Contrastive Learning of User Behavior Sequence for Context-Aware Document Ranking

Yutao Zhu[1], Jian-Yun Nie[1], Zhicheng Dou[2], Zhengyi Ma[2], Xinyu Zhang[3]
Pan Du[1], Xiaochen Zuo[2], and Hao Jiang[3]
[1] Université de Montréal, Montréal, Québec, Canada
[2] Gaoling School of Artificial Intelligence, Renmin University of China, Beijing, China
[3] Distributed and Parallel Software Lab, Huawei, Hangzhou, Zhejiang, China
yutao.zhu@umontreal.ca,nie@iro.umontreal.ca,{dou,zymaa,zuoxc}@ruc.edu.cn
du@youark.com,{zhangxinyu35,jianghao66}@huawei.com

ABSTRACT

Context information in search sessions has proven to be useful for capturing user search intent. Existing studies explored user behavior sequences in sessions in different ways to enhance query suggestion or document ranking. However, a user behavior sequence has often been viewed as a definite and exact signal reflecting a user's behavior. In reality, it is highly variable: user's queries for the same intent can vary, and different documents can be clicked. To learn a more robust representation of the user behavior sequence, we propose a method based on contrastive learning, which takes into account the possible variations in user's behavior sequences. Specifically, we propose three data augmentation strategies to generate similar variants of user behavior sequences and contrast them with other sequences. In so doing, the model is forced to be more robust regarding the possible variations. The optimized sequence representation is incorporated into document ranking. Experiments on two real query log datasets show that our proposed model outperforms the state-of-the-art methods significantly, which demonstrates the effectiveness of our method for context-aware document ranking.

CCS CONCEPTS

• **Information systems → Retrieval models and ranking**.

KEYWORDS

Context-aware Document Ranking, Contrastive Learning, User Behavior Sequence, Data Augmentation

ACM Reference Format:
Yutao Zhu, Jian-Yun Nie, Zhicheng Dou, Zhengyi Ma, Xinyu Zhang, Pan Du, Xiaochen Zuo, and Hao Jiang. 2021. Contrastive Learning of User Behavior Sequence for Context-Aware Document Ranking . In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management (CIKM '21), November 1–5, 2021, Virtual Event, QLD, Australia.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3459637.3482243

1 INTRODUCTION

Search engines have evolved from one-shot searches to consecutive search interactions with users [1]. To fulfil complex information needs, users will issue a sequence of queries, examine and interact with some of the results. User historical behavior or interaction history in a session is known to be very useful for understanding the user's information needs and to rank documents [4, 20, 32, 63].

Various studies exploited user behavior data for different purposes. For example, by analyzing search logs, researchers found that a user's search history provides useful information for understanding user intent during the search sessions [4]. To utilize the historical user behavior in document ranking, some early work explored query expansion and learning to rank techniques [4, 7, 25, 50]. More recently, various neural structures have been used to model the user behavior sequence. For example, a recurrent neural network (RNN) is proposed to model the historical queries and suggest the next query [51]. This structure has been extended to model both the historical queries and clicked documents, leading to further improvement on document ranking [2, 3]. Pre-trained language models have also been exploited to encode contextual information from user behavior sequences, and they achieved promising results [48].

All these studies tried to learn a prediction or representation model to capture the information hidden in the sequences. However, user behavior sequences have been viewed as definite and exact sequences. That is, an observed sequence is used as a positive sample and any unseen sequence is not used or is viewed as a negative sample. This strict view does not reflect the flexible nature of user's behavior in a session. Indeed, when interacting with a search engine, users do not have a definitive interaction pattern, nor a fixed query for an information need. All these are flexible and change greatly from a user to another, and from a search context to another. Similarly, user's click behaviors are also not definitive: one can click on different documents for the same information need, and can also click on irrelevant documents. The high variation is inherent in the user's interactions with a search engine. This characteristic has not been explicitly addressed in previous studies. One typically relied on a large amount of log data, hoping that strong patterns can emerge, while accidental variations (or noise) can be discarded. This is true to some extent when we have a large amount of log data and when we are only interested in the common patterns shared by users. However, the models strictly relying on the log data cannot fully capture the nuances in user behaviors and

CIKM '21, November 1–5, 2021, Virtual Event, QLD, Australia

ACM ISBN 978-1-4503-8446-9/21/11…$15.00
https://doi.org/10.1145/3459637.3482243

cope with the variations. A better approach is to view the data as they are, *i.e.*, they are just samples of possible query formulations and interactions, but much more are not shown in the logs.

To tackle this problem, in this work, we propose a **data augmentation** approach to generate possible variations from a search log. More specifically, we use three strategies to mask some terms in a query or document, delete some queries or documents, or reorder the sequence. These strategies reflect some typical variations in user's behavior sequences. The generated behavior sequences can be considered similar to the observed ones. We have, therefore, automatically tagged user behavior sequences in terms of similarity, which are precious for model training. In addition, we can generate more training data from search logs, which has always been a critical issue for research in this area. Based on the augmented data, we utilize **contrastive learning** to extract what is similar and dissimilar. More specifically, the contrastive model tries to pull the similar sequences (generated variants) closer and to distinguish them from semantically unrelated ones. Compared to the existing approaches based on search logs, we expect that contrastive learning can better cope with the inherent variations and generate more robust models to deal with new behavior sequences.

Contrastive learning is implemented with a pre-trained language model BERT [14] through encoding a sequence and its variants into a contextualized representation with a contrastive loss. The document ranking is then learned by a linear projection on top of the optimized sequence representation. With both the original sequences and corresponding variants modeled in the representation, the final ranking function can not only address the context information thoroughly, but also learn to cope with the inherent variations, hence generating better ranking results during prediction.

We conduct experiments on two large-scale real-world search log datasets (AOL and Tiangong-ST). Experimental results show that our proposed method outperforms the existing methods (including those exploiting search logs) significantly, which demonstrates the effectiveness of our approach.

Our contributions are three-fold:

(1) We design three different data augmentation strategies to construct similar sequences of observed user behavior sequences, which modify the original sequence at term, query/document, and behavior levels.

(2) We propose a self-supervised task with a contrastive learning objective based on the augmented behavior sequences to capture what is hidden behind the sequences and their variants, and to distinguish them from other unrelated sequences.

(3) Experiments on two large-scale real-world search log datasets confirm the effectiveness of our method. This study shows that contrastive learning with automatically augmented search logs is an effective way to alleviate the shortage of log data in IR research.

2 RELATED WORK

2.1 Exploiting Historical Log Data

Context information in sessions has shown to be useful in modeling user intent in search tasks [4, 20, 32, 64]. Early studies focused on extracting contextual features from users' search activities so as to characterize their search intent. For example, some keywords were extracted from users' historical queries and clicked documents and used to rerank the documents for the current query [50]. Statistical features and rule-based features were also introduced to quantify or characterize context information [57, 60]. However, these methods often rely on manually extracted features or handcrafted rules, which limits their application in different retrieval tasks.

Later, researchers started to build predictive models for users' search intent or future behavior. For example, a hidden Markov model was employed to model the evolution of users' search intent. Then, both document ranking and query suggestion were conducted based on the predicted user intent [6]. Reinforcement learning has also been applied to model user interactions in search tasks [22, 40]. Unfortunately, the predefined model space or state transition structure limits the learning of rich user-system interactions.

The development of neural networks generated various solutions for context-aware document ranking. Some researchers proposed a hierarchical neural structure with RNNs to model historical queries and suggest the next query [51]. This model is further extended with the attention mechanism to better represent sessions and capture user-level search behavior [11]. Recently, researchers found that jointly learning query suggestion and document ranking can boost the model's performance on both tasks [2]. In addition to leveraging historical queries, the historical clicked documents are also reported to be helpful in both query suggestion and document ranking [3].

More recently, large-scale pretrained language models, such as BERT [14], have achieved great performance on many NLP and IR tasks [33, 34, 38, 41]. Qu et al. [48] proposed to concatenate all historical queries, clicked documents, and unclicked documents as a long sequence and leveraged BERT as an encoder to compute their term-level representations. These representations were further combined with relative position embeddings and human behavior embeddings through another transformer-based structure to get the final representations. The ranking score is computed based on the representation of the special "[CLS]" token.

Our framework is also based on BERT, but we use contrastive learning to pretrain the model in a self-supervised manner. Theoretically, this strategy better leverages the available training data, which can also be applied to existing methods.

2.2 Contrastive Learning for IR

Contrastive learning aims to learn effective representation of data by pulling semantically close neighbors together and pushing apart other non-neighbors [26, 55]. It has been widely applied in computer vision [10, 53, 67] and NLP tasks [16, 19, 23, 59] and has proven its high efficiency in leveraging the training data without the need of annotation. What is required in contrastive learning is to identify semantically close neighbors. In visual representation, neighbors are commonly generated by two random transformations of the same image (such as flipping, cropping, rotation, and distortion) [10, 15]. Similarly, in text representation, data augmentation techniques such as word deletion, reordering, and substitution are applied to derive similar texts from a given text sequence [43, 59]. Although the principle of contrastive learning is well accepted, the ways to implement it are still under exploration, with the general guiding principles of *alignment* and *uniformity* [55].

As for pre-training, Chang et al. [8] designed several paragraph-level pre-training tasks and the Transformer models can improve over the widely-used BM25 [49]. Ma et al. [41] constructed a representative word prediction (ROP) task for pre-training BERT. Experimental results showed that the BERT model pre-trained with ROP and masked language model (MLM) tasks achieves great performance on ad-hoc retrieval. Our proposed sequence representation optimization stage can be treated as a pre-training stage because it is trained before document ranking (our main task). However, as we do not use external datasets, we do not categorize our method as a pre-training approach.

In this paper, we propose a contrastive learning objective to optimize the sequence representation for improving the downstream document ranking task. This first attempt opens the door to more future studies on applying contrastive learning to IR.

3 METHODOLOGY

Context-aware document ranking aims at using the historical user behavior sequence and the current query to rank a set of candidate documents. In this work, we design a new framework for this task. Our framework aims at optimizing the representation of the user behavior sequence before learning document ranking. As shown in Figure 1, our framework can be divided into two stages: (1) *sequence representation optimization* and (2) *document ranking*. In the first stage, we design a self-supervised task with contrastive learning objective to optimize the sequence representation. In the second stage, our model uses the optimized sequence representation and further learns the ranking model. We call our framework COCA – **CO**ntrastive learning for **C**ontext-**A**ware document ranking.

3.1 Notations

Before introducing the task and the model, we first provide the definitions of important concepts and notations. We present a user's search history as a sequence of M queries $Q = \{q_1, \cdots, q_M\}$, where each query q_i is associated with a submission timestamp t_i and the corresponding list of returned documents $D_i = \{d_{i,1}, \cdots, d_{i,M}\}$. Each query q_i is represented as the original text string that users submitted to the search engine. Q is ordered according to query timestamp t_i. Each document $d_{i,j}$ has two attributes: its text content and click label $y_{i,j}$ ($y_{i,j} = 1$ if it is clicked). In general, user clicks serve as a good proxy of relevance feedback [3, 30, 31, 48]. Given all available historical queries and clicked documents up to n turns, we denote the user behavior sequence as $H_n = \{q_1, d_1, \cdots, q_n, d_n\}$[1]. As reported in [3, 48], the unclicked documents are less helpful and may even introduce noise, so they are not considered in the user behavior sequence.

3.2 Overview

With the above concepts and notations, we briefly introduce the two stages in COCA as follows.

(1) **Sequence Representation Optimization.** As shown in the left side of Figure 1, our target is to obtain a better representation of the user behavior sequence H_n in this stage. To achieve this, we first construct two augmented sequences H'_n and H''_n from H_n with randomly selected augmentation strategies (Section 3.3.1). Such a pair of sequences are considered to be similar. Then a BERT encoder is applied to get the representations of these two sequences (Section 3.3.2). With the contrastive loss, the model learns to pull them close and push them away from other sequences in the same minibatch (Section 3.3.3). By comparing the two augmented sequences, the BERT encoder is forced to learn a more generalized and robust representation for sequences.

(2) **Document Ranking.** As shown in the right side of Figure 1, we aim to rank the relevant documents as high as possible in this stage. Given the current query q_i and the historical behavior sequence H_{i-1}, we treat $H_{i-1} \cup \{q_i\}$ as a sequence and the candidate document $d_{i,j}$ as another sequence. Then, we concatenate them together and use the BERT encoder trained in the first stage to generate a representation. The final ranking score is obtained by a linear projection on the representation. A cross-entropy loss is applied between the predicted ranking score and the click label $y_{i,j}$.

3.3 Sequence Representation Optimization

The user behavior sequence contains abundant information about the user intent. To optimize the representation of the user behavior sequence, we propose a self-supervised approach. Specifically, we apply a contrastive learning objective to pull close the representation of similar sequences and push apart different ones. The similar sequences are created by the three augmentation strategies described below.

3.3.1 *Augmentation Strategy.*

Inspired by the existing data augmentation strategies in NLP and image processing, we propose three strategies to construct similar sequences, namely term mask, query/document deletion, and behavior reordering (shown in Figure 2). These strategies correspond to three levels of variation in user behaviors, *i.e.*, term level, query/document level, and user behavior level.

(a) **Term Mask.** In natural language processing, the "word mask" or "word dropout" technique has been widely applied to avoid overfitting. It has been shown to improve the robustness of the sentence representation, *e.g.*, in sentence generation [5], sentiment analysis [12], and question answering [17]. Inspired by this, we propose to apply a random term mask operation over the user behavior sequence (including query terms and document terms) as one of the augmentation strategies for contrastive learning.

With the term-level augmentation strategy, we can obtain various user behavior sequences similar to the original one. The similar sequences only have minor differences in some terms. This aims to simulate the real search situations where users may issue slightly different queries for searching the same target, and a document may satisfy similar information needs. By contrasting similar sequences with others, the models can learn the importance of different terms in both queries and documents. Besides, it can also help the model to learn more generalized sequence representation by avoiding relying too much on specific terms.

Specifically, for a user behavior sequence $H_n = \{q_1, d_1, \cdots, q_n, d_n\}$, we first represent it as a term sequence $H_n = \{w_1, \cdots, w_{N_T}\}$, where N_T is the total number of terms. Then, we randomly mask a proportion γ of terms $T_n = \{t_1, \cdots, t_L\}$, where $L_{\text{tm}} = \lfloor N \cdot \gamma \rfloor$, and t_i is the index of term to be masked. If a term is masked, it is replaced by a special token "[T_MASK]", which is similar to the token "[MASK]" used in BERT [14]. Therefore, we formulate this augmentation

[1] Following previous studies [48], we only use one clicked document to construct the sequence.

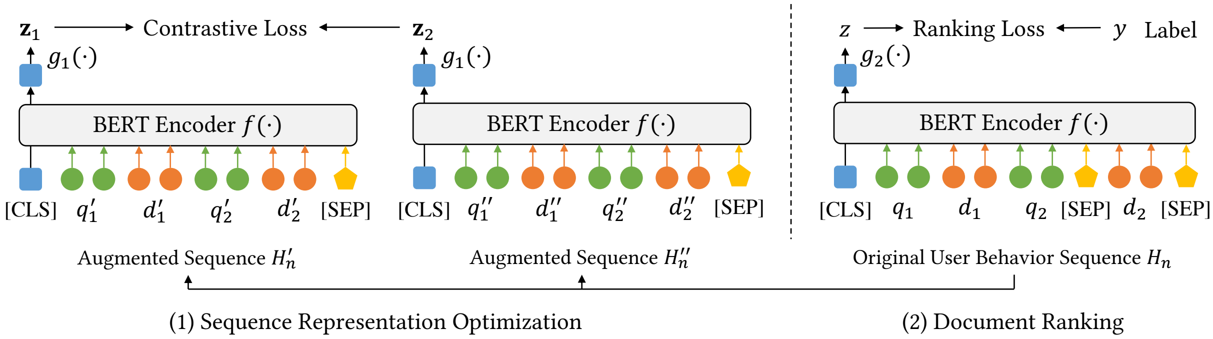
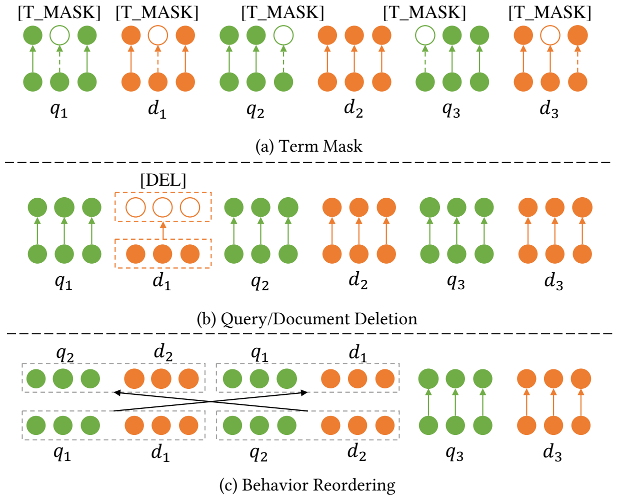

$\mathbf{z}_1$
Contrastive Loss
$\mathbf{z}_2$
$g_1(\cdot)$
$g_1(\cdot)$
BERT Encoder $f(\cdot)$
BERT Encoder $f(\cdot)$
[CLS] q_1' d_1' q_2' d_2' [SEP]
[CLS] q_1'' d_1'' q_2'' d_2'' [SEP]
Augmented Sequence H_n'
Augmented Sequence H_n''
z
Ranking Loss
y Label
$g_2(\cdot)$
BERT Encoder $f(\cdot)$
[CLS] q_1 d_1 q_2 [SEP] d_2 [SEP]
Original User Behavior Sequence H_n
(1) Sequence Representation Optimization
(2) Document Ranking

Figure 1: The illustration of COCA. The query-document sequence (H_n) is augmented with two different strategies, and the processed sequences are treated as a positive pair (H_n' and H_n''). Other augmented sequences in the same minibatch are used to construct negative pairs for them (not shown here). The contrastive objective is to pull close the representation of the two sequences in positive pairs and push apart the representation of others.


[T_MASK]
[T_MASK]
[T_MASK]
[T_MASK]
[T_MASK]
q_1 d_1 q_2 d_2 q_3 d_3
[DEL]
q_1 d_1 q_2 d_2 q_3 d_3
q_2 d_2 q_1 d_1
q_1 d_1 q_2 d_2 q_3 d_3

(a) Term Mask

(b) Query/Document Deletion

(c) Behavior Reordering

Figure 2: Three augmentation strategies used in COCA. We use the user behavior sequence with three query-document pairs as an example.

strategy as a function f^{tm} over the user behavior sequence H_n as:

$$f^{\text{tm}}(H_n) = \{\widehat{w}_1, \cdots, \widehat{w}_N\}, \tag{1}$$

$$\widehat{w}_t = \begin{cases} w_t, & t \notin T_n, \\ [\text{T\_MASK}], & t \in T_n. \end{cases} \tag{2}$$

(b) **Query/Document Deletion.** Random crop (deletion) is a common data augmentation strategy in computer vision to increase the variety of images [10, 53]. This operation can create a random subset of an original image and help the model generalize better. Inspired by this, we propose a query/document deletion augmentation operation for contrastive learning.

The query/document deletion strategy can improve the learning of sequence representation in two respects. First, after deletion, the resulting user behavior sequence becomes a similar one with the difference on some queries or documents. This reflects a type of variation in real query logs. By contrasting these similar sequences with others, the models are trained to learn the influence of the deleted queries or documents. Second, the generated incomplete sequence provides a partial view of the original sequence, which forces the model to learn a more robust representation without relying on complete information.

Specifically, for a user behavior sequence $H_n = \{q_1, d_1, \cdots, q_n, d_n\}$, we treat each query and document as a sub-sequence s and represent the sequence as $H_n = \{s_1, s_2, \cdots, s_{2n-1}, s_{2n}\}$. Then, we randomly delete a proportion μ of sub-sequences $R_n = \{r_1, \cdots, r_L\}$, where $L_{\text{del}} = \lfloor 2n \cdot \mu \rfloor$, and r_i is the index of the sub-sequence to be deleted. Different from the term mask strategy, if a query or document is deleted, the whole sub-sequence is replaced by a special token "[DEL]". This augmentation strategy is formulated as a function f^{del} on H_n and defined as:

$$f^{\text{del}}(H_n) = \{\widehat{s}_1, \cdots, \widehat{s}_{2n}\}, \tag{3}$$

$$\widehat{s}_r = \begin{cases} s_r, & r \notin R_n, \\ [\text{DEL}], & r \in R_n. \end{cases} \tag{4}$$

(c) **Behavior Reordering.** Many tasks assume the strict order of the sequence, *e.g.*, natural language generation [35, 54] and text coherence modeling [36, 37, 66]. However, we observe that the user search behavior sequence is much more flexible. For example, when users only have a vague search intent, they will issue several queries in a random order to obtain related information before making their real intent clear [22]. Besides, sometimes users may issue a repeated query when they miss some information, which is called re-finding behavior [42, 65]. Under this circumstance, we cannot assume the order of the queries is strict. To prevent the model from relying too much on the order information and make the model more robust to the newly issued query, we propose a behavior reordering strategy for contrastive learning. Different from the former two strategies, user behavior reordering does not reduce the information contained in the sequence. Models can focus on learning content representation in queries and documents rather than merely "remembering" their relative order.

For a user behavior sequence $H_n = \{q_1, d_1, \cdots, q_n, d_n\}$, we treat each query and its corresponding document as a behavior sub-sequence and denote it as $H_n = \{b_1, \cdots, b_n\}$, where $b_i = \{q_i, d_i\}$.

Then, we randomly select two behavior sub-sequences and switch their positions, and this operation is conducted $L_{\text{br}} = \lfloor n \cdot \eta \rfloor$ times, where η is the reordering ratio. Considering the randomly selected i-th pairwise position as (u_i, v_i), we switch b_{u_i} and b_{v_i}, which can be formulated as a function f^{br} on H_n:

$$f^{br}(H_n) = \{\widehat{b}_1, \cdots, \widehat{b}_n\} \tag{5}$$

$$\widehat{b}_j = \begin{cases} b_j, & j \neq u_i \text{ and } j \neq v_i, \\ b_{v_i}, & j = u_i, \\ b_{u_i}, & j = v_i. \end{cases} \tag{6}$$

3.3.2 Representation. Previous work has shown the effectiveness of applying BERT [14] for sequence representation [13, 45, 48, 52, 66]. In our framework, we also use the pre-trained BERT as an encoder to represent the augmented user behavior sequences (shown in the left side of Figure 1). For a user behavior sequence $H_n = \{q_1, d_1, \cdots, q_n, d_n\}$, following the design of vanilla BERT, we add special tokens "[CLS]" and "[SEP]" at the head and tail of the sequence, respectively. Besides, to further indicate the end of each query/document, we append a special token "[EOS]" at the end of it. Therefore, the input sequence X is represented as:

$$X = [\text{CLS}]q_1[\text{EOS}]d_1[\text{EOS}] \cdots q_n[\text{EOS}]d_n[\text{EOS}][\text{SEP}]. \tag{7}$$

Then, the embedding of each token, the positional embedding, and the segment embedding[2] are added together and input to BERT to obtain the contextualized representation. The output of BERT is a sequence of representations for all tokens, and we use the representation of "[CLS]" token as the sequence representation:

$$\mathbf{z} = g_1\big(\text{BERT}(X)_{[\text{CLS}]}\big), \tag{8}$$

where $\mathbf{z} \in \mathbb{R}^{768}$, and $g_1(\cdot)$ is a linear projection.

3.3.3 Training Objective. We apply a contrastive learning objective to optimize the user behavior sequence representation. A contrastive learning loss function is defined for the contrastive prediction task, *i.e.*, trying to predict the positive augmentation pair (H_i, H_j) in set $\{\mathcal{H}\}$. We construct the set $\{\mathcal{H}\}$ by randomly augmenting twice for all sequences in a minibatch. The strategy of each augmentation is randomly selected from our proposed three ones. Assuming a minibatch with size N, we can obtain a set $\{\mathcal{H}\}$ with size $2N$. The two augmented sequences from the *same* user behavior sequence form the positive pair, while all other sequences from the same minibatch are regarded as negative samples for them. Following previous work [10, 16, 19, 59], the contrastive learning loss for a positive pair is defined as:

$$l(i,j) = -\log \frac{\exp(\text{sim}(\mathbf{z}_i, \mathbf{z}_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{k \neq i} \exp(\text{sim}(\mathbf{z}_i, \mathbf{z}_k)/\tau)}, \tag{9}$$

where $\mathbb{1}_{k \neq i}$ is the indicator function to judge whether $k \neq i$ and τ is a hyperparameter representing temperature. The overall contrastive learning loss is defined as all positive pairs' losses in a minibatch:

$$\mathcal{L}_{\text{CL}} = \sum_{i=1}^{2N} \sum_{j=1}^{2N} m(i,j) l(i,j), \tag{10}$$

where $m(i,j) = 1$ when (H_i, H_j) is a positive pair, and $m(i,j) = 0$ otherwise.

From another perspective, the contrastive learning stage can be viewed as a kind of domain-specific post-training for pre-trained language models. As these contextualized language models are usually pre-trained on general corpora, such as the Toronto Books Corpus and Wikipedia, it is less effective to directly fine-tune these models on our downstream ranking task if there is a domain shift. Our contrastive learning stage can help the model on domain adaptation to further improve the ranking task. This strategy has shown to be effective in various tasks including reading comprehension [62] and dialogue generation [21, 56].

3.4 Context-aware Document Ranking

In the previous step, the BERT encoder has been optimized with the contrastive learning objective. We now incorporate this BERT encoding to learn the context-aware document ranking task.

3.4.1 Representation. Previous studies have applied BERT for ranking in a manner of sequence pair classification [13, 45, 48]. Different from the first stage, the ranking stage aims at measuring the relationship between the historical user behavior sequence $H_{n-1} = \{q_1, d_1, \cdots, q_{n-1}, d_{n-1}\}$, the current query q_n, and a candidate document $d_{n,i}$. Therefore, we treat $H_{n-1} \cup q_n$ as one sequence and $d_{n,i}$ as another sequence, and the input sequence Y is represented as:

$$Y = [\text{CLS}]q_1[\text{EOS}]d_1[\text{EOS}] \cdots q_n[\text{EOS}][\text{SEP}]d_{n,i}[\text{EOS}][\text{SEP}].$$

Afterwards, the embedding of each token, the positional embedding, and the segment embedding are added together and input to BERT. Note that Y contains two sequences, so we set their segment embeddings respectively as 0 and 1 to distinguish them. The output representation of "[CLS]" is used as the sequence representation to calculate the ranking score z as:

$$\mathbf{h} = \text{BERT}(Y)_{[\text{CLS}]}, \quad z = g_2(\mathbf{h}), \tag{11}$$

where $\mathbf{h} \in \mathbb{R}^{768}$, and $g_2(\cdot)$ is a linear projection to map the representation into a (scalar) score.

3.4.2 Optimization. Following previous studies [3, 48], we use the following cross-entropy loss to optimize the model:

$$\mathcal{L}_{\text{rank}} = -\frac{1}{N} \sum_{i=1}^{N} y_i \log z_i + (1 - y_i) \log(1 - z_i), \tag{12}$$

where N is the number of samples in the training set.

4 EXPERIMENTS

4.1 Datasets and Evaluation Metrics

We conduct experiments on two public datasets: AOL search log data[3] [46] and Tiangong-ST query log data [9].

For AOL search log, we use the one provided by Ahmad et al. [3]. The dataset contains a large number of sessions, and each session consists of several queries. In training and validation sets, there are

[2] Please refer to the original paper of BERT [14] for more details about these embeddings.

[3] We understand that the AOL dataset should normally not be used in experiments. We choose to use it here because it contains real human clicks, which fits our experiments well. MS MARCO Conversational Search dataset may be another possible dataset, but the sessions in it are artificially constructed rather than real search logs. So, we do not use the MS MARCO dataset in experiments.

Table 1: The statistics of the datasets used in the paper.

AOL	Training	Validation	Test
# Sessions	219,748	34,090	29,369
# Queries	566,967	88,021	76,159
Avg. # Query per Session	2.58	2.58	2.59
Avg. # Document per Query	5	5	50
Avg. Query Len	2.86	2.85	2.9
Avg. Document Len	7.27	7.29	7.08
Avg. # Clicks per Query	1.08	1.08	1.11
Tiangong-ST	**Training**	**Validation**	**Test**
# Sessions	143,155	2,000	2,000
# Queries	344,806	5,026	6,420
Avg. # Query per Session	2.41	2.51	3.21
Avg. # Document per Query	10	10	10
Avg. Query Len	2.89	1.83	3.46
Avg. Document Len	8.25	6.99	9.18
Avg. # Clicks per Query	0.94	0.53	(3.65)

five candidate documents for each query in the session. In the test set, 50 documents retrieved by BM25 [49] are used as candidates for each query in the session. All queries have at least one satisfied click in this dataset, and if there are more than one clicked documents, we use the first one in the list to construct the user behavior sequence.

Tiangong-ST dataset is collected from a Chinese commercial search engine. It contains web search session data extracted from an 18-day search log. Each query in the dataset has 10 candidate documents. In the training and validation sets, we use the clicked documents as the satisfied clicks. Some queries may have no satisfied click, we use a special token "[Empty]" for padding. For the test, the last query of each session is manually annotated with relevance scores, while other (previous) queries in the session have only click labels. Therefore, we construct two test sets based on the original test data as follows:

(1) Tiangong-ST-Click: In this test set, we only use the previous queries (*i.e.*, without the last query) and their candidate documents. Similar to AOL dataset, in this test scenario, all documents are labeled with "click" or "unclick", and the model is asked to rank the clicked documents as high as possible. Note that the query with no click document is not used for testing.

(2) Tiangong-ST-Human: In this test set, only the last query with human annotated relevance score is used. The score ranges from 0 to 4. More details can be found in [9].

The statistics of both datasets are shown in Table 1. Following previous studies [3, 18, 28, 29], to reduce memory requirements and speed up training, we only use the document title as its content.

Evaluation Metrics We use Mean Average Precision (MAP), Mean Reciprocal Rank (MRR), and Normalized Discounted Cumulative Gain at position k (NDCG@k, $k = \{1, 3, 5, 10\}$) as evaluation metrics. For Tiangong-ST-Human, since the candidate documents are provided by a commercial search engine, the irrelevant documents are expected to be limited. Hence, as suggested by the authors of [9], we only evaluate the results with NDCG@k. All evaluation results are calculated by TREC's evaluation tool (trec_eval) [24].

4.2 Baseline

We compare our method with several baseline methods, including those for (1) ad-hoc ranking and (2) context-aware ranking.

(1) **Ad-hoc ranking methods**. These methods do not use context information (historical queries and documents), and only current query is used for ranking documents. `KNRM` [61] performs fine-grained interaction between current query and candidate documents and obtain a matching matrix. The ranking features and scores are then calculated by a kernel pooling method. `ARC-I` [27] is a representation-based method. The query and document are represented by convolutional neural networks (CNNs), respectively. The score is calculated by a multi-layer perceptron (MLP). `ARC-II` [27] is an interaction-based method. A matching map is constructed from the query and document, based on which the matching features are extracted by CNNs. The score is also computed by an MLP. `Duet` [44] computes local and distributed representations of the query and document by several layers of CNNs and MLPs. Then, it integrates both interaction-based features and representation-based features to compute ranking scores.

(2) **Context-aware ranking methods**. These methods can leverage both context information and current query to rank candidate documents. `M-NSRF` [2] is a multi-task model, which jointly predicts the next query and ranks corresponding documents. The historical queries in a session are encoded by a recurrent neural network (RNN). The ranking score is computed based on the query representation, history representation, and document representation. `M-Match-Tensor` [2] is similar to `M-NSRF` but learns a contextual representation for each word in the queries and documents. The computation of ranking score is based on the word-level representation. `CARS` [3] is also a multi-task model, which learns query suggestion and document ranking simultaneously. Different from `M-NSRF`, this method also models the click documents in the history through an RNN. An attention mechanism is applied to compute representations for each query and document. The final ranking score is computed based on the representation of historical queries, clicked documents, current query, and candidate documents[4]. `HBA-Transformer` [48] (henceforth denoted as `HBA`) concatenates historical queries, clicked documents, and unclick documents into a long sequence and applies BERT [14] to encode them into representations. Then, a higher-level transformer structure with behavior embedding and relative position embedding is employed to further enhance the representation. Finally, the representation of the first token ("[CLS]") is used to calculate the ranking score. This is the state-of-the-art method in context-aware document ranking task. It is the most similar to our approach, but without contrastive learning.

4.3 Implement Details

We use PyTorch [47] and Transformers [58] to implement our model. The pre-trained BERT is provided by Huggingface[5]. The maximum number of tokens in the two stages are set as 128. Sequences with more than 128 tokens are truncated by popping query-document

[4] We will notice some slight discrepancies between our results and those of the original paper of CARS. This is due to different tie-breaking strategies in evaluation. Following [48], we use trec_eval while the authors of CARS uses an author-implemented evaluation.

[5] https://huggingface.co/bert-base-uncased

Table 2: Experimental results on all datasets. All baseline models are based on the code released in the original paper. The best performance and the second best performance are in bold and underlined, respectively. The improvement of COCA over the best baseline is given in the last column. † indicates COCA achieves significant improvements over all existing methods in paired t-test with p-value < 0.01.

Dataset	Metric	ARC-I	ARC-II	KNRM	Duet	M-NSRF	M-Match	CARS	HBA	COCA	Improv.
AOL	MAP	0.3361	0.3834	0.4038	0.4008	0.4217	0.4459	0.4297	0.5281	**0.5500**†	4.15%
	MRR	0.3475	0.3951	0.4133	0.4111	0.4326	0.4572	0.4408	0.5384	**0.5601**†	4.03%
	NDCG@1	0.1988	0.2428	0.2397	0.2492	0.2737	0.3020	0.2816	0.3773	**0.4024**†	6.65%
	NDCG@3	0.3108	0.3564	0.3868	0.3822	0.4025	0.4301	0.4117	0.5241	**0.5478**†	4.52%
	NDCG@5	0.3489	0.4026	0.4322	0.4246	0.4458	0.4697	0.4542	0.5624	**0.5849**†	4.00%
	NDCG@10	0.3953	0.4486	0.4761	0.4675	0.4886	0.5103	0.4971	0.5951	**0.6160**†	3.51%
Tiangong-ST-Click	MAP	0.6597	0.6729	0.6551	0.6745	0.6836	0.6778	0.6909	0.6957	**0.7481**†	7.53%
	MRR	0.6826	0.6954	0.6748	0.7026	0.7065	0.6993	0.7134	0.7171	**0.7696**†	7.32%
	NDCG@1	0.5315	0.5458	0.5104	0.5738	0.5609	0.5499	0.5677	0.5726	**0.6386**†	11.53%
	NDCG@3	0.6383	0.6553	0.6415	0.6511	0.6698	0.6636	0.6764	0.6807	**0.7445**†	9.37%
	NDCG@5	0.6946	0.7086	0.6949	0.6955	0.7188	0.7199	0.7271	0.7292	**0.7858**†	7.76%
	NDCG@10	0.7509	0.7608	0.7469	0.7621	0.7691	0.7646	0.7746	0.7781	**0.8180**†	5.13%
Tiangong-ST-Human	NDCG@1	0.7088	0.7131	0.7560	0.7577	0.7124	0.7311	0.7385	0.7612	**0.7769**	2.06%
	NDCG@3	0.7087	0.7237	0.7457	0.7354	0.7308	0.7233	0.7386	0.7518	**0.7576**	0.77%
	NDCG@5	0.7317	0.7379	0.7716	0.7548	0.7489	0.7427	0.7512	0.7639	**0.7703**	0.84%
	NDCG@10	0.8691	0.8732	0.8894	0.8829	0.8795	0.8801	0.8837	0.8896	**0.8932**	0.40%

pairs from the head. We use AdamW [39] optimizer in both stages. In the sequence representation optimization stage, both the term mask ratio and query/document deletion ratio are tuned from 0.1 to 0.9 and set as 0.6. As for behavior reordering, only one pair of positions are switched because the session is not long (on average 2.5 queries per sessions). The three strategies are randomly selected. Note that the reordering strategy can only be applied to sessions with more than one query. The batch size is set as 128, and the temperature is set as 0.1. We train the model for four epochs. The learning rate is set as 5e-5. In the document ranking stage, we apply a dropout layer on the sequence representation with the rate of 0.1. The learning rate is set as 5e-5 and linearly decayed during the training. We train the model for three epochs. All hyperparameters are tuned based on the performance on the validation set. Our code is released on GitHub at https://github.com/DaoD/COCA.

4.4 Experimental Results and Analysis

The experimental results are shown in Table 2. We can find COCA outperforms all existing methods. This result clearly demonstrates the superiority of our method. Based on the results, we can make the following observations.

(1) Among all models, COCA achieves the best results, which demonstrates its effectiveness on modeling user behavior sequence through contrastive learning. In general, the context-aware document ranking models perform better than ad-hoc ranking models. For example, on AOL dataset, the weak contextualized model M-NSRF can still outperform the strong ad-hoc ranking model KNRM. This indicates that modeling user behavior sequence is beneficial for understanding user intent and improving the ranking results.

(2) Compared with the RNN-based multi-task learning models (M-NSRF, M-Match-Tensor, and CARS), BERT-based methods (HBA and COCA) achieve better performance. Specifically, on AOL dataset, HBA and COCA improve the results by more than 15% in terms of all metrics. It is worth noting that HBA and COCA learn document ranking independently without supervision signals from query suggestion task. This result reflects the clear advantage of applying pre-trained language models (*e.g.*, BERT) in document ranking.

(3) HBA is the state-of-the-art method on context-aware document ranking task. It designs complex structures over a BERT encoder to consider user behavior in various aspects, including an intra-behavior attention on clicked documents and skipped documents; an inter-behavior attention on all turns; and an embedding indicates their relative positions. In contrast, our COCA only applies a standard BERT encoder and achieves significantly better performance (paired t-test with p-value < 0.01). Both MAP and MRR are improved about 4%. The key difference between them is the contrastive learning we use. The improvements of COCA over HBA directly reflects the advantage of using contrastive learning for behavior sequence representation.

(4) Intriguingly, the improvements of COCA on AOL and Tiangong-ST-Click are much more significant than that on Tiangong-ST-Human test set. The potential reasons fall in two aspects: (a) COCA is trained on data with click labels rather than relevance labels, and the construction of the user behavior sequence is also based on click labels. Therefore, the model is better at predicting click-based scores than relevance scores. (b) According to our statistics, there are more than 77.4% documents labeled as relevant (*i.e.*, their annotated relevance scores are larger than 1), so the base score is very high. Even the basic model ARC-I can achieve 0.7088 and 0.8691 in terms of NDCG@1 and NDCG@10. Without more accurate relevance labels for training, it is more difficult for our model to further improve relevance ranking.

4.5 Discussion

We further investigate the following research questions.

Table 3: Performance of COCA on the AOL dataset with different data augmentation strategies.

	MAP	MRR	NDCG@1	NDCG@3	NDCG@10
COCA (Full)	**0.5500**	**0.5601**	**0.4024**	**0.5478**	**0.6160**
None	0.5341	0.5445	0.3867	0.5296	0.5999
TM	0.5472	0.5576	0.4009	0.5444	0.6121
QDD	0.5452	0.5554	0.3969	0.5422	0.6110
TM + QDD	0.5492	0.5592	0.4005	0.5467	0.6155
TM + BR	0.5448	0.5550	0.3963	0.5414	0.6115
QDD + BR	0.5473	0.5576	0.3995	0.5444	0.6132

Table 4: Performance of COCA on the AOL dataset with different hyperparameters.

		CE (↓)	Acc.	MAP	MRR	NDCG@1	NDCG@10
Temperature τ	0.05	**0.5662**	79.62	0.5417	0.5521	0.3947	0.6078
	0.1	0.5823	83.56	**0.5500**	**0.5601**	**0.4024**	**0.6160**
	0.3	1.6240	**84.03**	0.5451	0.5552	0.3972	0.6116
	0.5	4.3226	69.85	0.5433	0.5536	0.3950	0.6031
	1.0	5.2148	62.33	0.5417	0.5522	0.3951	0.6073
Batch Size	16	0.7289	81.14	0.5380	0.5482	0.3897	0.6044
	32	0.7226	80.92	0.5447	0.5547	0.3972	0.6108
	64	0.7210	81.20	0.5432	0.5534	0.3951	0.6089
	128	**0.5823**	**83.56**	**0.5500**	**0.5601**	**0.4024**	**0.6160**

4.5.1 Influence of Data Augmentation Strategy. To study the effectiveness of our proposed sequence augmentation strategy, we test the performance on AOL with different combinations of strategies. The results are shown in Table 3. "None" means that we use the original BERT parameters for document ranking without our proposed sequence optimization stage. We denote the term mask strategy as "TM", query/document deletion as "QDD", and behavior reordering as "BR". Note that the reordering strategy can only apply to sequences with more than two query-document pairs, thus cannot work independently.

First, compared with no sequence optimization stage, optimizing sequence representation with any combination of our proposed strategies is helpful. This clearly demonstrates that our proposed method is effective in building a more robust representation. Second, the term mask works best and this single strategy can improve around 2.5% in MAP. This implies that learning user behavior sequences with similar queries and documents are very useful for document ranking. Finally, it is interesting to see that combining term mask and behavior reordering strategy (*i.e.*, "TM + BR") leads to a performance degradation compared with only using the term mask strategy. After checking the sequence representation optimization process, we find that the contrastive learning loss in this case is very low and the prediction accuracy is very high, which indicates that this combination is easy to overfit and cannot learn a good sequence representation.

4.5.2 Performance with Different Hyperparameters. As reported in recent work [10], the temperature and batch size are two important hyperparameters in contrastive learning. To investigate the impact of them, we train our model with different settings and test their performance. In addition to evaluating the performance of ranking, we also compute the loss value (cross-entropy, CE) and prediction accuracy in contrastive prediction. The results are shown in Table 4.

Considering temperature, according to Equation (9), a higher temperature will cause a higher loss, which are consistent with our results. However, a lower contrastive loss cannot always lead to a better performance. Indeed, $\tau = 0.1$ is the best choice for the document ranking task. Therefore, it is important to select a proper temperature for contrastive learning. Similar observations are also reported in other recent studies [10, 19].

As for batch size, we can see that contrastive learning benefits from larger batch sizes. According to a recent study [10], larger batch sizes can provide more negative examples, so that the convergence can be facilitated. Due to our limited hardware resources, the largest batch size we can handle is 128. We speculate that a larger batch size can bring more improvements.

4.5.3 Performance on Sessions with Different Lengths. To understand the impact of the session length on the final ranking performance, we categorize the sessions in the test set into three bins:

(1) Short sessions (with 1-2 queries) - 77.13% of the test set;
(2) Medium sessions (with 3-4 queries) - 18.19% of the test set;
(3) Long sessions (with 5+ queries) - 4.69% of the test set.

As we also consider sessions with only one query, the short sessions have a higher proportion than that provided in [3].

We compare COCA with Duet, CARS, HBA on AOL dataset and show the results regarding MAP and NDCG@3 in the left side of Figure 3. First, it is evident that COCA outperforms all context-aware baseline methods on all three bins of sessions. This suggests COCA's advantages in learning search context. Second, we can see the ad-hoc ranking method Duet performs worse than other context-aware ranking methods. This demonstrates once again that modeling the historical user behavior is essential for improving the document ranking performance. Third, we can observe that COCA performs relatively worse in long sessions than in short sessions. We hypothesize that those longer sessions are intrinsically more difficult, and similar trend in baseline methods can support this. This can be due to the fact that a long session may contain more noise or exploratory search. This is also shown by a larger improvement in the short sessions from COCA to the ad-hoc baseline ranker Duet than that in the long sessions (37.10% v.s. 26.83% in terms of MAP). This result implies that it may be useful to model the immediate search context rather than the whole context.

4.5.4 Effect of Modeling User Behavior Progression. It is important to study how the modeled search context helps document ranking when a search session progresses. We compare COCA with CARS and HBA at individual query positions in short (S), medium (M), and long (L) sessions. The results are reported in the right side of Figure 3. Due to the limited space, long sessions with more than seven queries are not presented.

It is noticeable that the ranking performance is improved steadily as a search session progresses, *i.e.*, more search context becomes available for predicting the next click. Both COCA and HBA benefit from it, while COCA improves faster by better exploiting the context. In contrast, the performance of CARS is unstable. This implies that BERT-based methods are much more effective in modeling search context. One interesting finding is that, when the search sessions

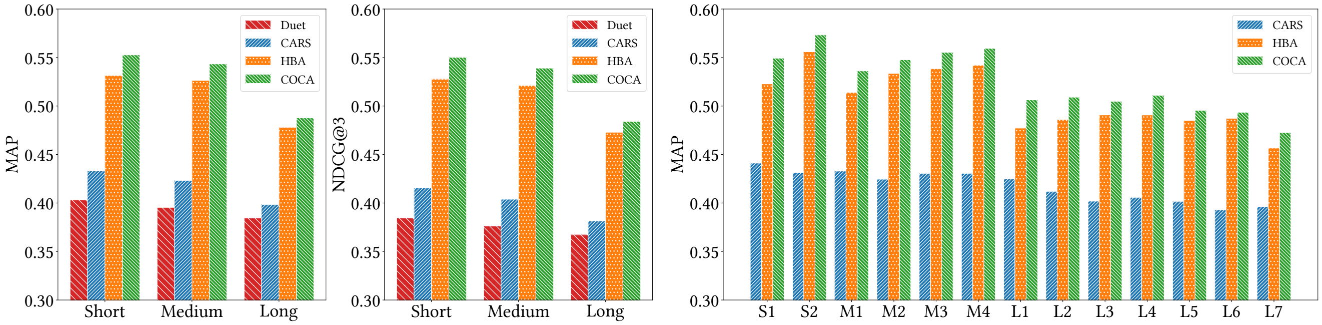


Figure 3: (Left) Performance on different lengths of sessions. (Right) Performance at different query positions in short (S1-S2), medium (M1-M4), and long sessions (L1-L7). The number after "S", "M", or "L" indicates the query index in the session.

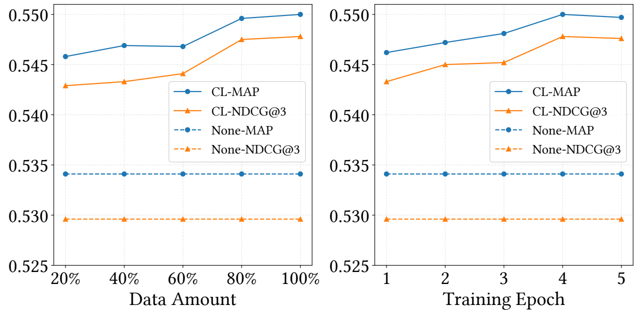


Figure 4: The performance with different training data amount and training epochs.

get longer (*e.g.*, from L4 to L7), the gain of COCA diminishes. We attribute this to the more noisy nature of long sessions.

4.5.5 Influence of Amount of Training Data. As reported by recent studies [10, 19], the amount of data for contrastive learning has a great impact on downstream task (*e.g.*, document ranking in our case). We investigate such influence by training the model with different proportions of data and different epochs. As a comparison, we also illustrate the performance of COCA without sequence representation optimization stage (denoted as "None").

We first reduce the number of training data used for contrastive learning[6]. The results are shown in the left side of Figure 4. It is clear that contrastive learning benefits from a larger amount of data. Surprisingly, our proposed sequence representation optimization stage can still work with only 20% of training data. This demonstrates the potential and effectiveness of learning better sequence representation for context-aware document ranking. We also train COCA with different number of epochs in the sequence optimization stage. The performance on document ranking is shown in the right side of Figure 4. The results suggest that the contrastive learning also benefits from larger training epochs. In our implementation, the data augmentation strategies are randomly selected in different epochs. Therefore, the sequence representation can be more fully learned. When training more than four epochs, the performance is stable without further improvement. Therefore, four epochs is the best choice in our experiments.

5 CONCLUSION AND FUTURE WORK

In this work, we aimed at learning better representation of user behavior sequence for context-aware document ranking. A self-supervised task with contrastive learning objective is introduced for optimizing sequence representation before learning document ranking. To construct positive pairs in contrastive learning, we proposed three data augmentation strategies at term, query/document, and user behavior level. These strategies can improve the generalization and robustness of sequence representation. The optimized sequence representation is used in document ranking task. We conducted comprehensive experiments on two large-scale search log datasets. The results clearly showed that our proposed method is very effective. In particular, our method with contrastive learning was shown to outperform the close competitor HBA without it.

This is the first attempt to utilize contrastive learning in IR and much remains to be explored. For example, it may be more appropriate to exploit recent history instead of the whole history. Query and document weighting in the history could also be a promising avenue.

ACKNOWLEDGMENTS

Zhicheng Dou is the corresponding author. This work was supported by a Discovery grant of the Natural Science and Engineering Research Council of Canada, National Natural Science Foundation of China (No. 61872370 and No. 61832017), Beijing Outstanding Young Scientist Program (No. BJJWZYJH012019100020098), Shandong Provincial Natural Science Foundation (No. ZR2019ZD06), and Intelligent Social Governance Platform, Major Innovation & Planning Interdisciplinary Platform for the "Double-First Class" Initiative, Renmin University of China. We also wish to acknowledge the support provided and contribution made by Public Policy and Decision-making Research Lab of Renmin University of China.

[6] Note that all models are trained four epochs with different number of data.

REFERENCES

[1] Eugene Agichtein, Ryen W. White, Susan T. Dumais, and Paul N. Bennett. 2012. Search, interrupted: understanding and predicting search task continuation. In *The 35th International ACM SIGIR conference on research and development in Information Retrieval, SIGIR '12, Portland, OR, USA, August 12-16, 2012*, William R. Hersh, Jamie Callan, Yoelle Maarek, and Mark Sanderson (Eds.). ACM, 315–324. https://doi.org/10.1145/2348283.2348328

[2] Wasi Uddin Ahmad, Kai-Wei Chang, and Hongning Wang. 2018. Multi-Task Learning for Document Ranking and Query Suggestion. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net. https://openreview.net/forum?id=SJ1nzBeA-

[3] Wasi Uddin Ahmad, Kai-Wei Chang, and Hongning Wang. 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 2019, Paris, France, July 21-25, 2019*, Benjamin Piwowarski, Max Chevalier, Éric Gaussier, Yoelle Maarek, Jian-Yun Nie, and Falk Scholer (Eds.). ACM, 385–394. https://doi.org/10.1145/3331184.3331246

[4] Paul N. Bennett, Ryen W. White, Wei Chu, Susan T. Dumais, Peter Bailey, Fedor Borisyuk, and Xiaoyuan Cui. 2012. Modeling the impact of short- and long-term behavior on search personalization. In *The 35th International ACM SIGIR conference on research and development in Information Retrieval, SIGIR '12, Portland, OR, USA, August 12-16, 2012*, William R. Hersh, Jamie Callan, Yoelle Maarek, and Mark Sanderson (Eds.). ACM, 185–194. https://doi.org/10.1145/2348283.2348312

[5] Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew M. Dai, Rafal Józefowicz, and Samy Bengio. 2016. Generating Sentences from a Continuous Space. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, CoNLL 2016, Berlin, Germany, August 11-12, 2016*, Yoav Goldberg and Stefan Riezler (Eds.). ACL, 10–21. https://doi.org/10.18653/v1/k16-1002

[6] Huanhuan Cao, Daxin Jiang, Jian Pei, Enhong Chen, and Hang Li. 2009. Towards context-aware search by learning a very large variable length hidden markov model from search logs. In *Proceedings of the 18th International Conference on World Wide Web, WWW 2009, Madrid, Spain, April 20-24, 2009*, Juan Quemada, Gonzalo León, Yoëlle S. Maarek, and Wolfgang Nejdl (Eds.). ACM, 191–200. https://doi.org/10.1145/1526709.1526736

[7] Ben Carterette, Paul D. Clough, Mark M. Hall, Evangelos Kanoulas, and Mark Sanderson. 2016. Evaluating Retrieval over Sessions: The TREC Session Track 2011-2014. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval, SIGIR 2016, Pisa, Italy, July 17-21, 2016*, Raffaele Perego, Fabrizio Sebastiani, Javed A. Aslam, Ian Ruthven, and Justin Zobel (Eds.). ACM, 685–688. https://doi.org/10.1145/2911451.2914675

[8] Wei-Cheng Chang, Felix X. Yu, Yin-Wen Chang, Yiming Yang, and Sanjiv Kumar. 2020. Pre-training Tasks for Embedding-based Large-scale Retrieval. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net. https://openreview.net/forum?id=rkg-mA4FDr

[9] Jia Chen, Jiaxin Mao, Yiqun Liu, Min Zhang, and Shaoping Ma. 2019. TianGong-ST: A New Dataset with Large-scale Refined Real-world Web Search Sessions. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019*, Wenwu Zhu, Dacheng Tao, Xueqi Cheng, Peng Cui, Elke A. Rundensteiner, David Carmel, Qi He, and Jeffrey Xu Yu (Eds.). ACM, 2485–2488. https://doi.org/10.1145/3357384.3358158

[10] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. 2020. A Simple Framework for Contrastive Learning of Visual Representations. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event (Proceedings of Machine Learning Research, Vol. 119)*. PMLR, 1597–1607. http://proceedings.mlr.press/v119/chen20j.html

[11] Wanyu Chen, Fei Cai, Honghui Chen, and Maarten de Rijke. 2018. Attention-based Hierarchical Neural Query Suggestion. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018*, Kevyn Collins-Thompson, Qiaozhu Mei, Brian D. Davison, Yiqun Liu, and Emine Yilmaz (Eds.). ACM, 1093–1096. https://doi.org/10.1145/3209978.3210079

[12] Andrew M. Dai and Quoc V. Le. 2015. Semi-supervised Sequence Learning. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, Corinna Cortes, Neil D. Lawrence, Daniel D. Lee, Masashi Sugiyama, and Roman Garnett (Eds.). 3079–3087. https://proceedings.neurips.cc/paper/2015/hash/7137debd45ae4d0ab9aa953017286b20-Abstract.html

[13] Zhuyun Dai and Jamie Callan. 2019. Deeper Text Understanding for IR with Contextual Neural Language Modeling. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, July 21-25, 2019*, Benjamin Piwowarski, Max Chevalier, Éric Gaussier, Yoelle Maarek, Jian-Yun Nie, and Falk Scholer (Eds.). ACM, 985–988. https://doi.org/10.1145/3331184.3331303

[14] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, Jill Burstein, Christy Doran, and Thamar Solorio (Eds.). Association for Computational Linguistics, 4171–4186. https://doi.org/10.18653/v1/n19-1423

[15] Alexey Dosovitskiy, Jost Tobias Springenberg, Martin A. Riedmiller, and Thomas Brox. 2014. Discriminative Unsupervised Feature Learning with Convolutional Neural Networks. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, Zoubin Ghahramani, Max Welling, Corinna Cortes, Neil D. Lawrence, and Kilian Q. Weinberger (Eds.). 766–774. https://proceedings.neurips.cc/paper/2014/hash/07563a3fe3bbe7e3ba84431ad9d055af-Abstract.html

[16] Hongchao Fang and Pengtao Xie. 2020. CERT: Contrastive Self-supervised Learning for Language Understanding. *CoRR* abs/2005.12766 (2020). arXiv:2005.12766 https://arxiv.org/abs/2005.12766

[17] Shubhashri G, Unnamalai N, and Kamalika G. 2018. LAWBO: a smart lawyer chatbot. In *Proceedings of the ACM India Joint International Conference on Data Science and Management of Data, COMAD/CODS 2018, Goa, India, January 11-13, 2018*, Sayan Ranu, Niloy Ganguly, Raghu Ramakrishnan, Sunita Sarawagi, and Shourya Roy (Eds.). ACM, 348–351. https://doi.org/10.1145/3152494.3167988

[18] Jianfeng Gao, Xiaodong He, and Jian-Yun Nie. 2010. Clickthrough-based translation models for web search: from word models to phrase models. In *Proceedings of the 19th ACM Conference on Information and Knowledge Management, CIKM 2010, Toronto, Ontario, Canada, October 26-30, 2010*, Jimmy Huang, Nick Koudas, Gareth J. F. Jones, Xindong Wu, Kevyn Collins-Thompson, and Aijun An (Eds.). ACM, 1139–1148. https://doi.org/10.1145/1871437.1871582

[19] Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple Contrastive Learning of Sentence Embeddings. *CoRR* abs/2104.08821 (2021). arXiv:2104.08821 https://arxiv.org/abs/2104.08821

[20] Songwei Ge, Zhicheng Dou, Zhengbao Jiang, Jian-Yun Nie, and Ji-Rong Wen. 2018. Personalizing Search Results Using Hierarchical RNN with Query-aware Attention. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, Torino, Italy, October 22-26, 2018*, Alfredo Cuzzocrea, James Allan, Norman W. Paton, Divesh Srivastava, Rakesh Agrawal, Andrei Z. Broder, Mohammed J. Zaki, K. Selçuk Candan, Alexandros Labrinidis, Assaf Schuster, and Haixun Wang (Eds.). ACM, 347–356. https://doi.org/10.1145/3269206.3271728

[21] Jia-Chen Gu, Tianda Li, Quan Liu, Zhen-Hua Ling, Zhiming Su, Si Wei, and Xiaodan Zhu. 2020. Speaker-Aware BERT for Multi-Turn Response Selection in Retrieval-Based Chatbots. In *CIKM '20: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, October 19-23, 2020*, Mathieu d'Aquin, Stefan Dietze, Claudia Hauff, Edward Curry, and Philippe Cudré-Mauroux (Eds.). ACM, 2041–2044. https://doi.org/10.1145/3340531.3412330

[22] Dongyi Guan, Sicong Zhang, and Hui Yang. 2013. Utilizing query change for session search. In *The 36th International ACM SIGIR conference on research and development in Information Retrieval, SIGIR '13, Dublin, Ireland - July 28 - August 01, 2013*, Gareth J. F. Jones, Paraic Sheridan, Diane Kelly, Maarten de Rijke, and Tetsuya Sakai (Eds.). ACM, 453–462. https://doi.org/10.1145/2484028.2484055

[23] Beliz Gunel, Jingfei Du, Alexis Conneau, and Ves Stoyanov. 2020. Supervised Contrastive Learning for Pre-trained Language Model Fine-tuning. *CoRR* abs/2011.01403 (2020). arXiv:2011.01403 https://arxiv.org/abs/2011.01403

[24] Christophe Van Gysel and Maarten de Rijke. 2018. Pytrec_eval: An Extremely Fast Python Interface to trec_eval. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018*, Kevyn Collins-Thompson, Qiaozhu Mei, Brian D. Davison, Yiqun Liu, and Emine Yilmaz (Eds.). ACM, 873–876. https://doi.org/10.1145/3209978.3210065

[25] Christophe Van Gysel, Evangelos Kanoulas, and Maarten de Rijke. 2016. Lexical Query Modeling in Session Search. In *Proceedings of the 2016 ACM on International Conference on the Theory of Information Retrieval, ICTIR 2016, Newark, DE, USA, September 12- 6, 2016*, Ben Carterette, Hui Fang, Mounia Lalmas, and Jian-Yun Nie (Eds.). ACM, 69–72. https://doi.org/10.1145/2970398.2970422

[26] Raia Hadsell, Sumit Chopra, and Yann LeCun. 2006. Dimensionality Reduction by Learning an Invariant Mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2006), 17-22 June 2006, New York, NY, USA*. IEEE Computer Society, 1735–1742. https://doi.org/10.1109/CVPR.2006.100

[27] Baotian Hu, Zhengdong Lu, Hang Li, and Qingcai Chen. 2014. Convolutional Neural Network Architectures for Matching Natural Language Sentences. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, Zoubin Ghahramani, Max Welling, Corinna Cortes, Neil D. Lawrence, and Kilian Q. Weinberger (Eds.). 2042–2050. https://proceedings.neurips.cc/paper/2014/hash/b9d487a30398d42ecff55c228ed5652b-Abstract.html

[28] Jizhou Huang, Wei Zhang, Yaming Sun, Haifeng Wang, and Ting Liu. 2018. Improving Entity Recommendation with Search Log and Multi-Task Learning. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial*

Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden, Jérôme Lang (Ed.). ijcai.org, 4107–4114. https://doi.org/10.24963/ijcai.2018/571

[29] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry P. Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *22nd ACM International Conference on Information and Knowledge Management, CIKM'13, San Francisco, CA, USA, October 27 - November 1, 2013*, Qi He, Arun Iyengar, Wolfgang Nejdl, Jian Pei, and Rajeev Rastogi (Eds.). ACM, 2333–2338. https://doi.org/10.1145/2505515.2505665

[30] Thorsten Joachims, Laura A. Granka, Bing Pan, Helene Hembrooke, and Geri Gay. 2005. Accurately interpreting clickthrough data as implicit feedback. In *SIGIR 2005: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Salvador, Brazil, August 15-19, 2005*, Ricardo A. Baeza-Yates, Nivio Ziviani, Gary Marchionini, Alistair Moffat, and John Tait (Eds.). ACM, 154–161. https://doi.org/10.1145/1076034.1076063

[31] Thorsten Joachims, Laura A. Granka, Bing Pan, Helene Hembrooke, Filip Radlinski, and Geri Gay. 2007. Evaluating the accuracy of implicit feedback from clicks and query reformulations in Web search. *ACM Trans. Inf. Syst.* 25, 2 (2007), 7. https://doi.org/10.1145/1229179.1229181

[32] Rosie Jones and Kristina Lisa Klinkner. 2008. Beyond the session timeout: automatic hierarchical segmentation of search topics in query logs. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management, CIKM 2008, Napa Valley, California, USA, October 26-30, 2008*, James G. Shanahan, Sihem Amer-Yahia, Ioana Manolescu, Yi Zhang, David A. Evans, Aleksander Kolcz, Key-Sun Choi, and Abdur Chowdhury (Eds.). ACM, 699–708. https://doi.org/10.1145/1458082.1458176

[33] Daniel Khashabi, Sewon Min, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi. 2020. UnifiedQA: Crossing Format Boundaries With a Single QA System. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings, EMNLP 2020, Online Event, 16-20 November 2020*, Trevor Cohn, Yulan He, and Yang Liu (Eds.). Association for Computational Linguistics, 1896–1907. https://doi.org/10.18653/v1/2020.findings-emnlp.171

[34] Omar Khattab and Matei Zaharia. 2020. ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, Jimmy Huang, Yi Chang, Xueqi Cheng, Jaap Kamps, Vanessa Murdock, Ji-Rong Wen, and Yiqun Liu (Eds.). ACM, 39–48. https://doi.org/10.1145/3397271.3401075

[35] Yuta Kikuchi, Graham Neubig, Ryohei Sasano, Hiroya Takamura, and Manabu Okumura. 2016. Controlling Output Length in Neural Encoder-Decoders. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, Jian Su, Xavier Carreras, and Kevin Duh (Eds.). The Association for Computational Linguistics, 1328–1338. https://doi.org/10.18653/v1/d16-1140

[36] Jiwei Li and Eduard H. Hovy. 2014. A Model of Coherence Based on Distributed Sentence Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, Alessandro Moschitti, Bo Pang, and Walter Daelemans (Eds.). ACL, 2039–2048. https://doi.org/10.3115/v1/d14-1218

[37] Jiwei Li and Dan Jurafsky. 2017. Neural Net Models of Open-domain Discourse Coherence. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, Martha Palmer, Rebecca Hwa, and Sebastian Riedel (Eds.). Association for Computational Linguistics, 198–209. https://doi.org/10.18653/v1/d17-1019

[38] Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019. Multi-Task Deep Neural Networks for Natural Language Understanding. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, Anna Korhonen, David R. Traum, and Lluís Màrquez (Eds.). Association for Computational Linguistics, 4487–4496. https://doi.org/10.18653/v1/p19-1441

[39] Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net. https://openreview.net/forum?id=Bkg6RiCqY7

[40] Jiyun Luo, Sicong Zhang, and Hui Yang. 2014. Win-win search: dual-agent stochastic game in session search. In *The 37th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '14, Gold Coast , QLD, Australia - July 06 - 11, 2014*, Shlomo Geva, Andrew Trotman, Peter Bruza, Charles L. A. Clarke, and Kalervo Järvelin (Eds.). ACM, 587–596. https://doi.org/10.1145/2600428.2609629

[41] Xinyu Ma, Jiafeng Guo, Ruqing Zhang, Yixing Fan, Xiang Ji, and Xueqi Cheng. 2021. PROP: Pre-training with Representative Words Prediction for Ad-hoc Retrieval. In *WSDM '21, The Fourteenth ACM International Conference on Web Search and Data Mining, Virtual Event, Israel, March 8-12, 2021*, Liane Lewin-Eytan, David Carmel, Elad Yom-Tov, Eugene Agichtein, and Evgeniy Gabrilovich (Eds.). ACM, 283–291. https://doi.org/10.1145/3437963.3441777

[42] Zhengyi Ma, Zhicheng Dou, Guanyue Bian, and Ji-Rong Wen. 2020. PSTIE: Time Information Enhanced Personalized Search. In *CIKM '20: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, October 19-23, 2020*, Mathieu d'Aquin, Stefan Dietze, Claudia Hauff, Edward Curry, and Philippe Cudré-Mauroux (Eds.). ACM, 1075–1084. https://doi.org/10.1145/3340531.3411877

[43] Yu Meng, Chenyan Xiong, Payal Bajaj, Saurabh Tiwary, Paul Bennett, Jiawei Han, and Xia Song. 2021. COCO-LM: Correcting and Contrasting Text Sequences for Language Model Pretraining. *CoRR* abs/2102.08473 (2021). arXiv:2102.08473 https://arxiv.org/abs/2102.08473

[44] Bhaskar Mitra, Fernando Diaz, and Nick Craswell. 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 2017, Perth, Australia, April 3-7, 2017*, Rick Barrett, Rick Cummings, Eugene Agichtein, and Evgeniy Gabrilovich (Eds.). ACM, 1291–1299. https://doi.org/10.1145/3038912.3052579

[45] Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage Re-ranking with BERT. *CoRR* abs/1901.04085 (2019). arXiv:1901.04085 http://arxiv.org/abs/1901.04085

[46] Greg Pass, Abdur Chowdhury, and Cayley Torgeson. 2006. A picture of search. In *Proceedings of the 1st International Conference on Scalable Information Systems, Infoscale 2006, Hong Kong, May 30-June 1, 2006 (ACM International Conference Proceeding Series, Vol. 152)*, Xiaohua Jia (Ed.). ACM, 1. https://doi.org/10.1145/1146847.1146848

[47] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (Eds.). 8024–8035. https://proceedings.neurips.cc/paper/2019/hash/bdbca288fee7f92f2bfa9f7012727740-Abstract.html

[48] Chen Qu, Chenyan Xiong, Yizhe Zhang, Corby Rosset, W. Bruce Croft, and Paul Bennett. 2020. Contextual Re-Ranking with Behavior Aware Transformers. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, Jimmy Huang, Yi Chang, Xueqi Cheng, Jaap Kamps, Vanessa Murdock, Ji-Rong Wen, and Yiqun Liu (Eds.). ACM, 1589–1592. https://doi.org/10.1145/3397271.3401276

[49] Stephen E. Robertson and Hugo Zaragoza. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Found. Trends Inf. Retr.* 3, 4 (2009), 333–389. https://doi.org/10.1561/1500000019

[50] Xuehua Shen, Bin Tan, and ChengXiang Zhai. 2005. Context-sensitive information retrieval using implicit feedback. In *SIGIR 2005: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Salvador, Brazil, August 15-19, 2005*, Ricardo A. Baeza-Yates, Nivio Ziviani, Gary Marchionini, Alistair Moffat, and John Tait (Eds.). ACM, 43–50. https://doi.org/10.1145/1076034.1076045

[51] Alessandro Sordoni, Yoshua Bengio, Hossein Vahabi, Christina Lioma, Jakob Grue Simonsen, and Jian-Yun Nie. 2015. A Hierarchical Recurrent Encoder-Decoder for Generative Context-Aware Query Suggestion. In *Proceedings of the 24th ACM International Conference on Information and Knowledge Management, CIKM 2015, Melbourne, VIC, Australia, October 19 - 23, 2015*, James Bailey, Alistair Moffat, Charu C. Aggarwal, Maarten de Rijke, Ravi Kumar, Vanessa Murdock, Timos K. Sellis, and Jeffrey Xu Yu (Eds.). ACM, 553–562. https://doi.org/10.1145/2806416.2806493

[52] Zhan Su, Zhicheng Dou, Yutao Zhu, Xubo Qin, and Ji-Rong Wen. 2021. Modeling Intent Graph for Search Result Diversification. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*, Fernando Diaz, Chirag Shah, Torsten Suel, Pablo Castells, Rosie Jones, and Tetsuya Sakai (Eds.). ACM, 736–746. https://doi.org/10.1145/3404835.3462872

[53] Yonglong Tian, Dilip Krishnan, and Phillip Isola. 2020. Contrastive Multiview Coding. In *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XI (Lecture Notes in Computer Science, Vol. 12356)*, Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (Eds.). Springer, 776–794. https://doi.org/10.1007/978-3-030-58621-8_45

[54] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (Eds.). 5998–6008. https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html

[55] Tongzhou Wang and Phillip Isola. 2020. Understanding Contrastive Representation Learning through Alignment and Uniformity on the Hypersphere. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020,*

13-18 July 2020, Virtual Event (Proceedings of Machine Learning Research, Vol. 119). PMLR, 9929–9939. http://proceedings.mlr.press/v119/wang20k.html

[56] Taesun Whang, Dongyub Lee, Dongsuk Oh, Chanhee Lee, Kijong Han, Dong-hun Lee, and Saebyeok Lee. 2020. Do Response Selection Models Really Know What's Next? Utterance Manipulation Strategies for Multi-turn Response Selection. *CoRR* abs/2009.04703 (2020). arXiv:2009.04703 https://arxiv.org/abs/2009.04703

[57] Ryen W. White, Paul N. Bennett, and Susan T. Dumais. 2010. Predicting short-term interests using activity-based search context. In *Proceedings of the 19th ACM Conference on Information and Knowledge Management, CIKM 2010, Toronto, Ontario, Canada, October 26-30, 2010*, Jimmy Huang, Nick Koudas, Gareth J. F. Jones, Xindong Wu, Kevyn Collins-Thompson, and Aijun An (Eds.). ACM, 1009–1018. https://doi.org/10.1145/1871437.1871565

[58] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. HuggingFace's Transformers: State-of-the-art Natural Language Processing. *CoRR* abs/1910.03771 (2019). arXiv:1910.03771 http://arxiv.org/abs/1910.03771

[59] Zhuofeng Wu, Sinong Wang, Jiatao Gu, Madian Khabsa, Fei Sun, and Hao Ma. 2020. CLEAR: Contrastive Learning for Sentence Representation. *CoRR* abs/2012.15466 (2020). arXiv:2012.15466 https://arxiv.org/abs/2012.15466

[60] Biao Xiang, Daxin Jiang, Jian Pei, Xiaohui Sun, Enhong Chen, and Hang Li. 2010. Context-aware ranking in web search. In *Proceeding of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2010, Geneva, Switzerland, July 19-23, 2010*, Fabio Crestani, Stéphane Marchand-Maillet, Hsin-Hsi Chen, Efthimis N. Efthimiadis, and Jacques Savoy (Eds.). ACM, 451–458. https://doi.org/10.1145/1835449.1835525

[61] Chenyan Xiong, Zhuyun Dai, Jamie Callan, Zhiyuan Liu, and Russell Power. 2017. End-to-End Neural Ad-hoc Ranking with Kernel Pooling. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Shinjuku, Tokyo, Japan, August 7-11, 2017*, Noriko Kando, Tetsuya Sakai, Hideo Joho, Hang Li, Arjen P. de Vries, and Ryen W. White (Eds.). ACM, 55–64. https://doi.org/10.1145/3077136.3080809

[62] Hu Xu, Bing Liu, Lei Shu, and Philip S. Yu. 2019. BERT Post-Training for Review Reading Comprehension and Aspect-based Sentiment Analysis. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, Jill Burstein, Christy Doran, and Thamar Solorio (Eds.). Association for Computational Linguistics, 2324–2335. https://doi.org/10.18653/v1/n19-1242

[63] Yujia Zhou, Zhicheng Dou, Bingzheng Wei, Ruobing Xie, and Ji-Rong Wen. 2021. Group based Personalized Search by Integrating Search Behaviour and Friend Network. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*, Fernando Diaz, Chirag Shah, Torsten Suel, Pablo Castells, Rosie Jones, and Tetsuya Sakai (Eds.). ACM, 92–101. https://doi.org/10.1145/3404835.3462918

[64] Yujia Zhou, Zhicheng Dou, and Ji-Rong Wen. 2020. Encoding History with Context-aware Representation Learning for Personalized Search. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, Jimmy Huang, Yi Chang, Xueqi Cheng, Jaap Kamps, Vanessa Murdock, Ji-Rong Wen, and Yiqun Liu (Eds.). ACM, 1111–1120. https://doi.org/10.1145/3397271.3401175

[65] Yujia Zhou, Zhicheng Dou, and Ji-Rong Wen. 2020. Enhancing Re-finding Behavior with External Memories for Personalized Search. In *WSDM '20: The Thirteenth ACM International Conference on Web Search and Data Mining, Houston, TX, USA, February 3-7, 2020*, James Caverlee, Xia (Ben) Hu, Mounia Lalmas, and Wei Wang (Eds.). ACM, 789–797. https://doi.org/10.1145/3336191.3371794

[66] Yutao Zhu, Kun Zhou, Jian-Yun Nie, Shengchao Liu, and Zhicheng Dou. 2021. Neural Sentence Ordering Based on Constraint Graphs. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*. AAAI Press, 14656–14664. https://ojs.aaai.org/index.php/AAAI/article/view/17722

[67] Chengxu Zhuang, Alex Lin Zhai, and Daniel Yamins. 2019. Local Aggregation for Unsupervised Learning of Visual Embeddings. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*. IEEE, 6001–6011. https://doi.org/10.1109/ICCV.2019.00610