

Query-Oriented Data Augmentation for Session Search

Haonan Chen , Zhicheng Dou , *Member, IEEE*, Yutao Zhu , and Ji-Rong Wen , *Senior Member, IEEE*

Abstract—Modeling contextual information in a search session has drawn more and more attention when understanding complex user intents. Recent methods are all data-driven, i.e., they train different models on large-scale search log data to identify the relevance between search contexts and candidate documents. The common training paradigm is to pair the search context with different candidate documents and train the model to rank the clicked documents higher than the unclicked ones. However, this paradigm neglects the symmetric nature of the relevance between the session context and document, i.e., the clicked documents can also be paired with different search contexts when training. In this work, we propose query-oriented data augmentation to enrich search logs and empower the modeling. We generate supplemental training pairs by altering the most important part of a search context, i.e., the current query, and train our model to rank the generated sequence along with the original sequence. This approach enables models to learn that the relevance of a document may vary as the session context changes, leading to a better understanding of users’ search patterns. We develop several strategies to alter the current query, resulting in new training data with varying degrees of difficulty. Through experimentation on two extensive public search logs, we have successfully demonstrated the effectiveness of our model.

Index Terms—Query-oriented data augmentation, session search, document ranking.

I. INTRODUCTION

AS SEARCH intents continue to grow in complexity, and the search behavior of users has undergone significant changes, transitioning from the use of single queries to engaging in multiple interactions with search engines. These interactions, including the queries issued and the documents clicked, form a search session. It has been shown that the information of a search session’s context can facilitate the comprehension of the actual search intent.

These years, many neural approaches have been proposed to model the session sequence and rank the candidate documents. These models aim to extract valuable information from the

Manuscript received 1 November 2023; revised 15 May 2024; accepted 22 June 2024. Date of publication 26 June 2024; date of current version 27 September 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 62272467, in part by the Public Computing Cloud, Renmin University of China, and in part by the fund for building world-class universities (disciplines) of Renmin University of China. Recommended for acceptance by S. Whang. (*Corresponding author: Zhicheng Dou.*)

The authors are with the Gaoling School of Artificial Intelligence, Engineering Research Center of Next-Generation Intelligent Search and Recommendation, Ministry of Education, Renmin University of China, Beijing 100872, China (e-mail: hnchen@ruc.edu.cn; dou@ruc.edu.cn; ytzhu@ruc.edu.cn; jrwen@ruc.edu.cn).

Digital Object Identifier 10.1109/TKDE.2024.3419131

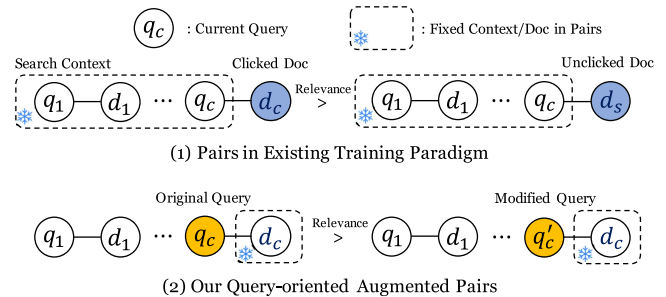


Fig. 1. An illustration of our augmented training pairs. The existing training paradigm constructs training samples by pairing different candidate documents with a fixed search context, while we pair fixed clicked document with the original search context and the one with the modified current query.

search context to predict users’ search intents. For instance, some models employed recurrent neural networks (RNNs) to model search behaviors sequentially within a session [1], [2]. Most recently, sophisticated pre-trained language models (PLMs), e.g., BERT [3] and BART [4], have also been applied to model contextual user behaviors and calculate ranking scores. All these models are trained on the search log data with each sample organized as a \langle search context, candidate document \rangle pair. As shown in the upper side of Fig. 1, during training, models learn to predict higher relevance scores for clicked documents and lower scores for unclicked documents [4], [5], [6]. While this training paradigm is intuitive and effective, it neglects an important fact —the relevance between the search context and candidate document is symmetric.

Let us analyze the relevance of a positive pair: For the search context, the clicked document can fulfill the search intent more effectively than others. This has been considered in the existing training paradigm (as shown in the upper part of Fig. 1). In contrast, for the clicked document, the search context should also be the one that matches its content the most. This aspect is unfortunately missed by existing methods, resulting in insufficient learning. To put it another way, existing methods are not able to teach the models that the relevance of a document could be different when the session context changes. Consider the following scenario: Suppose a user’s current query is “Artificial Intelligence” and their previous search was for “Machine Learning Algorithms”. In this situation, the user likely seeks information about AI algorithms or related topics. However, if the user’s previous search was for “Job Opportunities in Tech”, or if their current query changes to “Online Courses on

TABLE I
PERFORMANCE OF COCA WITH DIFFERENT QUERIES MISSING IN THE
TRAINING DATA

| Metric | w/o. q_c | | w/o. q_n & d_n | | w/o. q_{n-1} & d_{n-1} | | COCA |
|--------|------------|---------|--------------------|--------|----------------------------|--------|---------------|
| MAP | 0.4751 | -13.62% | 0.5452 | -0.88% | 0.5465 | -0.64% | 0.5500 |
| MRR | 0.4860 | -13.23% | 0.5555 | -0.83% | 0.5566 | -0.63% | 0.5601 |
| N@3 | 0.4631 | -15.46% | 0.5416 | -1.14% | 0.5429 | -0.90% | 0.5478 |
| N@10 | 0.5450 | -11.53% | 0.6103 | -0.93% | 0.6120 | -0.65% | 0.6160 |

Supposing q_c denotes the current query. The query sequence that contains n historical queries is $\{q_1, \dots, q_{n-1}, q_n, q_c\}$. The corresponding clicked documents of the historical queries are $\{d_1, \dots, d_{n-1}, d_n\}$. We remove the current query (q_c) and the last two query-document pairs (q_n & d_n , q_{n-1} & d_{n-1}), respectively. “NDCG@ k ” is referred to as “N@ k ”.

Programming”, the relevance ranking of the candidate documents should be different. The problem is even more severe for the PLM-based methods, as they always constructed training sequences by fixed search context and different candidate documents.

To address this problem, we propose to augment the training data by considering search context alterations (as shown in the lower part of Fig. 1), i.e., fixing the clicked document and identifying possible alternations of the search context to construct more training pairs. In general, a search context consists of three components: historical queries, corresponding clicked documents, and the current query. The decision to modify the current query is based on two key observations: (1) The current query is the most effective information in the search context to understand the user’s search intent. To support this, we conduct a preliminary experiment based on a well-known baseline COCA [7]: To evaluate the impact of the current query and the historical query-document pairs on the performance of COCA, we proceed by removing both the current query and the last two query-document pairs from the session history, respectively. The performance is evaluated in terms of Mean Average Precision (MAP), Mean Reciprocal Rank (MRR), and Normalized Discounted Cumulative Gain (NDCG) at position k (NDCG@ k), where k takes values from the set $\{3, 10\}$. The results shown in Table I indicate that the absence of the current query has the most significant impact on ranking performance. (2) Context-aware ranking models concentrate on representing the entire search behavior sequence, which may weaken its modeling of the current query. Enhancing the model’s ability to capture more fine-grained information from the current query is important.

More specifically, we propose a Query-oriented Data Augmentation method for Session Search (QASS). We generate new training samples by altering the current query to complement real-world search logs and facilitate model learning. In the training process, the generated samples serve as **negative** samples, given that the original samples are directly observed in the search log. Specifically, we consider altering the current query at two levels: (1) Term-level Modification. By changing (i.e., masking, replacing, or adding) some terms within the current query, the model can learn the impact of subtle variations in the query. (2) Query-level Replacement. We directly replace the current query with some queries mining from the search

log. In this process, we also consider the difficulty of query modification, inspired by recent studies in dense retrieval [8], [9], [10], where a mixture of negative documents in different difficulties can make the training process more stable [11], [12]. In particular, all samples generated by randomly sampled queries from the search log are considered “easy” negative samples. Then, we replace the current query with its historical queries in the search context. The generated samples are treated as “medium” negative samples because the replaced query is close to the current query (they appear in the same session). Similarly, the queries augmented by term-level modification are also used as “medium” negative samples. Finally, we mine some ambiguous queries of the current query by some heuristics. We use the generated samples as “hard” negative samples because the ambiguous queries are even closer to the current query than the historical ones. Through these strategies, we can generate negative sequences of varying difficulty with respect to the current queries. Our experiments on two public search logs (AOL [13] and Tiangong-ST [14]) demonstrate that QASS significantly outperforms existing models, indicating the effectiveness of our proposed query-oriented data augmentation method.

In summary, the contributions of the paper are as follows:

- (1) We identify the problem in current training paradigms, where the relevance from the perspective of the clicked document is overlooked. We propose to generate query-oriented data for session search by altering the current query, thereby enriching search logs and enabling models to learn users’ search patterns more comprehensively. It is the first time that negative sampling is performed on the query side rather than the document side for session search.
- (2) We develop various methods to generate negative training samples with varying difficulty. Different score margins are applied to identify their difficulty and coordinate these augmented pairs.
- (3) We design a heuristic for mining ambiguous queries, ensuring their similarity to the current query by considering the ranking of the clicked document in other sessions. Experimental results validate that these queries are more informative than other mined queries for learning users’ search intents.

II. RELATED WORK

A. Data Augmentation for Ranking

There are already some research works that designed various data augmentation strategies to facilitate information retrieval models [15], [16], [17], [18], [19], [20], [21], [22], [23], [24]. Moreover, data augmentation techniques can be applied to generate additional training data for document ranking models [25], [26], [27]. Through the generated synthetic data, the model can learn from a more diverse set of examples, which can improve its ability to rank documents effectively. Data augmentation can help in addressing issues like data sparsity, overfitting, and generalization, leading to better performance in document ranking tasks. For example, Li et al. [25] proposed an attention-based sequence-to-sequence model for POI recommendation. Specifically, they incorporated spatial and temporal information to augment the check-in datasets. Subsequently, an

encoder-decoder model is applied to learn the missing check-in. Yu et al. [26] designed an informative data generation model to address the data imbalance problem in learning to rank. Based on the adversarial autoencoder, they disentangled the relevance information from the latent representation and exploited query information to regularize the prior distribution. Qiu et al. [27] proposed a Learning to Augment (LTA) method to resolve the data imbalance issue. They proposed to generate informative data using a Gaussian Mixture Variational Autoencoder. Furthermore, they applied a teacher model to learn how to optimize their generation policy based on reinforcement learning. In Named Entity Recognition (NER) task, there are also some works utilizing data augmentation techniques to train better models [28], [29]. For example, COSINER [28] replaced entity mentions with alternatives, considering available training data and the contexts in which entities commonly occur.

B. Modeling Search Sessions

Some early works have resorted to statistical methods to study contextual information of search sessions. Shen et al. [30] employed context-sensitive retrieval-based algorithms that rely on statistical language models to effectively incorporate session context. Bennett et al. [31] demonstrated that a combination of historical behaviors and short-term behaviors can benefit the understanding of search intents in a statistical manner. White et al. [32] mined data from similar search sessions conducted by other users to identify documents that would be highly relevant. Van Gysel et al. [33] studied lexical query modeling in session search. They pointed out that context-aware methods are more effective than traditional query terms re-weighting. These traditional approaches have achieved great success. However, restricted by their non-parametric and statistic-based nature, they are not able to model user behaviors thoroughly.

The advent of deep learning has led to the emergence of numerous neural context-aware ranking models in recent years. Ahmad et al. [1] encoded sequential historical behaviors and candidate documents with RNNs and computed the ranking score based on the matching of their representations. They [2] complemented their work using attention mechanisms and jointly learning the ranking task and the query suggestion task. HBA-Transformer [34] concatenated the session sequence and used the popular PLM BERT as the encoder. They also designed a hierarchical behavior-aware module to capture interaction-based information. Zuo et al. [6] modeled multi-level historical query changes to obtain representations of sessions from multiple aspects. RICR [5] integrated representation and interaction. They employed Recurrent Neural Networks (RNNs) to effectively capture session sequences. This modeling technique was then utilized to augment the word-level interaction between the current query and candidate documents. COCA [7] employed data augmentation and contrastive learning methods to pre-train an enhanced BERT encoder for effectively modeling session sequences. HEXA [35] utilized heterogeneous graphs to capture information within a session and from other sessions. DCL [11] designed a curriculum learning framework that learns the matching between the session context and documents from easy to hard. Since it is a learning framework rather than a specific model

and it re-samples the negative documents, we will omit it in our comparisons. ASE [4] used a decoder and several generation tasks specifically designed for session search to enhance the ability of the encoder.

Our approach to data augmentation differs from that of COCA [7] in a significant way. While Zhu et al. treated the augmented sequences as positive examples of contrastive learning, we consider our generated sequences as negative examples in pair-wise training. This distinction arises from our focus on altering the most crucial behavior, namely the current query, rather than making slight changes to the session context, as suggested by Zhu et al. They argue that these minor changes should not affect the sequence representation significantly. In contrast, we focus on altering the most important behavior, i.e., the current query, and we believe our augmentation strategies (e.g., replacing it with a random query) should make the search intent change, even under the same session history. Our objective is to complement the existing training paradigm by generating alternate behavior sequences from the query side, whereas COCA aims to pre-train the encoders through an entirely separate training stage.

III. PROPOSED MODEL: QASS

In this work, we propose to pair various search contexts with the same clicked document, allowing the model to learn their distinct relevance. We choose to alter the current query to generate new search contexts, thus our approach can be treated as a query-oriented data augmentation method. Our data augmentation strategies involve altering terms in the current query (masking, replacing, or adding), and replacing the entire query with other queries mined from the search log (random queries, historical queries, and ambiguous queries). These mined/generated queries form negative training pairs with different difficulty levels, which can help stabilize the training process.

A. Important Notations

Before introducing our model, we first explain some important notations of session search. The target of this task is to model sequential user behaviors in a session to understand the user's search intent and rank the clicked documents as high as possible. We denote the user's queries in the session history H as $[q_1, q_2, \dots, q_n]$, and their corresponding clicked documents as $[d_1, d_2, \dots, d_n]$.¹ The current query is denoted as q_c , and its candidate document set is denoted as D . Furthermore, the clicked documents in D are denoted as d_c , and the skipped documents are d_s ($d_c \cup d_s = D$, $d_c \cap d_s = \emptyset$).

A context-aware document ranking model attempts to score $d \in D$ based on H and q_c as follows:

$$P(d) = P(H, q_c, d). \quad (1)$$

Existing training paradigm optimizes the model to rank $d_c \in d_c$ higher than $d_s \in d_s$, i.e., $P(H, q_c, d_c) > P(H, q_c, d_s)$. We enrich search logs with the data generated by altering the current query q_c to q'_c and let the model learn $P(H, q_c, d_c) > P(H, q'_c, d_c)$.

¹Following previous works [4], [7], we only use the first clicked document of each historical query.

TABLE II
EXAMPLES OF QUERIES GENERATED BY DIFFERENT DATA
AUGMENTATION STRATEGIES

| Level | Type | Query | Difficulty |
|-------|------------|------------------------------------|------------|
| - | Original | burlington wisconsin | - |
| Term | Mask | burlington [term_del] | Medium |
| Term | Replace | burlington becker | Medium |
| Term | Add | school burlington wisconsin | Medium |
| Query | Random | laugh factory nyc | Easy |
| Query | Historical | racine county history | Medium |
| Query | Ambiguous | burlington county jobs | Hard |

Search session:

H : [q_1 : racine county history, d_1 : racine county wi home]

q_c : burlington wisconsin, d_c : burlington wi official website

We take an actual session from the AOL search log as an example: H is {“racine county history” (q_1), “racine county wi home” (d_1)}, q_c is “burlington wisconsin”, and d_c is “burlington wi official website”. Texts in bold indicate unchanged terms.

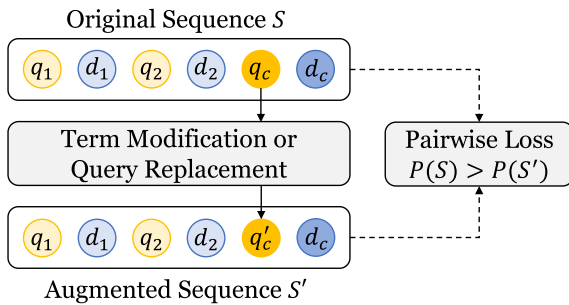


Fig. 2. Illustration of QASS. The current query q_c of the original user behavior sequence S is altered to construct the augmented sequence S' . The clicked document d_c is hypothesized to be more relevant to the original search context than the altered context (i.e., $P(S) > P(S')$).

B. Model Overview

In this section, we will provide a concise overview of the structure of our model. Our model is comprised of two stages:

(1) *Query-oriented Data Augmentation*: We aim to generate query-oriented data pairs to enrich the search log by altering the current query q_c . Sequences constructed from generated/mined queries are considered negative sequences compared to the actually observed ones. Specifically, as shown in Table II, we employ term-level modification and query-level replacement to generate various queries, which are then used to construct negative training pairs with different difficulties. For term-level modification, we change (i.e., mask, replace, or add) some terms of q_c , which enable QASS to learn fine-grained matching signals. For query-level replacement, we mine some queries from search logs (including random queries, historical queries, and ambiguous queries) and replace q_c with these queries, helping our model extract search intent at a higher level.

(2) *Jointly Training on All Data Pairs*: As illustrated in Fig. 2, with the generated and original data pairs ready, we use the pre-trained language model BERT to score all sequences and apply a pair-wise loss function to optimize the model. To identify training pairs of varying difficulty, we apply different score margins for them.

C. Query-Oriented Data Augmentation Strategies

As shown in Fig. 2 and Table II, we alter q_c at both the term and query levels. These modifications serve distinct purposes: the term-level modification introduces slight variations to the original query, which enhances the model’s capability to capture fine-grained interactions. On the other hand, query-level replacement directly changes the entire query, requiring the model to understand the query from a higher view.²

1) *Term-Level Modification*: Some existing works in Natural Language Processing (NLP) have used word-level augmentation to make the representations of sentences more robust [36], [37], [38]. A recent model for session search, COCA [7], also generates sequences for contrastive learning by masking terms within session sequence. Inspired by these studies, we use the term-level modification to the current query to construct additional data.

Most queries in actual search logs primarily comprise keywords [4], namely, they contain fewer than three terms (approximately 72.5% in AOL search log). Thus, subtle term-level modifications of q_c may result in noticeable changes in search intent. Supposing $q_c = \{w_1, \dots, w_t\}$, we design three term-level modification strategies:

(1) *Term Masking*: We randomly select an index k ranging in $[0, t]$, and mask the word w_k by replacing it with “[term_del]” (similar to the [MASK] token in BERT) as

$$q'_c = [w_1, \dots, w_{k-1}, [\text{term_del}], w_{k+1}, \dots, w_t]. \quad (2)$$

For example, q'_c “burlington [term_del]” is generated from q_c “burlington wisconsin” by term masking. Obviously, q'_c lacks the important information “wisconsin”, which makes the current search intent different, even with the same session history H . Thus, the generated sequence $[H, q'_c]$ should be less relevant to the clicked document d_c .

(2) *Term Replacing*: We first randomly select an index k ranging in $[0, t]$. Then, we randomly mine a term w_r from the training data of search log and replace w_k with w_r as

$$q'_c = [w_1, \dots, w_{k-1}, w_r, w_{k+1}, \dots, w_t], \quad (3)$$

where w_r should be different from w_k . Similar to term masking, this strategy may also change the search intent of the current query.

(3) *Term Adding*: First, we randomly select an index k ranging in $[0, t + 1]$. We then randomly mine a term w_a from the search log and insert it at the position k as

$$q'_c = [w_1, \dots, w_{k-1}, w_a, w_k, \dots, w_t]. \quad (4)$$

The added word may introduce a more complex search intent to q_c .

2) *Query-Level Replacement*: In addition to term-level augmentations, we alter the whole query from a higher view, i.e., mine some queries to directly replace q_c . Specifically, we mine three kinds of queries from the search log:

²Since QASS strives to emphasize that changes in the current query have a significant influence on the search intent of the search sequence even under the same history, we only alter the queries that have historical queries.

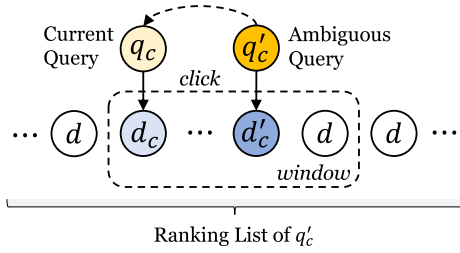


Fig. 3. Illustration of mining the ambiguous queries. We first use a ranking model to obtain a ranking list of all documents for each query. Then a window of negative documents is sampled around each query’s clicked document. If d_c is in the window of a query q_c , this query q'_c is an ambiguous query of q_c . The closer d_c is to the clicked document of q'_c (d'_c), the more ambiguous q'_c is to q_c .

(1) *Random Query*: The most straightforward way is to replace q_c with a randomly mined query from the search log. The sequences generated by this strategy may contain much noise, which can make our model more robust.

(2) *Historical Query*: A user’s search behaviors in the same session usually serve for a main search intent [2], [4], [7], [39], [40], while queries in the same session can usually represent different subtle intents. For example, as shown in Table II, q_1 and q_c both try to find the websites of certain cities in Wisconsin. However, q_1 tries to find the history of Racine county, whereas q_c wants to learn about Burlington. Thus, these two queries share some common intents but are still not identical, which makes q_1 a good replacement for q_c to train the model. Compared to the random query, the historical queries are more similar to the current query, so the generated training pairs are more difficult.

(3) *Ambiguous Query*: Inspired by a recent study on mining ambiguous documents for dense retrieval [8], we propose to mine some ambiguous queries to replace the current query. Building on this insight, our objective is to mine negative queries that strike a balance between being too challenging (possibly false negatives) and too easy (uninformative) to replace the current query. Intuitively, if the clicked document d_c under the current query q_c is ranked around the clicked document d'_c under another query q'_c by a ranking model, we can treat q'_c as an ambiguous query of q_c . This is because their respective clicked documents are very similar, indicating a potential overlap in search intent. As shown in Fig. 3, we first rank all documents for each query in the search log so that each query has a complete document ranking list. Subsequently, for each clicked document associated with a query, we sample a window of documents using the clicked document as the center. The documents within the window are highly relevant to the clicked document. Finally, in order to mine ambiguous queries for q_c , we choose the queries whose document window contains d_c . Following [11], to get the ranking list of each query with good quality and efficiency, we train a dense retriever based on the BERT [3] representation of queries and documents with the dot-product as the relevance score, namely

$$P(q, d) = \text{BERT}(q)_{[\text{CLS}]} \cdot \text{BERT}(d)_{[\text{CLS}]}. \quad (5)$$

We use in-batch negatives for training and FAISS [41] to achieve fast retrieval.

As yet, we have generated q'_c with various strategies. Then, we use q'_c to replace q_c and generate new search behavior sequences. Finally, they are combined with the clicked document d_c to form new negative training pairs. Based on the similarity between q'_c and q_c , we can categorize the difficulty of the generated negative pairs into three levels: (1) The random queries obtained in query-level replacement are used to form the least difficult negative pairs. This is intuitive since these queries are randomly sampled from other search sessions, which may have totally different search intents. (2) The historical queries and those generated by term-level modifications are used to construct negative pairs of medium difficulty. These altered or replaced queries may share some common intents with q_c yet are still different. Thus we consider them both as “medium negative” pairs for simplicity. (3) The ambiguous queries are used to form hard negative pairs. According to our heuristic of mining ambiguous queries, their search intent may be very close to the current query, so the generated negative pairs are the most difficult to distinguish.

D. Scoring and Training

1) *Sequence Scoring With BERT*: Pre-trained language models, such as BERT [3], are widely used in the task of session search [4], [7], [34]. We use BERT as the backbone model for a fair comparison with existing methods. Following [7], we first use some special tokens to concatenate behaviors in H : $I_H = q_1[\text{EOS}]d_1[\text{EOS}] \cdots q_n[\text{EOS}]d_n$, where the “[EOS]” token indicates the end of a query/document. Then, we append the current query q_c and the candidate document d to I_H : $I_S = [\text{CLS}]I_H[\text{EOS}]q_c[\text{EOS}][\text{SEP}]d[\text{EOS}][\text{SEP}]$, where [SEP] is the separator to identify the candidate document d , [CLS] is used for computing the sequence representation. Then we use BERT to compute the ranking score of the sequence $[H, q_c, d]$

$$P(H, q_c, d) = \text{MLP}(\text{BERT}_{[\text{CLS}]}(I_S)), \quad (6)$$

where MLP is a multi-layer perceptron used as a classifier to compute the ranking score.

2) *Jointly Optimizing All Training Pairs*: Taking the training of a query q_c as an example, the (search context, clicked document) pairs that are actually observed in the search log are used to construct the positive sequences, i.e., $S_p = [H, q_c, d_c]$. The training objective is to predict the ranking score of the positive sequence S_p to be higher than that of the negative sequence S_n . A standard pair-wise ranking function is often employed to optimize the model

$$\mathcal{L}(S_p, S_n) = \max(0, m - P(S_p) + P(S_n)), \quad (7)$$

where m is a hyperparameter, representing the minimum acceptable score margin between S_p and S_n .

In the original datasets, the skipped documents d_s are used to construct negative training pairs, i.e., $S_n = [H, q_c, d_s]$. They are also valuable for learning document ranking, so we keep them for training as follows:

$$\mathcal{L}_{\text{op}}(q_c) = \sum_{(d_c, d_s) \in D} \max(0, m_{\text{op}} - P(S_p) + P(H, q_c, d_s)),$$

where m_{op} is a margin hyperparameter, which is usually set to 1.0 for binary classification problems.

We also apply our data augmentation strategies to generate negative sequences, in which the current query q_c is altered to q'_c . These negative sequences can be represented as $S_n = [H, q'_c, d_c]$. The training objective is defined as

$$\mathcal{L}_{\text{cp}}(q_c) = \sum_{d_c \in D} \sum_{q'_c} \max(0, m_{\text{cp}} - P(S_p) + P(H, q'_c, d_c)),$$

where m_{cp} is the margin hyperparameter for constructed pairs.

To identify sequences generated by different strategies, we apply different margins during the training process. The margin can control the distance between S_p and S_n , i.e., the smaller the margin is, the more relevant q'_c is to d_c . Following the discussion of difficulty before, m_{cp} have three levels: m_{rq} (random queries) $> m_{\text{th}}$ (term-level modification and historical queries) $> m_{\text{aq}}$ (ambiguous queries). The intuition here is that the higher the ranking of d_c , the closer q'_c and d_c , i.e., the smaller the margin. The influence of the score margins is studied in Section V-D2. Specifically, for m_{aq} , we slightly tune each ambiguous query's margin based on the position of d_c in that ambiguous query's ranking list: $m_{\text{aq}} = (\text{pos}(d_c)/w_{\text{size}}) * 2 * \bar{m}_{\text{aq}}$, \bar{m}_{aq} is the average margin of ambiguous queries, and w_{size} is the size of the window of negative documents.

In summary, for a query q_c , we train the original data pairs and the augmented data pairs jointly

$$\mathcal{L}(q_c) = \mathcal{L}_{\text{cp}}(q_c) + \mathcal{L}_{\text{op}}(q_c). \quad (8)$$

IV. EXPERIMENTS

A. Datasets and Evaluation

1) *Datasets*: Following existing works [4], [5], [6], [7], we use two large-scale public search logs AOL [13] and Tiangong-ST [14] to compare the performance of QASS and baselines. In accordance with previous research, we opt for utilizing only the document title to ensure efficiency. The statistical data derived from the analysis of these two search logs is illustrated in Table III.

To use the AOL search log, we utilize the dataset provided by the authors of CARS [2]. It is important to note that every query in this log has a minimum of one click that meets the user's satisfaction criteria. In both the training and validation sets, there are five candidate documents for each query, while the testing set contains 50 candidate documents.

The Tiangong-ST search log was obtained from a Chinese commercial search engine. It comprises user sessions spanning a duration of 18 days, including their top 10 search results and click information. However, certain queries and documents in the Tiangong-ST dataset are incomplete, and to address this, we have used the placeholders “[empty_q]” and “[empty_d]” to indicate missing content. To create our test set, we have selected 2,000 sessions from the entire dataset. These sessions were chosen based on the condition that their last query possesses human relevance labels. For the remaining sessions, we have allocated the last 2,000 sessions as the validation set, while the remaining sessions form the training set. It is important to note that only

TABLE III
STATISTICS OF TWO SEARCH LOGS

| AOL | Training | Validation | Testing |
|------------------------|----------|------------|---------|
| # session | 219,748 | 34,090 | 29,369 |
| # query | 566,967 | 88,021 | 76,159 |
| avg. session length | 2.58 | 2.58 | 2.59 |
| avg. query length | 2.86 | 2.85 | 2.9 |
| avg. document length | 7.27 | 7.29 | 7.08 |
| # candidate per query | 5 | 5 | 50 |
| avg. # click per query | 1.08 | 1.08 | 1.11 |
| Tiangong-ST | Training | Validation | Testing |
| # session | 143,155 | 2,000 | 2,000 |
| # query | 344,806 | 5,026 | 6,420 |
| avg. session length | 2.41 | 2.51 | 3.21 |
| avg. query length | 2.89 | 1.83 | 3.46 |
| avg. document length | 8.25 | 6.99 | 9.18 |
| # candidate per query | 10 | 10 | 10 |
| avg. # click per query | 0.94 | 0.53 | 3.65 |

the last query in each session of the test set has an annotated relevance score, ranging from 0 to 4. This approach aligns with the methodology employed in recent studies [4], [11] as well as the original paper introducing this dataset [14]. Consequently, during testing, we will exclusively utilize queries that possess relevance labels. For more comprehensive information regarding this dataset, please refer to the original paper by Tiangong [14].

2) *Evaluation*: We evaluate the performance of QASS and baseline models using three metrics: Mean Average Precision (MAP), Mean Reciprocal Rank (MRR), and Normalized Discounted Cumulative Gain (NDCG) at position k (NDCG@ k), where k takes values from the set $\{1, 3, 5, 10\}$. These metrics provide a comprehensive assessment of the effectiveness of the models

$$\text{MAP} = \frac{1}{N} \sum_{i=1}^N \frac{1}{d_i^c} \sum_{j=1}^{d_i^c} \frac{j}{p_i^j},$$

$$\text{MRR} = \frac{1}{N} \sum_{i=1}^N \frac{1}{p_i^1},$$

$$\text{DCG}(\sigma) = \frac{1}{N} \sum_{i=1}^N \sum_{j=1}^{d_i^c} \frac{1}{\log(1 + p_i^j)},$$

$$\text{NDCG} = \frac{\text{DCG}(\sigma_{\text{actual}})}{\text{DCG}(\sigma_{\text{optimal}})},$$

where N represents the total number of queries in the evaluation set, d_i^c denotes the number of user clicks that occurred during the i th query, p_i^j represents the position of the j th click within the ranking list of the query i , and σ is the ranking list.

Note that the Tiangong-ST dataset's relevance labels are human-annotated, hence the evaluation of MAP and MRR may not be accurate. We focus on NDCG@ k measures as suggested by the latest works [4], [35] and the original authors of this dataset [14]. Specifically, we use the tool provided by TREC to compute these metrics (trec_eval [42]).

B. Baseline Models

Following [4], [5], [6], [7], we evaluate QASS against two types of baselines:

(1) *Ad-hoc ranking models*: These models consider only the information of the current query q_c and the candidate document d , i.e., neglect the session history.

- **BM25 [43]** BM25 is a traditional ranking algorithm that calculates the relevance between the current query q_c and d using probability.
- **ARC-I [44]** utilizes convolutional neural networks (CNNs) to model both q_c and d . Then it computes these representations' semantic similarity as the relevance score.
- **ARC-II [44]** uses 2D-CNNs to obtain the fine-grained interaction-based information of q_c and d .
- **KNRM [45]** first construct a word-level interaction matrix of q_c and d . Then it computes the relevance score based on soft matching signals by kernel pooling.
- **Duet [46]** effectively evaluates the score of d_c by incorporating a combination of interaction-based and representation-based features.

(2) *Context-aware ranking models*: These models utilize the session history to understand the search intent.

- **CARS [2]** jointly optimizes ranking and query suggestions. For ranking, it utilizes a Recurrent Neural Network (RNN) and attention to model sequential session behaviors.
- **HBA-Transformers [34]** uses BERT as the encoder of the session sequence. In addition, It proposes a hierarchical behavior-aware attention module that is applied to BERT in order to extract detailed interaction-based information.
- **HQCEN [6]** attempts to mine information from multi-granularity historical query change and utilize a query change classification task to help the ranking task.
- **RICR [5]** utilizes Recurrent Neural Networks (RNNs) to encode the historical session information, followed by leveraging this encoded representation to improve the word-level performance of q_c and d .
- **BERT [3]** is a pre-trained language model which is widely used in IR tasks.
- **COCA [7]** designs some data augmentation strategies to generate data for contrastive learning. Additionally, it employs pre-training of BERT to improve the representation of the session sequence.
- **ASE [4]** uses a decoder and three generation tasks to enhance the encoding of the session sequence. These generation tasks are designed specifically for session search.

C. Model Settings

We use BERT provided by Huggingface as QASS's backbone.³ We use AdamW [47] as the optimizer, and the training batch size is set as 588. We train our model for three epochs and set the learning rate as 4e-5 with linear decayed. For Tiangong-ST, we train data pairs of term-level modification after training other pairs because changes in Chinese characters may influence the training stability.

³<https://huggingface.co/bert-base-uncased>

For each strategy of our query-oriented data augmentation, there are two kinds of hyperparameters: the number of generated sequences and the score margins. (1) For term-level modification, we generate one sequence for each term-level strategy, i.e., three sequences in total. For query-level replacement, we mine three random queries for AOL (eight for Tiangong), all historical queries in the session, and the four most ambiguous queries for AOL (five for Tiangong). (2) We set m_{rq} as 1.0, m_{th} as 0.5, and \bar{m}_{aq} as 0.2. w_{size} is set as 50.

V. EXPERIMENTAL RESULTS AND ANALYSIS

A. Overall Results

The experimental results on two search logs are presented in Table IV. It is observed that ad-hoc ranking models generally perform worse than models with contexts, highlighting the significance of incorporating session context in the modeling process. Additionally, the following observations can be made:

(1) *Our model QASS outperforms all other baselines*: It outperforms ASE, a strong baseline that enhances BART using multiple generative tasks, by approximately 2.94% in terms of NDCG@1 on the AOL search log. This significant improvement showcases the capability of our generated data to enhance the existing training paradigm and provide valuable insights into user search patterns. Furthermore, it is worth noting that the improvements achieved by QASS on the AOL search log are more significant compared to those on the Tiangong-ST set. This intriguing phenomenon was also noticed in previous works [4], [7]. We believe the possible reasons are: (i) QASS are trained on click-based search logs rather than relevance-based. Our query-oriented data augmentation is also conducted on sequences of clicked documents, which makes QASS naturally perform better on predicting click behaviors than relevance scores. (ii) The initial score on Tiangong-ST is already high. According to statistical analysis of the test set, more than 77.4% of the candidate documents have relevance ratings greater than 1, meaning they are identified as relevant (Tiangong, citation). Even the basic neural model Duet achieves an impressive NDCG@10 score of 0.8829 on this dataset. Therefore, it becomes more challenging for our model, QASS, to show significant improvements on this particular dataset.

(2) *PLM-based models generally perform better than others*: For example, the BERT-based models (COCA and QASS) and the BART-based model BART outperform RNN-based multi-task model CARS by over 20% in terms of all metrics on the AOL search log. The PLM-based models perform better than CARS even without the auxiliary task of query suggestion, which demonstrates the effectiveness of PLMs in modeling session context.

B. Ablation Studies

To study the effectiveness of our data augmentation strategies, we conducted several ablation studies on AOL as follows:

- **QASS w/o. TM**. is QASS without the strategy of Term-level Modification (TM).

TABLE IV
OVERALL RESULTS ON TWO SEARCH LOGS

| Dataset | Metric | BM25 | ARC-I | ARC-II | Duet | KNRM | CARS | HBA | RICR | HQCN | BERT | COCA | ASE | QASS |
|-------------|--------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------------|---------------|
| AOL | MAP | 0.2200 [†] | 0.3361 [†] | 0.3834 [†] | 0.4008 [†] | 0.4038 [†] | 0.4297 [†] | 0.5281 [†] | 0.5338 [†] | 0.5448 [†] | 0.5471 [†] | 0.5500 [†] | <u>0.5650[†]</u> | 0.5750 |
| | MRR | 0.2271 [†] | 0.3475 [†] | 0.3951 [†] | 0.4111 [†] | 0.4133 [†] | 0.4408 [†] | 0.5384 [†] | 0.5450 [†] | 0.5549 [†] | 0.5572 [†] | 0.5601 [†] | <u>0.5752[†]</u> | 0.5850 |
| | N@1 | 0.1195 [†] | 0.1988 [†] | 0.2428 [†] | 0.2492 [†] | 0.2397 [†] | 0.2816 [†] | 0.3773 [†] | 0.3894 [†] | 0.3990 [†] | 0.3990 [†] | 0.4024 [†] | <u>0.4144[†]</u> | 0.4266 |
| | N@3 | 0.1862 [†] | 0.3108 [†] | 0.3564 [†] | 0.3822 [†] | 0.3868 [†] | 0.4117 [†] | 0.5241 [†] | 0.5267 [†] | 0.5441 [†] | 0.5440 [†] | 0.5478 [†] | <u>0.5682[†]</u> | 0.5789 |
| | N@5 | 0.2136 [†] | 0.3489 [†] | 0.4026 [†] | 0.4246 [†] | 0.4322 [†] | 0.4542 [†] | 0.5624 [†] | 0.5648 [†] | 0.5783 [†] | 0.5818 [†] | 0.5849 [†] | <u>0.6007[†]</u> | 0.6104 |
| | N@10 | 0.2481 [†] | 0.3953 [†] | 0.4486 [†] | 0.4675 [†] | 0.4761 [†] | 0.4971 [†] | 0.5951 [†] | 0.5971 [†] | 0.6070 [†] | 0.6123 [†] | 0.6160 [†] | <u>0.6283[†]</u> | 0.6373 |
| Tiangong-ST | N@1 | 0.6029 [†] | 0.7088 [†] | 0.7131 [†] | 0.7577 [†] | 0.7560 [†] | 0.7385 [†] | 0.7612 [†] | 0.7670 [‡] | 0.7739 [‡] | 0.7488 [†] | 0.7769 | <u>0.7884</u> | 0.7955 |
| | N@3 | 0.6646 [†] | 0.7087 [†] | 0.7237 [†] | 0.7354 [†] | 0.7457 [†] | 0.7386 [†] | 0.7518 [†] | 0.7636 [‡] | 0.7682 | 0.7541 [†] | 0.7576 [‡] | <u>0.7727</u> | 0.7742 |
| | N@5 | 0.7072 [†] | 0.7317 [†] | 0.7379 [†] | 0.7548 [†] | 0.7716 [†] | 0.7512 [†] | 0.7639 [†] | 0.7740 [‡] | 0.7783 [‡] | 0.7651 [†] | 0.7703 [†] | <u>0.7839</u> | 0.7861 |
| | N@10 | 0.8541 [†] | 0.8691 [†] | 0.8732 [†] | 0.8829 [‡] | 0.8894 [‡] | 0.8837 [‡] | 0.8896 [‡] | 0.8934 [‡] | 0.8976 | 0.8890 [‡] | 0.8932 [‡] | <u>0.8996</u> | 0.9010 |

“†” denotes the result performs significantly worse than our model in paired t-test with p -value < 0.01 and “‡” denotes a p -value < 0.05 level. We highlight the the best performance in bold and the second-best one underlined.

TABLE V
PERFORMANCES OF ABLATED MODELS ON AOL SEARCH LOG

| Metric | MAP | NDCG@1 | NDCG@3 |
|-------------|---------------|--------|---------------|
| QASS (Full) | 0.5750 | - | 0.5789 |
| w/o. TM | 0.5700 | -0.87% | 0.5736 |
| w/o. RQ | 0.5706 | -0.77% | 0.5739 |
| w/o. HQ | 0.5718 | -0.56% | 0.5779 |
| w/o. AQ | 0.5699 | -0.89% | 0.5735 |

- *QASS w/o. RQ.* is QASS without the mined Random Queries (RQ) of query-level replacement, i.e., easy negatives.
- *QASS w/o. HQ.* is QASS without the Historical Queries (HQ) of query-level replacement.
- *QASS w/o. AQ.* is QASS without the Ambiguous Queries (AQ), i.e., hard negatives.

Due to the space limitation, we only present the performance of MAP, NDCG@3, and NDCG@5. From the results in Table V, we can find that all ablated models perform worse than QASS, which demonstrates the effectiveness of our data augmentation strategies. Moreover, we can see:

(1) *Term-level modification can help QASS learn subtle modeling knowledge:* We propose to change some words of q_c to construct supplemental data pairs. By this means, we try to help our model learn that subtle variations over the original query can result in changes in search intent. This can enhance the model’s capability of capturing fine-grained information from the query. After abandoning the data pairs generated by this strategy, QASS’s performance decreases by about 0.71% in terms of NDCG@1. This indicates our term-level modification strategy can help training.

(2) *Generating data pairs with random queries can make our model more robust:* We randomly mine some queries from the search log to replace q_c . These random queries may contain much noise, and the data pairs constructed by them can make our model more robust. Specifically, removing these data makes the performance of QASS drop about 0.40% in terms of NDCG@1.

(3) *Ambiguous queries are more informative than other queries:* We attempt to mine some ambiguous queries to replace q_c by tracking the ranking position of d_c in other sessions. Discarding these data results in the greatest decline of QASS.

TABLE VI
PERFORMANCES OF AMBIGUOUS QUERIES SAMPLED BY DIFFERENT STRATEGIES FROM THE NEGATIVE DOCUMENT WINDOW (WITH DIFFERENT RANKING POSITIONS OF THEIR CLICKED DOCUMENTS OR DIFFERENT REPRESENTATION MODELS) ON AOL

| Strategies | MAP | MRR | NDCG@3 | NDCG@10 |
|---------------|---------------|---------------|---------------|---------------|
| Low + BERT | 0.5740 | 0.5844 | 0.5776 | 0.6370 |
| High + BERT | 0.5710 | 0.5812 | 0.5740 | 0.6347 |
| Medium + BERT | 0.5750 | 0.5850 | 0.5789 | 0.6373 |
| Medium + BM25 | 0.5739 | 0.5840 | 0.5773 | 0.6372 |

The performance of ambiguous queries used in QASS is in bold.

This demonstrates that the data generated by ambiguous queries are the most informative ones.

C. Influence of Ambiguous Query Sampling Strategies

As illustrated in Section III-C2, we mine ambiguous queries q'_c from the search log by tracking the ranking position of d_c in other sessions. We sample a window of negative documents (w_n) around the clicked document of q'_c (i.e., d'_c), and treat the proximity between d_c and d'_c as the ambiguity of q'_c . We believe they are more informative than other generated queries, as demonstrated in the previous section. In this section, we will further study the effectiveness of ambiguous queries sampled by different strategies from w_n .

We conduct our experiments by sampling ambiguous queries with different settings. We treat the position of d_c in w_n as the difficulty of distinguishing q'_c and q_c , i.e., the higher d_c in w_n , the harder q'_c . We first test different queries whose corresponding clicked documents d'_c are in w_n : (1) “Low queries” are queries where d_c is ranked low in their window, that is, d_c is less relevant to q'_c than d'_c according to dot product by BERT. (2) “Medium queries” are queries where d_c is ranked in the medium part of w_n , i.e., around d'_c . These queries’ d'_c are closer to d_c than low/high ranking ones, thus they are more ambiguous. (3) “High queries” are queries where d_c is ranked high in w_n , i.e., more relevant to q'_c . We also test different retrieval models (BERT or BM25) for calculating query-document relevance for all queries. We apply different strategies to sample the same number of ambiguous queries from w_n and show the results in Table VI. From the results, we can have the following findings:

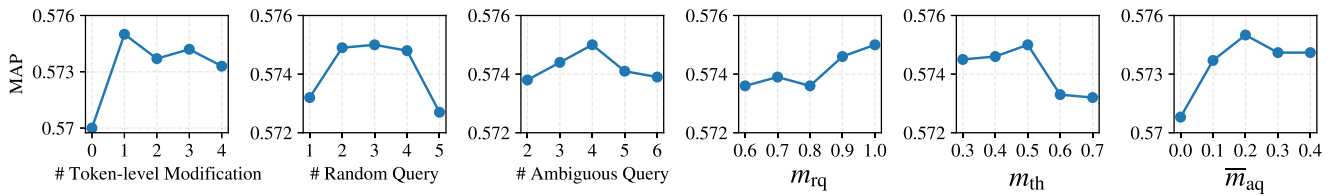


Fig. 4. Influence of the number of generated data pairs and the score margins.

(1) *Medium-ranking (Ambiguous) queries are more informative*: It is clear to see that data pairs constructed from medium-ranking (ambiguous) queries perform better than both low-ranking and high-ranking queries. We consider the reason is that the likelihood of false negatives resulting from the negative pairs created by ambiguous queries is reduced. These negatives are neither too difficult (perhaps false negatives) nor too simple (uninformative).

(2) *The dense model BERT outperforms BM25 in terms of getting a high-quality ranking list*: The results show that using BERT as the dense retriever when generating ranking lists performs better than the sparse retriever BM25. The speculation is that we try to mine ambiguous queries at the semantic level rather than the term level (which we have already done in term-level modification), and the dense model BERT can better represent semantic-level information.

D. Influence of Hyperparameters

1) *The Number of Generated Data Pairs*: We propose some data augmentation strategies to generate additional data pairs by altering q_c . We examined how the number of generated data pairs by each strategy (excluding historical queries, which were all used and not treated as a hyperparameter) influenced the results. We tuned the numbers within the range of 0 to 10, with increments of 1. Due to space constraints, we only present the MAP performance on the AOL dataset and display results for five tuned values. The patterns observed on the other datasets and metrics are similar and thus not included in this discussion.

As illustrated in the left section of Fig. 4, our data augmentation strategies' performance initially increases to reach optimal values, and then starts to decline. We believe there is a trade-off: If we generate too few data pairs, our model cannot fully model user search patterns. However, QASS may overfit the generated data if the number is too large.

2) *Score Margins of Varying Difficulty Training Pairs*: We propose to coordinate the training of data pairs of varying difficulty by different score margins. We tune the margins in the range $[0, 1.0]$ with the step of 0.1. We conduct experiments and present the performance of MRR on the AOL search log to investigate the influence of margins.

As shown in the right part of Fig. 4, the performances of different margins all increase to optimal values and then drop (except for m_{rq} which is already set as the maximum value). There is also a trade-off: If the margin is too small, our model will become sensitive and wrongly give high scores for the augmented sequences that are irrelevant to the clicked document.

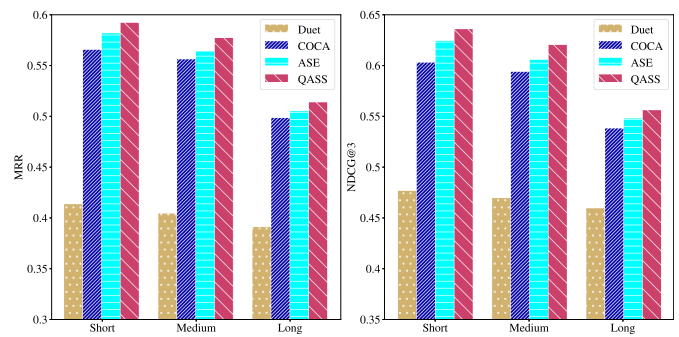


Fig. 5. Performances on different lengths of sessions on AOL search log.

However, when the margin is too large, our matching model cannot handle strongly relevant distractors.

E. Performances on Different Session Lengths and Query Positions

Following existing works [2], [4], [5], [7], [11], we conduct experiments on different lengths of sessions to analyze our model's performance. We divide the AOL search log into three kinds of data:

- Short sessions, which consist of 2 queries, account for 66.5% of the test set.
- Medium sessions, which consist of 3-4 queries, account for 27.24% of the test set.
- Long sessions, which consist of 5 or more queries, account for 6.26% of the test set.

We conduct a comparison of the performance of our proposed model, QASS, with several baseline models including Duet, COCA, and ASE, using split data. The results presented in Fig. 5 allowed us to make the following observations: (1) Context-aware ranking models consistently outperform the ad-hoc ranking model Duet across all lengths of sessions. This finding further supports the notion that incorporating session context is beneficial in understanding user search intents. (2) Our model, QASS, demonstrates superior performance compared to all other models across all lengths of session data. This result highlights the effectiveness of our proposed query-oriented data augmentation technique. (3) All models' performances decline as session lengths become longer, but QASS has a smaller drop ratio. This is because QASS constructs additional data pairs by altering queries that have history, which helps our model learn that the same session history can represent different search intents with different current queries. The results further demonstrate the

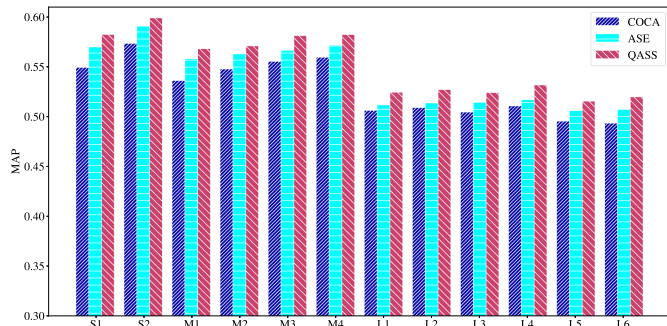


Fig. 6. We evaluated the performances at different query positions in short (S), medium (M), and long (L) sessions. The numbers appended to “S,” “M,” and “L” indicate the query position within the session.

effectiveness of our model in terms of modeling search patterns of session search.

We also conduct experiments on queries of different positions to study the performance of QASS of modeling session progression. We compared the performance of our model, QASS, with COCA and ASE. The results are depicted in Fig. 6, and based on these results, we can draw the following observations: (1) The performance of all models improved as the session continues. This can be attributed to the availability of more session histories, which further highlights the importance of modeling session context in improving the performance of ranking models. (2) QASS outperforms baselines at all positions, which demonstrates our query-oriented data augmentation’s effectiveness again. Besides, QASS performs especially better at the posterior queries in sessions. This is because QASS emphasizes the modeling of the most important behavior in the session, i.e., the current query, by altering it to construct training pairs, which helps understand search intents when there are lots of behaviors available (long history). (3) It is intriguing that all performance drops from L4 to L7. As stated in [4], [7], these long sessions are believed to be challenging exploratory or extremely complicated search tasks, which are naturally hard to resolve.

F. Cost Analysis

In this section, we will analyze the cost of QASS across three stages: preprocessing, training, and inference.

For the preprocessing stage, most augmentation strategies are rule-based, except for ambiguous query mining. The primary cost here is the time required for mining ambiguous queries, which is approximately 20 minutes for the AOL dataset and 10 minutes for the Tiangong-ST dataset.

For the training stage, the additional time cost in this stage comes from the augmented data pairs. For each original pair, we generate three sequences for term-level modifications, three random queries, four most ambiguous queries, and all historical queries within the session. With an average session length of about three, the average number of sequences generated from historical queries is $(0 + 1 + 2)/3 = 1$ in average. Thus, the total augmented sequences are $3 + 3 + 4 + 1 = 11$ in average. As a result, QASS requires approximately 11 times the training time compared to the naive BERT model.

For the inference stage, QASS incurs the same inference cost as BERT-based models since our augmentation process affects only the training stage. This ensures that QASS achieves better performance than BERT-based models while maintaining a similar online cost, which makes it suitable for practical use.

VI. CONCLUSION AND FUTURE WORK

In this study, we aimed to enhance search logs by incorporating augmented training pairs through query alterations. This approach allowed our model to learn that the relevance of a document can vary when the session context changes, thus improving our understanding of users’ search patterns. The symmetric relevance between the candidate document and the search context was overlooked by the existing training paradigm. To generate negative sequences for pair-wise training with the original sequence, we modified the current query at both term and query levels. This involved masking, replacing, and adding terms, as well as substituting the query with mined queries from the search log (random queries, historical queries, and ambiguous queries). The difficulty of the mined/generated queries varied based on their similarity to the original query. Additionally, we employed different score margins to coordinate the data pairs generated from various data augmentation strategies. Our approach was evaluated through experiments on two publicly available search logs, and the results demonstrated its effectiveness.

Despite the contributions of our work, there are still some limitations that need to be addressed in future research:

- We used term-level modification and query-level replacement to alter the current query. There are more sophisticated data augmentation strategies to be designed, e.g., masking the word that has the highest attention score.
- QASS was implemented based on BERT. However, our approach can be applied to other base models, e.g., the encoder of BART. In future work, we plan to conduct experiments on different base models to further study our approach’s effectiveness and applicability.
- In this work, we only implemented data augmentation on queries that have session history. For queries that do not have a history (i.e., ad-hoc queries), we need to design a special augmentation strategy, which may help our model learn more about users’ search patterns.
- For negative training pairs of varying difficulty, we plan to try curriculum learning as a more advanced approach to coordinate them.
- For representing queries and documents, we will investigate the performance of our augmentation strategies on more advanced general embedding models, such as E5 [48] or BGE [49].

REFERENCES

- [1] W. U. Ahmad, K. Chang, and H. Wang, “Multi-task learning for document ranking and query suggestion,” in *Proc. 6th Int. Conf. Learn. Representations*, Vancouver, BC, Canada, 2018, pp. 1–14. [Online]. Available: <https://openreview.net/forum?id=SJ1nzBeA->
- [2] W. U. Ahmad, K. Chang, and H. Wang, “Context attentive document ranking and query suggestion,” in *Proc. 42nd Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, Paris, France, 2019, pp. 385–394, doi: 10.1145/3331184.3331246.

- [3] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol.*, J. Burstein, C. Doran, and T. Solorio, Eds., Minneapolis, MN, USA, Association for Computational Linguistics, 2019, pp. 4171–4186, doi: [10.18653/v1/n19-1423](https://doi.org/10.18653/v1/n19-1423).
- [4] H. Chen, Z. Dou, Y. Zhu, Z. Cao, X. Cheng, and J.-R. Wen, "Enhancing user behavior sequence modeling by generative tasks for session search," in *Proc. 31st ACM Int. Conf. Inf. Knowl. Manage.*, 2022, pp. 180–190.
- [5] H. Chen, Z. Dou, Q. Zhu, X. Zuo, and J.-R. Wen, "Integrating representation and interaction for context-aware document ranking," *ACM Trans. Inf. Syst.*, vol. 41, 2022, Art. no. 21, doi: [10.1145/3529955](https://doi.org/10.1145/3529955).
- [6] X. Zuo, Z. Dou, and J. Wen, "Improving session search by modeling multi-granularity historical query change," in *Proc. 15th ACM Int. Conf. Web Search Data Mining*, Tempe, AZ, USA, 2022, pp. 1534–1542.
- [7] Y. Zhu et al., "Contrastive learning of user behavior sequence for context-aware document ranking," in *Proc. 30th ACM Int. Conf. Inf. Knowl. Manage.*, Queensland, Australia, 2021, pp. 2780–2791, doi: [10.1145/3459637.3482243](https://doi.org/10.1145/3459637.3482243).
- [8] K. Zhou et al., "SimANS: Simple ambiguous negatives sampling for dense text retrieval," 2022, *arXiv:2210.11773*, doi: [10.48550/arXiv.2210.11773](https://doi.org/10.48550/arXiv.2210.11773).
- [9] V. Karpukhin et al., "Dense passage retrieval for open-domain question answering," in *Proc. Conf. Empir. Methods Natural Lang. Process.*, Association for Computational Linguistics, 2020, pp. 6769–6781, doi: [10.18653/v1/2020.emnlp-main.550](https://doi.org/10.18653/v1/2020.emnlp-main.550).
- [10] L. Xiong et al., "Approximate nearest neighbor negative contrastive learning for dense text retrieval," in *Proc. 9th Int. Conf. Learn. Representations*, Austria, 2021, pp. 1–16. [Online]. Available: <https://openreview.net/forum?id=zeFrfgyZln>
- [11] Y. Zhu, J. Nie, Y. Su, H. Chen, X. Zhang, and Z. Dou, "From easy to hard: A dual curriculum learning framework for context-aware document ranking," in *Proc. 31st ACM Int. Conf. Inf. Knowl. Manage.*, Atlanta, GA, USA, 2022, pp. 2784–2794, doi: [10.1145/3511808.3557328](https://doi.org/10.1145/3511808.3557328).
- [12] J. Zhan, J. Mao, Y. Liu, J. Guo, M. Zhang, and S. Ma, "Optimizing dense retrieval model training with hard negatives," in *Proc. 44th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, Canada, 2021, pp. 1503–1512, doi: [10.1145/3404835.3462880](https://doi.org/10.1145/3404835.3462880).
- [13] G. Pass, A. Chowdhury, and C. Torgeson, "A picture of search," in *Proc. 1st Int. Conf. Scalable Inf. Syst.*, Hong Kong, 2006, pp. 1–es, doi: [10.1145/1146847.1146848](https://doi.org/10.1145/1146847.1146848).
- [14] J. Chen, J. Mao, Y. Liu, M. Zhang, and S. Ma, "TianGong-ST: A new dataset with large-scale refined real-world web search sessions," in *Proc. 28th ACM Int. Conf. Inf. Knowl. Manage.*, Beijing, China, 2019, pp. 2485–2488, doi: [10.1145/3357384.3358158](https://doi.org/10.1145/3357384.3358158).
- [15] H. S. Nugraha and S. Suyanto, "Typographic-based data augmentation to improve a question retrieval in short dialogue system," in *Proc. Int. Seminar Res. Inf. Technol. Intell. Syst.*, 2019, pp. 44–49.
- [16] T. Wang and P. Isola, "Understanding contrastive representation learning through alignment and uniformity on the hypersphere," in *Proc. 37th Int. Conf. Mach. Learn.*, PMLR, 2020, pp. 9929–9939. [Online]. Available: <http://proceedings.mlr.press/v119/wang20k.html>
- [17] K. He, H. Fan, Y. Wu, S. Xie, and R. B. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Seattle, WA, USA, 2020, pp. 9726–9735, doi: [10.1109/CVPR42600.2020.00975](https://doi.org/10.1109/CVPR42600.2020.00975).
- [18] Y. Li, Z. Liu, C. Xiong, and Z. Liu, "More robust dense retrieval with contrastive dual learning," in *Proc. ACM SIGIR Int. Conf. Theory Inf. Retrieval*, Canada, F. Hasibi, Y. Fang, and A. Aizawa, Eds., 2021, pp. 287–296, doi: [10.1145/3471158.3472245](https://doi.org/10.1145/3471158.3472245).
- [19] Y. Yang, N. Jin, K. Lin, M. Guo, and D. Cer, "Neural retrieval for question answering with cross-attention supervised data augmentation," in *Proc. 59th Annu. Meeting Assoc. Comput. Linguistics 11th Int. Joint Conf. Natural Lang. Process.*, C. Zong, F. Xia, W. Li, and R. Navigli, Eds., Association for Computational Linguistics, 2021, pp. 263–268, doi: [10.18653/v1/2021.acl-short.35](https://doi.org/10.18653/v1/2021.acl-short.35).
- [20] N. Yang, F. Wei, B. Jiao, D. Jiang, and L. Yang, "xMoCo: Cross momentum contrastive learning for open-domain question answering," in *Proc. 59th Annu. Meeting Assoc. Comput. Linguistics 11th Int. Joint Conf. Natural Lang. Process.*, C. Zong, F. Xia, W. Li, and R. Navigli, Eds., Association for Computational Linguistics, 2021, pp. 6120–6129, doi: [10.18653/v1/2021.acl-long.477](https://doi.org/10.18653/v1/2021.acl-long.477).
- [21] T. Fu, X. E. Wang, M. F. Peterson, S. T. Grafton, M. P. Eckstein, and W. Y. Wang, "Counterfactual vision-and-language navigation via adversarial path sampler," in *Proc. 16th Eur. Conf. Comput. Vis.*, Glasgow, U.K., Springer, 2020, pp. 71–86, doi: [10.1007/978-3-030-58539-6_5](https://doi.org/10.1007/978-3-030-58539-6_5).
- [22] Y. Chen et al., "Cross-language sentence selection via data augmentation and rationale training," in *Proc. 59th Annu. Meeting Assoc. Comput. Linguistics 11th Int. Joint Conf. Natural Lang. Process.*, C. Zong, F. Xia, W. Li, and R. Navigli, Eds., Association for Computational Linguistics, 2021, pp. 3881–3895, doi: [10.18653/v1/2021.acl-long.300](https://doi.org/10.18653/v1/2021.acl-long.300).
- [23] L. Yao, B. Yang, H. Zhang, B. Chen, and W. Luo, "Domain transfer based data augmentation for neural query translation," in *Proc. 28th Int. Conf. Comput. Linguistics*, D. Scott, N. Bel, and C. Zong, Eds., Barcelona, Spain, 2020, pp. 4521–4533, doi: [10.18653/v1/2020.coling-main.399](https://doi.org/10.18653/v1/2020.coling-main.399).
- [24] L. Chen, H. Zhang, J. Xiao, X. He, S. Pu, and S. Chang, "Counterfactual critic multi-agent training for scene graph generation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Seoul, South Korea, 2019, pp. 4612–4622, doi: [10.1109/ICCV.2019.00471](https://doi.org/10.1109/ICCV.2019.00471).
- [25] Y. Li, Y. Luo, Z. Zhang, S. W. Sadiq, and P. Cui, "Context-aware attention-based data augmentation for POI recommendation," in *Proc. 35th IEEE Int. Conf. Data Eng. Workshops*, Macao, China, 2019, pp. 177–184, doi: [10.1109/ICDEW.2019.00-14](https://doi.org/10.1109/ICDEW.2019.00-14).
- [26] Q. Yu and W. Lam, "Data augmentation based on adversarial autoencoder handling imbalance for learning to rank," in *Proc. 33rd AAAI Conf. Artif. Intell., 31st Innov. Appl. Artif. Intell. Conf., 9th AAAI Symp. Educ. Adv. Artif. Intell.*, Honolulu, Hawaii, USA, AAAI Press, 2019, pp. 411–418, doi: [10.1609/aaai.v33i01.3301411](https://doi.org/10.1609/aaai.v33i01.3301411).
- [27] Z. Qiu, Y. Jian, Q. Chen, and L. Zhang, "Learning to augment imbalanced data for re-ranking models," in *Proc. 30th ACM Int. Conf. Inf. Knowl. Manage.*, Queensland, Australia, G. Demartini, G. Zuccon, J. S. Culpepper, Z. Huang, and H. Tong, Eds., 2021, pp. 1478–1487, doi: [10.1145/3459637.3482364](https://doi.org/10.1145/3459637.3482364).
- [28] I. Bartolini, V. Moscato, M. Postiglione, G. Sperli, and A. Vignali, "COSINER: Context similarity data augmentation for named entity recognition," in *Proc. 15th Int. Conf. Similarity Search Appl.*, Bologna, Italy, T. Skopal, F. Falchi, J. Lokoc, M. L. Sapino, I. Bartolini, and M. Patella, Eds., Springer, 2022, pp. 11–24, doi: [10.1007/978-3-031-17849-8_2](https://doi.org/10.1007/978-3-031-17849-8_2).
- [29] I. Bartolini, V. Moscato, M. Postiglione, G. Sperli, and A. Vignali, "Data augmentation via context similarity: An application to biomedical named entity recognition," *Inf. Syst.*, vol. 119, 2023, Art. no. 102291, doi: [10.1016/j.is.2023.102291](https://doi.org/10.1016/j.is.2023.102291).
- [30] X. Shen, B. Tan, and C. Zhai, "Context-sensitive information retrieval using implicit feedback," in *Proc. 28th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, Salvador, Brazil, 2005, pp. 43–50, doi: [10.1145/1076034.1076045](https://doi.org/10.1145/1076034.1076045).
- [31] P. N. Bennett et al., "Modeling the impact of short- and long-term behavior on search personalization," in *Proc. 35th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, Portland, OR, USA, 2012, pp. 185–194, doi: [10.1145/2348283.2348312](https://doi.org/10.1145/2348283.2348312).
- [32] R. W. White, W. Chu, A. H. Awadallah, X. He, Y. Song, and H. Wang, "Enhancing personalized search by mining and modeling task behavior," in *Proc. 22nd Int. World Wide Web Conf.*, Rio de Janeiro, Brazil, 2013, pp. 1411–1420, doi: [10.1145/2488388.2488511](https://doi.org/10.1145/2488388.2488511).
- [33] C. V. Gysel, E. Kanoulas, and M. de Rijke, "Lexical query modeling in session search," in *Proc. ACM Int. Conf. Theory Inf. Retrieval*, Newark, DE, USA, B. Carterette, H. Fang, M. Lalmas, and J. Nie, Eds., 2016, pp. 69–72, doi: [10.1145/2970398.2970422](https://doi.org/10.1145/2970398.2970422).
- [34] C. Qu, C. Xiong, Y. Zhang, C. Rosset, W. B. Croft, and P. Bennett, "Contextual re-ranking with behavior aware transformers," in *Proc. 43rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, China, 2020, pp. 1589–1592, doi: [10.1145/3397271.3401276](https://doi.org/10.1145/3397271.3401276).
- [35] S. Wang, Z. Dou, and Y. Zhu, "Heterogeneous graph-based context-aware document ranking," in *Proc. 16th ACM Int. Conf. Web Search Data Mining*, New York, NY, USA, 2023, pp. 724–732, doi: [10.1145/3539597.3570390](https://doi.org/10.1145/3539597.3570390).
- [36] S. R. Bowman, L. Vilnis, O. Vinyals, A. M. Dai, R. Józefowicz, and S. Bengio, "Generating sentences from a continuous space," in *Proc. 20th SIGNLL Conf. Comput. Natural Lang. Learn.*, Berlin, Germany, Y. Goldberg and S. Riezler, Eds., 2016, pp. 10–21, doi: [10.18653/v1/k16-1002](https://doi.org/10.18653/v1/k16-1002).
- [37] A. M. Dai and Q. V. Le, "Semi-supervised sequence learning," in *Proc. Adv. Neural Inf. Process. Syst.*, Montreal, QC, Canada, 2015, pp. 3079–3087. [Online]. Available: <https://proceedings.neurips.cc/paper/2015/hash/7137debd45ae4d0ab9aa953017286b20-Abstract.html>
- [38] S. G. U. N. and K. G., "LAWBO: A smart lawyer chatbot," in *Proc. ACM India Joint Int. Conf. Data Sci. Manage. Data*, Goa, India, 2018, pp. 348–351, doi: [10.1145/3152494.3167988](https://doi.org/10.1145/3152494.3167988).
- [39] R. Jones and K. L. Klinkner, "Beyond the session timeout: Automatic hierarchical segmentation of search topics in query logs," in *Proc. 17th ACM Conf. Inf. Knowl. Manage.*, Napa Valley, CA, USA, 2008, pp. 699–708, doi: [10.1145/1458082.1458176](https://doi.org/10.1145/1458082.1458176).

- [40] H. Wang, Y. Song, M. Chang, X. He, R. W. White, and W. Chu, "Learning to extract cross-session search tasks," in *Proc. 22nd Int. World Wide Web Conf.*, Rio de Janeiro, Brazil, 2013, pp. 1353–1364, doi: [10.1145/2488388.2488507](https://doi.org/10.1145/2488388.2488507).
- [41] J. Johnson, M. Douze, and H. Jégou, "Billion-scale similarity search with GPUs," *IEEE Trans. Big Data*, vol. 7, no. 3, pp. 535–547, Jul. 2021, doi: [10.1109/TBDDATA.2019.2921572](https://doi.org/10.1109/TBDDATA.2019.2921572).
- [42] C. V. Gysel and M. de Rijke, "Py trec_eval: An extremely fast python interface to trec_eval," in *Proc. 41st Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, Ann Arbor, MI, USA, 2018, pp. 873–876, doi: [10.1145/3209978.3210065](https://doi.org/10.1145/3209978.3210065).
- [43] S. E. Robertson and H. Zaragoza, "The probabilistic relevance framework: BM25 and beyond," *Found. Trends Inf. Retrieval*, vol. 3, no. 4, pp. 333–389, 2009, doi: [10.1561/1500000019](https://doi.org/10.1561/1500000019).
- [44] B. Hu, Z. Lu, H. Li, and Q. Chen, "Convolutional neural network architectures for matching natural language sentences," in *Proc. Adv. Neural Inf. Process. Syst.*, Montreal, QC, Canada, 2014, pp. 2042–2050. [Online]. Available: <https://proceedings.neurips.cc/paper/2014/hash/b9d487a30398d42ecff55c228ed5652b-Abstract.html>
- [45] C. Xiong, Z. Dai, J. Callan, Z. Liu, and R. Power, "End-to-end neural ad-hoc ranking with kernel pooling," in *Proc. 40th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, Shinjuku, Tokyo, Japan, 2017, pp. 55–64.
- [46] B. Mitra, F. Diaz, and N. Craswell, "Learning to match using local and distributed representations of text for web search," in *Proc. 26th Int. Conf. World Wide Web*, Perth, Australia, 2017, pp. 1291–1299, doi: [10.1145/3038912.3052579](https://doi.org/10.1145/3038912.3052579).
- [47] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proc. 7th Int. Conf. Learn. Representations*, New Orleans, LA, USA, 2019, pp. 1–18. [Online]. Available: <https://openreview.net/forum?id=Bkg6RiCqY7>
- [48] L. Wang et al., "Text embeddings by weakly-supervised contrastive pre-training," 2022, *arXiv:2212.03533*.
- [49] S. Xiao, Z. Liu, P. Zhang, and N. Muennighoff, "C-pack: Packaged resources to advance general chinese embedding," 2023, *arXiv:2309.07597*.



Zhicheng Dou (Member, IEEE) received the BS and PhD degrees in computer science and technology from Nankai University, in 2003 and 2008, respectively. He is currently a professor with the Renmin University of China. He worked with Microsoft Research Asia from 2008 to 2014. His current research interests are information retrieval, natural language processing, and Big Data analysis. He received the Best Paper Runner-Up Award from SIGIR 2013, and the Best Paper Award from AIRS 2012. He served as the program co-chair of the short paper track for SIGIR 2019.



Yutao Zhu received the BS and MS degrees from the Renmin University of China, and the PhD degree from the University of Montreal. He is currently a postdoc with the Renmin University of China. His current research interests are large language models and information retrieval. He received the Best Paper Award from CCIIR 2021 and the Google Scholarship for UdeM, in 2019. He served as the PC member of several top-tier conferences, such as ACL, SIGIR, SIGKDD, AAAI, EMNLP, etc.



Haonan Chen received the BE degree in computer science and technology from the Harbin Institute of Technology, in 2017. He is currently working toward the PhD degree with the Gaoling School of Artificial Intelligence, Renmin University of China. His research interests include information retrieval.



Ji-Rong Wen (Senior Member, IEEE) received the BS, and MS degrees from the Renmin University of China, and the PhD degree from the Chinese Academy of Science, in 1999. He is a professor with the Renmin University of China. He was a senior researcher and research manager with Microsoft Research from 2000 to 2014. His main research interests include web data management, information retrieval (especially web IR), and data mining.