From Easy to Hard: A Dual Curriculum Learning Framework for Context-Aware Document Ranking

Yutao Zhu  
Jian-Yun Nie  
University of Montreal, Montreal Quebec, Canada  
yutaozhu94@gmail.com  
nie@iro.umontreal.ca

Yixuan Su  
Language Technology Lab, University of Cambridge, Cambridge, United Kingdom  
ys484@cam.ac.uk

Haonan Chen  
Gaoling School of Artificial Intelligence, Renmin University of China, Beijing, China  
hnchen@ruc.edu.cn

Xinyu Zhang  
Huawei Poisson Lab, Hangzhou Zhejiang, China  
zhangxinyu35@huawei.com

Zhicheng Dou  
Gaoling School of Artificial Intelligence, Renmin University of China, Beijing, China  
dou@ruc.edu.cn

ABSTRACT

Contextual information in search sessions is important for capturing users’ search intents. Various approaches have been proposed to model user behavior sequences to improve document ranking in a session. Typically, training samples of (search context, document) pairs are sampled randomly in each training epoch. In reality, the difficulty to understand user’s search intent and to judge document’s relevance varies greatly from one search context to another. Mixing up training samples of different difficulties may confuse the model’s optimization process. In this work, we propose a curriculum learning framework for context-aware document ranking, in which the ranking model learns matching signals between the search context and the candidate document in an easy-to-hard manner. In so doing, we aim to guide the model gradually toward a global optimum. To leverage both positive and negative examples, two curricula are designed. Experiments on two real query log datasets show that our proposed framework can improve the performance of several existing methods significantly, demonstrating the effectiveness of curriculum learning for context-aware document ranking.

CCS CONCEPTS

• Information systems → Retrieval models and ranking.

KEYWORDS

Curriculum Learning; Sample Difficulty; Context-aware Document Ranking

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ACM Reference Format:
Yutao Zhu, Jian-Yun Nie, Yixuan Su, Haonan Chen, Xinyu Zhang, and Zhicheng Dou. 2022. From Easy to Hard: A Dual Curriculum Learning Framework for Context-Aware Document Ranking. In Proceedings of the 31st ACM International Conference on Information and Knowledge Management (CIKM ’22), October 17–21, 2022, Atlanta, GA, USA. ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3511808.3557328

1 INTRODUCTION

Users’ search behaviors have evolved from one-shot queries to multiple interactions with search engines [1]. To fulfill a complex information retrieval task, users may issue a series of queries, examine and interact with some results. Many studies have shown that historical user behavior or search activities can be leveraged to improve document ranking, especially when users’ search intent is ambiguous [3, 5, 18, 31, 64, 66].

Previous studies have exploited user search behaviors for understanding user intent and improving document ranking within a session [5, 8, 52]. Earlier research explored query expansion and learning-to-rank techniques [5, 8, 21, 50]. More recently, many neural architectures have been developed to model user behavior sequences and capture user search intent [2, 3, 46, 52, 65]. For example, a hierarchical RNN with attention mechanism was used to model the historical queries and the corresponding clicked documents, leading to better document ranking [2, 52]. Researchers also discovered that learning query suggestion as a supplementary task is also beneficial for document ranking [3]. Recently, pre-trained language models have also been used to capture search intent from user behavior sequences [46, 65].

Although the approaches proposed are different, they all rely on the information extracted from previous search logs, for example, (search context, document) pairs that are considered as positive or negative samples. The search context usually aggregates the current query and the previous queries (and interacted documents in some cases). All positive pairs are assumed to be samples of equal importance that reflect relevance, and are put together in the same pool for sampling a training batch. The same for negative pairs that reflect irrelevance. While it is true that positive and negative examples are generally useful for training a good ranking model, it is not true that they are equally useful at different training stages. In some cases, the relevance (or irrelevance) relation in a pair of (search context, document) is obvious, while in other cases, it is
more subtle. Let us illustrate this by some examples. In the left part of Figure 1, we show two search contexts (sessions) containing queries and the documents clicked by the user. Let us try to understand the search intent in each of the search context. In the first case, the relevance relation between the search context formed by \( q_1 \) and the clicked document \( d^{+}_1 \) is clear because both are about the singer "Clay Aiken". In comparison, in the second case, the relation between the search context formed by \( \{q_1, d^{+}_1, q_2\} \) and the clicked document \( d^{+}_2 \) is more difficult to capture. The underlying intent is "Chanel's designer handbags", which can only be understood with the help of the historical query \( q_1 \) and the clicked document \( d^{+}_1 \). If a human is asked to learn from these two samples, the relevance signal in the first one is much easier to capture than in the second. For a training process, the situation is similar: These pairs represent different levels of difficulty for a training process to digest. When a human learner is presented with samples of mixed difficulties, he/she can be confused because signals from different samples may appear inconsistent, or they do not have sufficient knowledge to understand difficult samples. Recent studies in machine learning also showed that learning with a batch of samples of mixed difficulties may disturb the optimization, especially when the network is deep [4]. In this case, curriculum learning, i.e., learning from easy samples before hard samples, becomes particularly useful.

A similar problem occurs when negative samples are considered. As can be seen in the right side of Figure 1, for the second search context, the negative candidate \( d^{-}_2 \) contains information about "designer handbags". Compared with \( d^{-}_1 \), we can see that \( d^{-}_2 \) is much harder than \( d^{-}_1 \) to be recognized as a negative sample. Therefore, it is also desirable to learn from easy negative samples before hard ones. If we compare \( d^{-}_2 \) with the positive document \( d^{+}_2 \), we can learn that "chanel" is an important term reflecting the user’s real intent. Such comparison/contrast is also common and critical for human learning, especially for discriminating similar concepts [13].

Motivated by the observations above, we propose a novel training framework that takes into account the levels of difficulty in the training samples. Our framework is inspired by both curriculum learning [22, 44] and contrastive learning [11, 17, 54, 55]. Curriculum learning simulates the human recognition process, i.e., learning with easier samples first and more difficult samples later. It has achieved great performance on various tasks, such as image classification [19, 22], natural language understanding [61], and ad-hoc retrieval [44]. Contrastive learning aims at learning representations such that similar samples stay close to each other, while dissimilar ones are far apart. By comparing similar and dissimilar samples, the model can be better optimized to capture their differences.

Concretely, we treat the search context and the candidate document as a pair and optimize the model by contrasting positive pairs (with clicked documents) and negative pairs (with unclicked documents). We design a dual curriculum learning framework incorporating two complementary curricula for positive and negative pairs, respectively. In the curriculum for positive pairs, sampling is restricted to easy pairs in early steps, and then extended gradually to the whole set of samples, so that hard pairs can also be learned. In the curriculum for negative pairs, we do the opposite: sampling from all pairs in early steps, then restricting gradually to hard pairs in late steps. These strategies are inspired by similar human learning: we select learning (positive) examples from easy to hard, but want to contrast them with all the negative examples at the beginning to have a better idea of the general differences between the positives and negatives. Toward the end of the learning, as the obvious differences have been learned, the easy negatives cannot provide effective learning signals, so we focus on distinguishing hard negative examples (these hard ones are demonstrated to be beneficial for model optimization [60, 62]). Our two curricula intend to follow the same principle during the training process.

The curriculum strategy can be used in any existing approach. We integrate it into three state-of-the-art approaches for context-aware document ranking. We conduct experiments on two large-scale search log datasets (AOL [42] and Tiangong-ST [10]). Experimental results show that our curriculum learning method significantly improves three strong baselines. The consistent performance gains demonstrate the effectiveness and wide applicability of our approach. Our further experiments show that both positive and negative curricula are beneficial to the ranking effectiveness.

Our contributions are three-fold:

1. We propose a novel curriculum learning framework for context-aware document ranking, in which the difficulty of training samples is taken into account.

2. We devise two complementary curricula for learning user intent from positive and negative pairs of (search context, candidate document). By learning them in an easy-to-hard manner, the model’s performance can be improved gradually.

3. Experimental results on two large-scale benchmark datasets show significant improvements. The experiments also confirm the broad applicability, flexibility, and high robustness of our method.

## 2 RELATED WORK

### 2.1 Context-Aware Document Ranking

Context information in sessions has shown to be beneficial in modeling user search intent [5, 18, 31]. Early research focused on characterizing users’ search intent by extracting contextual features from their search behaviors [50, 56, 58]. However, because these methods are built on handcrafted rules or manually collected features, they can only be used for a limited number of retrieval tasks. Researchers also developed predictive models for users’ search intent or future behavior [6], but the learning of complex user-system interactions is limited by the predefined features.

The recent development of deep neural networks has triggered new approaches to context-aware document ranking. For example, researchers exploited hierarchical RNN-based architectures to model the sequence of historical queries [27, 52, 57]. These architectures were further enhanced by attention mechanisms to better capture search behaviors [12]. It is also found that learning query suggestion and document ranking jointly can boost the performance on both tasks [2]. Besides, historical clicked documents are also reported to be helpful in predicting user search behaviors [3].

Recently, pre-trained language models, such as BERT [14], have achieved promising results on several NLP and IR tasks [33, 34, 37, 38]. Some researchers proposed concatenating all historical queries and candidate documents into a long sequence to compute a sequence representation using BERT, based on which the ranking
Figure 1: Illustration of DCL’s utilization of samples. The left side shows positive pairs of different difficulties. In the early training steps, the model can only learn from easy pairs (e.g., the first pair). Then, as the training progresses, harder pairs (e.g., the second pair) are added. The right side is the curriculum of negative pairs. All negative candidates ($d_{1,1}^-, d_{1,2}^-, d_{1,3}^-$) of easy cases are used for training in the early steps, while only hard negatives ($d_{2,1}^-$) are used in later steps.

score is determined [46]. Furthermore, contrastive learning has shown to be beneficial for optimizing the BERT encoder in context-aware document ranking [65].

Different from the studies above, we focus on improving the model optimization process by curriculum learning rather than designing new architectures or supplementary tasks for context-aware document ranking. Our work is orthogonal to the above approaches and can be combined with them. In fact, in most existing approaches, sampling of training data is necessary in the optimization process. This is typically done by random sampling or by selecting hard samples. Our work will show that selecting samples by curriculum from easy to hard can better optimize the existing models.

### 2.2 Curriculum Learning for IR

In the context of human learning, it is common to follow a curriculum that regulates the ordering and content of the education materials [35, 43, 51]. With this strategy, students can leverage previously learned concepts to help them learn new and more difficult ones. Inspired by research in cognitive science [49], researchers proposed machine learning algorithms based on a curriculum [4, 15]. The core idea is to train the model using easy samples first and increase the difficulty along the training process. Such a curriculum learning (CL) strategy has achieved great performance on several tasks, such as image classification [19, 22], machine translation [45, 63], dialogue generation [53], and natural language understanding [61].

In the area of IR, CL research is still in its early stage. The first attempt applied CL to learning-to-rank (LTR) [16], however, without much success. Later, researchers found that manually collected features in LTR can be a source of noise, and the CL strategy is more suitable for neural ranking models [44]. More recently, several heuristics have been proposed to determine the difficulty of different answers, based on which a CL-based method is used. This led to improved performance in answer ranking [39].

All existing CL-based methods for IR tasks are designed for organizing positive samples, but the influence of negative samples is neglected. We notice that some studies have focused on selecting hard negative samples for IR tasks [36, 47, 62], but they did not use CL. In this paper, we propose two complementary and contrastive curricula to enhance the model’s learning using both positive and negative context-document pairs. Our experiments show that regulating the learning pace of both positive and negative samples is very effective.

## 3 METHODOLOGY

The goal of context-aware document ranking is to rank a list of candidate documents using the search context. In this work, we propose a Dual Curriculum Learning (DCL) framework for this task. As shown in Figure 1, our framework consists of two complementary curricula for learning positive and negative context-document pairs. In each curriculum, the pairs are sorted according to their difficulty so that the model can learn them from easy to hard.

### 3.1 Notations and Task Definition

We first define some important concepts and notations before introducing our framework. A user’s search behavior is represented as a sequence of $n$ interactions $H_n = [q_1, d_1^+, \ldots, q_n, d_n^+]$, where each query $q_i$ is associated with a corresponding clicked document $d_i^+$. If there are several clicked documents, each of them is associated with the query to form a separate pair in the sequence. Each query $q_i$ is represented by the original text string submitted to the search engine, while each clicked document $d_i$ is represented by its text content. All queries are ordered according to the timestamps. For convenience, we further denote $C = [q_1, d_1^+, \ldots, q_n]$ as a search context when $q_n$ is submitted after a series of queries and interacted documents. When training a ranking model, for each positive clicked document, a set of $m$ unclicked documents are selected as
negative candidates. As a result, the candidate document set for the search context $C$ is represented as $D = \{d^n_1, d^n_2, \ldots, d^n_m\}$ (the subscript $n$ in $d^n_1$ will be omitted).

With the above notations, the task of context-aware document ranking can be defined as: ranking the candidate document set $D$ based on the search context $C$ so as to rank the clicked document $d^+$ as high as possible.

To make it clear, henceforth, we will call the pair $(C, d^+)$ a positive pair, whereas the pairs $(C, d^n_j)_{j=1}^m$ are negative pairs.

Notice that in this paper, we leverage user logs to learn a ranking model because user clicks can serve as a good proxy of relevance feedback [3, 28, 29, 46]. However, the same method can be used with human relevance judgments if available.

### 3.2 Overview

Our framework consists of two complementary curricula as follows:

1. **Curriculum of Positive Pairs.** As shown in the left side of Figure 1, our first curriculum is designed for positive pairs. The target is to teach the model how to understand users’ intent by capturing matching clues between search context and clicked documents. To achieve this goal, we first sort all positive pairs in the training set according to their difficulty and then let the model learn from easy ones to hard ones. By this means, the model can gradually increase its ability to capture matching signals.

2. **Curriculum of Negative Pairs.** We also design a curriculum for negative pairs (right side of Figure 1) to enhance the model’s ability to identify the mismatching information between search context and negative documents. Specifically, we progressively increase the difficulty of the negative candidate documents associated with each selected positive pair in the training set. In this way, the model is encouraged to gradually distinguish more subtle mismatching signals between search context and negative documents.

The principles used in the two curricula have some differences. The learning from positive pairs aims at identifying the relevance signals. In the curriculum, the model can gradually capture signals from shallow and easy matching (such as similar terms) to deep and hard matching (such as semantic information). On the contrary, a group of negative documents is associated with a specific positive pair, so they provide supplementary mismatching signals. As the curriculum progresses, the negative document becomes more similar to the positive document, so the model has to capture more fine-grained clues to distinguish them. Overall, we train the model with the two complementary curricula simultaneously so that its capability of modeling users’ intent can be gradually enhanced. The whole process is shown in Figure 2.

It is worth noting that our DCL is a general training framework that works by organizing the learning order of the training samples. Therefore, it can be applied to various base models to improve their performance (this will be shown in our experiments).

### 3.3 Dual Curriculum Learning Framework

The implementation of our approach involves several key concepts, which we examine below.

**How does curriculum learning work?** When training neural networks, a mini-batch of training samples is usually randomly (i.e., uniformly – every sample is selected with the same likelihood) sampled from the training set and used for optimizing the model at a step. Curriculum learning, on the other hand, aims at adjusting the order of the samples so that they are learned according to a predefined pace rather than at random. In our framework, we design two curricula for learning positive pairs and negative pairs from easy to hard, respectively. Following the paradigm of curriculum learning [22, 44], each curriculum is defined by two functions:

- A **difficulty** function determines the difficulty of samples so that the samples can be sorted according to their difficulty.
- A **pacing** function controls the learning pace. Essentially, it adjusts the sampling space to control the difficulty of the training samples at each step.

The curricula and their combination are described below.

#### 3.3.1 Curriculum of Positive Pairs

The green part of Figure 2 illustrates the curriculum of positive pairs. Positive pairs are sorted from easy to hard according to a difficulty function. Thereafter, we gradually enlarge the sampling space so that more difficult positive samples will be included. The key is to define an appropriate difficulty function.

**Difficulty Function.** Different heuristics can be used to determine the difficulty of documents for a query [7, 39]. We consider
two factors for measuring the difficulty of each positive training pair \((C, d^+): (1)\) The first factor is the ranking score \(M(C, \cdot)\) between \(C\) and \(d^+\) (see details in Section 3.3.4): A higher \(M(C, d^+)\) indicates that it is easier to select \(d^+\) based on \(C\). (2) We also consider the position of the clicked document in the ranked list: A higher position indicates that it is easier to select it out of all documents. Formally, the difficulty for \((C, d^+)\) is computed as follows:

\[
d_p(C, d^+) = \max_{\varepsilon \in [1, |D|]} \frac{M(C, d^+)}{\max_{(C_i, d^+_{i}) \in D} M(C_i, d^+_i)},
\]

where \(D\) is the training set. The first term is the ranking position of \(d^+\) under the search context \(C\); the second term is the normalized ranking score of \((C, d^+)\). In this function, the difficulty is dominated by the position of the clicked document (the first term), while the normalized ranking score (the second term) makes an effect only when different \((C, d^+)\) pairs are ranked at the same position. This particular definition of difficulty leads to good experimental performance among different alternatives we tested.

**Pacing Function.** The pacing function determines how the training process transitions from easy to hard pairs. Following previous studies on curriculum learning [22, 44], we define the pacing function \(f_p(t)\) with respect to the training step \(t\). The value of \(f_p(t)\) is a proportion, and only the first \(f_p(t) \times |D|\) positive pairs can be used at the training step \(t\). \(f_p(t)\) is defined as follows:

\[
f_p(t) = \min \left( 1.0, \left( \frac{1}{t \cdot \alpha T} + \delta^k \right)^{\frac{1}{\beta T}} \right),
\]

where \(T\) is the total number of training steps, \(\alpha, \delta \in (0, 1)\) and \(k \in [1, +\infty)\) are hyperparameters. As shown by the green line in Figure 3, this function has the following properties: (1) the initial value \(f_p(0) = 0\) is \(\delta\), so that the model can only use some easy pairs in the first training step; (2) it increases monotonically so that harder pairs are added to the training set gradually; (3) when it reaches \(\alpha T\) steps \((f_p(\alpha T) = 1)\), all pairs in the corpus can be used for training.

### 3.3.2 Curriculum of Negative Pairs

Negative samples are also very important for learning a ranking model [62]. By comparing positive and negative pairs, the model can learn what matching signals are vital in a contrastive manner.

The orange part of Figure 2 shows the curriculum of negative pairs. Similar to positive pairs, the negative pairs are also arranged according to their difficulties (more details later). By gradually constraining the sampling space, the model will focus on more difficult samples in later steps.

**Difficulty Function.** Similar to the positive curriculum, we use the relevance between the search context and the negative candidate as the difficulty of the negative pair. A negative candidate with a high ranking score to the search context is deemed to be hard to distinguish. For a negative pair \((C, d^-)\), its difficulty is defined as:

\[
d_n(C, d^-) = M(C, d^-),
\]

where \(M(\cdot, \cdot)\) is a scoring model similar to that used in the curriculum of positive pairs. Different from the difficulty function \(f_p(t)\) in Equation 2, it is unnecessary to introduce the ranking position and normalization operation, because all negative candidates are associated with the same query; their ranking positions are determined by the ranking scores.

It is worth noting that we follow previous studies [3] to select negative candidates: the unclicked documents ranked around the clicked document (within a window) are considered as negative samples. By this means, we can avoid using too trivial or too hard negative documents. The obtained list of negative pairs is denoted as \(L\), in which all pairs are sorted according to their difficulties descendingly.

**Pacing Function.** The pacing function \(f_n(t)\) for negative pairs is designed in a similar way to that for positive pairs. \(f_n(t)\) adjusts the sampling space from which the negative sequences are sampled. It decreases with \(t\). At time step \(t\), we sample from the first \(f_n(t) \times |L|\) negative pairs for training. \(f_n(t)\) is defined as:

\[
f_n(t) = \max \left( \eta, 1 + \eta \cdot \left( \frac{1 - \eta^k}{\beta T} + \eta^k \right)^{\frac{1}{\beta T}} \right),
\]

where \(\beta, \eta \in (0, 1)\) are hyperparameters. As shown by the orange line in Figure 3, this function is similar to \(f_p(t)\), but makes opposite effect (\(i.e., decreasing rather than increasing\) along with \(t\)).

**Remark.** As shown in Figure 2 and Figure 3, we design the curriculum of negative pairs in a manner opposite to that of positive pairs, namely we gradually focus on only hard negative pairs. This is because: (1) All positive pairs are collected from human click data, which are very valuable for learning the real user intent. So, we choose to enlarge the positive sampling space and use all positive samples. By this means, we can avoid using too trivial or too hard negative documents. This strategy can lead to a more robust optimization in retrieval performance [62].

### 3.3.3 Combination of Two Curricula

DCL trains ranking models with the two curricula simultaneously. Specifically, for a training step \(t\), we build a batch of training data as follows: First, we select a batch of positive pairs \((C, d^+)\) according to the pacing function \(f_p(t)\). Then, for each search context \(C\) in the positive pairs, we sample \(m\) negative candidate documents based on the pacing function \(f_n(t)\). The process is summarized in Algorithm 1.

The whole training process of our framework naturally simulates two learning paradigms of human beings. On the one hand, the learning material is organized from easy to hard, which has been
demonstrated to be effective for both animal training and human learning. In our case, such a training process is beneficial for model optimization. On the other hand, in cognitive science, learning through comparison is also an effective way to understand new concepts [23]. In our framework, the model can learn by contrasting positive and negative candidate documents.

3.3.4 Selection of Scoring Model. In Equation (1) and (3), DCL applies a scoring model $M(\cdot, \cdot)$ to measure the ranking score between the search context and the candidate document. In our experiments, we use two different methods for $M(\cdot, \cdot)$: (1) BM25 [48] and (2) a BERT-based score commonly used in dense retrieval [32, 34]. Since we need to compute the ranking score between a search context and all documents in the training set, we apply dot-product based on BERT representations for fast computation:

$$M(C, d) = \text{BERT}(C)_{[\text{CLS}]} \cdot \text{BERT}(d)_{[\text{CLS}]}.$$  \hspace{1cm} (5)

To achieve better performance, we fine-tune BERT encoders on positive sequences with in-batch negatives [32]. Then, we can obtain the representations of all the search contexts and documents. Afterwards, we use FAISS [30] to compute $M(C, d)$ efficiently.

4 EXPERIMENTS

4.1 Datasets and Evaluation Metrics

Following previous work [3, 9, 46, 65], we conduct experiments on two public datasets: AOL search log data [42] and Tiangong-ST query log data [10].\(^2\) Another possibility is the MS MARCO Conversational Search dataset [41], but the sessions in it are artificially constructed rather than derived from real search logs and the corresponding clicked documents are unavailable. Therefore, we do not use the MS MARCO dataset in experiments.

We use the AOL dataset constructed by Ahmad et al. [3]. The dataset contains a large number of sessions, each of which consists of several queries. Each query in the training and validation sets has five candidate documents. The test set uses 50 documents retrieved by BM25 [48] as candidates for each query. All queries have at least one satisfied click in this dataset.

\(^2\)We understand that the AOL dataset should normally not be used in experiments. We still use it because there are not many datasets available that fit our experiments well.

| Table 1: The statistics of the datasets. The number in parentheses is the average number of relevant documents. |
|---|
| **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 | 5 |
| 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 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 ten candidate documents. In the training and validation sets, we use the clicked documents as the satisfied clicks. For queries with no satisfied clicks, we use a special token "[Empty]" for padding. In the test set, the candidate documents for the last query of each session have been manually annotated with relevance scores from from 0 to 4, which are used for evaluation. More details can be found in [10].

The statistics of both datasets are shown in Table 1. In Tiangong-ST test set, the relevance scores are provided instead of click labels, so we report the average number of documents with positive relevance scores (≥ 1). Following previous studies [3, 25, 26, 65], to reduce memory requirements and speed up training, we only use the document title as its content.

**Evaluation Metrics.** Similar to previous studies [3, 46, 65], we use Mean Average Precision (MAP), Mean Reciprocal Rank (MRR), and Normalized Discounted Cumulative Gain at position $k$ (NDCG@k, $k \in \{1, 3, 5, 10\}$) as evaluation metrics. In AOL, as clicked documents are used instead of manually judged documents, these measures reflect the capability of a model to rank clicked documents high. All evaluation results are obtained using the TREC’s standard evaluation tool (trec_eval) [20].

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** only use the current query without context information (historical queries and documents) for document ranking.

- **ARC-I** [24] is a representation-based approach. Query and document representations are generated by CNNs. The ranking score is determined by a multi-layer perceptron (MLP).
- **ARC-II** [24] is an interaction-based method. A matching map is constructed from the query and document, from which CNNs extract matching features. The score is also computed by an MLP.

**KNRM** [59] constructs a matching matrix by performing fine-grained interaction between
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 [46, 65], we use trec_eval while the authors of CARS use their own implementation.

Table 2: Experimental results on two datasets. All results using our framework (X+DCL) outperforms the original results (X) significantly at $p$-value < 0.01 with Bonferroni correction in paired t-test.

| Model          | MAP   | MRR   | NDCG@1 | NDCG@3 | NDCG@5 | NDCG@10 | MAP   | MRR   | NDCG@1 | NDCG@3 | NDCG@5 | NDCG@10 |
|----------------|-------|-------|--------|--------|--------|---------|-------|-------|--------|--------|--------|---------|
| ARC+I          | 0.3361| 0.3475| 0.1988 | 0.3108 | 0.3489 | 0.3953  | 0.8580| 0.7088| 0.7087 | 0.7317 | 0.7317 | 0.8691  |
| ARC+II         | 0.8384| 0.3951| 0.2428 | 0.3564 | 0.4026 | 0.4486  | 0.8611| 0.7227| 0.7237 | 0.7379 | 0.7379 | 0.8732  |
| KNRM           | 0.4038| 0.4133| 0.2397 | 0.3868 | 0.4322 | 0.4761  | 0.8709| 0.7261| 0.7473 | 0.7505 | 0.7624 | 0.8891  |
| Duet           | 0.4008| 0.4111| 0.2492 | 0.3822 | 0.4246 | 0.4675  | 0.8663| 0.7273| 0.7577 | 0.7354 | 0.7548 | 0.8829  |
| M-NSRF         | 0.4217| 0.4326| 0.2737 | 0.4025 | 0.4458 | 0.4886  | 0.8517| 0.7064| 0.7124 | 0.7308 | 0.7489 | 0.8795  |
| M-Match        | 0.4459| 0.4572| 0.3020 | 0.4301 | 0.4697 | 0.5103  | 0.8529| 0.7211| 0.7311 | 0.7233 | 0.7427 | 0.8801  |
| CAR5           | 0.4297| 0.4408| 0.2816 | 0.4117 | 0.4542 | 0.4971  | 0.8556| 0.9268| 0.7385 | 0.7386 | 0.7512 | 0.8837  |
| HBA            | 0.5281| 0.5384| 0.3773 | 0.5241 | 0.5624 | 0.5951  | 0.8615| 0.9316| 0.7612 | 0.7518 | 0.7639 | 0.8986  |
| HBA+DCL        | 0.5599| 0.5693| 0.4074 | 0.5626 | 0.5961 | 0.6242  | 0.8986| 0.9538| 0.8069 | 0.7985 | 0.8130 | 0.9122  |
| Improv.        | +6.02%| +5.74%| +7.98% | +7.35% | +5.99% | +4.89%  | +4.31%| +2.38%| +6.00% | +6.21% | +6.43% | +2.54%  |
| RCR            | 0.5338| 0.5450| 0.3894 | 0.5267 | 0.5648 | 0.5971  | 0.8147| 0.8937| 0.7670 | 0.7636 | 0.7740 | 0.8934  |
| RCR+DCL        | 0.5630| 0.5742| 0.4219 | 0.5589 | 0.5943 | 0.6257  | 0.8963| 0.9498| 0.7995 | 0.7925 | 0.8078 | 0.9089  |
| Improv.        | +5.47%| +5.36%| +8.35% | +6.11% | +5.22% | +4.79%  | +10.02%| +6.28%| +4.24% | +3.78% | +4.37% | +1.73%  |
| COCA           | 0.5500| 0.5601| 0.4024 | 0.5478 | 0.5849 | 0.6160  | 0.8623| 0.9382| 0.7769 | 0.7756 | 0.7793 | 0.8932  |
| COCA+DCL       | 0.5794| 0.5888| 0.4281 | 0.5841 | 0.6167 | 0.6432  | 0.8990| 0.9501| 0.7936 | 0.7922 | 0.8077 | 0.9088  |
| Improv.        | +5.35%| +5.12%| +6.38% | +6.63% | +5.44% | +4.42%  | +3.21%| +1.27%| +2.15% | +4.57% | +4.86% | +1.75%  |

4.3 Experimental Results

Experimental results are shown in Table 2. We choose three recently proposed methods (i.e., HBA, RCR, and COCA) as the base model and train them with our proposed DCL framework. The corresponding results are reported as “X+DCL”. To avoid the influence of randomness, we set three different random seeds and report the average performance. The standard deviation is less than 1e-3 for all results, which is omitted in the table. In general, our DCL significantly improves the performance of three base models in terms of all evaluation metrics on both datasets. This result clearly demonstrates DCL’s superiority. We also have the following observations.

(1) The context-aware document ranking models generally perform better than ad-hoc ranking methods. For instance, on the AOL dataset, the weak contextualized model M-NSRF can still outperform the strong ad-hoc ranking model KNRM. This indicates that modeling user historical behavior is beneficial for understanding the user intent and determining the desired documents. For the three strong context-aware models, DCL can further improve their performance greatly, showing its effective utilization of training samples via curriculum learning.

(2) Compared with RNN-based methods (such as RCR and CARS), BERT-based methods (COCA and HBA) perform better. It is noticeable

3We 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 [46, 65], we use trec_eval while the authors of CARS use their own implementation.

4https://github.com/DaoD/DCL
that DCL can bring improvements for both kinds of methods. Specifically, it improves the results by more than 4.7% and 1.2% in terms of all metrics on AOL and Tiangong-ST, respectively. This result demonstrates the wide applicability of our method to different base models and reflects the evident advantage of learning samples from easy to hard in context-aware document ranking.

(3) COCA is the state-of-the-art approach to context-aware document ranking. It involves a contrastive learning pre-training stage to help the BERT encoder learn more robust representations for user behavior sequences. In comparison, our DCL is a general training framework without a specific pre-training step, making it more efficient in practice. In addition, COCA trained with DCL achieves a new state-of-the-art performance in context-aware document ranking task, showing the usefulness to combine both curriculum learning and contrastive learning.

### 4.4 Discussion

We further discuss several aspects of our proposed DCL. These analyses are based on the results on the AOL dataset, while we have similar findings on the Tiangong-ST dataset.

#### 4.4.1 Impact of Both Curricula

As we propose two curricula for learning on positive and negative pairs, to validate their effectiveness, we conduct an ablation study by disabling each of them from COCA+DCL (i.e., the sampling is done among all samples). The results are shown in Table 3. We can observe:

- First, we can see that both curricula are useful. Applying any of them leads to performance improvement. When no curriculum learning is used, we observe large drops in performance. This directly validates our assumption in this paper that learning from easy to hard samples can guide the model in a good learning direction.
- Second, the curriculum of positive pairs brings slightly higher improvement than that of negative pairs. This suggests that the ability to capture positive matching signals is more critical than being able to discard negative signals. A possible explanation is that positive signals are more focused while the negative ones are diffuse.

Furthermore, to investigate the influence of samples’ difficulty changes during the training, we replace the curriculum of negative pairs by only using easy or hard pairs, and the curriculum of positive pairs is disabled to avoid additional influence. The experimental results are shown in the bottom of Table 3.

#### 4.4.2 Influence of Scoring Models

We proposed two scoring models for 

| Pos. Neg. | MAP | MRR | NDCG@1 | NDCG@3 | NDCG@10 |
|----------|-----|-----|--------|--------|---------|
| × ×      | 0.5500 | 0.5601 | 0.4024 | 0.5478 | 0.6160 |
| ✓ ×      | 0.5750 | 0.5843 | 0.4222 | 0.5811 | 0.6391 |
| ✓ ✓      | 0.5794 | 0.5888 | 0.4281 | 0.5841 | 0.6432 |
| × Easy   | 0.5240 | 0.5350 | 0.3677 | 0.5200 | 0.5939 |
| × Hard   | 0.5698 | 0.5792 | 0.4173 | 0.5741 | 0.6339 |

Table 3: Ablation study of COCA+DCL. Mark “✓” and “×” indicate whether a curriculum is used or not. “Easy” or “hard” means only easy/hard samples are used for training.

**Table 4: Performance with different difficulty scorers.**

| Pos. Neg. | MAP | MRR | NDCG@1 | NDCG@3 | NDCG@10 |
|----------|-----|-----|--------|--------|---------|
| None None | 0.5500 | 0.5601 | 0.4024 | 0.5478 | 0.6160 |
| BM25 BM25 | 0.5661 | 0.5763 | 0.4159 | 0.5697 | 0.6293 |
| BM25 BERT | 0.5600 | 0.5701 | 0.4081 | 0.5632 | 0.6244 |
| BERT BM25 | 0.5794 | 0.5888 | 0.4281 | 0.5841 | 0.6432 |
| BERT BERT | 0.5652 | 0.5735 | 0.4152 | 0.5684 | 0.6290 |

Figure 4: Performance with different hyperparameters.

As can be seen, training with only hard negatives is even better than using all negatives (first row in the table). This finding is consistent with existing studies on using hard negatives to facilitate the optimization of dense retrievers [47, 60, 62]. However, only using easy samples for training makes the performance drop sharply. This is because the easy negatives cannot provide sufficient “contrastive signals” for learning the matching between search context and candidate documents. This is also why we design our curriculum of negative pairs as gradually shrinking to only hard negatives (details are presented in Section 3.3.2). Finally, dynamically adjusting the learning difficulty through curriculum is beneficial for model training (e.g., MAP is improved from 0.5698 to 0.5740). This demonstrates again the effectiveness of applying curriculum learning.

#### 4.4.3 Influence of Hyperparameters

In DCL, \(\delta\) (in Equation 2) and \(\eta\) (in Equation 4) are two hyperparameters that control the degree...
of difficulty in the initial set of positive pairs and in the set of final negative pairs. They are determined according to the validation set. We show their impact in Figure 4.

As we can see, when $\delta$ is small ($< 0.4$, i.e., using very easy positive pairs at beginning), the performance is high. When $\delta$ becomes large (i.e., including more difficult positive pairs at beginning), the performance drops. This confirms our hypothesis that the optimization process can be confused with more difficult pairs at beginning. When $\delta = 1.0$, the curriculum of positive pairs is disabled, so all positive pairs are learned in a random order. We can see that this common strategy used in the previous studies is suboptimal.

For negative pairs, when $\eta$ is too small, we end the training process with a very small subset of the training data consisting of highly-ranked negative documents. In this case, the sampled documents may contain false negatives. The best performance is obtained with $\eta = 0.7$, i.e., some mixture of easy and hard negative pairs is used at the end. However, when $\eta$ is too large (i.e., 1.0), all negative pairs are randomly sampled during the whole training, the model’s performance also decreases because of the disabled curriculum effect. These observations confirm the impact of both curricula and suggest that the right degree of difficulty in the initial and final pools of samples may influence the effectiveness of DCL.

### 4.4.4 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.

We compare COCA+DCL with Duet, HBA, RICR, and COCA and show the results regarding MAP and NDCG@3 in the left side of Figure 5. First, COCA+DCL improves the performance of COCA across all three session bins. This shows DCL’s high robustness for different kinds of search context. Second, we can see the ad-hoc ranking method Duet performs worse than other context-aware ranking methods. This highlights once again that modeling historical user behavior is essential for improving document ranking performance. Third, COCA+DCL performs better on short sessions than on long ones. We hypothesize that those longer sessions are inherently more difficult to understand, and a similar trend in baseline methods may corroborate this. This can be due to the fact that a long session may contain more diverse intents or exploratory search.

### 4.4.5 Effect of Modeling User Behavior Progression

It is important to study how the modeled search context contributes to document ranking as a search session progresses. We compare COCA+DCL with HBA, RICR, and COCA at individual query positions in short (S), medium (M), and long (L) sessions. The results are presented in the right side of Figure 5. Due to space limitations, long sessions with more than seven queries are omitted.

We can see that the ranking performance generally improves when the short and medium sessions progress (e.g., S2 is higher than S1) because more search context information becomes available for predicting the next click. It benefits COCA+DCL and two baselines (COCA and HBA), while COCA+DCL improves more rapidly by better exploiting the context. One interesting observation is that, when the search sessions become longer (e.g., from L4 to L7), the gain of DCL decreases. We attribute this to the noisier nature of long sessions.

### 5 CONCLUSION AND FUTURE WORK

In this work, we proposed a novel curriculum learning framework for context-aware document ranking. Two complementary curricula were designed for learning positive and negative context-document pairs in an easy-to-hard manner. With these curricula, the model’s capability of capturing matching signals and identifying mismatching signals is gradually enhanced. We conducted experiments with three recently proposed methods on two large-scale datasets. The results clearly demonstrate the effectiveness and wide applicability of our framework. Besides, we also investigated the influence of different settings on applying curriculum learning to context-aware document ranking. Our work is an early attempt to apply curriculum learning to IR, and there is still much space to be explored. For example, it may be useful to mine the most valuable negative candidate documents. Considering query and document weighting in computing the difficulty is also a future direction.

**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), Beijing Outstanding Young Scientist Program (No. BJJWZYJH012019100020098), and Beijing Academy of Artificial Intelligence (BAAI).
