Hard Negatives or False Negatives: Correcting Pooling Bias in Training Neural Ranking Models

Yinqiong Cai
CAS Key Lab of Network Data Science and Technology, ICT, CAS
University of Chinese Academy of Sciences
Beijing, China
caiyinqiong18s@ict.ac.cn

Jiafeng Guo*
CAS Key Lab of Network Data Science and Technology, ICT, CAS
University of Chinese Academy of Sciences
Beijing, China
guojiafeng@ict.ac.cn

Yixing Fan
CAS Key Lab of Network Data Science and Technology, ICT, CAS
University of Chinese Academy of Sciences
Beijing, China
fanyixing@ict.ac.cn

Qingyao Ai
Dept. CS&T, Beijing National Research Center for Information Science and Technology, Tsinghua University
Beijing, China
aiqy@tsinghua.edu.cn

Ruqing Zhang
CAS Key Lab of Network Data Science and Technology, ICT, CAS
University of Chinese Academy of Sciences
Beijing, China
zhangruqing@ict.ac.cn

Xueqi Cheng
CAS Key Lab of Network Data Science and Technology, ICT, CAS
University of Chinese Academy of Sciences
Beijing, China
cxq@ict.ac.cn

ABSTRACT

Neural ranking models (NRMs) have become one of the most important techniques in information retrieval (IR). Due to the limitation of relevance labels, the training of NRMs heavily relies on negative sampling over unlabeled data. In general machine learning scenarios, it has shown that training with hard negatives (i.e., samples that are close to positives) could lead to better performance. Surprisingly, we find opposite results from our empirical studies in IR. When sampling top-ranked results (excluding the labeled positives) as negatives from a stronger retriever, the performance of the learned NRM becomes even worse. Based on our investigation, the superficial reason is that there are more false negatives (i.e., unlabeled positives) in the top-ranked results with a stronger retriever, which may hurt the training process; The root is the existence of *pooling bias* in the dataset constructing process, where annotators only judge and label very few samples selected by some basic retrievers. Therefore, in principle, we can formulate the false negative issue in training NRMs as learning from labeled datasets with pooling bias. To solve this problem, we propose a novel Coupled Estimation Technique (CET) that learns both a relevance model and a selection model simultaneously to correct the pooling bias for training NRMs. Empirical results on three retrieval benchmarks show that NRMs trained with our technique can achieve significant gains on ranking effectiveness against other baseline strategies.

*Jiafeng Guo is the corresponding author.

CIKM '22, October 17–21, 2022, Atlanta, GA, USA

ACM ISBN 978-1-4503-9236-5/22/10…$15.00
https://doi.org/10.1145/3511808.3557343

CCS CONCEPTS

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

KEYWORDS

Neural Ranking Models; Negative Sampling; Pooling Bias

ACM Reference Format:
Yinqiong Cai, Jiafeng Guo, Yixing Fan, Qingyao Ai, Ruqing Zhang, and Xueqi Cheng. 2022. Hard Negatives or False Negatives: Correcting Pooling Bias in Training Neural Ranking Models. 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/10.1145/3511808.3557343

1 INTRODUCTION

To balance the effectiveness and efficiency, modern information retrieval (IR) systems generally employ a multi-stage architecture where a *retriever* (e.g., BM25 [38], ANCE [47]) is firstly used to quickly retrieve a few potentially relevant documents from a large collection and then a *ranker* (e.g., monoBERT [31], CEDR [27]) is employed to further analyze and re-rank these documents for precise ranking [29]. Recently, deep learning techniques have been widely applied to construct the rankers, and these neural ranking models (NRMs) have shown advanced performance in modern IR systems [16]. Similar to other deep learning models, most existing NRMs are data-hungry. Thus, with limited relevance data in IR, i.e., sparse-labeled documents and a large number of unlabeled documents for each query, the training of NRMs heavily relies on negative sampling over unlabeled data [9, 30]. To facilitate NRMs training, different negative sampling strategies have been explored [18, 27]. Among them, the most popular strategy is to sample negatives from the top-ranked results (excluding the labeled positives) returned by the retriever [24, 27].

Compared to random negative samples, studies in the machine learning community have shown that training with hard negative

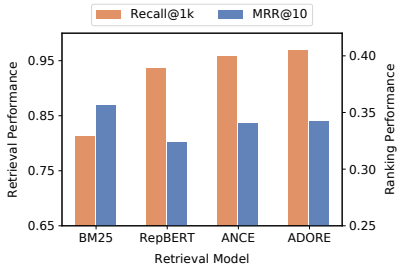

Figure 1: Performance on the MS MARCO Passage Ranking task for four different retrievers (Recall@1k) and BERT-base rankers [31] (MRR@10). Negatives for rankers training are randomly sampled from top-1000 passages returned by corresponding retrievers.

samples (i.e., negatives that are similar to positives) can better improve the performance of deep learning models [39]. In IR scenarios, such hard negatives can be easily obtained from the top-ranked results of a stronger retriever, such as RepBERT [50], ANCE [47], and ADORE [49], which has more sophisticated structure and much better recall capacity than traditional statistical retrievers such as BM25. Surprisingly, as shown in our empirical experiments on the MS MARCO Passage Ranking task (see Figure 1), directly sampling hard negatives from an improved retriever for NRMs training could lead to inferior ranking performance. For example, RepBERT [50] is a better retriever than BM25 (e.g., the Recall@1k of RepBERT is 15% better than BM25), but the BERT-base ranker [31] trained with negatives sampled from RepBERT is 9% worse than the same ranker trained with samples from BM25 on MRR@10.

To investigate the cause of this phenomenon, we trace back to the hard negative samples from different retrievers for NRMs training. Interestingly, we observe that the training data constructed with stronger retrievers is more likely to be riddled with false negatives (i.e., unlabeled positives). This observation is consistent with the previous study [3] which found that the top-ranked passages returned by the retrievers on the MS MARCO leaderboard often appear as good as, or even superior to, the qrels (i.e., positive samples labeled by annotators explicitly). Without proper treatment, the increasing rate of false negatives in the training data would inevitably hurt the NRMs training process [13, 52]. To this end, one may image that a direct solution to this problem is to apply some heuristic or statistical methods [13] to classify the false negatives from the negative samples during training.

However, if we take a deeper look at the construction of the training data, we may find that the above false negative problem is actually related to the *pooling bias*[1] in the labeling process in IR. The pooling bias happens when annotators are instructed to judge and label documents within a small set pooled by some basic retrievers (e.g., Bing for the MS MARCO dataset construction [30]). In other words, the labeling process is *biased* by basic retrievers one have chosen. Under this setting, all the documents outside the pooled set are left unjudged, leading to the potential existence of unlabeled positives. When these unlabeled positives are sampled as negatives, the above false negative problem emerges. The stronger the sampler is, the severer the problem becomes. Therefore, we argue that the root of the above false negative problem is the pooling bias in IR, and we refer more detailed discussions to Section 3.2.

In this way, we no longer treat the false negative issue in training NRMs as a simple classification problem, but rather formulate it as a learning problem from the biased dataset. Inspired by previous studies on unbiased learning-to-rank [1, 19, 45], we propose a Coupled Estimation Technique (CET) to solve the NRM training problem from the labeled dataset with pooling bias. Specifically, CET attempts to train a selection model to estimate the selection propensity of each document, and a relevance model to estimate the true relevance degree of each document given the query. We demonstrate that these two models can be trained in an unbiased way with the help of each other over the biased dataset. Therefore, CET employs a coupled learning procedure to learn both models simultaneously. Based on this, we are able to weaken false negatives and derive high quality hard negatives for NRMs training.

To evaluate the effectiveness of CET, we conduct extensive experiments on three retrieval benchmarks. Empirical results demonstrate that NRMs learned with CET can achieve significantly better performance over that learned with state-of-the-art techniques to address the false negative issue. Moreover, CET is shown to be effective in training different rankers with hard negatives from a variety of retrievers. In addition, unlike baseline techniques that suffer from high sensitivity w.r.t hyperparameters (which are usually set according to the prior information about the distribution of false negatives in the dataset), CET is demonstrated to be more stable and robust in general.

2 RELATED WORK

In this section, we present topics related to our work: bias in information retrieval and negative sampling strategies in IR.

2.1 Bias in Information Retrieval

Data for IR tasks are usually collected in two ways: explicit labeling by annotators (i.e., labeled data) and implicit feedback from users (i.e., click data). We review typical bias problems in them.

Bias in labeled data. Early IR systems are built on the relevance data constructed with a pooling process [41]: existing retrieval models are firstly used to get a document pool, and then annotators are instructed to label their relevance. The idea behind the pooling technique is to find enough relevant documents such that the relevance data is sufficiently complete and unbiased when only partial documents are judged [7]. However, with the growth of the document set size and the development of IR techniques, the assumption of approximately complete labeling becomes invalid [3, 8, 43]. In this case, the pooling bias problem is identified by researchers, which would underestimate the effectiveness of new IR systems with standard evaluation [5]. For example, Webber and Park [46] proposed to estimate the degree of bias against a new retriever to adjust the evaluation score. Differing from the above early works, recently researchers realize the detrimental effect of pooling bias to IR models training [3, 43], where unlabeled positives, called false negatives in this work, would make the learned models biased and hurt their performance. However, these pioneers only discussed this problem preliminarily and intuitively. In this work, we will show a exhaustive analysis and give a formal definition to it.

[1]Note that the pooling bias in this study focuses on the effect on model training, not the evaluation.

Bias in click data. Due to the cost of annotators labeling, click data is easily collected and has developed to be a critical resource for IR models training. However, various bias problems, such as position bias [19], selection bias [33, 45] and presentation bias [48], make it difficult to be directly leveraged as training data. For example, position bias is caused by the position where a document is displayed to users, making higher ranked documents more likely to be clicked. Ovaisi et al. [33] claimed that users rarely have the chance or energy to check about all documents in the lists. In this case, lower-ranked relevant documents have zero probability of being clicked, thus leading to the selection bias. To make full use of click data, studies on unbiased learning-to-rank [2] have attracted a lot of attention. Among them, the counterfactual estimation technique from causal inference, such as inverse propensity weighting (IPW) [1, 19, 45] and Heckman correction [33], is used to solve these bias problems in click data. Different from these studies, in this work, we focus on the bias in labeled data, where we do not possess any prior knowledge on the labeling process and cannot explicitly quantify the bias distribution with user studies.

2.2 Negative Sampling Strategies in IR

In practice, training data for IR models usually consist of sparse-labeled documents for a set of queries and a large number of unlabeled documents. Among the labeled documents, there are often limited [7, 35] or even no [30, 36] samples with explicit negative labels. Therefore, negative sampling over unlabeled documents is usually necessary for training IR models.

To find informative negatives for models training, many methods have been explored in IR. Random sampling from the document set is the simplest and most direct way to get negatives, which is a widely used strategy in existing works [18, 20]. Nonetheless, such approach is sub-optimal because random negatives have been proven to be too easy to learn effective models for generalizing to sophisticated testing cases [22, 39]. Instead, negative samples that are more difficult to be distinguished from positive ones are more desired. There have been studies showing that these *hard* negative samples could improve model generalization and accelerate convergence [32, 40]. However, efficiently identifying such informative negative samples emerges as a challenge since it is computationally infeasible to examine all possible samples. Among current works in the IR field, the most commonly used hard negatives are the top-ranked documents of a strong retriever [37, 47, 51].

Recently, with the prevalence of pre-training methods, the availability of stronger retrievers [15] shows new potentials to provide hard negatives for IR models training. Meanwhile, researchers begin to realize the severity of false negatives for neural retrievers and rankers training [4, 13, 14, 28]. For example, Ding et al. [13] found that 70% of the top-ranked passages returned by their retriever are actually positives or highly relevant in MS MARCO [30]. To address this problem, there have been several studies trying to filter false negatives during the negative sampling process with some heuristic methods. For example, RocketQA [13] proposed to train another ranking model in advance to evaluate relevance scores for all unlabeled documents and sets a hard threshold empirically to remove those potentially false negatives. Additionally, RANCE [34] presented a special sampling technique to filter false negatives. It estimates the negative sampling distribution relying on a separate dataset with complete relevance labeling, which, unfortunately, is usually not available in practice. These two works are highly related to this study, and we implement two baselines based on them respectively to compare with our method.

3 PROBLEM DESCRIPTION

In this section, we briefly review the general training paradigm of NRMs on labeled datasets and introduce the pooling bias in detail. We show how this pooling bias leads to the false negative problem in negative sampling as observed in the Introduction Section. Based on these analysis, we formulate the false negative issue in training NRMs as a learning problem over biased labeled datasets.

3.1 Training NRMs on Labeled Dataset

We begin by introducing the general setting of NRMs learning with fully labeled data, also referred to as the **Full-Information Setting** [19]. Ideally, given a query $q \in \mathcal{Q}$, the relevance r_i of each document $d_i \in \mathcal{D}$ is known beforehand. For simplicity, we assume that the relevance is binary (i.e., $r \in \{0, 1\}$) and one can easily extend it to the multi-level relevance case. In practice, the pairwise learning setting tends to be more effective for ranking models training and has been widely adopted [6, 17]. Thus, here we consider that the ranking model R is defined on a query-document pair, and the loss function is defined on a triple (q, d_i, d_j). Let r_i^+ and r_j^- represent that d_i is relevant (i.e., $r_i = 1$) and d_j is irrelevant (i.e., $r_j = 0$) for q. Let x_i and x_j denote feature vectors from d_i and d_j as well as q. The risk function for NRMs learning is defined as:

$$\mathcal{R}_{full}(R) = \int L\left(R(x_i), R(x_j)\right)\ dP(x_i, r_i^+, x_j, r_j^-). \tag{1}$$

Given the fully labeled dataset, the ranker R can be learned by minimizing the empirical risk function as follows:

$$\hat{R}_{full} = \arg\min_R \sum_{q \in \mathcal{Q}} \sum_{(d_i, d_j) \in \mathcal{D}} L\left(R(x_i), R(x_j)\right), \tag{2}$$

where the document pair (d_i, d_j) denotes that d_i is a document with $r_i = 1$ and d_j is a document with $r_j = 0$ for the query q.

However, in practice, it is impossible to judge and label all the documents in the corpus for each query. To reduce the annotation effort, existing methods usually apply some basic retrievers to select a small set of documents for labeling [41]. That is, for each query, only a few selected documents can be judged and labeled by annotators. In this case, NRMs are trained under a **Partial Information Setting** [33], where only a part of relevance information for each query is available and most remains unobserved. With this labeled dataset for training NRMs, except for the very few labeled documents, all unlabeled documents are usually deemed to be negatives (i.e., irrelevant). Then the risk function and minimization of empirical risk function could be reformalized as:

$$\begin{aligned}\mathcal{R}_{part}(R) = &\int L\left(R(x_i), R(x_j)\right)\ dP(x_i, l_i^+, x_j, l_j^-) \\ &+ \int L\left(R(x_i), R(x_j)\right)\ dP(x_i, l_i^+, x_j, ul_j),\end{aligned} \tag{3}$$

$$\hat{R}_{part} = \arg\min_R \sum_{q \in \mathcal{Q}} \sum_{\substack{d_i \in \mathcal{D}^+ \\ d_j \in \mathcal{D}^- \cup \mathcal{D}^{ul}}} L\left(R(x_i), R(x_j)\right), \tag{4}$$

where l_i^+ and l_j^- denote that d_i is labeled as relevant (i.e., $l_i = 1$) and d_j is labeled as irrelevant (i.e., $l_j = 0$) for q respectively, ul_j denotes that d_j is an unlabeled document for q, $\mathcal{D} = \mathcal{D}^+ \cup \mathcal{D}^- \cup \mathcal{D}^{ul}$, and these three document sets are disjointing. In practice, the labeled negatives (i.e., $\mathcal{D}^-$) are often missing [30, 36], thus it is essential to utilize documents in $\mathcal{D}^{ul}$ for learning effective NRMs [18, 24, 27].

3.2 Pooling Bias in Labeled Data

As mentioned in Section 3.1, the construction of labeled datasets relies on some basic retrievers, called pooling systems, to select a small set of most promising documents for labeling. The pooling technique is very important for IR since it can significantly reduce manual effort for labeling. Therefore, it has been widely adopted in benchmark construction [7, 30]. While the pooling technique does save labeling efforts, it introduces the undesired bias into labeled data, namely pooling bias here. That is, the labeled data is biased by the preference of basic retrievers used in the pooling process. A direct consequence of this pooling bias is the potential existence of unlabeled positives, i.e., some relevant documents might not be preferred by basic retrievers and thus not be selected for labeling.

When the unlabeled data are sampled as negatives in training NRMs, those unlabeled positives become false negatives if they are unfortunately selected by the sampler. In general, unlabeled positives are the minority in the massive unlabeled data. Therefore, when the random sampling strategy is adopted, the false negative issue would not be severe. This explains why the traditional training paradigm with negative sampling works for NRMs. However, previous studies have shown that uniformly sampling negatives from $\mathcal{D}^{ul}$ often fails to learn an effective ranking model, since random negatives are too easy to produce effective parameter updates [22, 39]. Therefore, more advanced studies employ the hard negative sampling strategy, where the top-ranked results (excluding labeled positives) returned by a strong retriever are used as negatives for NRMs training [18, 27]. However, such sampling strategies would increase the selection probability of false negatives, since unlabeled positives are more likely to be ranked at top positions by a stronger retriever [12]. This in turn will hurt NRMs training, which is exactly the observation we have mentioned in Figure 1.

Based on the above analysis, we can see that the pooling bias is the root of the false negative issue in training NRMs with hard negative sampling strategies. Therefore, directly identifying the false negatives from unlabeled data with some classification models may not touch the heart of the problem. In principle, we can formulate the false negative issue in training NRMs as a learning problem from labeled datasets with pooling bias.

4 OUR APPROACH

In this section, we introduce our approach in detail. We first analyze the problem with a counterfactual learning framework (Section 4.1). Then we propose a coupled estimation technique (Section 4.2) to solve the NRM learning problem with the biased dataset.

4.1 Bias Correction Analysis

Existing works on bias correction mostly focus on the click data [2, 19], where the click of a document is affected by its position, popularity, etc. Pooling bias discussed in this study is that the label of a document is affected by whether it is selected during the pooling process. Based on the common ground, we follow the inverse propensity weighting (IPW) framework in [19, 45] to solve the pooling bias in labeled datasets.

In this work, we focus on the typical case in popular large-scale retrieval benchmarks [21, 30, 36], where only relevant documents in the judgement pool are labeled by annotators and there is no explicitly labeled negatives (i.e., $\mathcal{D}^- = \varnothing$). As a result, it is unavailable which documents have been selected during the pooling process, which makes it challenging for addressing the pooling bias. With this labeled dataset, we define three notations for each query-document pair (q, d) in it. Besides the relevance r and labeling l defined in Section 3.1, we use $s \in \{0, 1\}$ to indicate whether the document d is selected into the judgement pool for the query q. The merely available information for the labeled dataset is l, and here, we consider the noise-free labeling where a document with $l = 1$ must be relevant and selected.

Considering that annotators have been well instructed, for the pair (q, d_i) denoted as x_i, we have the following premise:

$$P(l_i^+ \mid x_i) = P(s_i^+ \mid x_i) \cdot P(r_i^+ \mid x_i), \tag{5}$$

where s_i^+ denotes that the document d_i is selected (i.e., $s_i = 1$) for q. Besides, for an unlabeled document d_j, it is more likely to be irrelevant if it has higher probability being selected, that is,

$$P(r_j^- \mid ul_j, x_j) \propto P(s_j^+ \mid x_j). \tag{6}$$

Then, we can learn a bias corrected relevance model with an IPW-based risk function and empirical risk function on the labeled dataset with $\mathcal{D}^- = \varnothing$:

$$\begin{aligned}
\mathcal{R}_{IPW}(R) &= \int \frac{L(R(x_i), R(x_j))}{\frac{P(s_i^+|x_i)}{P(s_j^+|x_j)}}\, dP(x_i, l_i^+, x_j, ul_j) \\
&\propto \iint \frac{L(R(x_i), R(x_j))}{\frac{P(l_i^+|x_i)}{P(r_i^+|x_i)} \cdot \frac{1}{P(r_j^-|ul_j, x_j)}}\, dP(x_i, l_i^+)\, dP(x_j, ul_j) \\
&= \iint \frac{L(R(x_i), R(x_j))\, dP(x_i, l_i^+)\, dP(x_j, ul_j)}{\frac{P(l_i^+|x_i)}{P(r_i^+|x_i)} \cdot \frac{P(ul_j|x_j)}{P(r_j^-, ul_j|x_j)}} \\
&= \iint \frac{L(R(x_i), R(x_j))\, dP(x_i, l_i^+)\, dP(x_j, ul_j)}{\frac{P(l_i^+|x_i)}{P(r_i^+|x_i)} \cdot \frac{P(ul_j|x_j)}{P(r_j^-|x_j)}} \\
&= \iint L(R(x_i), R(x_j))\, dP(x_i, r_i^+)\, dP(x_j, r_j^-) \\
&= \int L(R(x_i), R(x_j))\, dP(x_i, r_i^+, x_j, r_j^-) \\
&= \mathcal{R}_{full}(R),
\end{aligned} \tag{7}$$

$$\hat{R}_{IPW} = \arg\min_R \sum_{q \in Q} \sum_{\substack{d_i \in \mathcal{D}^+ \\ d_j \in \mathcal{D}^{ul}}} \frac{L(R(x_i), R(x_j))}{\frac{P(s_i^+|x_i)}{P(s_j^+|x_j)}}. \tag{8}$$

For Eq. (7), it is assumed that the relevance and labeling of d_i are independent from d_j (empirical results show that it works well with this assumption, even one may think that it is not strictly correct), the second step uses Eq. (5) & (6), the third step uses the conditional probability formula, and the fourth step uses the premise that a labeled document must be relevant (i.e., $l^+ \Rightarrow r^+$, and thus $r^- \Rightarrow ul$ with $\mathcal{D}^- = \varnothing$). With Eq. (7), it implies that the model R optimized

with the IPW empirical risk minimization on labeled datasets can produce the same relevance model trained with the relevance r,

$$\arg\min_R \mathcal{R}_{IPW}(R) = \arg\min_R \mathcal{R}_{full}(R). \tag{9}$$

To optimize the relevance model with Eq. (8), we need to estimate the selection probability $P(s_i^+ \mid x_i)$ for each document $d_i \in \mathcal{D}$. Ideally, we can learn a selection model S with the risk function and minimization of empirical risk function:

$$\begin{aligned} \mathcal{R}_{full}(S) &= \int L(S(x_i), S(x_j))\, dP(x_i, s_i^+, x_j, s_j^-), \\ \hat{S}_{full} &= \arg\min_S \sum_{q\in Q} \sum_{(d_i,d_j)\in\mathcal{D}} L(S(x_i), S(x_j)), \end{aligned} \tag{10}$$

where s_i^+ and s_j^- denote that d_i is selected (i.e., $s_i = 1$) and d_j is not selected (i.e., $s_j = 0$) by pooling systems for the query q, and the document pair (d_i, d_j) denotes that d_i is a document with $s_i = 1$ and d_j is a document with $s_j = 0$ for q. However, similar to the relevance information, the selection information is also partial with $\mathcal{D}^- = \emptyset$, since only the labeled relevant document's selection information is available and an unlabeled document may be just irrelevant but not unselected. Nevertheless, for the unlabeled document d_j, it is more likely to be unselected if it has higher relevance probability, that is,

$$P(s_j^- \mid ul_j, x_j) \propto P(r_j^+ \mid x_j). \tag{11}$$

Then, similar to the relevance model, the bias corrected selection model could be learned with an IPW-based risk function:

$$\mathcal{R}_{IPW}(S) = \int \frac{L(S(x_i), S(x_j))}{\frac{P(r_i^+|x_i)}{P(r_j^+|x_j)}}\, dP(x_i, l_i^+, x_j, ul_j), \tag{12}$$

$$\hat{S}_{IPW} = \arg\min_S \sum_{q\in Q} \sum_{\substack{d_i\in\mathcal{D}^+ \\ d_j\in\mathcal{D}^{ul}}} \frac{L(S(x_i), S(x_j))}{\frac{P(r_i^+|x_i)}{P(r_j^+|x_j)}}. \tag{13}$$

Based on Eq. (5) & (11) and the fact that a labeled document must be selected (i.e., $l^+ \Rightarrow s^+$, and thus $s^- \Rightarrow ul$ with $\mathcal{D}^- = \emptyset$), Eq. (12) is easily proven as Eq. (7) and the details are omitted here.

With Eq. (8) & (13), we can find that the key of obtaining a debiased relevance model is to estimate $P(s_i^+ \mid x_i)$, while the key of building a debiased selection model is to estimate $P(r_i^+ \mid x_i)$ for each (q, d_i) in the dataset. This indicates that the estimation of selection propensities and relevance scores for documents is coupled with each other, where a better selection model can help to train a better relevance model and vice versa. Based on this observation, we propose a coupled estimation technique to train both models simultaneously to correct the pooling bias.

4.2 Coupled Estimation Technique

Based on the above analysis, we propose a Coupled Estimation Technique (CET) to train NRMs on labeled datasets. As shown in Fig. 2, the framework of CET includes three components:

- The relevance model R (implemented with NRMs) that estimates the relevance score for the pair (q, d);
- The selection model S that estimates the propensity of the document d being selected into the judgement pool for the query q during annotators labeling process;
- The coupled learning algorithm that optimizes the relevance model and selection model jointly on the labeled dataset.

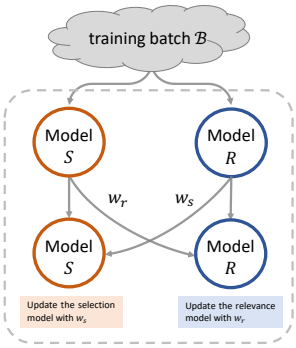

Figure 2: The framework of Coupled Estimation Technique (CET). The model S and model R denote the selection model and the relevance model respectively.

Relevance Model. In this work, we implement the relevance model R with a BERT-based architecture parameterized by $\boldsymbol{\theta}$. Following previous studies [31], we concatenate the query q and the document d with special delimiting tokens and feed them into Transformer layers [44]. Then, a multi-layer perceptron (MLP) function is applied over the hidden state of the special token [CLS] to obtain the relevance score. Let $R_{\boldsymbol{\theta}}(q, d)$ be the relevance score for the query-document pair (q, d), which can be estimated with,

$$R_{\boldsymbol{\theta}}(q, d) = MLP(BERT_{cls}([CLS] + q + [SEP] + d + [SEP])). \tag{14}$$

Then, we use the pairwise cross entropy loss as L in Eq. (4) for the relevance model training:

$$\mathcal{L}(R) = \frac{1}{|Q|} \sum_{q\in Q} \sum_{\substack{d_i\in\mathcal{D}^+ \\ d_j\in\mathcal{D}^{ul}}} -\log \frac{e^{R_{\boldsymbol{\theta}}(q,d_i)}}{e^{R_{\boldsymbol{\theta}}(q,d_i)} + e^{R_{\boldsymbol{\theta}}(q,d_j)}}. \tag{15}$$

Selection Model. According to the construction process of labeled datasets, the pooling systems used to select documents are usually basic retrievers. Based on this principle, we build the selection model S with the same architecture as R. Specifically, for the selection model S (parameterized by $\boldsymbol{\phi}$), let $S_{\boldsymbol{\phi}}(q, d)$ be the estimated selection propensity for the pair (q, d), which is calculated with the same method as Eq. (14) but with a different set of parameters. Then we train the model S by minimizing the following loss function:

$$\mathcal{L}(S) = \frac{1}{|Q|} \sum_{q\in Q} \sum_{\substack{d_i\in\mathcal{D}^+ \\ d_j\in\mathcal{D}^{ul}}} -\log \frac{e^{S_{\boldsymbol{\phi}}(q,d_i)}}{e^{S_{\boldsymbol{\phi}}(q,d_i)} + e^{S_{\boldsymbol{\phi}}(q,d_j)}}. \tag{16}$$

Coupled Learning Algorithm. As discussed in Section 4.1, the estimation of selection propensities and relevance scores for documents is coupled with each other. Thus, we employ a coupled learning algorithm to train two models simultaneously and achieve bias correction learning. During the coupled learning process, each model estimates the bias weight for the training data of the other model with their current parameters, and then injects the weight into loss functions for next update. In this way, the relevance model R and the selection model S are promoted mutually. An overview of the complete algorithm is shown in Algorithm 1.

We first initialize all parameters (i.e., $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$) of two models. For each pair (q, d_i), we estimate its relevance score and selection

Algorithm 1: The coupled learning algorithm.

Input: query set $\mathcal{Q}$, document set $\mathcal{D}$, the labeling l, α, τ
Output: relevance model R, selection model S

Initialize parameters $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ in models R and S;
repeat
  Sample a batch of triples $(q, d_i, d_j) \in \mathcal{B}$ from $\mathcal{D}$ with l ;
  for $(q, d_i, d_j) \in \mathcal{B}$ **do**
    Estimate $w_r(q, d_i, d_j)$ with Eq. (18);
    Estimate $w_s(q, d_i, d_j)$ with Eq. (19);
  end
  Compute $\mathcal{L}^{'}(R)$ on $\mathcal{B}$ with Eq. (20);
  Compute $\mathcal{L}^{'}(S)$ on $\mathcal{B}$ with Eq. (21);
  $\boldsymbol{\theta} = \boldsymbol{\theta} + \alpha \cdot \frac{\partial \mathcal{L}^{'}(R)}{\partial \boldsymbol{\theta}}, \boldsymbol{\phi} = \boldsymbol{\phi} + \alpha \cdot \frac{\partial \mathcal{L}^{'}(S)}{\partial \boldsymbol{\phi}}$;
until *Convergence*;
return R, S

propensity with current parameters and convert them into probability distributions:

$$P(r_i^+ \mid x_i) = \frac{e^{R_{\boldsymbol{\theta}}(q,d_i)}}{\sum_{d_k \in \mathcal{D}} e^{R_{\boldsymbol{\theta}}(q,d_k)}},\ P(s_i^+ \mid x_i) = \frac{e^{S_{\boldsymbol{\phi}}(q,d_i)}}{\sum_{d_k \in \mathcal{D}} e^{S_{\boldsymbol{\phi}}(q,d_k)}}. \quad (17)$$

As shown in Eq. (17), the use of the softmax function assumes that the examination probabilities on different documents in $\mathcal{D}$ will sum up to 1, which is not true in practice. This, however, does not hurt the effectiveness of models training. In fact, the predicted values of $P(r_i^+ \mid x_i)$ and $P(s_i^+ \mid x_i)$ have a minor effect on the bias correction learning as long as their relative proportions are correct. Thus, with Eq. (8) & (13), the actual bias correction weights used in CET are:

$$w_r(q, d_i, d_j) = \frac{P(s_j^+ \mid x_j)}{P(s_i^+ \mid x_i)} = \frac{e^{S_{\boldsymbol{\phi}}(q,d_j)/\tau}}{e^{S_{\boldsymbol{\phi}}(q,d_i)/\tau}}, \quad (18)$$

$$w_s(q, d_i, d_j) = \frac{P(r_j^+ \mid x_j)}{P(r_i^+ \mid x_i)} = \frac{e^{R_{\boldsymbol{\theta}}(q,d_j)/\tau}}{e^{R_{\boldsymbol{\theta}}(q,d_i)/\tau}}, \quad (19)$$

where $\tau > 0$ is a hyperparameter to control the scale of bias weights. By injecting the bias weights into Eq. (15) & (16), we can obtain the IPW loss functions:

$$\mathcal{L}^{'}(R) = \frac{1}{|\mathcal{Q}|} \sum_{q \in \mathcal{Q}} \sum_{\substack{d_i \in \mathcal{D}^+ \\ d_j \in \mathcal{D}^{ul}}} -w_r(q, d_i, d_j) \cdot \log \frac{e^{R_{\boldsymbol{\theta}}(q,d_i)}}{e^{R_{\boldsymbol{\theta}}(q,d_i)} + e^{R_{\boldsymbol{\theta}}(q,d_j)}}, \quad (20)$$

$$\mathcal{L}^{'}(S) = \frac{1}{|\mathcal{Q}|} \sum_{q \in \mathcal{Q}} \sum_{\substack{d_i \in \mathcal{D}^+ \\ d_j \in \mathcal{D}^{ul}}} -w_s(q, d_i, d_j) \cdot \log \frac{e^{S_{\boldsymbol{\phi}}(q,d_i)}}{e^{S_{\boldsymbol{\phi}}(q,d_i)} + e^{S_{\boldsymbol{\phi}}(q,d_j)}}. \quad (21)$$

Then, parameters $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ are updated with $\mathcal{L}^{'}(R)$ and $\mathcal{L}^{'}(S)$ respectively. This process is repeated until the algorithm converges. After the training process, only the relevance model R is retained to evaluate relevance scores for query-document pairs in the dataset. It should be noted that although training two models simultaneously will bring higher computational cost, the evaluation process is the same inference latency as usual.

5 EXPERIMENTAL SETTINGS

This section presents the experimental settings, including datasets, evaluation metrics, implementation details, and baselines.

5.1 Datasets Description

We conduct experiments on three retrieval benchmarks to evaluate the effectiveness of CET:

- **MS MARCO [30]:** The *passage ranking task* provides about 503k queries paired with relevant passages for training. Each query is associated with sparse relevance labels of one (or very few) passages labeled as relevant and no passages explicitly labeled as irrelevant. It also has approximately 7k queries in dev and test sets. For relevance labels creation, annotators label relevant passages from the top-10 passages retrieved by a existing retrieval system at Bing. The *document ranking task* has about 367k training queries, 5k dev queries, and 5k test queries. The document that produces a relevant passage is viewed as relevant.
- **TREC DL [9]:** TREC 2019 Deep Learning Track has the same training and dev set as MS MARCO, but replaces the test set with a novel set produced by TREC. It contains 43 test queries for both passage and document ranking tasks. Especially, NIST constructs separate pools for them and uses depth pooling for more complete labeling.
- **DuReader$_{retrieval}$ [36]:** DuReader$_{retrieval}$ is a Chinese dataset for passage retrieval, which contains over 90K queries and 8M passages. For training queries, annotators label relevant passages within the top-5 retrieved results by Baidu Search. To reduce false negatives in the dev and test sets, more and stronger retrievers are used for the pooling process. As a result, the average labeled relevant passages for training and dev queries are 2.57 and 4.93.

5.2 Evaluation Metrics

We conduct the evaluation following the official settings. For MS MARCO and TREC DL, the ranking results of top-1000 passages and top-100 documents are compared, and we use the Mean Reciprocal Rank (MRR@10 and MRR@100) for MS MARCO and Normalized Discounted Cumulative Gain (NDCG@10 and NDCG@100) for TREC DL as previous works [25, 26]. For the DuReader$_{retrieval}$ dataset, we report the ranking results of top-50 passages with MRR@10 and Recall@1 as officials [36].

5.3 Implementation Details

For experiments on MS MARCO and TREC DL, unless otherwise specified, we implement the relevance model and selection model with the BERT-base model [11] and parameters are initialized with the checkpoint released by Google[2]. We adopt the popular Transformers library[3] for implementations. We truncate the input sequence to a maximum of 256 tokens and 512 tokens for passage ranking and document ranking tasks respectively. For the document ranking task, we concatenate url, title, and body fields if they are available, and experiment with the FirstP setting [10]. Negatives for passage ranking and document ranking tasks are uniformly sampled from the top-1000 and top-100 results (excluding the labeled positives) returned by given retrievers (see Section 6.1 & 6.3).

For experiments on DuReader$_{retrieval}$, we use Ernie-base [42] as the initialization for both relevance model and selection model. We set the maximal length of input sequence as 384 [36]. Negatives

[2]https://github.com/google-research/bert
[3]https://github.com/huggingface/transformers

Table 1: Ranking performance on the MS MARCO and TREC DL datasets. The highest value for every column is highlighted in bold and the statistically significant (p < 0.05) improvements of our method over ANCE+BERT$_{DML}$ and ANCE+BERT$_{RANCE}$ are marked with the asterisk † and ‡ respectively.

Method	MS MARCO Dev				TREC DL Test			
	Passage Ranking		Document Ranking		Passage Ranking		Document Ranking	
	MRR@10	MRR@100	MRR@10	MRR@100	NDCG@10	NDCG@100	NDCG@10	NDCG@100
BM25 + BERT	0.3562	0.3642	0.3841	0.3898	0.6958	0.6182	**0.6420**	**0.5379**
ANCE + BERT	0.3403	0.3493	0.4091	0.4165	0.6971	0.5946	0.6268	0.4900
ANCE + BERT$_{zhan}$	0.3486	0.3597	0.3903	0.4003	0.6905	0.5968	0.6115	0.4620
ANCE + BERT$_{RocketQA}$	0.3476	0.3583	0.4083	0.4147	0.7018	0.6143	0.6257	0.4893
ANCE + BERT$_{DML}$	0.3458	0.3545	0.4110	0.4188	0.7015	0.6021	0.6272	0.4961
ANCE + BERT$_{RANCE}$	0.3513	0.3616	0.4118	0.4183	0.7032	0.6174	0.6270	0.4953
ANCE + BERT$_{CET}$	**0.3638**†‡	**0.3743**†‡	**0.4233**†‡	**0.4311**†‡	**0.7243**†‡	**0.6398**†‡	0.6416†‡	0.5289†‡

for models training are uniformly sampled from the top-50 results (excluding the labeled positives) of retrievers (see Section 6.1).

We use the AdamW optimizer with $\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 10^{-8}$, and learning rate is 3×10^{-6} with the rate of linear scheduling warm-up at 0.1. The hyperparameter τ is set to 1.0 unless noted otherwise. For all experiments, each positive sample is paired with one negative, and no special tricks are used for training. The batch size is set to 64 and 32 for passage and document ranking, and we run all experiments on Nvidia Tesla V100-32GB GPUs.

5.4 Baselines

We compare the BERT ranking model learned with CET (i.e., BERT$_{CET}$) against that learned with the following methods:

- **BERT:** It trains the naive BERT ranker without any special techniques for addressing the false negative problem.
- **BERT$_{zhan}$:** The BERT ranker is trained with BM25 negatives and evaluated with the results of stronger retrievers. This training-evaluating inconsistency setting is employed by Zhan et al. [50] for alleviating the false negative problem in models training.
- **BERT$_{RocketQA}$:** The denoising technique in RocketQA [13] is used for BERT rankers training. We first train a naive BERT ranker on biased data and predict relevance scores for unlabeled documents, then we randomly sample negatives with relevance scores less than η for the denoised BERT ranker training.
- **BERT$_{RANCE}$:** The sampling technique to remove false negatives in RANCE [34] is used for BERT rankers training. Following RANCE, we estimate the probability $P_{relevant}(r)$ using the dev/test set labeled with depth pooling, and then we sample negatives from the retrieval results according to $P_{relevant}(r)$.
- **BERT$_{DML}$:** The dynamic multi-granularity learning method [52] is used for BERT rankers training to address the false negatives. All parameters are set the same as in [52], except that only one negative is sampled for each positive to achieve fair comparisons.

6 RESULTS AND ANALYSIS

We empirically evaluate NRMs learned with CET to address following research questions:

- **RQ1:** Can NRMs learned with CET achieve better performance as compared with other SOTA learning techniques?
- **RQ2:** Could CET work well for training different NRMs with different retrievers for negative sampling?

Table 2: Recall performance of BM25 and ANCE on the MS MARCO and TREC DL datasets.

Model	MS MARCO Dev		TREC DL Test	
	Passage	Document	Passage	Document
	Recall@1k	Recall@100	Recall@1k	Recall@100
BM25	0.8140	0.7564	0.6778	0.3871
ANCE	0.9587	0.8927	0.6610	0.2664

- **RQ3:** How does the hyper-parameter τ in CET affect the effectiveness of learned NRMs?
- **RQ4:** How does CET affect the training process of NRMs on labeled datasets with pooling bias?

6.1 Empirical Comparison with Baselines

To answer **RQ1**, we compare CET with baseline methods on three retrieval datasets to verify the effectiveness of CET.

For MS MARCO and TREC DL, we train rankers with negatives sampled from two retrievers, i.e., BM25 and ANCE [47]. We use BM25-retrieved results released by officials, and for ANCE, we use the code and checkpoint (FirstP) released by Xiong et al.[4] to obtain the retrieval results. The recall performance of them are reported in Table 6, and the ranking performance of BERT$_{CET}$ and baselines are shown in Table 1. From the results we can observe that:

(1) **BM25+BERT *vs* ANCE+BERT:** For the MS MARCO dataset, recall of ANCE improves by 18% compared with BM25 on both tasks (see Table 6). However, the BERT ranker trained with ANCE negatives (i.e., ANCE+BERT) does not improve compared with that trained with BM25 negatives (i.e., BM25+BERT) on the passage ranking task, and improves slightly over the recall gain on the document ranking task. It indicates that without addressing the false negative, ranking models cannot benefit from stronger retrievers for negative sampling. Besides, it is worth mention that on the document ranking task of TREC DL, Recall@100 of ANCE is 31% lower than BM25 (see Table 6), causing ANCE+BERT performs significantly worse than BM25+BERT.

(2) **ANCE+BERT *vs* ANCE+BERT$_\star$:** For BERT rankers trained with special techniques to solve the false negative issue, we collectively call them BERT$_\star$ here for simplicity. Among them, ANCE+BERT$_{zhan}$ performs better than ANCE+BERT on the

[4] https://github.com/microsoft/ANCE

MS MARCO passage ranking task. However, this method generally fails on the other three datasets. With the denoising technique, $\text{BERT}_{RocketQA}$ can alleviate false negatives on passage ranking task, but it does not work for document ranking. ANCE+BERT_{DML} could slightly improve the performance over ANCE+BERT, consisting with the result in [52]. BERT_{RANCE} and BERT_{CET} both achieve improvements over naive BERT rankers on all the passage ranking and document ranking tasks.

(3) **BERT_{DML} *vs* BERT_{RANCE} *vs* BERT_{CET}:** To further analyze the result, ANCE+BERT_{DML} and ANCE+BERT_{RANCE} do not show advantages over ANCE+BERT on the document ranking task. Different from them, ANCE+BERT_{CET} improves significantly than ANCE+BERT on all tasks. Overall, among three methods, CET performs best and shows stable superiority against all baseline methods. Besides, it is worth noting that the gain of CET is more significant on the passage ranking task. For example, for the MS MARCO dataset, ANCE+BERT_{CET} improves about 6.9% and 3.5% on MRR@10 over ANCE+BERT for the passage ranking and document ranking tasks respectively.

For the $\text{DuReader}_{retrieval}$, we train rankers based on the retrieved results by BM25 and a strong retriever (called DE in [36]). We directly use the DE-retrieved results released by officials, and we get BM25-retrieved results with Anserini[5] since the official result is unavailable. The retrieval performance in terms of Recall@50 of BM25 and DE is 0.6635 and 0.9115, respectively. The ranking performance is shown in Table 3. From the results, we can find that:

(1) **BM25+BERT *vs* DE+BERT:** With stronger retrievers to obtain hard negatives for rankers training (e.g, Recall@50 of DE improves 37% over BM25), DE+BERT could improve 33% on MRR@10 over BM25+BERT. We speculate that it is because the false negative problem in $\text{DuReader}_{retrieval}$ is slighter than that in MS MARCO dataset. This could be further proved by comparing the number of labeled relevant passages before and after depth pooling for two datasets [9, 36].

(2) **DE+BERT *vs* DE+$\text{BERT}_{\star}$:** In spite of the slighter bias problem in $\text{DuReader}_{retrieval}$, BERT rankers learned with special techniques to solve the false negative could further improve the ranking performance than the naive BERT ranker. Especially, compared with all baselines, the performance improvement of CET is the most significant, where the gain on MRR@10 and Recall@1 are 2.2% and 4.4% respectively.

In summary, BERT rankers learned with CET perform better on ranking effectiveness over baselines. Empirically, our method is able to correct pooling bias in training NRMs on labeled datasets.

6.2 Results with Different Rankers

To answer **RQ2**, we conduct experiments to investigate whether CET works well with more advanced ranking models. Specifically, we take three NRMs, i.e., PROP [24], CEDR [27], and PARADE [23] as the ranker, and train them with negatives sampled from ANCE retrieved results on the MS MARCO document ranking task. For PROP, we use the checkpoint released by authors[6] as the initialization of the ranking model. For CEDR, we choose the best variant, CEDR-KNRM[7], to conduct experiments. For PARADE, we use the BERT ranker fine-tuned on MS MARCO passage ranking task (released by authors[8]) to initialize PARADE-Max's PLM component.

Table 3: Ranking performance on $\text{DuReader}_{retrieval}$. The highest value for each column is highlighted in bold and the statistically significant ($p < 0.05$) improvements of our method over DE+$\text{BERT}_{RocketQA}$ are marked with †.

Model	MRR@10	Recall@1
BM25 + BERT	0.5391	0.4675
DE + BERT	0.7168	0.6210
DE + BERT_{DML}	0.7171	0.6230
DE + BERT_{RANCE}	0.7196	0.6265
DE + $\text{BERT}_{RocketQA}$	0.7223	0.6300
DE + BERT_{CET}	**0.7323**†	**0.6485**†

Table 4: MRR@100 of different rankers learned with or without CET on MS MARCO document ranking task.

Model	BERT	PROP	CEDR-KNRM	PARADE-Max
w/o CET	0.4165	0.4293	0.4317	0.4278
w/ CET	0.4311	0.4400	0.4403	0.4373

Table 4 reports the performance comparison of four rankers learned with or without CET. From the results, we can observe that even learning without CET, three advanced ranking models all perform better than the BERT ranker, which verifies the effectiveness of these techniques. On the other hand, it is obvious that these models could achieve better ranking performance after equipped with CET. It indicates that CET is effective for correcting pooling bias in training different NRMs.

6.3 Results with Different Retrievers

To answer **RQ2**, we further investigate whether CET works well with negatives sampled from different retrievers. For this purpose, we take two other strong retrievers, i.e., RepBERT [50] and ADORE [49], along with ANCE for experiments. For RepBERT, we train it on the document ranking task following [50] and adopt their open-source codes[9]. For other experiments, we use the code and checkpoint released by authors[4,9,10] to obtain the retrieval results.

Here, we report the ranking performances on MS MARCO and TREC DL with the above three retrievers for negative sampling, and results are depicted in Fig. 3. We can observe that with different retrievers for negative sampling, BERT rankers learned with CET all improve significantly than that without CET. It indicates that CET works well for addressing false negatives from different retrievers. Furthermore, comparing results on two tasks, the gain of CET is more obvious on the passage ranking task, e.g., averagely 8.5% *vs* 3.5% on passage ranking and document ranking for MS MARCO.

6.4 Parameter Sensitivity Analysis

To answer **RQ3**, we evaluate ANCE+BERT_{CET} with different values of τ to investigate its impact on models effectiveness.

[5] http://anserini.io/
[6] https://github.com/Albert-Ma/PROP
[7] https://github.com/Georgetown-IR-Lab/cedr
[8] https://github.com/canjiali/PARADE
[9] https://github.com/jingtaozhan/RepBERT-Index
[10] https://github.com/jingtaozhan/DRhard

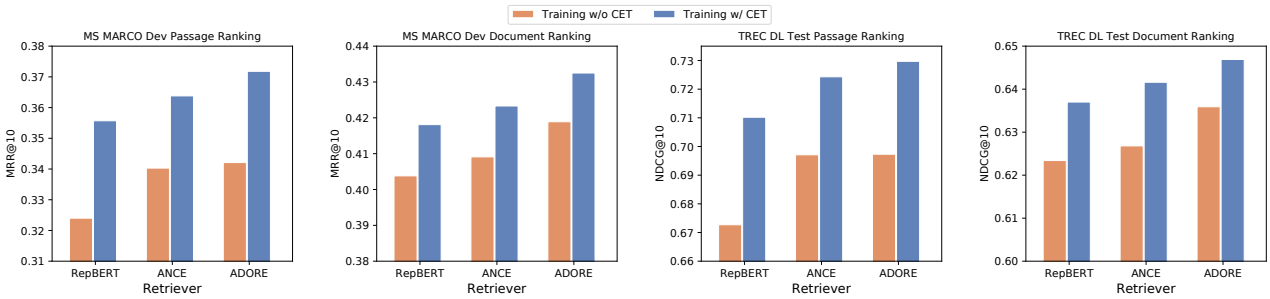

Figure 3: Performance of BERT rankers learned with/without CET along with three different retrievers to sample negatives.

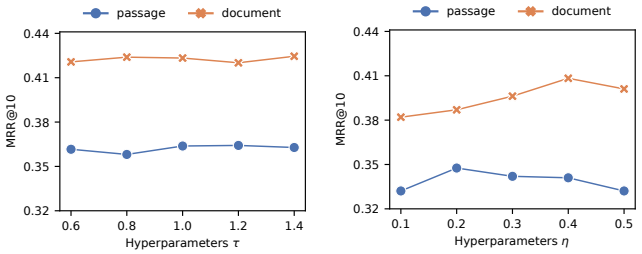

Figure 4: Performance of BERT$_{CET}$ (left) and BERT$_{RocketQA}$ (right) w.r.t their hyperparameters on MS MARCO dataset.

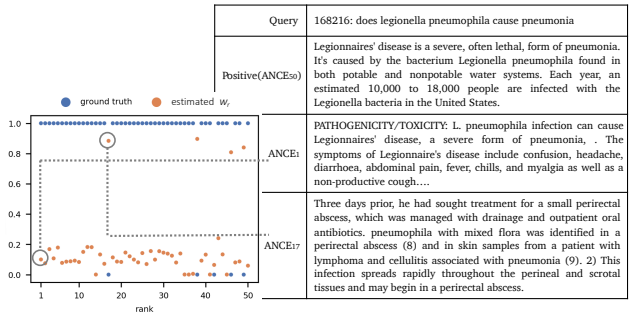

Figure 5: A case to show the ground truth relevance and estimated w_r of top-50 documents retrieved by ANCE.

As shown in Fig. 4, we report MRR@10 of ANCE+BERT$_{CET}$ and ANCE+BERT$_{RocketQA}$ with different choices for τ and η on both tasks of MS MARCO. We can see that the impact of τ is slight on the performance of BERT rankers learned with CET, while the denoising technique in RocketQA [13] is highly sensitive w.r.t its hyperparameters η (e.g., the performance variance of ANCE+BERT$_{RocketQA}$ is about 10 times larger than that of ANCE+BERT$_{CET}$ on the passage ranking task). We suspect that it is because the value of η directly determines the extent of denoising. If it is set too large, false negatives cannot be fully removed, and conversely, hard negatives would be filtered at the same time. Both of cases would hurt rankers training. Especially, compared with the result in Table 1, we can find that if η is set unreasonably, the BERT ranker trained with the denoising technique would be worse than that without denoising. By comparison, CET is much stable and robust in performance.

6.5 Probing Analysis

To answer **RQ4**, we investigate how NRMs perform in distinguishing hard negatives and false negatives during the training process. Specifically, we select a query q from the passage ranking task of TREC DL Test set. Then, we take the selection model checkpoint (Step 150k) to estimate $w_r(q, d_i, d_j)$ for the top-50 documents retrieved by ANCE as d_j, paired with one of labeled relevant documents (ranked at position 51 by ANCE) as d_i. We show the distribution of the ground truth relevance and estimated w_r in Fig. 5.

With more complete labeled ground truth in the TREC DL test set, we can see that 45 of the top-50 retrieval results by ANCE are relevant documents for Query #168216. It further indicates that there are riddled false negatives in the training set, where usually one (at most four) passages are labeled as relevant. Besides, it shows a negative correlation between the ground truth and estimated w_r for the top-50 documents. That is, during the coupled learning process, the selection model would estimate lower w_r if d_j is an unlabeled relevant document. It implies that the relevance model learned with the IPW loss function (i.e., Eq. (20)) could relax the penalty to false negatives. This intuitively displays how CET learns to adaptively distinguish false negatives and hard negatives and achieve bias correction learning.

7 CONCLUSION

In this work, we formulate the false negative issue in training NRMs as learning from labeled datasets with pooling bias. To address this problem, we follow the inverse propensity weighting learning framework and propose a Coupled Estimation Technique. CET learns both a relevance model and a selection model simultaneously with a coupled learning algorithm on the biased dataset. During the training process, the ranking model could adaptively distinguish hard negatives and false negatives with the selection propensities estimated by the selection model and achieve bias correction learning. Empirical results on three retrieval benchmarks show that NRMs learned with CET achieve significant gains on ranking effectiveness against baseline methods. In the future, we would like to extend this method to the listwise setting, where multiple negative samples are considered simultaneously to estimate the bias weights. Moreover, we would also try to apply this method in the first-stage retrieval to address the pooling bias problem.

ACKNOWLEDGMENTS

This work was funded by the National Natural Science Foundation of China (NSFC) under Grants No. 61902381 and 62006218, the Youth Innovation Promotion Association CAS under Grants No. 20144310, and 2021100, the Young Elite Scientist Sponsorship Program by CAST under Grants No. YESS20200121, and the Lenovo-CAS Joint Lab Youth Scientist Project.

Table 5: The statistics of three datasets.

Datasets	Task	Train	Dev	Test	Document
MS MARCO/TREC DL	passage ranking	502,939	6,980	43	8,841,823
	document ranking	367,013	5,193	43	3,213,835
DuReader$_{retrieval}$	passage ranking	86,395	2000	-	8,096,665

Table 6: Recall performance of BM25 and ANCE on the MS MARCO and TREC DL datasets.

Model	Recall@1	Recall@50
BM25	0.1285	0.6635
DE	0.4025	0.9115

REFERENCES

[1] Qingyao Ai, Keping Bi, Cheng Luo, Jiafeng Guo, and W Bruce Croft. 2018. Unbiased learning to rank with unbiased propensity estimation. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 385–394.

[2] Qingyao Ai, Jiaxin Mao, Yiqun Liu, and W Bruce Croft. 2018. Unbiased learning to rank: Theory and practice. In *Proceedings of the 27th ACM CIKM*. 2305–2306.

[3] Negar Arabzadeh, Alexandra Vtyurina, Xinyi Yan, and Charles LA Clarke. 2021. Shallow pooling for sparse labels. *arXiv preprint arXiv:2109.00062* (2021).

[4] Negar Arabzadeh, Alexandra Vtyurina, Xinyi Yan, and Charles LA Clarke. 2022. Shallow pooling for sparse labels. *Information Retrieval Journal* (2022), 1–21.

[5] Chris Buckley, Darrin Dimmick, Ian Soboroff, and Ellen Voorhees. 2006. Bias and the limits of pooling. In *Proceedings of the 29th annual international ACM SIGIR*. 619–620.

[6] Christopher JC Burges. 2010. From ranknet to lambdarank to lambdamart: An overview. *Learning* 11, 23-581 (2010), 81.

[7] Charles L Clarke, Nick Craswell, and Ian Soboroff. 2009. *Overview of the trec 2009 web track*. Technical Report. WATERLOO UNIV (ONTARIO).

[8] Charles LA Clarke, Nick Craswell, Ian Soboroff, et al. 2004. Overview of the TREC 2004 Terabyte Track.. In *TREC*, Vol. 4. 74.

[9] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Ellen M Voorhees. 2020. Overview of the trec 2019 deep learning track. *arXiv preprint arXiv:2003.07820* (2020).

[10] Zhuyun Dai and Jamie Callan. 2019. Deeper text understanding for IR with contextual neural language modeling. In *Proceedings of the 42nd International ACM SIGIR*. 985–988.

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[12] Jingtao Ding, Yuhan Quan, Quanming Yao, Yong Li, and Depeng Jin. 2020. Simplify and Robustify Negative Sampling for Implicit Collaborative Filtering. *arXiv preprint arXiv:2009.03376* (2020).

[13] Yingqi Qu Yuchen Ding, Jing Liu, Kai Liu, Ruiyang Ren, Xin Zhao, Daxiang Dong, Hua Wu, and Haifeng Wang. 2020. RocketQA: An Optimized Training Approach to Dense Passage Retrieval for Open-Domain Question Answering. *arXiv preprint arXiv:2010.08191* (2020).

[14] Luyu Gao, Zhuyun Dai, and Jamie Callan. 2021. Rethink training of BERT rerankers in multi-stage retrieval pipeline. *arXiv preprint arXiv:2101.08751* (2021).

[15] Jiafeng Guo, Yinqiong Cai, Yixing Fan, Fei Sun, Ruqing Zhang, and Xueqi Cheng. 2022. Semantic models for the first-stage retrieval: A comprehensive review. *ACM Transactions on Information Systems (TOIS)* 40, 4 (2022), 1–42.

[16] Jiafeng Guo, Yixing Fan, Liang Pang, Liu Yang, Qingyao Ai, Hamed Zamani, Chen Wu, W Bruce Croft, and Xueqi Cheng. 2020. A deep look into neural ranking models for information retrieval. *Information Processing & Management* 57, 6 (2020), 102067.

[17] Ziniu Hu, Yang Wang, Qu Peng, and Hang Li. 2019. Unbiased lambdamart: an unbiased pairwise learning-to-rank algorithm. In *The WWW Conference*. 2830–2836.

[18] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM CIKM*. 2333–2338.

[19] Thorsten Joachims, Adith Swaminathan, and Tobias Schnabel. 2017. Unbiased learning-to-rank with biased feedback. In *Proceedings of the Tenth ACM WSDM*. 781–789.

[20] Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense Passage Retrieval for Open-Domain Question Answering. *arXiv preprint arXiv:2004.04906* (2020).

[21] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics* 7 (2019), 453–466.

[22] Vihan Lakshman, Choon Hui Teo, Xiaowen Chu, Priyanka Nigam, Abhinandan Patni, Pooja Maknikar, and SVN Vishwanathan. 2021. Embracing Structure in Data for Billion-Scale Semantic Product Search. *arXiv preprint arXiv:2110.06125* (2021).

[23] Canjia Li, Andrew Yates, Sean MacAvaney, Ben He, and Yingfei Sun. 2020. PARADE: Passage representation aggregation for document reranking. *arXiv preprint arXiv:2008.09093* (2020).

[24] Xinyu Ma, Jiafeng Guo, Ruqing Zhang, Yixing Fan, Xiang Ji, and Xueqi Cheng. 2021. Prop: Pre-training with representative words prediction for ad-hoc retrieval. In *Proceedings of the 14th ACM WSDM*. 283–291.

[25] Xinyu Ma, Jiafeng Guo, Ruqing Zhang, Yixing Fan, Yingyan Li, and Xueqi Cheng. 2021. B-PROP: bootstrapped pre-training with representative words prediction for ad-hoc retrieval. In *Proceedings of the 44th International ACM SIGIR*. 1513–1522.

[26] Zhengyi Ma, Zhicheng Dou, Wei Xu, Xinyu Zhang, Hao Jiang, Zhao Cao, and Ji-Rong Wen. 2021. Pre-training for Ad-hoc Retrieval: Hyperlink is Also You Need. In *Proceedings of the 30th ACM CIKM*. 1212–1221.

[27] Sean MacAvaney, Andrew Yates, Arman Cohan, and Nazli Goharian. 2019. CEDR: Contextualized embeddings for document ranking. In *Proceedings of the 42nd International ACM SIGIR*. 1101–1104.

[28] Joel Mackenzie, Matthias Petri, and Alistair Moffat. 2021. A Sensitivity Analysis of the MSMARCO Passage Collection. *arXiv preprint arXiv:2112.03396* (2021).

[29] Irina Matveeva, Chris Burges, Timo Burkard, Andy Laucius, and Leon Wong. 2006. High accuracy retrieval with multiple nested ranker. In *Proceedings of the 29th ACM SIGIR*. 437–444.

[30] Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A human generated machine reading comprehension dataset. In *CoCo@ NIPS*.

[31] Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage Re-ranking with BERT. *arXiv preprint arXiv:1901.04085* (2019).

[32] Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. 2016. Deep metric learning via lifted structured feature embedding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4004–4012.

[33] Zohreh Ovaisi, Ragib Ahsan, Yifan Zhang, Kathryn Vasilaky, and Elena Zheleva. 2020. Correcting for selection bias in learning-to-rank systems. In *Proceedings of The Web Conference 2020*. 1863–1873.

[34] Prafull Prakash, Julian Killingback, and Hamed Zamani. 2021. Learning Robust Dense Retrieval Models from Incomplete Relevance Labels. In *Proceedings of the 44th International ACM SIGIR*. 1728–1732.

[35] Tao Qin and Tie-Yan Liu. 2013. Introducing LETOR 4.0 datasets. *arXiv preprint arXiv:1306.2597* (2013).

[36] Yifu Qiu, Hongyu Li, Yingqi Qu, Ying Chen, Qiaoqiao She, Jing Liu, Hua Wu, and Haifeng Wang. 2022. DuReader_retrieval: A Large-scale Chinese Benchmark for Passage Retrieval from Web Search Engine. *arXiv preprint arXiv:2203.10232* (2022).

[37] Ruiyang Ren, Yingqi Qu, Jing Liu, Wayne Xin Zhao, Qiaoqiao She, Hua Wu, Haifeng Wang, and Ji-Rong Wen. 2021. RocketQAv2: A Joint Training Method for Dense Passage Retrieval and Passage Re-ranking. *arXiv preprint arXiv:2110.07367* (2021).

[38] Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends® in Information Retrieval* 3, 4 (2009), 333–389.

[39] Joshua Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. 2020. Contrastive learning with hard negative samples. *arXiv preprint arXiv:2010.04592* (2020).

[40] Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 815–823.

[41] Karen Spark-Jones. 1975. Report on the need for and provision of an'ideal'information retrieval test collection. *Computer Laboratory* (1975).

[42] Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. 2019. Ernie: Enhanced representation through knowledge integration. *arXiv preprint arXiv:1904.09223* (2019).

[43] Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A Heterogenous Benchmark for Zero-shot Evaluation of Information Retrieval Models. *arXiv preprint arXiv:2104.08663* (2021).

[44] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. 5998–6008.

[45] Xuanhui Wang, Michael Bendersky, Donald Metzler, and Marc Najork. 2016. Learning to rank with selection bias in personal search. In *Proceedings of the 39th International ACM SIGIR*. 115–124.

[46] William Webber and Laurence AF Park. 2009. Score adjustment for correction of pooling bias. In *Proceedings of the 32nd international ACM SIGIR*. 444–451.

[47] Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. 2020. Approximate nearest neighbor negative contrastive learning for dense text retrieval. *arXiv preprint arXiv:2007.00808* (2020).

[48] Yisong Yue, Rajan Patel, and Hein Roehrig. 2010. Beyond position bias: Examining result attractiveness as a source of presentation bias in clickthrough data. In *Proceedings of the 19th international conference on World wide web*. 1011–1018.

[49] Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Jiafeng Guo, Min Zhang, and Shaoping Ma. 2021. Optimizing Dense Retrieval Model Training with Hard Negatives. *arXiv preprint arXiv:2104.08051* (2021).

[50] Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Min Zhang, and Shaoping Ma. 2020. RepBERT: Contextualized text embeddings for first-stage retrieval. *arXiv preprint arXiv:2006.15498* (2020).

[51] Hang Zhang, Yeyun Gong, Yelong Shen, Jiancheng Lv, Nan Duan, and Weizhu Chen. 2021. Adversarial Retriever-Ranker for dense text retrieval. *arXiv preprint arXiv:2110.03611* (2021).

[52] Xuanyu Zhang and Qing Yang. 2021. DML: Dynamic Multi-Granularity Learning for BERT-Based Document Reranking. In *Proceedings of the 30th ACM CIKM*. 3642–3646.