Mitigating Pooling Bias in E-commerce Search via False Negative Estimation

Xiaochen Wang*
The Pennsylvania State University
University Park, PA, USA
xcwang@psu.edu

Xiao Xiao
Instacart
San Francisco, CA, USA
xiao.xiao@instacart.com

Ruhan Zhang
Instacart
San Francisco, CA, USA
ruhan.zhang@instacart.com

Xuan Zhang
Instacart
San Francisco, CA, USA
rachel.zhang@instacart.com

Taesik Na
Instacart
San Francisco, CA, USA
taesik.na@instacart.com

Tejaswi Tenneti
Instacart
San Francisco, CA, USA
tejaswi.tenneti@instacart.com

Haixun Wang
Instacart
San Francisco, CA, USA
haixun.wang@instacart.com

Fenglong Ma
The Pennsylvania State University
University Park, PA, USA
fenglong@psu.edu

Abstract

Efficient and accurate product relevance assessment is critical for user experiences and business success. Training a proficient relevance assessment model requires high-quality query-product pairs, often obtained through negative sampling strategies. Unfortunately, current methods introduce pooling bias by mistakenly sampling false negatives, diminishing performance and business impact. To address this, we present Bias-mitigating Hard Negative Sampling (BHNS), a novel negative sampling strategy tailored to identify and adjust for false negatives, building upon our original False Negative Estimation algorithm. Our experiments in the Instacart search setting confirm BHNS as effective for practical e-commerce use. Furthermore, comparative analyses on public dataset showcase its domain-agnostic potential for diverse applications.

CCS Concepts

• **Information systems → Business intelligence**.

Keywords

e-commerce search, pooling bias, negative sampling

ACM Reference Format:
Xiaochen Wang, Xiao Xiao, Ruhan Zhang, Xuan Zhang, Taesik Na, Tejaswi Tenneti, Haixun Wang, and Fenglong Ma. 2024. Mitigating Pooling Bias in E-commerce Search via False Negative Estimation. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '24), August 25–29, 2024, Barcelona, Spain.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3637528.3671630

*This work was performed when the author was at Instacart.

This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike International 4.0 License.

KDD '24, August 25–29, 2024, Barcelona, Spain
© 2024 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0490-1/24/08
https://doi.org/10.1145/3637528.3671630

1 Introduction

In the rapidly evolving landscape of e-commerce search, efficient and accurate inference on query-product relevance plays a crucial role in enhancing user experiences and driving business success. Recently, more and more giant companies such as Instacart have used advanced techniques to enhance the performance of e-commerce searches. However, training such sophisticated search relevance models typically necessitates datasets of high quality and ample quantity, which is usually impractical in the e-commerce scenario due to the following reasons. On the one hand, annotating training data is time-consuming and labor-intensive, constraining the volume of labeled data [12]. On the other hand, query-product pairs for annotation are typically pre-collected through a basic information retrieval system [3, 12]. It results in a practical obstacle that labeled data are dominated by the relevant pairs, while irrelevant cases are scarce, thus harming the performance of the relevance assessment model.

To address these issues, researchers at Instacart have leveraged a transformer-based model with a negative sampling strategy to streamline the training [27]. Negative sampling methods facilitate the generation of irrelevant query-product pairs to complement the positive instances. This approach ensures the provision of a well-balanced training dataset that accurately mirrors the intricacies encountered in real-world e-commerce contexts. It not only aids in augmenting the model's capacity to distinguish relevant from irrelevant query-product pairs but also significantly contributes to the overall resilience and efficacy of the e-commerce search system.

While negative sampling strategies effectively facilitate the training, they introduce another challenge: *the emergence of **pooling bias** during the negative sampling process.* In this context, pooling bias refers to the utilization of unlabeled positive pairs as false negatives, which detrimentally impacts the model's performance [1, 3, 25]. Conventional negative samplers assume randomly selected query-product pairs as negative (irrelevant), ignoring the possibility of false negatives. An illustration example is shown in Figure 1.

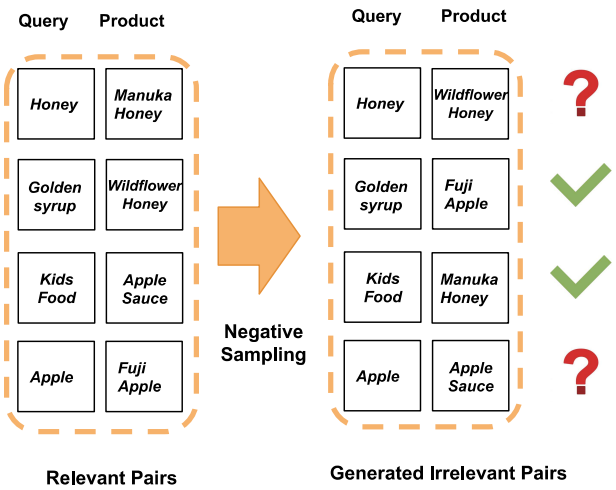

Figure 1: In e-commerce scenario, conventional negative sampling usually assumes the irrelevance between original query and sampled products, which introduces pooling bias by producing false negative pairs such as [Honey, Wildflower Honey] or [Apple, Apple Sauce]. These samples are wrongly labeled as irrelevant pairs, thus harm the performance of search model.

This issue becomes more pronounced for hard negative samplers that explicitly select products with high similarity to the given query, as these query-product pairs have a higher chance of being false negatives without proper control. The presence of false negatives in the training data significantly impacts the performance of the search embedding model by introducing noise, potentially leading to the overfitting issue of the model to the true positives and over-penalization of popular products. Therefore, mitigating pooling bias by appropriately addressing false negatives introduced during sampling is crucial for e-commerce search.

To tackle this problem, in this paper, we present a novel Bias-mitigating Hard Negative Sampling (BHNS) strategy, incorporating a simple yet effective way to estimate the probability of a query-product pair being false negative, i.e., False Negative Estimation. In particular, we leverage semantic similarity between queries to calculate the false negative likelihood for query-product pairs. We then apply the estimation of false negative likelihood to eliminate pooling bias by (1) directing the sampling through regularizing the hard negative sampler and (2) guiding the training via generating pseudo labels for enhancement. We combine these two approaches and build our bias-mitigation sampling strategy. We utilize the proposed BHNS to optimize cross encoder, which takes concatenated query-product pairs as input and serves as advanced relevance assessment model [6, 21].

To validate the efficacy of our proposed BHNS, we conduct three experiments: (1) validation with public datasets, (2) offline test with Instacart's in-house data, and (3) test as part of Instacart's search system. We compare BHNS with conventional negative sampling strategy as well as baselines specifically designed to handle false negative bias to examine the efficacy of BHNS. Through our experimental results and case study, our proposed BHNS demonstrates superior performance, thus providing strong evidence that validates the hypotheses underlying False Negative Estimation and supports the correctness of the rationale behind the design of BHNS.

Table 1: Notation table.

Notation	Meaning
$\mathbf{e}_i^q/\mathbf{e}_j^p$	the embedding of a query q_i/ a product p_j
$r_{i,j}$	the known annotated relevance score between q_i and p_j
$\hat{r}_{i,j}$	the similarity score between q_i and p_j
$\tilde{r}_{i,k}$	the estimated relevance score between augmented negative pair q_i and p_k
$\theta_{i,j}$	the estimated probability of a pair q_i and p_k being as a false negative sample

2 Background

The general relevance assessment task can be formulated as follows: Each training batch is composed of two distinct components: a set of queries $\mathcal{Q}$ and a corresponding set of products $\mathcal{P}$. Given a search query $q_i \in \mathcal{Q}$ and a product $p_j \in \mathcal{P}$, the model is responsible for assessing the degree of relevance between q_j and p_j, which is denoted by $\hat{r}_{i,j}$. In this section, we briefly introduce the existing query-product relevance model at Instacart, empowered by conventional negative sampling strategies and more effective cross-encoder architecture in development, as the technical context of this work. Table 1 lists the key notations used in this paper.

2.1 Search Model at Instacart

2.1.1 Search Embedding. Instacart has an effective and efficient embedding model to measure the relevance between queries and products, which is widely used in search retrieval, ranking, and many other surfaces [27]. The search embedding model follows a typical bi-encoder design, where queries and products are passed through their corresponding Sentence Transformer-based encoders[1], denoted as BE_Q and BE_P, independently. The input query-product pair (q_i, p_j) is transformed into dense embeddings as follows:

$$\begin{aligned} \mathbf{e}_i^q &= \text{BE}_Q(q_i), \\ \mathbf{e}_j^p &= \text{BE}_P(p_j). \end{aligned} \tag{1}$$

The estimated relevance score between a query and a product is computed via cosine similarity between their embeddings, i.e.,

$$\hat{r}_{i,j} = \text{sim}(\mathbf{e}_i^q, \mathbf{e}_j^p). \tag{2}$$

During the training phase, similarity scores can be efficiently calculated through a series of matrix transpositions and multiplications, making the process computationally efficient. For the inference stage, the product and known query embeddings can be pre-computed offline and cached, while tail query embeddings are generated in real-time. During inference, relevant products for an incoming search query are retrieved using the query's embedding via approximate nearest neighbor search (ANN), and the estimated relevance score $\hat{r}_{i,j}$ is passed to the ranker service as a feature to determine the final ranking of p_j for search term q_i.

[1] https://huggingface.co/sentence-transformers

Although the supervised training of the search embedding model benefits from supervised learning using ground-truth annotation score $r_{i,j}$, it still suffers from the insufficiency of negative cases. In e-commerce search, positive pairs (q_i, p_j) can be easily obtained from past search logs with converted products. However, high-quality negative cases, which should have been able to help the model acquire the capability of distinguishing irrelevant products from relevant ones, are naturally absent.

2.1.2 Negative Sampling. A straightforward solution is selecting products that do not result in search conversions as negative examples for a given query. Nevertheless, unconverted products may not necessarily be irrelevant to the search query because users' decisions can be swayed by many factors other than relevance, such as price, size, visual presentation, personal preference for flavor or brand, item position, etc. This is confirmed in our data verification process in examining search logs from Instacart, a leading e-commerce player in online grocery shopping, where **we found more than 4% of query-product pairs from the search logs with zero historical conversions are labeled as positive (relevant) by human annotators**.

A common alternative approach to address the lack of negative samples issue is in-batch negative sampling. Given a query q_i and in-batch products $\mathcal{P}$, let $\mathcal{P}_i^+$ be the set of products whose relevance with the query q_i are explicitly labeled. Then the vanilla sampling process aims to randomly select K products from the difference of sets $\mathcal{P}$ and $\mathcal{P}_i^+$, i.e., $\mathcal{P} - \mathcal{P}_i^+$, as negative sample set $\mathcal{P}_i^-$, satisfying

$$\tilde{r}_{i,k} = 0, \forall p_k \in \mathcal{P}_i^- \tag{3}$$

where $\tilde{r}_{i,k} = 0$ denotes the irrelevance between query q_i and any randomly sampled product p_k in negative sample set $\mathcal{P}_i^-$.

Although the naive random sampling technique can generate negative pairs, it fails to utilize the most informative negative samples, resulting in insufficient learning. To avoid this issue, several studies [9, 22] have tried to use "hard" negative samples during the sampling procedure due to their informativeness. At Instacart, we implement the selection of hard negative samples using our search embedding technique introduced in Section 2.1.1 and determining by the similarity scores between products and queries calculated by Eq. (2). The hard negative sampling process with a negative size equal to K can be represented as follows:

$$\begin{aligned} \tilde{r}_{i,k} &= 0, \forall p_k \in \mathcal{P}_i^- = argmax_{\text{top}K}\{\hat{r}_{i,j}\}, \\ \hat{r}_{i,j} &= \text{sim}(\mathbf{e}_i^q, \mathbf{e}_j^p), \forall p_j \in \mathcal{P} - \mathcal{P}_i^+. \end{aligned} \tag{4}$$

However, indiscriminately assuming irrelevance between a given query q_i and in batch product $p_k \in \mathcal{P}_i^-$ will inevitably introduce false negatives. Our case study based on the Instacart search log shows that for the query "*Honey*", the product "*Honey 100%*", which is definitely relevant to the query but is not the product in this pair, will be retrieved as the top-ranked hard negative sample by the hard negative sampler mentioned in Eq. (4), clearly proving the existence of pooling bias. The pooling bias diminishes the quality of augmented negative pairs, impeding the model from conducting precise relevance assessment and necessitating a concrete solution.

2.2 Cross Encoder in E-commerce Search

Previous work [20] has demonstrated the cross-encoder as a more powerful solution to capture the relationship between two input sentences compared with the bi-encoder. The cross-encoder takes the concatenation of query and product sentences as input and directly outputs the relevance score after a series of transformations. In recent exploration at Instacart, transformer-based cross-encoder[2], denoted as CE, is adopted for relevance assessment as follows:

$$\hat{r}_{i,j} = \text{CE}([q_i; p_j]) \tag{5}$$

where $[\cdot;\cdot]$ denotes the concatenation operation.

Since the bi-encoder model encodes the query and the product separately, once the model is trained, the embedding for a given query is fixed regardless of which product it is paired with. In contrast, in the cross-encoder, the output of each intermediate layer is determined jointly by both query and product, which allows more interactions between the two and a richer context to be extracted.

However, the cross-encoder does not output embedding of either query or product, thus presents two primary limitations. First, while training the cross-encoder, similarity score calculation via matrix multiplication, as mentioned in Eq. (2), is precluded due to the lack of corresponding embeddings. The score can only be obtained through aggregating queries and products in the input stage, proposing a significant computational challenge. The problem becomes particularly severe when the parameter K in Eqs. (3) and (4) is large, constraining the scale of negative sampling. Second, instead of simply computing the cosine similarity between the cached embeddings, the cross-encoder requires the full computation of a transformer model on the fly during inference time, which results in an $O(mn)$ computational complexity that can significantly increase the latency of the system, where m denotes the number of queries and n is the number of products. Given this limitation, in e-commerce, CE is often adopted for re-ranking only, while bi-encoders are retained for retrieval and first-stage ranking. In this work, we follow this tradition and adopt CE as part of a re-ranker for a small subset of top products pre-selected by the retrieval engine and first-stage ranking.

3 Methodology

To alleviate the pooling bias problem introduced above, we propose a new metric for evaluating the likelihood of false negatives, namely False Negative Estimation. Based on this metric, we design our Bias-mitigating Hard Negative Sampling strategy, depicted in Figure 2. Source code is available[3].

3.1 False Negative Estimation

Given a query q_i and in-batch off-diagonal product p_j, our goal is to estimate the likelihood that (q_i, p_j) is a false negative pair, i.e., the likelihood that p_j is actually relevant to q_i, which is denoted as $\theta_{i,j}$. Given the fact that there is no explicit knowledge regarding the ground truth for the relationship between such in-batch pairs, we turn to estimate this relationship by leveraging Instacart's bi-encoder search embedding model mentioned in Section 2.1.1 by

[2] https://www.sbert.net/examples/applications/cross-encoder/
[3] https://github.com/XiaochenWang-PSU/BHNS_Cross_Encoder

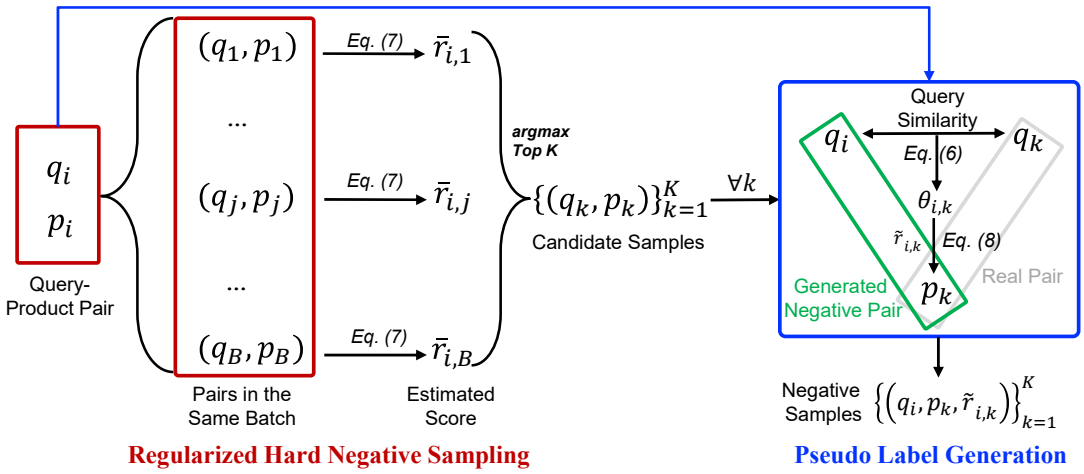

Figure 2: Bias-mitigating Hard Negative Sampling. By leveraging False Negative Estimation, the sampler detects potential false negatives and functions by (1) reducing the weight of potential false negatives during hard negative sampling and (2) informing the model of the likelihood of false negatives through the pseudo label.

following the hypothesis: *The more similar two queries are, the more likely they share the same relevant product.*

The intuition behind this hypothesis is straightforward. Given the same target product, different users may formulate their search queries differently based on their context [2]. Since the search embedding model is trained on historical search conversions, it naturally projects these different queries with similar semantic meanings to be close to each other, as well as to the product that they converted on, in the semantic space. Similarly, if two queries are semantically independent, then it is likely that their correspondences with the same product are also mutually independent.

Based on this intuition, we do not directly estimate the relevance of a query-product pair (q_i, p_j) based on the similarity between their embeddings $\mathbf{e}_i^q$ and $\mathbf{e}_j^p$. Instead, we use the query q_t that is positively related to the product p_j as the bridge and take the similarity between two queries q_i and q_t into consideration. Since the annotation score of the pair (q_t, p_j) is known, which is $r_{t,j}$, we only need to calculate the similarity between q_i and q_t using Eq. 2. Finally, we can estimate the false negative probability of a pair (q_i, p_j) as follows:

$$\theta_{i,j} = \frac{1}{T_j} \sum_{t=1}^{T_j} r_{t,j} * \text{sim}(\mathbf{e}_i^q, \mathbf{e}_t^q), \forall p_j \in \mathcal{P}_i^-, \tag{6}$$

where T_j denotes the number of queries positively related to the product p_j in this batch. The fusion of knowledge derived from the pre-trained bi-encoder and information conveyed through the ground truth annotations renders the estimation comprehensive and robust. The computation of $\text{sim}(\mathbf{e}_i^q, \mathbf{e}_t^q)$ can be performed dynamically in the data loading step without introducing significant time-consumption, as it is in the time complexity of $O(n)$.

We utilize the estimation of false negative samples in two ways for optimizing the conventional hard negative sampling process (Eq. (4)), i.e., sampling regularization and pseudo label generation, which will be introduced, respectively.

3.2 Sampling Regularization

Compared with vanilla negative sampling (Eq. (3)), the hard negative sampling strategy (Eq. (4)) is even more likely to introduce false negative pairs. To alleviate the problem, we propose to utilize False Negative Estimation as an additional regularization term to generate negative samples, described as follows:

$$\begin{aligned} \tilde{r}_{i,k} &= 0, \forall p_k \in \mathcal{P}_i^- = argmax_{\text{topK}}\{\bar{r}_{i,j}\}, \\ \bar{r}_{i,j} &= (1 - \theta_{i,j})^\tau * \text{sim}(\mathbf{e}_i^q, \mathbf{e}_j^p), \forall p_j \in \mathcal{P} - \mathcal{P}_i^+, \end{aligned} \tag{7}$$

where τ is a temperature hyperparameter curving the weight of $\theta_{i,j}$. Here, False Negative Estimation serves as a regularization term. A larger value of $\theta_{i,j}$ implies that the pair (q_i, p_j) has a high likelihood of being false negative, and Eq. (7) ensures that the pair has less of a chance to be erroneously selected as a hard negative sample, thus balancing informativeness and robustness and mitigating pooling bias prevalent in conventional hard negative sampling.

3.3 Pseudo Label Generation

Although reducing the score for potential false negatives diminishes pooling bias, unsupervised regularization cannot guarantee the extinction of all false negatives. To further eliminate the remaining pooling bias caused by these omitted cases, we design a dual insurance mechanism. Instead of indiscriminately assuming off-diagonal query-product pairs (q_i, p_j) are irrelevant as in vanilla negative sampling (Eq. (3)) and hard negative sampling (Eq. (4)), we adopt the False Negative Estimation (Eq. (6)) to estimate the relevance between (q_i, p_j) that serves as pseudo label for training, i.e.,

$$\tilde{r}_{i,k} = \theta_{i,k}, \forall p_k \in \mathcal{P}_i^-. \tag{8}$$

This pseudo label has the following desirable properties when $T_j = 1$ in Eq. (6):

- When $\text{sim}(\mathbf{e}_i^q, \mathbf{e}_j^q)$ equals 1, the two queries q_i and q_j are equivalent, and their relationship with product p_k is also equivalent, i.e., $\tilde{r}_{i,k} = \tilde{r}_{j,k}$.

- When $\text{sim}(\mathbf{e}_i^q, \mathbf{e}_j^q)$ equals 0, the two queries are independent from each other. Given that the product p_k is relevant to q_j, it is likely irrelevant to q_i, and the pair (q_i, p_k) is unlikely to be a false negative when we set $\tilde{r}_{i,k}$ to 0.

In this way, the pseudo label generated via False Negative Estimation provides an extra correction on the regularized hard negative sampling strategy (Eq. (7)), further reducing the likelihood of introducing pooling bias in indiscriminate negative labeling. The combination of regularization and pseudo label based on False Negative Estimation, namely Bias-mitigating Hard Negative Sampling, will be evaluated in the next section. Pseudo codes of the algorithm is provided in Algorithm 1.

Algorithm 1 Sampling Procedure

Require: Training batch $\mathrm{B}_{tra} = \{(q_0, p_0, r_{00}), \cdots, (q_B, p_B, r_{BB})\}$, frozen pretrained sentence transformer BE_Q and BE_P.

```
for query i = 1 to B do
    for product j = 1 to B do
        Calculate r̃_{i,j} based on Eq. (7);
    end for
    Select top K candidate pair set {(q_k, p_k)}_{k=0}^{K} according to Eq. (7);
    for product k = 1 to K do
        Obtain θ_{i,k} according to Eq. (6);
        Calculate r̃_{i,k} according to Eq. (8);
        Construct new pair {(q_i, p_k, r̃_{i,k})};
    end for
end for
```

4 Experiment

We conduct experiments on both public and private datasets to demonstrate the efficacy of our proposed Bias-mitigating Hard Negative Sampling strategy. Besides, we incorporate Bias-mitigating Hard Negative Sampler with Instacart search system to validate the practical value of our proposed approach.

4.1 Semantic Textual Similarity

To examine the effectiveness of False Negative Estimation, we first conduct our experiment on the Semantic Textual Similarity (STS) task using the benchmark dataset[4]. In the STS task, the similarity between two pieces of text is expected to be predicted by a model, analogous to the general search use cases.

4.1.1 Implementation. MiniLM-L3-v2 [5] and MiniLM-L-2-v2 [6] serve as bi-encoder and cross encoder in our experiments. During the training, the bi-encoder is frozen. We set the temperature hyperparameter τ to 2.

4.1.2 Evaluation Metric. As the STS Benchmark evaluation dataset does not allow list-wise comparison, we decide to adopt the evaluation metrics utilized in script provided in the document of cross encoder[7], which are Pearson correlation coefficient and Spearman correlation coefficient. Besides, we leverage the Area under the ROC Curve (AUROC) as an additional evaluation metric for a more fair comparison. These three metrics perform pair-wise evaluation simultaneously.

Table 2: Experiments on STS Benchmark Dataset. The highest score is shown in boldface.

K Valuae	Method	Pearson	Spearman	AUROC
$K = 2$	VNS	67.61	77.05	90.18
	HNS	66.74	74.57	89.24
	STAR [30]	59.95	72.04	88.28
	FALSE [32]	70.56	77.10	90.12
	CET [3]	75.87	75.21	88.96
	BHNS (ours)	**78.32**	**77.37**	**90.64**
$K = 4$	VNS	67.53	76.67	89.99
	HNS	67.09	74.11	88.93
	STAR [30]	62.93	72.30	88.52
	FALSE [32]	70.76	76.55	89.76
	CET [3]	75.78	75.12	88.91
	BHNS (ours)	**77.97**	**76.91**	**90.34**
$K = 8$	VNS	67.49	76.12	89.90
	HNS	71.76	74.81	89.19
	STAR [30]	65.57	73.01	88.56
	FALSE [32]	71.85	76.07	89.43
	CET [3]	76.53	75.45	89.07
	BHNS (ours)	**77.30**	**76.37**	**90.05**

4.1.3 Baselines. We compare our proposed Bias-mitigating Hard Negative Sampling (BHNS) with Vanilla Negative Sampling (VNS) and Hard Negative Sampling (HNS), they serve as natural baselines widely used in diverse scenarios. Furthermore, we extend our experiment by adding the following advanced baselines specially designed for handling false negatives:

- **STAR** (**S**table **T**raining **A**lgorithm for dense **R**etrieval) [30] is a training pattern developed to combine random negative sampling with hard negative sampling to optimize the training process of the bi-encoder. In this study, we transfer their strategy of performing combination to the case of cross encoder.
- **FALSE** (**F**alse neg**A**tive samp**L**es aware contra**S**tive l**E**arning model) [32] is a strategy of detecting false negative samples in an unsupervised contrastive learning scenario. We transfer this method to the supervised training of the cross-encoder by adopting their idea of capturing false negatives based on inter-sample similarity.
- **CET** (**C**oupled **E**stimation **T**echnique) [3] leverages a subordinate cross encoder, i.e., observer, to monitor and correct false negative bias during the training of the primary cross encoder, i.e. reranker.

To perform a fair comparison, we also choose MiniLM-L-2-v2 as the cross-encoder for baselines. For methods involving hard negative sampling, i.e., HNS, FALSE and STAR, we adopt the bi-encoder same to BHNS (MiniLM-L3-v2) to guide the selection of hard negatives.

4.1.4 Results. We test different sampling sizes to validate the superiority as well as the robustness of our proposed BHNS. The

[4] http://ixa2.si.ehu.eus/stswiki/index.php/STSbenchmark
[5] https://huggingface.co/sentence-transformers/paraphrase-MiniLM-L3-v2
[6] https://huggingface.co/cross-encoder/ms-marco-MiniLM-L-2-v2
[7] https://github.com/UKPLab/sentence-transformers/blob/master/examples/training/cross-encoder/training_stsbenchmark.py

Table 3: Ablation Study on STS Benchmark Dataset. K is set to 2 and the highest score is shown in boldface.

	Pearson	Spearman	AUROC
Pseudo Label Generation	77.07	**77.75**	90.43
Sampling Regulation	71.57	75.13	89.52
BHNS (ours)	**78.32**	77.37	**90.64**

results are listed in Table 2, where K is the negative sampling size. Comparison across different sampling strategies implies that hard negative sampling may not contribute to decent performance due to the introduced false negative bias, while BHNS equipped with our proposed BHNS is enabled to handle the pooling bias, thus performing better via the bias mitigation.

While efficiency can be regarded as a potential trouble as we introduce additional computations for bias exclusion, we observe 32% - 38% training time increase, compared with VNS method. Considering that inference speed will not be influenced, we confirm that our method contributes to superior performance with an acceptable compromise on efficiency.

It is also worth noting that BHNS outperforms CET, which involves an additional cross-encoder to observe and eliminate false negatives. It indicates that leveraging a frozen bi-encoder to simply estimate the likelihood of false negatives is superior to supervised sophisticated approaches demanding extra training, showcasing the soundness of the hypothesis we propose in Section 3.1, as well as validating the rationale behind our design.

4.1.5 Ablation Study. We also conduct ablation study on each component in BHNS. The results are presented in Table 3. While both modules manifest competitive effectiveness compared with baselines, the pseudo label generation module performs better. Nevertheless, the combination of both modules, i.e., BHNS, is capable of achieving a more stable and superior performance. This study further validates that our design is non-redundant.

4.2 Offline Experiment

4.2.1 Dataset. Datasets used for training and evaluating the cross-encoder model come from Instacart's ongoing data collection effort to benchmark search relevance, where a random sample of search queries and the products ranked in top display positions are sent for human annotation every month to judge the relevance between queries and products. The annotators categorize each (query, product) pair into five categories:

- **Strongly Relevant**: The product is exactly the type of product the search query is asking for.
- **Relevant**: The product fits the search query, but there are likely better alternatives.
- **Somewhat Relevant**: The product is not exactly what the search query asks for, but it is understandable why it shows up.
- **Not Relevant**: The product is not what the search query asks for, and it is unclear why it shows up.
- **Offensive**: The product is unacceptable and creates a bad user experience.

To help the model grasp the finer granularity in relevance, we convert the above categories into soft labels in the training dataset by taking "Strongly Relevant" as 1, "Relevant" as 0.5, "Somewhat Relevant" as 0.2, and "Not Relevant/Offensive" as 0.1. Scores from multiple annotators for the same (query, product) pair are averaged.

Annotated classes in the evaluation dataset are converted with a coarser stroke, with "Strongly Relevant" and "Relevant" as 1, and "Somewhat Relevant"/"Not Relevant"/"Offensive" as 0.1, in accordance with our previous work [27].

This golden dataset with human annotation is of extremely high quality, but it can introduce bias in model training in that all samples come from products occupying top display positions in search. As a result, they are much more relevant to the search query than other products not selected in the sample. To ensure that the model is capable of correctly scoring less relevant products, we further expand both the training and the evaluation datasets by including an additional 20% (query, product) pairs where the query and the product are randomly and independently sampled from all eligible queries and products from the full dataset. These random negative pairs get a score of 0, based on the assumption that, on average, they are even less relevant than the "Not Relevant"/ "Offensive" products that appeared in top search positions.

4.2.2 Implementation. The present search embedding model serving as the bi-encoder for experiments is originally initialized with a MiniLM-L3-H384-uncased model[8] pretrained on paraphrase dataset[9]. It is further finetuned using Instacart's in-house dataset extracted from the search log as described in the reference [27]. The search embedding model is frozen during the training of the cross-encoder model and is only leveraged for guiding negative sampling.

The cross-encoder is initialized with a pretrained MiniLM-L-2-V2 model[10]. The chunk size of negative sampling is set to 4. The temperature hyperparameter τ is set to 2 for our method.

4.2.3 Evaluation Metric. We focus on the two evaluation metrics: Normalized Discounted Cumulative Gain at M (NDCG@M), which measures the effectiveness of the model in a list-wise setting, and Area under the ROC Curve (AUROC), which measures the model's pair-wise performance. For NDCG, we aggregate the data by search queries and evaluate the model's ability to rank the products for each query by their relevance at M = 5, 10, or 20. For AUROC, we convert the classes "Strongly Relevant" and "Relevant" to positive, and the classes "Somewhat Relevant"/"Not Relevant"/"Offensive" as well as the additional random (query, product) pairs to negative, to make the labels binary.

4.2.4 Baselines. We compare our proposed Bias-mitigating Hard Negative Sampling (BHNS) method with Vanilla Negative Sampling (VNS, Eq. (3)) and Hard Negative Sampling (HNS, Eq. (4)) strategies as baselines. We set the chunk size of sampled negatives to 4 for all sampling strategies implemented for fair comparisons.

4.2.5 Results. We conduct our experiment as described above and obtain results listed in Table 4. The comparison showcases the advantage of adopting a false negative sampling strategy, as the cross-encoder equipped with any negative sampling methods outperforms the one without them on nearly all the metrics. Compared

[8] https://huggingface.co/nreimers/MiniLM-L6-H384-uncased
[9] https://www.sbert.net/examples/training/paraphrases/README.html
[10] https://huggingface.co/cross-encoder/ms-marco-MiniLM-L-2-v2

Table 4: Offline Experiments on Instacart Search Dataset. The highest score is shown in boldface. "w/o NS" means the approach without using the negative sampling technique.

	NDCG@5	NDCG@10	NDCG@20	AUROC
w/o NS	89.79	90.88	92.65	91.14
VNS	90.42	91.43	93.09	94.09
HNS	90.62	91.53	93.19	90.34
BHNS	**90.96**	**91.91**	**93.47**	**95.06**

Table 5: Sampled negative products using the query "honey".

Approach	Product Name	Score
VNS	bowls, compostable	0
	potato and wheat chips, cheddar, cheese crispers	0
	garlic	0
	classic white premium baking chips	0
HNS	honey, orange blossom	0
	peanuts, honey roasted	0
	moisturizing liquid hand soap milk & honey	0
	greek strained yogurt with honey	0
BHNS	frozen dairy dessert cookies & cream	0
	dark chocolate, sea salt & almonds, oat milk, 55% cocoa	0
	honey roasted breakfast cereal, whole grain, family size	0.019
	powered donuts pop'ettes	0

to VNS, HNS does not contribute to performance increase in terms of AUROC, and the performance enhancement on NCDG is minimal, implying that the pooling bias introduced during the sampling process impedes the thorough exploitation of those informative true hard negatives. As an extension, our proposed Bias-mitigating Hard Negative Sampler addresses this shortcoming of HNS by introducing False Negative Estimation, and thus assists cross encoder to manifest more competitive performance.

4.2.6 Case Study. To further demonstrate how our Bias-mitigating Hard Negative Sampling alleviates the pooling bias in the sampling process, we perform an intuitive case study to compare BHNS with conventional negative sampling approaches, i.e., VNS and HNS. Given the query "*honey*", we retrieve corresponding negative products using VNS, HNS, and BHNS, the results are listed in Table 5.

It is obvious that the vanilla negative sampling method (VNS) is incapable of capturing informative hard negative samples, thus failing to boost the cross-encoder via learning from valuable sources. The hard negative sampling (HNS) strategy, although effectively collecting informative products, assumes the irrelevance between query and sampled products, thus unexpectedly generating false negatives and introducing the pooling bias. Our method, Bias-mitigating Hard Negative Sampling, successfully avoids probable false negatives by assigning lower scores to them. Also, even though potential false negative products are collected, the pseudo label generation mechanism specially designed to tackle this scenario is able to hint the model about the potential bias, thus achieving the most promising performance.

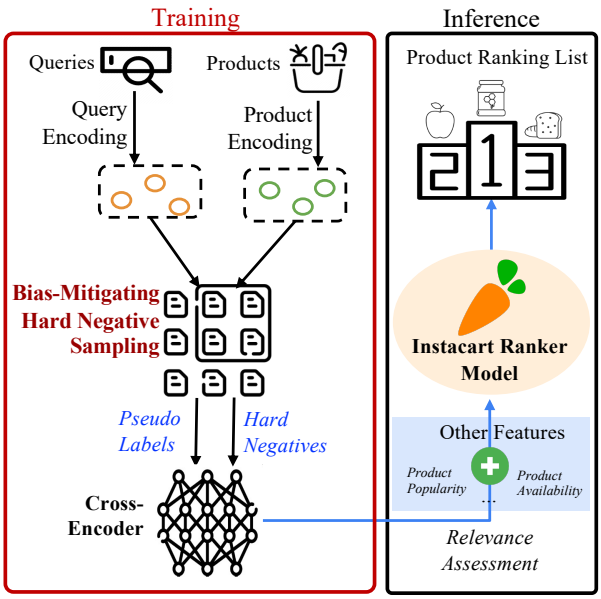

Figure 3: Overview of Instacart's ranking system incorporated with BHNS-boosted cross encoder. The cross encoder is independently trained with Instacart search dataset (see Section 4.2.1). During the inference time, the output of cross encoder boosted by BHNS serves as a necessary feature in the Instacart ranking system, facilitating the ranking process along with other features.

4.3 Experiment with Instacart Search System

To establish the real-world application of BHNS, we incorporate the cross-encoder trained in Section 4.2 with Instacart's search system and examine its impact on the relevance of search results, as shown in Figure 3.

4.3.1 Search System at Instacart. The search system at Instacart consists of a retrieval stage and a ranking stage. The retrieval stage casts a broad net and retrieves a large number of products related to the search query from multiple sources using both sparse and dense signals. It is followed by a multi-step ranking stage, where ranking models with increasing complexities are applied to filter out less relevant products at each step. This cascade pattern ensures that only high-quality search results are presented to the end user. Given the complexity and the latency implication of the cross-encoder model, it is tested as part of the final ranking stage, where the top 50 items get reranked to maximize search conversion.

The final ranker is a deep & cross network (DCN) model [26] with a variety of features, such as how relevant the product is to the search query, the product's popularity and availability, the user's personal preference, etc. In the baseline ranker model serving in production, the query-product cosine similarity from the existing search embedding model [27] is one of the dominant relevance features. In this experiment, we use the similarity score derived from the cross-encoder model, both with and without BHNS, and retrain the DCN ranker model using the same ranking objective. The retrained ranking is used as a replacement for the existing

Table 6: Evaluation of Instacart's Ranking Model Performance on Search Logs. The evaluation involves comparing the performance of Instacart's ranking model on search logs. In this experiment, the ranker without BHNS uses the relevance score returned by the cross encoder with a Vanilla Negative Sampling strategy. This relevance feature is replaced with the BHNS-cross-encoder score for the purpose of the experiment.

Metric	Reranker w/o BHNS	Reranker w. BHNS
AUC	86.39	**86.46**
MRR	54.21	**54.31**
NDCG@5	72.19	**72.25**
NDCG@10	75.41	**75.45**
NDCG@20	76.47	**77.03**

ranker in this test, while other components of the search pipeline remain unchanged.

4.3.2 Evaluation Metrics. In addition to evaluation metrics involved in Table 4, we extend to cover more evaluation metric, i.e., Mean Reciprocal Rank (MRR), for more comprehensive evaluation.

4.3.3 Results. We benchmark the performance of the new ranker model against the baseline model on a sample of recent searches from Instacart's business, and compare the two models' performance in ranking, where converted products are taken as positive labels. The results are listed in Table 6. While the other features going into the two ranking models remain identical, the cross encoder boosted by Bias-mitigating Hard Negative Sampling strategy proves to be effective in significantly improving both list-wise and pair-wise ranking metrics, outperforming the one without BHNS. The comparison further demonstrates the superiority of Bias-mitigating Hard Negative Sampling in handling pooling bias in real-world e-commerce scenario.

5 Related Work

5.1 Cross Encoder for Search

Compared with a typical bi-encoder, the cross-encoder has demonstrated its more advanced capability in performing ranking [23]. After the emergence of BERT [5], researchers take a very initial step on ranking with cross encoder by proposing monoBERT [18]. Researchers further propose to utilize an advanced cross-encoder for instructing bi-encoder through knowledge distillation [4, 14–16, 29]. Several studies in various domains also propose to combine the bi-encoder with the cross-encoder in a cascade pattern, where the bi-encoder retriever provides informative negatives to facilitate the training of cross-encoder serving as ranking [11, 13, 31], however, this concept does not function in our application scenario according to our internal experiments.

5.2 Hard Negative Sampling

Previous works have thoroughly analyzed the benefit of introducing hard negatives. Comparison between random negative sampling and hard negative sampling demonstrates the effectiveness of heuristical sampling strategy [10]. Researchers propose their method to gather global negatives via dense retrieval model [28]. In the setting of contrastive learning, previous works [22, 24, 33, 34] also report that hard negatives are far more informative compared with random negatives, inducing more competitive model performance.

While the advantage of exploiting hard negatives has been acknowledged, researchers also notice the problem co-occurring with the sampling procedure. It is discerned that false negatives are introduced through the negative sampling process, thus harming the performance of models. Researchers mention that introducing hard negatives indistinguishably might result in performance degradation of bi-encoder architecture, and propose to leverage cross encoder for correcting the bias [19]. Adding random negatives along with hard negatives is also proposed as an effective solution to alleviate the false negative problem [30]. A recent study on recommender system [17] proposes to leverage the clustering approach to eliminate false negatives and thus obtain so-called "Real Hard Negatives". However, the method is designed to handle sequential information and is not suitable for our pair-wise ranking scenario.

Model leveraging similarity between positive and negative cases to estimate the probability of false negatives is also proposed in the domain of computer vision [32]. A study in the domain of knowledge graphs also proposes that labels generated by clustering algorithms are capable of guiding models to avert false negatives [8]. Similarly, supervised models rely on data augmentation and are utilized for identifying potential false negatives [7]. However, these methods require the availability of trainable embedding of entities, whose implementations are unfeasible while the cross-encoder is applied. It is also proposed that the introduction of an additional IPW-based risk model can eliminate the false negative [3], while its simultaneous training of two cross-encoder patterns will introduce considerable additional time consumption.

As the conclusion of this section, most of the previous works focus on false negative problems in the setting of bi-encoder, while research on cross-encoder regarding negative sampling is rare due to the lack of independent embedding of query/product that can guide the heuristic sampling process.

6 Conclusion

In this paper, we propose a novel method, referred to as False Negative Estimation, to mitigate pooling bias by identifying potential false negative samples that arise during the training of a cross-encoder for e-commerce retrieval at Instacart. False Negative Estimation utilizes query similarity to estimate the likelihood that a sampled query-product pair is a false negative. To comprehensively address this issue, we have developed two modules, namely the sampling regularization and pseudo label generation, each designed to mitigate the pooling bias introduced by false negatives. By combining these two modules, we establish the Bias-mitigating Hard Negative Sampling, denoted as BHNS. To validate the effectiveness of the proposed BHNS, we conduct experiments on both public data and Instacart private data. The success of BHNS on the STS Benchmark dataset implies its applicability extends beyond the e-commerce domain. Through a series of experiments conducted within the Instacart environment, we have demonstrated that BHNS significantly enhances the performance of the cross-encoder. This enhancement holds considerable promise for its application in the e-commerce setting.

References

[1] Negar Arabzadeh, Alexandra Vtyurina, Xinyi Yan, and Charles LA Clarke. 2022. Shallow pooling for sparse labels. *Information Retrieval Journal* 25, 4 (2022), 365–385.

[2] Anne Aula. 2003. Query Formulation in Web Information Search.. In *ICWI*. 403–410.

[3] 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 & Knowledge Management*. 118–127.

[4] Jaekeol Choi, Euna Jung, Jangwon Suh, and Wonjong Rhee. 2021. Improving bi-encoder document ranking models with two rankers and multi-teacher distillation. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*. 2192–2196.

[5] 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).

[6] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural Collaborative Filtering. In *Proceedings of the 26th International Conference on World Wide Web* (Perth, Australia) *(WWW '17)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 173–182. https://doi.org/10.1145/3038912.3052569

[7] Tri Huynh, Simon Kornblith, Matthew R Walter, Michael Maire, and Maryam Khademi. 2022. Boosting contrastive self-supervised learning with false negative cancellation. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*. 2785–2795.

[8] Sang-Hyun Je. 2022. Entity Aware Negative Sampling with Auxiliary Loss of False Negative Prediction for Knowledge Graph Embedding. *arXiv preprint arXiv:2210.06242* (2022).

[9] Yannis Kalantidis, Mert Bulent Sariyildiz, Noe Pion, Philippe Weinzaepfel, and Diane Larlus. 2020. Hard negative mixing for contrastive learning. *Advances in Neural Information Processing Systems* 33 (2020), 21798–21809.

[10] Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, 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).

[11] Jie Lei, Xinlei Chen, Ning Zhang, Mengjiao Wang, Mohit Bansal, Tamara L Berg, and Licheng Yu. 2022. Loopitr: Combining dual and cross encoder architectures for image-text retrieval. *arXiv preprint arXiv:2203.05465* (2022).

[12] Dan Li and Evangelos Kanoulas. 2017. Active sampling for large-scale information retrieval evaluation. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. 49–58.

[13] Junnan Li, Ramprasaath Selvaraju, Akhilesh Gotmare, Shafiq Joty, Caiming Xiong, and Steven Chu Hong Hoi. 2021. Align before fuse: Vision and language representation learning with momentum distillation. *Advances in neural information processing systems* 34 (2021), 9694–9705.

[14] Sheng-Chieh Lin, Jheng-Hong Yang, and Jimmy Lin. 2020. Distilling dense representations for ranking using tightly-coupled teachers. *arXiv preprint arXiv:2010.11386* (2020).

[15] Sheng-Chieh Lin, Jheng-Hong Yang, and Jimmy Lin. 2021. In-batch negatives for knowledge distillation with tightly-coupled teachers for dense retrieval. In *Proceedings of the 6th Workshop on Representation Learning for NLP (RepL4NLP-2021)*. 163–173.

[16] Yuxiang Lu, Yiding Liu, Jiaxiang Liu, Yunsheng Shi, Zhengjie Huang, Shikun Feng Yu Sun, Hao Tian, Hua Wu, Shuaiqiang Wang, Dawei Yin, et al. 2022. Ernie-search: Bridging cross-encoder with dual-encoder via self on-the-fly distillation for dense passage retrieval. *arXiv preprint arXiv:2205.09153* (2022).

[17] Haokai Ma, Ruobing Xie, Lei Meng, Xin Chen, Xu Zhang, Leyu Lin, and Jie Zhou. 2023. Exploring False Hard Negative Sample in Cross-Domain Recommendation. In *Proceedings of the 17th ACM Conference on Recommender Systems*. 502–514.

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

[19] Yingqi Qu, Yuchen Ding, Jing Liu, Kai Liu, Ruiyang Ren, Wayne 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).

[20] Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084* (2019).

[21] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian Personalized Ranking from Implicit Feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence* (Montreal, Quebec, Canada) *(UAI '09)*. AUAI Press, Arlington, Virginia, USA, 452–461.

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

[23] Guilherme Rosa, Luiz Bonifacio, Vitor Jeronymo, Hugo Abonizio, Marzieh Fadaee, Roberto Lotufo, and Rodrigo Nogueira. 2022. In defense of cross-encoders for zero-shot retrieval. *arXiv preprint arXiv:2212.06121* (2022).

[24] Afrina Tabassum, Muntasir Wahed, Hoda Eldardiry, and Ismini Lourentzou. 2022. Hard negative sampling strategies for contrastive representation learning. *arXiv preprint arXiv:2206.01197* (2022).

[25] 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).

[26] Ruoxi Wang, Bin Fu, Gang Fu, and Mingliang Wang. 2017. Deep & cross network for ad click predictions. In *Proceedings of the ADKDD 17'*. 1–7.

[27] Yuqing Xie, Taesik Na, Xiao Xiao, Saurav Manchanda, Young Rao, Zhihong Xu, Guanghua Shu, Esther Vasiete, Tejaswi Tenneti, and Haixun Wang. 2022. An Embedding-Based Grocery Search Model at Instacart. *arXiv preprint arXiv:2209.05555* (2022).

[28] 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).

[29] Hansi Zeng, Hamed Zamani, and Vishwa Vinay. 2022. Curriculum learning for dense retrieval distillation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1979–1983.

[30] Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Jiafeng Guo, Min Zhang, and Shaoping Ma. 2021. Optimizing dense retrieval model training with hard negatives. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1503–1512.

[31] 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).

[32] Zhaoyang Zhang, Xuying Wang, Xiaoming Mei, Chao Tao, and Haifeng Li. 2022. FALSE: False negative samples aware contrastive learning for semantic segmentation of high-resolution remote sensing image. *IEEE Geoscience and Remote Sensing Letters* 19 (2022), 1–5.

[33] Yao Zhou, Haonan Wang, Jingrui He, and Haixun Wang. 2021. From Intrinsic to Counterfactual: On the Explainability of Contextualized Recommender Systems. *CoRR* abs/2110.14844 (2021). https://arxiv.org/abs/2110.14844

[34] Yao Zhou, Jianpeng Xu, Jun Wu, Zeinab Taghavi Nasrabadi, Evren Körpeoglu, Kannan Achan, and Jingrui He. 2021. PURE: Positive-Unlabeled Recommendation with Generative Adversarial Network. In *KDD '21: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, Singapore, August 14-18, 2021*. ACM, 2409–2419.