ranking list of documents. This list of documents is treated as the source of feedback context. Then, the learning goal is transformed to refine the ranking list using feedback for better performance. The ranking refinement seeks to consider more context information from feedback.

3.2 Autoencoder-based ranking refinement model

In this section, we introduce more details of our model for context-aware ranking refinement. We adopt autoencoders to refine the ranking list, because autoencoders have proved to be effective in generating high-quality document features in relevant tasks. In our method, the learned features using autoencoders aim to fully capture the context information of each individual query for better performance.

3.2.1 Autoencoders

Autoencoders are a classical kind of the building blocks of neural networks. An autoencoder inputs the original data, and represents the data in its hidden embedding layers, and outputs the reconstructed data. The hidden embedding of data has been usually used as learned features in different tasks. The usefulness of learned features can be measured by the reconstruction capability of autoencoders. Therefore, we can formalize the encoding and decoding process of our tailored autoencoders as follows.

$$\mathbf{y} = f(\mathbf{W}_1\mathbf{x} + \mathbf{b}_1) \tag{2}$$

$$\hat{\mathbf{x}} = f(\mathbf{W}_2\mathbf{y} + \mathbf{b}_2) \tag{3}$$

where $\mathbf{x}$ and $\hat{\mathbf{x}}$ are the input data and output data of an autoencoder, respectively. $\mathbf{y}$ is the hidden data embedding, namely, the learned features. $\mathbf{W}_1$, $\mathbf{W}_2$, $\mathbf{b}_1$ and $\mathbf{b}_2$ are parameters of the encoding and decoding neural networks. f denotes any nonlinear activation function. In our model, we use denoising autoencoders with tied weights, namely $\mathbf{W}_1 = \mathbf{W}_2$, to accelerate the training process and avoid overfitting. The reason for using denoising autoencoders lies in its good capability in reconstructing the ranking data, and yielding more robust ranking models.

3.2.2 Attentive loss for ranking feature weighting

Our model use a tailored loss function to learn an autoencoder -based ranking features. Since the loss function directly affects the feature effectiveness, how to design the loss function is a crucial issue in our work. For one thing, we hope that the loss function can measure the reconstruction capability of autoencoder. For another, we seek to tailor the original loss functions of autoencoder for ranking documents. Therefore, we propose to integrate the attention mechanism into the loss function of autoencoder. The tailored loss function can iteratively capture ranking semantics for learning enhanced ranking models. To consider the reconstruction capability of an autoencoder, we first introduce the original version of autoencoder loss function as follows.

$$\text{loss} = \sum_{i=1}^{n} ||x_i - \hat{x_i}||_2^2 \tag{4}$$

In Equation (4), $\mathbf{x}$ and $\hat{\mathbf{x}}$ are the input data and output data of an autoencoder, respectively, with n training samples. Distance measures, such as Euclidean distance, can be adopted to model the difference between $\mathbf{x}$ and $\hat{\mathbf{x}}$. The learning target of an original autoencoder is to reduce the loss for an optimal solution. However, in ranking scenario, the loss function

cannot work well, because this loss function ignore the difference of feature importance in the learning process. Thus, the accumulated loss may miss some important ranking features and meanwhile pay more attention on other unimportant ranking features, which may largely reduce the ranking performance. To avoid this problem, we would like to consider feature importance in reconstructing the ranking data using autoencoders.

Our previous work has measured the feature importance-based solely on an pre-trained ranker (Xu et al. 2017, 2019), which may suffer from the problem of unbalanced query performance in the final ranking list. To overcome this problem, in this work, we introduce the attention mechanism to precisely evaluate the feature importance by dynamically adjusting the feature importance in the autoencoder-enhanced re-ranking process. Specifically, we extend the Bregman divergence (Banerjee et al. 2004) using attentive feature weighting for better training the ranking refinement model. Furthermore, we use the following modified loss function for learning the ranking features.

$$loss = \sum_{i=1}^{n} \Theta^T (x_i - \hat{x_i})^2 \tag{5}$$

$$\Theta^T = softmax(\theta^T (x_i - \hat{x_i})^2) \tag{6}$$

where θ is a weighting vector obtained from the ListNet-based initial ranking model. We adopt the softmax function as the attention to generate a set of balanced feature weights Θ for loss computation. The learned weights using attention mechanism would pay more attention on the useful features in model optimization. In the training process, $\mathbf{W}_1$, $\mathbf{W}_2$, $\mathbf{b}_1$, $\mathbf{b}_2$ are continuously updated for learning useful ranking features. The learning process not only uses the ideal ranking list obtained from document labels for ListNet training, but also unsupervisedly reconstructs the data using autoencoders. Therefore, we can learn the final ranking model in a semi-supervised way for context-aware ranking refinement.

3.2.3 Context-aware hybrid listwise query constraints

In previous sections, we have modeled the ranking context using pseudo-relevance feedback, and measure the feature importance using an attention mechanism. To further improve the learned features, we consider the incorporation of query constraints into our model. Query constraints are rooted in the optimization of learning to rank. In search engine optimization, ranking can be optimized by improving the average performance on a large amount of user queries. Therefore, in the field of learning to rank, training data is divided based on different queries. Namely, each training query is related to a set of documents represented as document feature vectors. The learning target is to sort the documents with respect to the same query in a user friendly way and fulfill user's information needs to the largest extent. Although the pre-defined loss has considered the context and feature importance, query-level information is largely ignored, which motivates us to further modify the loss function of an autoencoder.

To fully consider the query-level information, we model it as different query constraints and integrate these query constraints as a hybrid one for better performance. We first model the query constraints as an item in our loss function. This item is designed to assign query-level weights on different queries. Namely, we assign more weights on the queries that are not well performed and less weights on the queries that have learned useful features so as to adjust the learned model for balanced learning. In this way, the learned models can generate useful features for all the queries.

Specifically, we use the initial ListNet ranker to model the query constraints. For a given query, based on the input data, ListNet produces one ranking list of documents, denoted as l^q_{in}. Based on the output data, we obtain another ranking list of documents, denoted as l^q_{out}. The original loss function of autoencoders directly measures the reconstruction capability using the input and output data. To measure the query-level difference, we can measure the distance between l^q_{in} and l^q_{out} for better considering the characteristics of the ranking scenario. Formally, we represent the distance between l^q_{in} and l^q_{out} as $\eta(l^q_{in}, l^q_{out})$, and incorporate $\eta(l^q_{in}, l^q_{out})$ into our loss function as follows.

$$loss = \sum_{q \in Q} \eta(l^q_{in}, l^q_{out}) \left(\sum_{i=1}^{n(q)} \theta^T (\hat{x_i} - x_i)^2 \right) \tag{7}$$

For any query q with $n(q)$ documents, we accumulate its loss value, and update the parameters $\mathbf{W}_1$, $\mathbf{W}_2$, $\mathbf{b}_1$, $\mathbf{b}_2$ in the loss function in iterations. The effectiveness of $\eta(l^q_{in}, l^q_{out})$ is proportional to the modeling of query-level information. Therefore, we use two ways to formalize this item by considering two listwise loss functions in learning to rank. In learning to rank, listwise approach can model the learning loss either based on computing the divergence of two ranking lists, or based on directly optimizing ranking metrics. Therefore, we would like to model the query constraints of ranking-oriented autoencoders in these two ways. To compute the difference of two ranking lists, we use cross-entropy of l^q_{in} and l^q_{out} in the following way.

$$\eta_{ce}(l^q_{in}, l^q_{out}) = \sum_{j=1}^{n(q)} P_{l^q_{in}}(j) log(P_{l^q_{out}}(j)) \tag{8}$$

The probability P can be achieved by ListNet-based permutation probability. This equation accumulates the cross-entropy of documents in l^q_{in} and l^q_{out}, which is treated as the

difference of these two lists. To directly optimize ranking metrics, we resort to the ranking performance. Namely, ranking performances of l^q_{in} and l^q_{out} are used to directly measure their distance, which can be achieved with any evaluation metrics $Eval$, such as $NDCG@k$. we formalize this idea in the following equation.

$$\eta_{oe}(l^q_{in}, l^q_{out}) = \frac{|Eval_{in} - Eval_{out}|}{Eval_{in}} \tag{9}$$

In Equation (9), $Eval_{in}$ denotes the ranking performance of l^q_{in}, and $Eval_{out}$ denotes the ranking performance of l^q_{out}. We obtain the performances by using the ListNet ranker. To comprehensively consider these two query constraints, we combine them as a hybrid one, which can fully consider the query constraints in constructing ranking refinement models.

$$\eta = \lambda\eta_{ce} + (1 - \lambda)\eta_{oe} \tag{10}$$

Based on Eq. (10), we model the hybrid listwise query constraints, and integrate the constraints in learning more useful query-level ranking features. We treat this loss function as our final loss function in feature learning, and used the learned features for learning more context-aware ranking semantics.

3.3 Network Details of our Framework

To help understand our framework for easily reproduction, we provide the details of network architecture of our framework in Table 1. The inputs of our framework contain a query set, each query corresponds to a document set, and each document is represented as a feature vector. The dimension of the original feature vectors is 45 for the MQ2007 and MQ2008 collections and 46 for the OHSUMED collection. The document feature vectors are inputted into a pseudo-relevance feedback (PRF) enhanced encoder. The PRF-enhanced encoder is optimized with attentive feature weighting and feedback ranking context. Then, we re-construct the ranking feature space with a query-level decoder. The dimension of the learned single-layer hidden representation is 60% or 70% of the original features, which is tuned in our experiments. $Tanh$ is used as the activation function for both the encoder and the decoder. The decoder is optimized with the hybrid listwise ranking loss including a performance change loss and a measure optimization loss based on Eq. (10). The learned context-aware features are then fed into the training of the final ranking model for better performance.

4 Experiments and analysis

4.1 Experimental settings

We evaluate the proposed model in this section. Our experiments are conducted on three LETOR collections, OHSUMED, MQ2007 and MQ2008 (Qin et al. 2010; Qin and Liu 2013). These LETOR collections are released by Microsoft.[1] The reported ranking performances are evaluated using retrieval metrics including Precision@k, NDCG@k and Mean Average Precision (MAP). We conduct fivefold cross-validations using the official divisions of these collections. The reported performances are averaged based on ten-time training for fair comparisons. NDCG@10 is used as $Eval$ in our experiments. In our experiments, we aim to answer three research questions:

RQ1 Can the proposed model improve ranking performance over state-of-the-art baselines?
RQ2 Does the proposed model achieve good performance using different ranking methods?
RQ3 How do selected parameters affect the final ranking performance of the proposed model?

4.2 Overall ranking performance of different models

In this section, we report the overall ranking performance of different models to answer **RQ1** in Tables 2, 3 and 4. In this group of experiments, we use ListNet not only to train the initial ranker, but also to train the ranking models. In Tables 2, 3 and 4, $original$ refers to the learning process with only the original document features. $denoising$ refers to the learned models using classical denoising autoencoders. $QSA\text{-}listOE$ (Xu et al. 2017) refers to the learned models using autoencoders integrated with query constraints by directly optimizing ranking metrics. $QSA\text{-}listCE$ (Xu et al. 2019) refers to the learned models using autoencoders integrated with query constraints by cross-entropy based performance change. $QSA\text{-}hybrid$ refers to the proposed model using the hybrid query constraints defined in Eq. (10). $+context$ refers to the context-aware models based on pseudo-relevance feedback. The results show that our context-aware models generally outperforms other models without context information. The proposed model with the hybrid query constraints achieves the optimal performance, which demonstrates that context information and hybrid query constraints can jointly contribute to improving the ranking performance.

Furthermore, we used deep learning based state-of-the-art ranking models: DSSM (Huang et al. 2013), DRMM (Guo et al. 2016) and Duet (Mitra et al. 2017) for comparison,

[1] http://research.microsoft.com/enus/um/people/letor/.

Table 1 The detailed architecture representing the hyper-parameters of our framework

MQ2007/MQ2008	Input dim.	Output dim.	Activation function	Num. of layers
Encoder	46	28/32	Tanh	1
Decoder	28/32	46	Tanh	1
OHSUMED	**Input dim.**	**Output dim.**	**Activation function**	**Num. of layers**
Encoder	45	27/31	Tanh	1
Decoder	27/31	45	Tanh	1

Table 2 Averaged ranking performance of different models on OHSUMED collection

Model	P@5	p@10	NDCG@5	NDCG@10	MAP	SD
Original	0.5502	0.4975	0.4432	0.4410	0.4457	0.0032
Denoising-AD	0.5283	0.4925	0.4355	0.4294	0.4381	0.0022
QSA-listOE	0.5774	0.5142	0.4779	0.4574	0.4537	0.0015
QSA-listCE	0.5752	0.5150	0.4755	0.4601	0.4542	0.0017
QSA-hybrid	0.5801	0.5168	0.4779	0.4612	0.4550	0.0021
QSA-listOE+context	0.5789†	0.5153	0.4770†	0.4591	0.4544	0.0014
QSA-listCE+context	0.5761	0.5146	0.4764	0.4613†	0.4550†	0.0013
QSA-hybrid+context	**0.5815**†	**0.5177**†	**0.4781**†	**0.4622**†	**0.4561**†	0.0012

Bold values in these tables indicate the best performance of each column
Significant improvement over QSA-listCE is marked with a dagger † based on two-tailed paired t test ($p \leq 0.05$). SD refers to the standard deviation of MAP

Table 3 Averaged ranking performance of different models on MQ2007 collection

Model	P@5	p@10	NDCG@5	NDCG@10	MAP	SD
Original	0.4126	0.3798	0.4170	0.4440	0.4652	0.0022
Denoising-AD	0.4135	0.3788	0.4209	0.4460	0.4683	0.0021
QSA-listOE	0.4194	0.3819	0.4260	0.4496	0.4720	0.0012
QSA-listCE	0.4203	0.3805	0.4276	0.4480	**0.4762**	0.0013
QSA-hybrid	0.4211	0.4025	0.4280	0.4501	0.4755	0.0012
QSA-listOE+context	0.4212†	0.4164†	0.4276	0.4503†	0.4737	0.0011
QSA-listCE+context	0.4257†	0.4138†	0.4253	0.4491†	0.4748	0.0010
QSA-hybrid+context	**0.4268**†	**0.4186**†	**0.4307**†	**0.4511**†	**0.4762**	0.0009

Bold values in these tables indicate the best performance of each column
Significant improvement over QSA-listCE is marked with a dagger † based on two-tailed paired t test ($p \leq 0.05$). SD refers to the standard deviation of MAP

Table 4 Averaged ranking performance of different models on MQ2008 collection

Model	P@5	p@10	NDCG@5	NDCG@10	MAP	SD
Original	0.3426	0.2476	0.4747	0.2303	0.4775	0.0030
Denoising-AD	0.3436	0.2473	0.4751	0.2291	0.4850	0.0027
QSA-listOE	0.3515	0.2487	0.4839	0.2333	0.4929	0.0018
QSA-listCE	0.3535	0.2490	0.4845	0.2413	**0.4987**	0.0017
QSA-hybrid	0.3528	0.2490	0.4851	0.2405	0.4974	0.0014
QSA-listOE+context	0.3520	0.2492	0.4860†	0.2357	0.4935	0.0013
QSA-listCE+context	**0.3564**†	0.2501†	0.4857†	0.2408	0.4982	0.0012
QSA-hybrid+context	0.3557†	**0.2516**†	**0.4875**†	**0.2425**†	0.4979	0.0012

Bold values in these tables indicate the best performance of each column
Significant improvement over QSA-listCE is marked with a dagger † based on two-tailed paired t test ($p \leq 0.05$). SD refers to the standard deviation of MAP

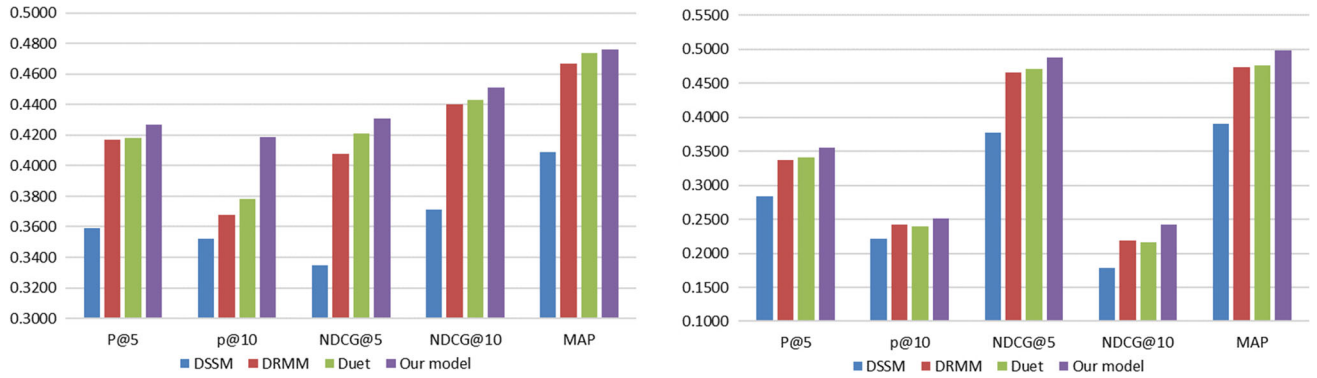

Fig. 2 Comparisons with state-of-the-art models on MQ2007 (left) and MQ2008 (right) collections

shown in Fig. 2. This group of experiments is conducted on MQ2007 and MQ2008, because these two collections provide the original textual documents for training the neural ranking models. The results show that DSSM is relatively lower than other models, and the performances of DRMM and Duet are better than DSSM. Furthermore, the proposed model achieved the best performance among all other models on both collections. The results demonstrate that our model can effectively learn more useful context-aware document features for promoting the overall ranking effectiveness.

4.3 Ranking effectiveness of ranking approaches

We further use four ranking approaches to examine the generalization ability of our feature generation framework. This group of experiments are used to answer **RQ2**. The compared learning to rank approaches include the pointwise Random Forests model, the pairwise RankBoost model (Freund et al. 1998), the listwise ListNet model (Cao et al. 2007) and the listwise LambdaMART model (Burges 2010). We use all these approaches to train different ranking models using the context-free and context-aware autoencoders on OHSUMED collection. The experimental results are shown in Fig. 3, in which we observe that our context-aware framework yields relatively good performance among all methods, which further demonstrates the effectiveness and robustness of context information in ranking refinement.

4.4 Parameter selection

To answer **RQ3**, we examine the impact of parameters. The parameters include the feature dimensionality and the interpolation parameter for the hybrid query constraints. The feature dimensionality is tuned based on the scale of the features in LETOR. We switch the dimensionality from 0 to 90% to observe the performance change. The interpolation parameter is the ratio of two listwise query constraints, which is tuned from 0.1 to 0.9 in our experiments. The experimental results are shown in Fig. 4. Ranking performance in this group of experiments is evaluated based on mean average precision. From the figure, we observe that 60–70% of the generated features contribute the most to the ranking performance. When λ is set to be 0.7 or 0.8, optimal performance was achieved.

4.5 Further discussion

Based on the experimental results, we observe that our method not only outperforms other competing baseline models and our previous methods, but also yields consistent performance improvement among different ranking methods. Therefore, we summarize the reasons of performance improvement in this section and provide more insights on the proposed method for future optimizations.

In general, compared with the previous works, the proposed method has three advantages: the PRF-based context modeling, attentive feature weighting and the hybrid query constraints. The PRF-based context modeling captures more local ranking contexts from the top-ranked documents in an initial ranking, which conveys useful information in autoencoder-based re-ranking. Attentive feature weighting accurately measures the feature importance in reconstructing the inputs of autoencoders. Instead of solely using a pre-trained ranker, the attention mechanism dynamically adjusts the feature importance in the autoencoder-enhanced re-ranking process, which assigns more weights on the high-quality ranking features for building more effective ranking models. The hybrid query constraints effectively encodes more comprehensive query-level ranking information, which not only considers the cross-entropy ranking loss of directly optimizing evaluation measures, but also makes the most use of the change of ranking performance. Therefore, these advantages jointly contribute to the overall ranking performance and produce consistent improvements.

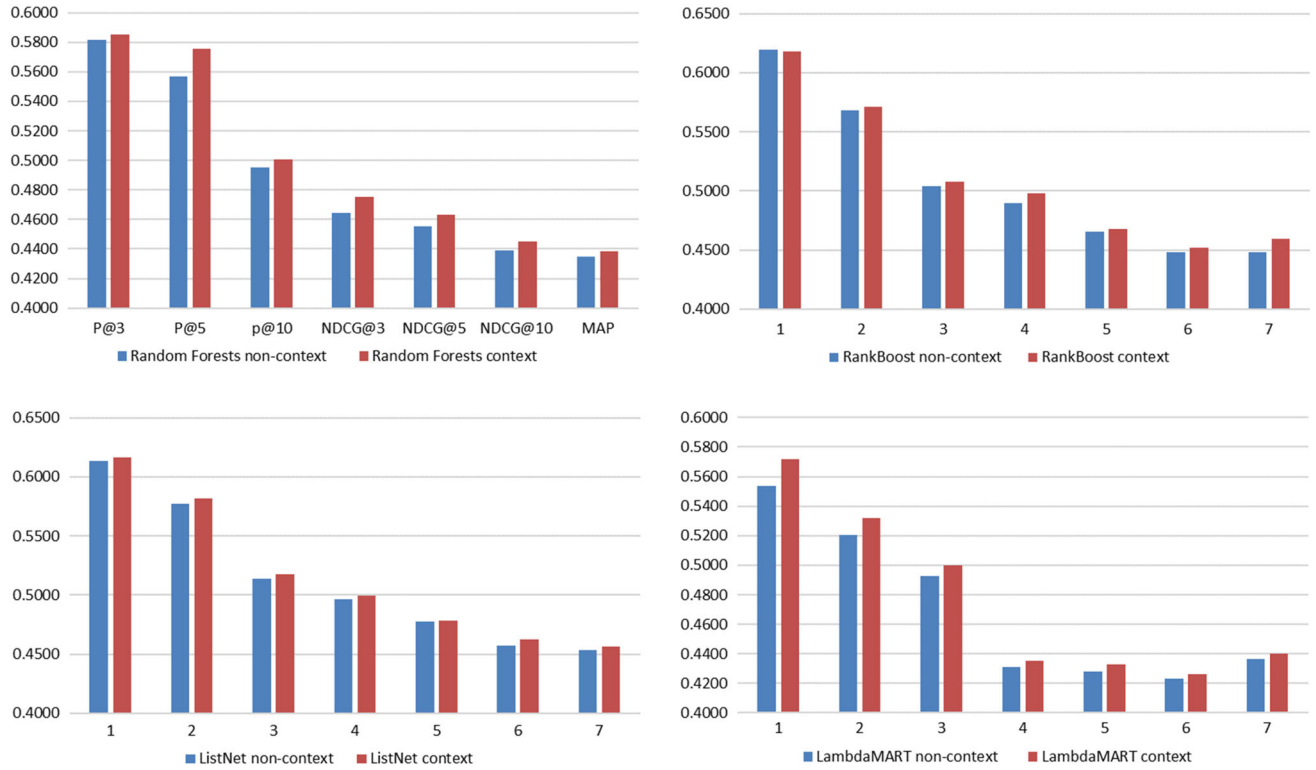

Fig. 3 Performance of the context-aware models using different learning to rank methods

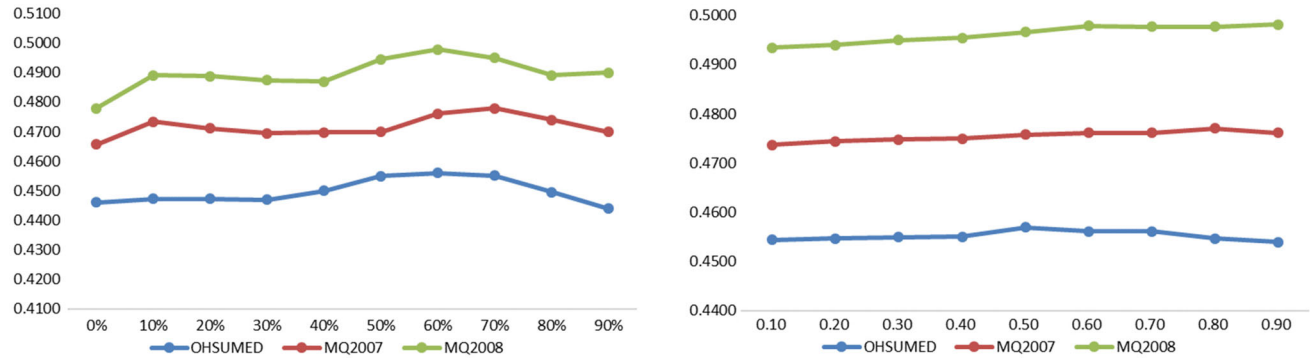

Fig. 4 Parameter selection on three collections. Left: the ratio of generated features. Right: the interpolation parameter λ

5 Conclusions

We propose a context-aware ranking refinement method using query-level autoencoders in this work. Our method captures the local context information of individual queries based on pseudo-relevance feedback for ranking refinement. Context information of queries are advantageous in capturing the characteristics of different queries. To refine the context-aware ranking lists, we propose an attentive semi-supervised autoencoder based on hybrid query constraints. The attention mechanism is used to precisely measure the feature importance in the learning process. Two types of listwise query constraints are combined as a hybrid one to capture the characteristics of relevant documents for different queries. We validate the effectiveness of the proposed model on three public learning to rank benchmark collections. The experimental results demonstrate the effectiveness of the proposed model in improving ranking performance. Our future work will further investigate the proposed model in other domain-specific IR tasks.

Author Contributions Bo Xu was involved in the conceptualization, methodology, writing—original draft preparation. Hongfei Lin contributed to the formal analysis and funding acquisition. Yuan Lin helped in writing—reviewing and editing and experiments. Kan Xu contributed to writing—reviewing and editing and validation.

Funding This work is partially supported by grant from the Natural Science Foundation of China (No. 62006034), Natural Science Foundation of Liaoning Province (No. 2021-BS-067) and the Fundamental Research Funds for the Central Universities (No. DUT21RC(3)015).

Data availability Enquiries about data availability should be directed to the authors.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Ethical approval This manuscript does not contain any studies with human participants or animals performed by any of the authors. We have read and have abided by the statement of ethical standards for manuscripts.

Informed consent The submitted manuscript has obtained informed consent from all authors.

References

Ahmad WU, Chang KW, Wang H (2019) Context Attentive Document Ranking and Query Suggestion. In: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR, pp 385–394. https://doi.org/10.1145/3331184.3331246

Ai Q, Bi K, Guo J, Croft WB (2018) Learning a deep listwise context model for ranking refinement. In: Proceedings of the 41st international ACM SIGIR conference on research and development in information retrieval. ACM, pp 135–144

Al-Asadi MA, Tasdemir S (2021) Empirical comparisons for combining balancing and feature selection strategies for characterizing football players using FIFA video game system. IEEE Access 9:149266–149286

Al-Asadi MA, Tasdemir S (2022) Predict the value of football players using FIFA video game data and machine learning techniques. IEEE Access 10:22631–22645

Banerjee A, Merugu S, Dhillon IS, Ghosh J (2004) Clustering with Bregman divergences. J Mach Learn Res 6(4):1705–1749

Bengio Y (2009) Learning deep architectures for AI. Found Trends Mach Learn 2(1):1–127

Burges Christopher J, Ragno Robert, Le Quoc V (2007) Learning to rank with nonsmooth cost functions. In: Advances in neural information processing systems (NIPS), pp 193–200

Burges CJC (2010) From RankNet to LambdaRank to LambdaMART: an overview. Learning 11(23–581):81

Burges C, Shaked T, Renshaw E, Lazier A, Deeds M, Hamilton N, Hullender G (2005) Learning to rank using gradient descent. In: International conference on machine learning (ICML), pp 89–96

Busolin F, Lucchese C, Nardini FM, Orlando S, Perego R, Trani S (2021) Learning early exit strategies for additive ranking ensembles. In: Diaz F, Shah C, Suel T, Castells P, Jones R, Sakai T (eds) SIGIR '21: the 44th international ACM SIGIR conference on research and development in information retrieval, Virtual Event, Canada, July 11–15, 2021. ACM, pp 2217–2221

Cao Z, Qin T, Liu TY, Tsai MF, Li H (2007) Learning to rank: from pairwise approach to listwise approach. In: International conference on machine learning (ICML), pp 129–136

Chen M, Liu C, Sun J, Hoi SC (2021) Adapting interactional observation embedding for counterfactual learning to rank. In: Diaz F, Shah C, Suel T, Castells P, Jones R, Sakai T (eds) SIGIR '21: the 44th international ACM SIGIR conference on research and development in information retrieval, Virtual Event, Canada, July 11–15, 2021. ACM, pp 285–294

Chen L, Wu L, Zhang K, Hong R, Wang M (2021) Set2setrank: collaborative set to set ranking for implicit feedback based recommendation. In: Diaz F, Shah C, Suel T, Castells P, Jones R, Sakai T (eds) SIGIR '21: the 44th international ACM SIGIR conference on research and development in information retrieval, Virtual Event, Canada, July 11–15, 2021. ACM, pp 585–594

Choi J, Jung E, Suh J, Rhee W (2021) Improving bi-encoder document ranking models with two rankers and multi-teacher distillation. In: Diaz F, Shah C, Suel T, Castells P, Jones R, Sakai T (eds) SIGIR '21: the 44th international ACM SIGIR conference on research and development in information retrieval, Virtual Event, Canada, July 11–15, 2021. ACM, pp 2192–2196

Fan Y, Guo J, Lan Y, Xu J, Zhai C, Cheng X (2018) Modeling diverse relevance patterns in ad-hoc retrieval. In: Proceedings of the 41st international ACM SIGIR conference on research and development in information retrieval, pp 375–384

Feng Y, Xu J, Lan Y, Guo J, Zeng W, Cheng X (2018) From greedy selection to exploratory decision-making: diverse ranking with policy-value networks. In: Proceedings of the 41st international ACM SIGIR conference on research and development in information retrieval, pp 125–134

Formal T, Piwowarski B, Clinchant S (2021) SPLADE: sparse lexical and expansion model for first stage ranking. In: Diaz F, Shah C, Suel T, Castells P, Jones R, Sakai T (eds) SIGIR '21: the 44th international ACM SIGIR conference on research and development in information retrieval, Virtual Event, Canada, July 11–15, 2021. ACM, pp 2288–2292

Freund Y, Iyer R, Schapire RE, Singer Y (1998) An efficient boosting algorithm for combining preferences. In: International conference on machine learning (ICML), pp 170–178

Friedman JH (2001) Greedy function approximation: a gradient boosting machine. Ann Statist 29(5): 1189–1232

Goeric H (2016) Learning to rank with deep neural networks. Master's thesis, University of Leuven

Guo J, Fan Y, Ai Q, Croft WB(2016) A deep relevance matching model for ad-hoc retrieval. In: Proceedings of the 25th ACM international on conference on information and knowledge management (CIKM). ACM, pp 55–64

Hansen C, Hansen C, Alstrup S, Grue Simonsen J, Lioma C (2019) Neural check-worthiness ranking with weak supervision: finding sentences for fact-checking. CoRR arXiv:1903.08404

He X, He Z, Du X, Chua TS (2018) Adversarial personalized ranking for recommendation. In: Proceedings of the 41st international ACM SIGIR conference on research and development in information retrieval, pp 355–364

Huang PS, He X, Gao J, Deng L, Acero A, Heck L (2013) Learning deep structured semantic models for web search using clickthrough data. In: Proceedings of the 22nd ACM international on conference on information and knowledge management (CIKM). ACM, pp 2333–2338

Joachims T, Swaminathan A, Schnabel T (2017) Unbiased learning-to-rank with biased feedback. In: Proceedings of the 10th ACM international conference on web search and data mining (WSDM). ACM, pp 781–789

Kim M, Ko Y (2021) Self-supervised fine-tuning for efficient passage re-ranking. In: Demartini G, Zuccon G, Shane Culpepper J, Huang Z, Tong H (eds) CIKM '21: the 30th ACM international conference on information and knowledge management, virtual event, Queensland, Australia, November 1–5, 2021. ACM, pp 3142–3146

Kim Y, Rahimi R, Bonab H, Allan J (2021) Query-driven segment selection for ranking long documents. In: Demartini G, Zuccon G, Shane Culpepper J, Huang Z, Tong H (eds) CIKM '21: the

30th ACM international conference on information and knowledge management, virtual event, Queensland, Australia, November 1–5, 2021. ACM, pp 3147–3151

Lavrenko V, Croft WB (2001) Relevance-based language models. In: International ACM SIGIR conference on research and development in information retrieval, vol 51, no 2, pp 120–127

Lee Y, Kim KE (2021) Dual correction strategy for ranking distillation in top-n recommender system. In: Demartini G, Zuccon G, Shane Culpepper J, Huang Z, Tong H (eds) CIKM '21: the 30th ACM international conference on information and knowledge management, virtual event, Queensland, Australia, November 1–5, 2021. ACM, pp 3186–3190

Li J, Luong T, Jurafsky D (2015) A hierarchical neural autoencoder for paragraphs and documents. In: International joint conference on natural language processing, vol 1, pp 1106–1115

Liu T-Y (2009) Learning to rank for information retrieval. Found Trends Inf Retr 3(3):225–331

Lucchese C, Nardini FM, Perego R, Orlando S, Trani S (2018) Selective gradient boosting for effective learning to rank. In: Proceedings of the 41st international ACM SIGIR conference on research and development in information retrieval, pp 155–164

MacAvaney S, Yates A, Hui K, Frieder O (2019) Content-based weak supervision for ad-hoc re-ranking. arXiv Information retrieval

Mehrotra R, Yilmaz E (2015) Representative and informative query selection for learning to rank using submodular functions. In: Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval. ACM, pp 545–554

Mitra B, Diaz F, Craswell N (2017) Learning to match using local and distributed representations of text for web search. In: Proceedings of the 26th international conference on World Wide Web (WWW), pp 1291–1299

Niu S, Lan Y, Guo J, Cheng X, Geng X (2014) What makes data robust: a data analysis in learning to rank. In: Proceedings of the 37th international ACM SIGIR conference on research and development in information retrieval. ACM, pp 1191–1194

Qin T, Liu T-Y, Jun X, Li H (2010) LETOR: a benchmark collection for research on learning to rank for information retrieval. Inf Retr J 13(4):346–374

Qin T, Liu TY (2013) Introducing LETOR 4.0 datasets. Computer Science

Robertson Stephen E, Sparck Jones K (1976) Relevance weighting of search terms. J Am Soc Inf Sci 27(3):129–146

Rosset C, Mitra B, Xiong C, Craswell N, Song X, Tiwary S (2019) An axiomatic approach to regularizing neural ranking models. In: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp 981–984. https://doi.org/10.1145/3331184.3331296

Salton G, Buckley C (1997) Improving retrieval performance by relevance feedback. J Am Soc Inf Sci 41(4):355–364

Schuth A, Oosterhuis H, Whiteson S, de Rijke M (2016) Multileave gradient descent for fast online learning to rank. In: Proceedings of the 9th ACM international conference on Web Search and Data Mining (WSDM). ACM, pp 457–466

Sedhain S, Menon AK, Sanner S, Xie L (2015) Autorec: autoencoders meet collaborative filtering. In: Proceedings of the 24th international conference on World Wide Web (WWW). ACM, pp 111–112

Shao J, Ji S, Yang T (2019) Privacy-aware document ranking with neural signals. In: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR, pp 305–314. https://doi.org/10.1145/3331184.3331189

Tax N, Bockting S, Hiemstra D (2015) A cross-benchmark comparison of 87 learning to rank methods. Inf Process Manage 51(6):757–772

Tran A, Yang T, Ai Q (2021) ULTRA: an unbiased learning to rank algorithm toolbox. In: Demartini G, Zuccon G, Shane Culpepper J, Huang Z, Tong H (eds) CIKM '21: the 30th ACM international conference on information and knowledge management, virtual event, Queensland, Australia, November 1–5, 2021. ACM, pp 4613–4622

Wang B, Klabjan D (2017) An attention-based deep net for learning to rank. arXiv preprint arXiv:1702.06106

Wang H, Langley R, Kim S, McCord-Snook E, Wang H (2018) Efficient exploration of gradient space for online learning to rank. In: Proceedings of the 41st international ACM SIGIR conference on research and development in information retrieval, pp 145–154

Wang H, Shi X, Yeung DY (2015) Relational stacked denoising autoencoder for tag recommendation. In: The association for the advancement of artificial intelligence (AAAI), pp 3052–3058

Wang J, Yu L, Zhang W, Gong Y, Xu Y, Wang B, Zhang P, Zhang D (2017) IRGAN: a minimax game for unifying generative and discriminative information retrieval models. In: Proceedings of the 40th international ACM SIGIR conference on research and development in information retrieval. ACM, pp 515–524

Wu L, Hu D, Hong L, Liu H (2018) Turning clicks into purchases: revenue optimization for product search in E-commerce. In: Proceedings of the 41st international ACM SIGIR conference on research and development in information retrieval, pp 365–374

Xu B, Lin H, Lin Y, Xu K (2017) Learning to rank with query-level semi-supervised autoencoders. In: Proceedings of the 26th ACM on conference on information and knowledge management (CIKM). ACM, pp 2395–2398

Xu B, Lin H, Lin Y, Xu K (2019) Incorporating query constraints for autoencoder enhanced ranking. Neurocomputing 356:142–150

Yang Z, Yan S, Lad A, Liu X, Guo W (2021) Cascaded deep neural ranking models in linkedin people search. In: Demartini G, Zuccon G, Shane Culpepper J, Huang Z, Tong H (eds) CIKM '21: the 30th ACM international conference on information and knowledge management, virtual event, Queensland, Australia, November 1–5, 2021. ACM, pp 4312–4320

Yin D, Hu Y, Tang J, Daly T, Zhou M, Ouyang H, Chen J, Kang C, Deng H, Nobata C, Langlois JM (2016) Ranking relevance in yahoo search. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. ACM, pp 323–332

Yoon S, Shin J, Jung K (2018) Learning to rank question-answer pairs using hierarchical recurrent encoder with latent topic clustering. In: Proceedings of the 2018 conference of the North American chapter of the association for computational linguistics: human language technologies (NAACL-HLT), pp 1575–1584

Zhai C, Lafferty JD (2004) A study of smoothing methods for language models applied to information retrieval. ACM Trans Inf Syst 22(2):179–214

Zhai S, Zhang ZM (2016) Semisupervised autoencoder for sentiment analysis. In: The association for the advancement of artificial intelligence (AAAI), pp 1394–1400

Zhuang F, Luo D, Yuan NJ, Xie X, He Q (2017) Representation learning with pair-wise constraints for collaborative ranking. In: Proceedings of the 10th ACM international conference on web search and data mining (WSDM). ACM, pp 567–575

Zhu Y, Nie JY, Dou Z, Ma Z, Zhang X, Du P, Zuo X, Jiang H (2015) Representation learning via semi-supervised autoencoder for multi-task learning. In: IEEE international conference on data mining (ICDM). IEEE, pp 1141–1146

Zhu Y, Nie JY, Dou Z, Ma Z, Zhang X, Du P, Zuo X, Jiang H (2021) Contrastive learning of user behavior sequence for context-aware document ranking. In: Demartini G, Zuccon G, Shane Culpepper J, Huang Z, Tong H (eds) CIKM '21: the 30th ACM international conference on information and knowledge management, virtual event, Queensland, Australia, November 1–5, 2021. ACM, pp 2780–2791