CoDIME: A Counterfactual Approach for Dimension Importance Estimation through Click Logs

Guglielmo Faggioli
University of Padova
Padova, Italy

Nicola Ferro
University of Padova
Padova, Italy

Raffaele Perego
ISTI-CNR
Pisa, Italy

Nicola Tonellotto
University of Pisa
Pisa, Italy

Abstract

Contextual dense representation models for text marked a shift in text processing, enabling a richer semantic understanding of the text and more effective Information Retrieval. These models project pieces of text into a latent space, describing them in terms of shared latent concepts, which are not explicitly tied to the text's content. Previous work has shown that certain dimensions of such dense text representations can be irrelevant and detrimental to retrieval effectiveness depending on the information need specified in the query. Higher effectiveness can be achieved by performing retrieval within a linear subspace that excludes these dimensions. Dimension IMportance Estimators (DIMEs) are models designed to identify such harmful dimensions, refining the representations of queries and documents to retain only the useful ones. Current DIMEs rely either on pseudo-relevance feedback, which often delivers inconsistent effectiveness, or on explicit relevance feedback, which is challenging to collect. Inspired by counterfactual modelling, we introduce Counterfactual DIMEs (CoDIMEs), designed to leverage noisy implicit feedback to assess the importance of each dimension. The CoDIME framework presented here approximates the relationship between a document's click frequency and its interaction with a given query dimension through a linear model. Empirical evaluations demonstrate that CoDIME outperforms traditional pseudo-relevance feedback-based DIMEs and surpasses other unsupervised counterfactual methods that utilize implicit feedback.

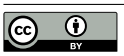
CCS Concepts

• **Information systems** → **Document representation**; **Query representation**; **Retrieval models and ranking**.

Keywords

Dimension Importance Estimators, Dense Information Retrieval, Counterfactual Modeling

ACM Reference Format:
Guglielmo Faggioli, Nicola Ferro, Raffaele Perego, and Nicola Tonellotto. 2025. CoDIME: A Counterfactual Approach for Dimension Importance Estimation through Click Logs. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '25), July 13–18, 2025, Padua, Italy*. ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3726302.3729926



SIGIR '25, July 13–18, 2025, Padua, Italy

ACM ISBN 979-8-4007-1592-1/2025/07
https://doi.org/10.1145/3726302.3729926

1 Introduction

Dense text representations have demonstrated remarkable capability in capturing semantic meaning, emerging as the dominant technology across numerous text-related tasks in Information Retrieval (IR) and Natural Language Processing (NLP). Early approaches to digital term representation for IR tasks primarily relied on lexical one-hot encoding techniques [36] and retrieval heuristics such as BM25 [34] and TF-IDF [22]. These methods represent a text as a sparse vector with a dimensionality equal to the size of the entire vocabulary of terms. Most vector elements are set to zero, with non-zero values corresponding only to the terms within the text. This type of representation offers several advantages. Its high sparsity facilitates efficient storage and computation, and its straightforward structure makes it highly interpretable. As a downside, it is affected by the semantic gap problem, i.e., the difference between the digital representation of a text and its human interpretation. For instance, it struggles to account for synonyms or polysemous words, leading to challenges in disambiguating word meanings or retrieving documents containing query term synonyms. Dense term representations, like Word2Vec [28] and GloVe [32], partially address the limitations of traditional methods. However, the advent of contextual term representations, exemplified by models such as BERT [9], marked a transformative shift in text encoding, enabling a richer semantic understanding. These representation models are based on neural networks that project the text onto a dense representation space where semantically similar contents tend to be arranged closely. While these novel representations are more effective than traditional lexical approaches in handling the semantic gap, they are far less interpretable, even if the dimensions of the representations are assumed to be associated with some latent semantic meaning. Starting from this, Faggioli et al. [13] propose the so-called *Manifold Clustering Hypothesis* which states that "*High-dimensional representations of queries and documents relevant to them tend to lie in a query-dependent lower-dimensional manifold of the representation space*". This hypothesis combines the well-known clustering [39] and manifold [1] hypotheses and states that it is possible to find a query-wise subspace of the dense representation space where the retrieval is more effective, i.e., where the representations of the query and its relevant documents are more aligned. While Faggioli et al. postulate that such a subspace can be an arbitrary

manifold, to validate their hypothesis empirically, they reduce the complexity of the problem by focusing only on linear subspaces, i.e., spaces obtained by zeroing some of the dimensions from the representation. To do so, Faggioli et al. define the concept of Dimension IMportance Estimator (DIME): a model explicitly meant to estimate the query-dependent importance of each dimension to preserve only the most important ones while discarding the others. In particular, Faggioli et al. propose DIMEs based on Pseudo-Relevance Feedback (PRF) or relying on explicit feedback. The former is known to have variable and not always consistent effectiveness, especially when it comes to dense models [26]. The latter, on the other hand, can be much more challenging to gather. To overcome this limitation, in this paper, we propose to employ an intermediate relevance signal, more reliable than PRF and far more available than explicit feedback: *implicit feedback*. Implicit feedback leverages the analysis of user interactions, such as clicks and dwell times, to infer weak relevance signals for retrieved content. A key source of this data are *query logs*, which, however, are generally not publicly accessible due to their added value for companies and the substantial presence of personally identifiable information[1]. The common practice in the IR community is thus to resort to synthetic query logs simulating realistic user interactions with an IR system. Also in this study, we employ a set of simulated click logs to estimate the frequency of clicks on the links in a Search Engine Result Page (SERP). By relying on such click frequencies, we devise a counterfactual modelling of the click probabilities. This model is then used as a source of implicit feedback information, and we exploit it to determine the importance of the dimensions in the dense representation space, thus instantiating a set of novel Counterfactual DIMEs (CoDIMEs). In particular, we design a set of *linear* CoDIMEs that quantifies the importance of a dimension by considering the characteristics of a linear model that regresses the documents' click frequency on the interaction between the query and the documents on such a dimension, i.e., the product of the representations on such dimensions. Empirically, we show that such models overcome the original DIME approach and can achieve state-of-the-art effectiveness using multiple backbone dense models and several commonly used IR test-beds.

Compared to CoRocchio [44], a state-of-the-art counterfactual approach, the CoDIME framework achieves up to +0.235 nDCG@10 points, moving from 0.404 to 0.639 (+58%) (Dragon and Robust '04) and +0.117 nDCG@100 points, moving from 0.356 to 0.473 (+33%) (Dragon and Robust '04). The remainder of this work is organised as follows. Section 2 introduces the relevant literature upon which we construct the CoDIME framework. Section 3 reports the description of the CoDIME approach and the methodological part of this paper. Finally, Section 4 details on the experimental evaluation, while in Section 5, we draw the conclusions.

2 Background and Related Work

In this section, we introduce the related work and the theoretical background underlying the development of the CoDIME framework. We start by introducing some notation. Let q be a query and $d \in C$ a document from a corpus C. A dense encoder ϕ is used to represent the documents and the queries. The encoder ϕ takes in input a text and projects it into a h-dimensional real space $\mathbb{R}^h$. Thus we write $\phi(q) = \mathbf{q}$ and $\phi(d) = \mathbf{d}$, i.e., the latent representations of q and d, respectively. The retrieval is operated by ranking the documents according to the dot product between the query and the documents' representations. In the following, we illustrate the related works on dense IR and DIME (Section 2.1), and relevance feedback (Section 2.2), while in Section 2.3 we provide background and discuss related works on implicit feedback and its biases, concluding with an existing application of the implicit feedback modeling in dense IR (Section 2.4).

2.1 Dense IR

Traditionally, dense IR systems are divided into three main categories: bi-encoders, cross-encoders, and late interaction models [43]. Regardless of the category, the most recent and effective solutions are based on the transformers [40] architecture and BERT [9]. Cross-encoders, such as Electra [5] jointly project documents and queries within the same latent space, thus preventing the pre-computation of an index data structure. On the other hand, late interaction models, such as ColBERT [23], are based on matching the contextual representation of the single terms in the documents and queries and tend to be less efficient from the space perspective. Finally, bi-encoder models, or dual encoders, use two separate neural networks to encode documents and queries. These networks can be identical (symmetric bi-encoders) or distinct (asymmetric bi-encoders). This architecture enables the precomputation of document representations, which can be stored in specialised index structures such as FAISS [10] for efficient retrieval. While bi-encoders might be slightly less effective than some alternative approaches, they balance efficiency and effectiveness.

Dimension IMportance Estimators. When using bi-encoders for retrieval, Faggioli et al. [13] proposed the manifold clustering hypothesis, which posits the existence of a query-dependent subspace within the dense embedding space where encoding is more effective. While Faggioli et al. conjecture that the optimal subspace could be an arbitrary manifold, they also observe that the hypothesis holds even when a linear subspace—i.e., a subspace of the original space with certain dimensions removed—is employed. Building on this hypothesis, Faggioli et al. define the concept of DIMEs. A DIME $u(\mathbf{q}|\theta) \in \mathbb{R}^h$ is a model that takes in input the representation of a query $\mathbf{q}$, possibly some additional information θ, and outputs an h-dimensional real-valued vector where the i-th element describes the "importance" of the i-th dimension. Empirical observations by Faggioli et al. reveal that truncating the query and document representations to retain only the top $\alpha \cdot h$ most important dimensions, where $\alpha \in (0,1)$ is a parameter, leads to improved retrieval performance compared to using the full representation. Most of the DIMEs proposed by Faggioli et al. were based on either PRF signal or shallow active feedback. In this paper, we investigate how to build a set of DIMEs that rely on implicit feedback and counterfactual modelling.

2.2 Relevance Feedback

In the most classical definition, relevance feedback approaches employ some form of feedback to direct the retrieval towards relevant documents, for example, by expanding a query with terms that appear in relevant documents. Relevance feedback approaches are based on real user feedback or Pseudo-Relevance Feedback (PRF). Approaches based on users' feedback are further divided into approaches that use explicit feedback, where the user explicitly

[1] https://en.wikipedia.org/wiki/AOL_search_log_release

identifies relevant and non-relevant documents, e.g., Rocchio algorithm [35], and implicit feedback, where implicit signals, such as the users' clicks or dwell time, are used as feedback. In the case of PRF, the information on which documents are thought to be relevant, also known as pseudo-relevant signal, is either generated by the model itself, e.g., RM3 [24], or might be generated through external tools, such as Large Language Models (LLMs) [27]. While approaches based on explicit feedback rely on exact signals, their main limitation is data scarcity. On the contrary, approaches based on PRF rely on the most available data but often have unstable performance, heavily dependent on the quality model that generated the pseudo-relevant signal. A mid-ground is represented by implicit feedback. In this case, the feedback is collected passively, without explicit user input, and thus can be acquired in large amounts. Nevertheless, it is more reliable than the PRF because it is provided directly by a human. Implicit feedback comes with its challenges, such as noise and the bias that the user might introduce when interacting with the documents.

2.3 Implicit Feedback & Biases

Users continuously generate data as implicit feedback when interacting with an IR system by issuing queries, examining SERPs, and interacting with documents they perceive as relevant.

Implicit feedback is far more abundant than other types of feedback, such as editorial or crowd-sourced feedback, because it is easier and less expensive to collect. Additionally, implicit feedback reflects users' genuine preferences, free from the influence of crowd-workers or editorial assessors. However, implicit feedback is subject to significant biases that must be addressed before using it. For example, a user might accidentally interact with a document and later determine it to be irrelevant or not recognize a relevant document and ignore it. Similarly, the order in which documents are presented to the user affects the chance that the user will interact with them. Finally, only a small subset of documents can be reasonably shown to the user in response to a query: we cannot collect any feedback on documents not shown. To leverage implicit feedback effectively, an IR system must account for these biases and implement strategies to mitigate their impact.

A large body of literature [19, 21, 29–31, 42, 44–46] agrees on three sources of bias affecting the interactions between the user and the SERP: the *position bias*, the *selection bias*, and the *relevance bias*.

To formalize the implicit feedback from a user searching for a given query q, we consider three events for any given document d: E is the 'examination' of the document d, R is the 'relevance' of d in response to q, and C corresponds to the 'click' of the user on d. While E and C are two binary events (either the user examines or clicks on the document, or not), the relevance R is typically modelled as a categorical event, e.g., not relevant, partially relevant, relevant, and highly relevant. Finally, K corresponds to the rank at which a document d is ranked in response to the query q by the IR system.

The *position bias* describes the tendency of the users to examine the ranked documents based on their position. For example, following previous studies on eye-tracking [20], it is more likely that a user will inspect documents ranked higher. Thus, the probability that a user will examine a document depends on the position at which the document is presented, i.e., $P(E=1|K)$. Following previous studies [19, 21], it is common to model this probability as inversely proportional to the position, i.e., $P(E=1|K) = (1/K)^{\eta}$, where the parameter η describes the 'patience' of the user. A patient user that will look at many documents before stopping can be represented using a small η. Vice-versa, an impatient user can be modeled with a large η. One of the most common de-biasing approaches, the Inverse Propensity Score (IPS) weighting, consists of dividing the observed probability by the estimated propensity score [16, 21].

The *selection bias* corresponds to the fact that only some documents will be selected by the IR system to be shown to the user. This bias can be modelled by saying that $P(E=1|K')=0$ for all $K' > K$.

The final bias to account for is the *relevance bias*. This bias models the fact that relevant documents are more likely to be clicked. In particular, the probability of a click on a document in response to a query is conditioned on the document being examined and its relevance in response to the query, i.e., $P(C|E=1,R)$. How this probability is estimated corresponds to defining multiple user models. For example, we can imagine the *perfect user* for which $P(C|E=1,R) \propto R$. The perfect user clicks on a document proportionally to its relevance: they will never click on a non-relevant document and always click on a relevant one. Similarly, we can define a "noisy" user who might click on non-relevant documents or miss relevant ones.

To combine all the different biases, following previous literature [21, 33], we model the probability $P(C)$ that a click will occur as: $P(C) = P(E=1|K) \times P(C|E=1,R)$. In other words, the probability of the click corresponds to the probability of the document being examined and that the document is clicked, given that it is examined and has a certain relevance. The advantage of being able to define such a model and define mathematically the bias underlying the clicks of the user is that it allows us to define a *counterfactual* framework [17], where we can account for the bias and remove it from the observational data that can be collected through a real click-log, without the need for interventions, such as modifying the SERP for different users. To estimate such probabilities in a practical scenario, several approaches can be adopted, such as the use of a supervised click model [2–4, 12, 14].

2.4 PRF via Implicit Feedback for Dense IR

An interesting application of the counterfactual framework aims to combine the information gained by considering the clicked documents with dense IR. The most prominent example is the Counterfactual Rocchio (CoRocchio) approach [44]. The CoRocchio approach extends Rocchio's algorithm [35] applied to dense IR system proposed by Li et al. [25] to take into account implicit feedback instead of explicit relevance feedback.

Let $\mathcal{R} = \{d_1, ..., d_k\}$ denote the SERP of length k returned in response to the query q. Assume a set of users $\mathcal{U}$ has an information need that can be expressed using the query q. After issuing the query to the IR system and obtaining the SERP $\mathcal{R}$, each user interacts with it, producing a click log $C_u = \{c_{u,d_1}, ..., c_{u,d_k}\}$. More in detail, the value of $c_{u,d}$ is 1 if the user u clicks on the document $d \in \mathcal{R}$, and 0 otherwise.

Given the dense representation of the query $\mathbf{q}$ and free parameters $\beta_1, \beta_2 > 0$, the CoRocchio approach constructs a new query representation $\mathbf{q}^*$ as:

$$\mathbf{q}^* = \beta_1 \mathbf{q} + \beta_2 \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \sum_{d \in \mathcal{R}} \frac{\mathbf{d}}{p(E=1|i)} c_{u,d},$$

where $\mathbf{d}$ is the dense representation of the document d while $p(E=1|i)$ is the estimated examination propensity at rank i. Following previous literature, Zhuang et al. [44] approximate $p(E=1|i)$ as $(1/i)^{\eta}$. In other terms, according to the CoRocchio approach, the new representation of the query $\mathbf{q}^*$ is a linear combination of the original representation of the query with the representations of the documents clicked, weighted by their click frequency debiased through the IPS. Zhuang et al. prove that if every relevant passage has a positive examination propensity, CoRocchio provides an unbiased estimate of the query concerning the propensity.

3 The CoDIME Framework

This section introduces the CoDIME methodology and its theoretical definition. Following the notation introduced in Section 2, we refer to the list of the top k documents retrieved in response to a query q as $\mathcal{R}=\{d_1,...,d_k\}$. All users $\mathcal{U}$ who submitted the query q interact with $\mathcal{R}$, each one generating a click log C_u such that $c_{u,d}$ is either 1 or 0, depending on whether the user clicked on document $d \in \mathcal{R}$ or not. Based on this historical information, we can compute the observed frequency of clicks of $\mathcal{U}$ for q and d as: $\hat{f}_d = \frac{1}{|\mathcal{U}|}\sum_{u\in\mathcal{U}} c_{u,d}$. In other terms, $\hat{f}_d$ describes the proportion of clicks received by a document $d \in \mathcal{R}$ retrieved in response to q. Following the implicit feedback modelling described in Section 2.3, we can expect this frequency to be somewhat correlated with the relevance of the top k documents retrieved in response to q, but also with the position at which the document is shown in the $\mathcal{R}$. As a consequence, we need to debias it. Akin to CoRocchio, we debias click frequencies using the IPS [16, 21]. The debiased click frequency is thus defined as: $f_d = \hat{f}_d \cdot (1/k)^{-\eta}$ Where k is the position at which the document d was observed in $\mathcal{R}$ and η is the propensity parameter. The debiased click frequencies describe how likely it is that a document is clicked, regardless of where it is placed in the SERP. Given a query q and a document d and their respective dense representations $\mathbf{q}$ and $\mathbf{d}$, we can define the interaction H_d between them as the Hadamard product between their representations. The interaction is a vector, and $H_{d,i}$, the i-th element of H_d, indicates how much the query and the document interact on the i-th dimension. Assuming each dimension corresponds to a latent concept, observing strong interaction between the query and the document on such dimension indicates that the concept is prime for the query and the document. Conversely, weak interaction suggests either the document does not concern the concept, or the concept is irrelevant to the query. By construction, the dot product between the query and the document corresponds to the sum of the interaction elements in H_d. Suppose the query and the document interact strongly on several dimensions: we can assume they are aligned, and the document is likely ranked highly.

Finally, we define $\left(f_{d_1},...,f_{d_k}\right)$ as the list of debiased click frequencies of the top k documents for q, included in $\mathcal{R}$.

3.1 Magnitude-based CoDIME Approaches

Based on our definition of the debiased click frequencies f_d and the interaction H_d, we can define two CoDIMEs that employ the magnitude of the interaction as an indicator of the importance of the dimensions.

Weighted Average CoDIME. This CoDIME, dubbed CoDIME$_{wavg}$, is inspired by CoRocchio [44], and it relies on computing the weighted centroid of the interaction matrix to identify the most relevant dimensions. More in detail, the importance of each dimension is computed as the average of the interactions between the query and the documents on such dimensions weighted by the debiased click frequency of the documents. Formally, given a dimension $i \in \{1,...,h\}$ of a representation, its importance according to CoDIME$_{wavg}$ is defined as:

$$CoDIME_{wavg}(i) = \frac{1}{k}\sum_{d\in\mathcal{R}} H_{d,i}\cdot f_d$$

In other terms, the importance of all dimensions can be derived by considering the centroid of the documents in the click log, weighted by their debiased click frequency. This mimics the feedback used by CoRocchio. The major difference between the two approaches is how the feedback is used. In the case of CoRocchio, this vector is linearly combined with the query vector. For CoDIME$_{wavg}$, the feedback vector is used to identify the most relevant dimensions of the query representation and discard all the dimensions that are not important, leaving the others unaltered.

Weighted Max CoDIME. The Weighted Average CoDIME described above might fail to treat unclicked documents correctly. Assume that a single document of those shown to the user is clicked. This would deflate the Weighted Average CoDIME, as several inputs of the average sum would be zeros. Similarly, assume a faceted query is issued to the IR system. In such cases, documents might have orthogonal representations, insisting on different dimensions, all equally important for the query. When using the average, the contribution of such dimensions might be decreased due to the orthogonality of the documents. To mitigate these phenomena, we propose to replace the mean aggregation with the maximum. The Weighted Max CoDIME, or CoDIME$_{wmax}$, is formally defined as follows:

$$CoDIME_{wmax}(i) = \max_{d\in\mathcal{R}} H_{d,i}\cdot f_d$$

3.2 Linear CoDIME Approaches

Linear CoDIME approaches estimate the importance of a dimension for a query by examining the linear correlation between the dimension's interaction and the debiased click frequency on the documents. In other words, consider a scenario where the query and a document exhibit strong interaction on a particular dimension. This indicates that the dimension heavily impacts the document's position within the ranking constructed in response to the query. Consider now the debiased click frequency on such a document. If the document is clicked often—regardless of its position— the document is more likely to be relevant and should be ranked high. Vice-versa, if this document is rarely clicked, excluding it from the top-ranked ones would be better. Thus, if the interaction on a dimension between a clicked-often document and the query is big, then such a dimension is likely to be important. On the contrary, if the document is clicked often but the interaction is small, or if the document is clicked rarely and the interaction is large, there is a misalignment that suggests the dimension is noisy and should be removed from the representation.

Correlation CoDIME. The first linear CoDIME is inspired by the *Oracle DIME* as proposed by Faggioli et al. [13]. More in detail, we define $\overline{f}$ and $\overline{H}_i$ as the mean debiased click frequency and the interaction on the i-th component for a given query and the corresponding

retrieved documents, respectively. Called ρ the Pearson's correlation, the Correlation CoDIME, or CoDIME$_{corr}$, is defined as follows:

$$CoDIME_{corr}(i) = \rho\big((f_{d_1},\dots,f_{d_k}),(H_{d_1,i},\dots,H_{d_k,i})\big) = \frac{\sum_{d\in\mathcal{R}}(f_d-\overline{f})(H_{d,i}-\overline{H}_i)}{\sqrt{\sum_{d\in\mathcal{R}}(f_d-\overline{f})^2}\sqrt{\sum_{d\in\mathcal{R}}(H_{d,i}-\overline{H}_i)^2}}.$$

This CoDIME quantifies the linear correlation between the interactions on a given dimension and the debiased click frequencies as their Pearson's ρ correlation. If the interaction on a dimension between the query and the documents aligns with the debiased click frequencies on such documents, the importance will be 1, and the dimension likely belongs to the optimal subspace. If the interaction and the debiased click probability are uncorrelated, the importance of the dimension will be zero. Finally, when the interaction and the debiased click probability have a negative relation, the importance is negative, and the dimension will likely be discarded.

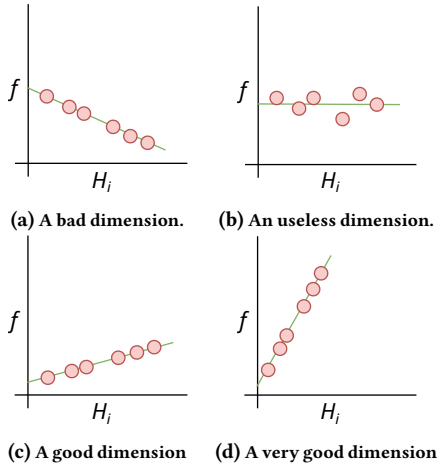
Slope CoDIME. One of the major limitations of the Correlation CoDIME is that it cannot consider how fast the interactions and the click frequencies tend to vary. In fact, the linear model that best fits the points might be more or less steep. A steeper linear model indicates that the dimension is better at separating the good and the bad documents. Vice-versa, if the linear model grows slowly, it is harder to separate documents clicked often from rarely clicked documents. The value of the Correlation CoDIME does not depend on such steepness, but only on how well a linear model fits the data. Therefore, we propose a second linear CoDIME that explicitly quantifies the dimension's importance based on the slope of the linear model that best fits the data according to the Ordinary Least Square (OLS) approach.

In more detail, let us call $\mathbf{H}_i \in \mathbb{R}^{k\times 2}$ a matrix such that its first column contains k 1s and the second column contains the values $H_{d_1,i},\dots,H_{d_k,i}$. This is the regressor matrix, while we treat $\mathbf{f} = [f_{d_1},\dots,f_{d_k}]^\top$ as the response variable. We fit a linear model using the OLS approach by computing $\mathbf{b}_i \in \mathbb{R}^2$: $\mathbf{b}_i = (\mathbf{H}_i^\top \mathbf{H}_i)^{-1}\mathbf{H}_i^\top \mathbf{f}$. Since we added a column of ones to the regressor matrix, the first element $b_{i,1}$ of $\mathbf{b}_i$ is the intercept of the OLS linear model while the second element $b_{i,2}$ of $\mathbf{b}_i$ is the slope.[2] The CoDIME$_{slope}$ is defined as:

$$CoDIME_{slope}(i) = b_{i,2}.$$

Figure 1 reports an illustrative visual comparison between different dimensions and how they would be considered based on the Linear CoDIMEs. Each plot represents a different dimension, and each dot represents a document. The debiased click frequency is reported on the y-axis of the figure, while, on the x-axis, we have the interaction between the document and the query (i.e., the values in the interaction H_i). Based on our intuition, the scenario depicted in Figure 1a describes a harmful dimension. The query and documents interaction and the debiased click frequency are inversely proportional. This means that the dimension pushes up documents with low debiased click frequency and ranks low those clicked often. Correlation and Slope CoDIME would assign a negative score to this dimension and likely remove it. Subsequently, Figure 1b illustrates what happens for a non-informative dimension: the debiased click frequency and the interaction values are completely uncorrelated. This suggests

[2]We also experimented with a linear model without the intercept, obtaining slightly inferior empirical results.



(a) A bad dimension. **(b) An useless dimension.**

(c) A good dimension **(d) A very good dimension**

Figure 1: A comparison between dimensions. Each dot represents a document in $\mathcal{R}$. The y-axis reports the debiased click frequencies $(f_{d_1},\dots,f_{d_k})$ and the x-axis reports the interaction values $H_{d_1,i},\dots,H_{d_k,i}$. The more the query-documents interaction on a dimension separates often and rarely clicked documents, the better such dimension is.

that the dimension does not help separate documents that are more likely to be clicked and documents that would not be clicked. Both Linear CoDIMEs would assign a score close to zero to this dimension. Finally, Figures 1c and 1d depict dimensions on which the interaction between the query and the documents helps separate documents that are likely to be clicked from those that are not. The major difference between Figure 1c and 1d lies in the scores that would be assigned by the CoDIMEs. In both cases, the Correlation CoDIME would assign a score of 1, as the linear correlation is positive and perfect. Vice versa, the Slope CoDIME would assign a larger score to the dimension illustrated in Figure 1d. This behavior lets us recognize better dimensions that best separate frequently and rarely clicked documents.

3.3 On the optimal dimension cutoff

All DIMEs and our CoDIMEs are based on preserving the most important dimensions of the query and documents' representations while discarding the least important ones by setting them to 0. This approach requires a cutoff α to decide how many dimensions should be kept. Faggioli et al. [13] did not investigate how to tune such α: as a result, they reported the performance at various α levels. We propose to choose the optimal α based on cross-validation. More in detail, we divide the queries in m folds (5 in our experimental section), identify the best α by averaging the results across $m-1$ folds, use the performance corresponding to such α on the remaining fold as the test effectiveness, and repeat the procedure using each fold as the test. In practice, this can be implemented using a historical set of annotated queries.

4 Experimental Evaluation

4.1 Experimental Setup

to assess the proposed counterfactual strategy we consider three well-known state-of-the-art dense encoders: Contriever [18], TAS-B [15], and Dragon [26], fine-tuned on MSMARCO.[3] While Contriever and TAS-B are based on a symmetric query and document

[3]We use the model weights publicly available on https://huggingface.co/

encoder, Dragon uses a different encoder for queries and documents. Experiments are conducted on three well-known TREC collections: TREC Robust 2004 (Robust '04) [41], TREC Deep Learning 2019 (DL '19) [8], and TREC Deep Learning 2020 (DL '20) [7]. The Robust '04 collection contains 249 topics and is based on the Tipster disks 4 and 5 corpus of documents, minus the congressual records. Notice that, since the dense models have been trained on a different corpus and with different types of texts – passages instead of documents – this collection can be considered as an out-of-domain scenario. On the contrary, DL '19 and DL '20 contain 43 and 54 topics respectively and are based on the MSMARCO passages corpus; therefore, they are in-domain applications of the dense models.

The parameter η, describing the user's patience in clicking a document of the SERP, is set to 1, while the maximum depth of inspection is set to 20 documents unless specified differently. Furthermore, the experiments are conducted by repeating 1000 times for each topic the simulation of the click log. Differently from Faggioli et al. [13], our CoDIMEs strategies choose the fraction α of representation dimension retained by applying a 5-fold cross-validation on the validation set (see Section 3.3). The code and the data are publicly released.[4]

4.2 Click Logs Simulation

The CoDIME approaches are based on historical user feedback needed to instantiate the counterfactual framework and estimate the click probabilities. In a real-world deployment with a consistent user base, click logs are easy to collect and use for our purposes. In our experimental analysis, we use the TREC Deep Learning (DL) collections, which are based on the MSMARCO dataset. While the MSMARCO dataset has an associated click log, the ORCAS dataset [6], such a click log is not suited for our case study. This log reports, in fact, information about clicked query and document pairs, but this information is not aligned with the TREC DL collections. Moreover, the clicks refer to entire documents and not to passages as the TREC DL collections, and any detail on the documents' presentation order is missing, preventing us from estimating the propensity bias. Other datasets, such as the one used for the Personalized Web Search Challenge[5], do not report the textual content of the documents and queries, nor the relevance assessments. These two characteristics make it impossible to compute the dense vectors used as the basis of the CoDIME framework, and it prevents us from computing the measures traditionally used in offline evaluation.

Thus, following the previous literature [19, 21, 29–31, 42, 44–46] on counterfactual implicit feedback and learning to rank, we simulate the interaction of the users with the documents to generate a set of synthetic click logs. More in detail, following previous work, to generate the synthetic click log, we need to simulate i) the selection bias, ii) the position bias, and iii) the relevance bias.

The selection bias is implemented by assuming that every user interacts and inspects the SERP up to the document in position k'. To simulate the click propensity, akin to the literature on counterfactual learning-to-rank [19, 21], we model $\hat{p}_{e,i}$, the probability of examination, as inversely proportional to the position, i.e., $\hat{p}_e(k) = \left(\frac{1}{k}\right)^{\eta}$. To simulate the relevance bias, we need to model $\hat{p}_r(q,d)$, the probability that the user will click on a document d, given its relevance to q. More in detail, we consider three ideal user models: the *perfect user* (P) whose click probability is directly proportional to the relevance of the document; the *binarized user* (B) that, clicks on a non-relevant or partially relevant document with probability 0.1 and clicks on a relevant or highly relevant document with probability 1; the *near random user* (R) that clicks on a non-relevant document with probability 0.4 and clicks on a highly relevant document with probability 0.6. Table 1 reports the click probabilities for the user models described above for a four-grade relevance assessments collection. For the perfect and near-random users, the probabilities are a linear spacing between the minimum and maximum probabilities, respectively 0 and 1 and 0.4 and 0.6, with as many steps as the relevance grades. For the binarized user, the click probability of a document with relevance within the lowest half of the grades is set to 0.1; otherwise, it is set to 1.

Thus, the simulated click probability is computed as: $\hat{p}_c(q,d,k) = \hat{p}_r(q,d) \cdot \hat{p}_e(k)$. In other terms, to simulate the click of a user on a document d retrieved in position k in response to the query q, we combine, by multiplying, the probability that the user will click on such document given its relevance to the query (i.e., the relevance bias) and the probability that the user will click on a document in position k, regardless of its relevance (i.e., the position bias).

Table 1: Click probabilities for the simulation for a four-grade relevance labels collection (e.g., DL '19 and DL '20).

	document relevance			
user model	non-relevant	partially relevant	relevant	highly relevant
Perfect (P)	0.00	0.33	0.67	1.00
Binarized (B)	0.10	0.10	1.00	1.00
Near Random (R)	0.40	0.47	0.53	0.60

4.3 Considered baselines

Vector PRF (VPRF) [25]. We employ the Rocchio variant of VPRF described by Li et al. [25] which combines linearly the centroid of the top-k documents retrieved with the query vector. We use $\alpha = 0.4$ and $\beta = 0.6$, following Li et al.. Furthermore, we test both with $k = 20$ (VPRF-20) to be comparable with the other approaches, but also with $k = 5$ (VPRF-5), being the most effective setting according to Li et al.. An important note on VPRF is that it is ineffective by construction with asymmetric encoders such as Dragon. In fact, for such encoders, documents' and queries' representations lie on two different latent spaces; thus, their linear combination produces a resulting vector that is semantically not meaningful. To address this limitation, for Dragon, during the query expansion phase, we employ the query encoder for both the query and the feedback documents.

LLM DIME [13]. The LLM DIME employs a LLM generated pseudo-relevant document as feedback. In particular, according to this DIME, the importance of the i-th dimension is the interaction between the query and the document on such dimension, i.e., the product of the values of the query and document representations on that dimension. To generate the pseudo-relevant documents, we use LLama 3.1 [11] with 70B parameters.[6]

PRF DIME [13]. The PRF DIME estimates the importance of a dimension as the magnitude in such dimension of the centroid of the interaction vectors between the top-k documents and the query.

[4] https://github.com/guglielmof/25-SIGIR-FFPT

[5] https://www.kaggle.com/c/yandex-personalized-web-search-challenge

[6] https://huggingface.co/meta-llama/Llama-3.1-70B-Instruct

CoRocchio [44]. The approach is described in Section 2.4. Akin to VPRF, this approach combines the representation of the query and the documents and fails with asymmetric encoders. Therefore, for asymmetric encoders, we use the query encoder for both the query and the feedback documents.

Since our approach is unsupervised and does not require any training, we do not compare with supervised Counterfactual Learning-to-Rank solutions [19, 21, 29–31, 42, 45, 46].

4.4 Performance

Table 2: Performance comparison on DL '19. In bold the most effective approach, underlined the runner-up. The top-tier according to an ANOVA with Tukey's HSD post-hoc test at a significance level of 0.05 is marked with *. P, B, and R are the Perfect, Binarized, and Near Random user models.

	Contriever			Dragon			TAS-B		
	P	B	R	P	B	R	P	B	R
	nDCG@10								
retrieval only	0.674	0.674	0.674	0.740	0.740	0.740	0.717	0.717	0.717
VPRF-5	0.664	0.664	0.664	0.752	0.752	0.752	0.721	0.721	0.721
VPRF-20	0.636	0.636	0.636	0.732	0.732	0.732	0.667	0.667	0.667
DIME$_{LLM}$	0.742	0.742	0.742*	0.767	0.767	0.767	0.749	0.749	0.749
DIME$_{PRF}$	0.668	0.668	0.668	0.740	0.740	0.740	0.717	0.717	0.717
CoRocchio	0.804*	0.766*	0.632	0.830	0.824*	0.724	0.810*	0.780*	0.665
CoDIME$_{wavg}$	0.759	0.747	0.752*	0.764	0.780	0.762	0.749	0.749	0.747
CoDIME$_{wmax}$	0.774	0.760*	0.730	0.748	0.751	0.742	0.800*	0.774*	0.774*
CoDIME$_{corr}$	0.851*	0.828*	**0.810***	0.891*	**0.854***	0.831*	0.856*	0.835*	0.804*
CoDIME$_{slope}$	**0.855***	**0.829***	0.809*	**0.897***	0.854*	**0.842***	**0.863***	**0.839***	**0.821***
	nDCG@20								
retrieval only	0.655	0.655	0.655	0.726	0.726	0.726	0.679	0.679	0.679
VPRF-5	0.656	0.656	0.656	0.743	0.743	0.743*	0.701	0.701	0.701
VPRF-20	0.643	0.643	0.643	0.717	0.717	0.717	0.651	0.651	0.651
DIME$_{LLM}$	0.710	0.710	0.710	0.746	0.746	0.746*	0.724	0.724	0.724*
DIME$_{PRF}$	0.655	0.655	0.655	0.726	0.726	0.726	0.682	0.682	0.682
CoRocchio	0.761*	0.729*	0.634	0.805*	0.796*	0.721	0.771*	0.746*	0.642
CoDIME$_{wavg}$	0.724	0.719*	0.711*	0.728	0.742	0.717	0.700	0.715	0.704
CoDIME$_{wmax}$	0.745*	0.740*	0.710	0.735	0.732	0.737	0.747	0.738*	0.734*
CoDIME$_{corr}$	**0.796***	**0.786***	**0.777***	0.830*	0.815*	0.792*	0.807*	0.781*	0.760*
CoDIME$_{slope}$	0.796*	0.774*	0.770*	**0.838***	**0.817***	**0.793***	**0.821***	**0.785***	**0.775***
	nDCG@50								
retrieval only	0.642	0.642	0.642	0.686	0.686	0.686	0.650	0.650	0.650
VPRF-5	0.644	0.644	0.644	0.718	0.718	0.718*	0.677	0.677*	0.677*
VPRF-20	0.637	0.637	0.637	0.698	0.698	0.698	0.641	0.641	0.641
DIME$_{LLM}$	0.686*	0.686*	0.686*	0.709	0.709	0.709*	0.693*	0.693*	0.693*
DIME$_{PRF}$	0.647	0.647	0.647	0.686	0.686	0.686	0.653	0.653	0.653
CoRocchio	0.737*	0.708*	0.625	0.772*	**0.767***	0.698	0.741*	0.723*	0.626
CoDIME$_{wavg}$	0.681*	0.684*	0.665*	0.696	0.704	0.681	0.659	0.671*	0.666
CoDIME$_{wmax}$	0.712*	0.706*	0.675*	0.689	0.693	0.692	0.722*	0.708*	0.711*
CoDIME$_{corr}$	**0.743***	**0.735***	**0.721***	**0.778***	0.764*	0.751*	0.745*	0.725*	0.712*
CoDIME$_{slope}$	0.741*	0.721*	0.715*	0.766*	0.763*	**0.754***	**0.747***	**0.729***	**0.733***
	nDCG@100								
retrieval only	0.634	0.634	0.634	0.673	0.673	0.673	0.637	0.637	0.637
VPRF-5	0.647	0.647	0.647	0.703	0.703	0.703*	0.667	0.667*	0.667*
VPRF-20	0.636	0.636	0.636	0.692	0.692	0.692*	0.636	0.636	0.636
DIME$_{LLM}$	0.676*	0.676*	0.676*	0.700	0.700	0.700*	0.684*	0.684*	0.684*
DIME$_{PRF}$	0.643	0.643	0.643	0.673	0.673	0.673	0.647	0.647	0.647
CoRocchio	0.733*	0.703*	0.628	**0.757***	**0.750***	0.689*	**0.726***	0.710*	0.621
CoDIME$_{wavg}$	0.669	0.675*	0.655	0.669	0.683	0.668	0.626	0.652	0.644
CoDIME$_{wmax}$	0.698*	0.692*	0.666*	0.673	0.676	0.669	0.702*	0.696*	0.686*
CoDIME$_{corr}$	0.734*	0.716*	**0.708***	0.747*	0.741*	**0.729***	0.725*	**0.711***	0.703*
CoDIME$_{slope}$	**0.735***	**0.723***	0.699*	0.739*	0.739*	0.726*	0.723*	**0.711***	**0.712***

As a first experiment, we report the comparison in terms of effectiveness between the different approaches. Tables 2, 3, and 4 report the effectiveness of our solution and the competitors on DL '19, DL '20 and Robust '04 collections, respectively. In bold, we report the highest performance achieved, underlined the runner-up. At the same time, the symbol * denotes the top tier of systems (i.e., the set of systems deemed statistically not distinguishable) according to an ANalysis Of the VAriance (ANOVA) [37] with Tukey's Honestly Significant Differences (HSD) [38] pairwise multiple comparison test and significance level of 0.05. The columns (P, B, and R) correspond respectively to *Perfect*, *Binarized*, and *Near Random* users. Within each measure, the first line reports the performance of the base encoder. Notice that some of the baselines (the encoder itself, VPRF, DIME$_{LLM}$ and DIME$_{PRF}$) are not based on users' clicks, and thus, they are not affected by the user model, and their performance is the same across all user models.

Table 3: Performance comparison on DL '20. In bold the most effective approach, underlined the runner-up. The top-tier according to an ANOVA with Tukey's HSD post-hoc test at a significance level of 0.05 is marked with *. P, B, and R are the Perfect, Binarized, and Near Random user models.

	Contriever			Dragon			TAS-B		
	P	B	R	P	B	R	P	B	R
	nDCG@10								
retrieval only	0.672	0.672	0.672	0.718	0.718	0.718	0.684	0.684	0.684
VPRF-5	0.693	0.693	0.693	0.722	0.722	0.722	0.695	0.695	0.695
VPRF-20	0.647	0.647	0.647	0.707	0.707	0.707	0.649	0.649	0.649
DIME$_{LLM}$	0.717	0.717	0.717	0.750	0.750	0.750	0.712	0.712	0.712
DIME$_{PRF}$	0.663	0.663	0.663	0.718	0.718	0.718	0.679	0.679	0.679
CoRocchio	0.793*	0.780*	0.633	0.833	0.825*	0.703	0.807*	0.785*	0.628
CoDIME$_{wavg}$	0.741	0.757	0.745	0.738	0.731	0.740	0.713	0.719	0.718
CoDIME$_{wmax}$	0.774*	0.752	0.756*	0.732	0.731	0.732	0.794*	0.787*	0.779*
CoDIME$_{corr}$	**0.822***	0.822*	0.797*	0.883*	**0.872***	0.838*	0.849*	0.829*	0.808*
CoDIME$_{slope}$	0.820*	**0.824***	**0.806***	**0.885***	0.863*	**0.839***	**0.861***	**0.848***	**0.813***
	nDCG@20								
retrieval only	0.662	0.662	0.662	0.692	0.692	0.692	0.658	0.658	0.658
VPRF-5	0.667	0.667	0.667	0.701	0.701	0.701	0.666	0.666	0.666
VPRF-20	0.644	0.644	0.644	0.692	0.692	0.692	0.633	0.633	0.633
DIME$_{LLM}$	0.678	0.678	0.678	0.715	0.715	0.715	0.683	0.683	0.683
DIME$_{PRF}$	0.672	0.672	0.672	0.692	0.692	0.692	0.661	0.661	0.661
CoRocchio	0.746*	0.733*	0.625	0.784*	0.774*	0.688	0.755*	0.725*	0.608
CoDIME$_{wavg}$	0.698	0.712*	0.699	0.700	0.684	0.692	0.659	0.649	0.648
CoDIME$_{wmax}$	0.714	0.705	0.706*	0.711	0.705	0.703	0.730*	0.730*	0.724*
CoDIME$_{corr}$	**0.767***	**0.754***	**0.750***	**0.806***	**0.801***	**0.774***	**0.783***	0.746*	**0.757***
CoDIME$_{slope}$	0.756*	0.749*	0.749*	0.801*	0.795*	0.767*	0.782*	**0.776***	0.751*
	nDCG@50								
retrieval only	0.635	0.635	0.635	0.671	0.671	0.671	0.631	0.631	0.631
VPRF-5	0.643	0.643	0.643	0.681	0.681	0.681	0.640	0.640	0.640
VPRF-20	0.622	0.622	0.622	0.675	0.675	0.675	0.609	0.609	0.609
DIME$_{LLM}$	0.657	0.657*	0.657*	0.686	0.686	0.686*	0.663	0.663	0.663*
DIME$_{PRF}$	0.642	0.642	0.642	0.671	0.671	0.671	0.631	0.631	0.631
CoRocchio	**0.707***	**0.691***	0.607	**0.745***	**0.732***	0.663	0.718*	0.691*	0.593
CoDIME$_{wavg}$	0.651	0.662*	0.647	0.671	0.671	0.671	0.615	0.613	0.614
CoDIME$_{wmax}$	0.673*	0.665*	0.672*	0.677	0.675	0.675	0.683*	0.688*	0.675*
CoDIME$_{corr}$	0.698*	**0.691***	**0.689***	0.743*	0.728*	**0.720***	0.715*	0.696*	**0.697***
CoDIME$_{slope}$	0.693*	0.688*	0.682*	0.735*	0.731*	0.718*	**0.721***	**0.714***	0.694*
	nDCG@100								
retrieval only	0.628	0.628	0.628	0.659	0.659	0.659	0.633	0.633	0.633
VPRF-5	0.639	0.639	0.639	0.674	0.674	0.674*	0.644	0.644	0.644*
VPRF-20	0.618	0.618	0.618	0.667	0.667	0.667	0.616	0.616	0.616
DIME$_{LLM}$	0.661*	0.661*	0.661*	0.678	0.678	0.678*	0.659	0.659	0.659*
DIME$_{PRF}$	0.637	0.637	0.637	0.659	0.659	0.659	0.638	0.638	0.638*
CoRocchio	**0.700***	**0.690***	0.605	**0.738***	**0.725***	0.658	**0.715***	0.692*	0.595
CoDIME$_{wavg}$	0.644	0.652	0.642	0.659	0.659	0.659	0.622	0.622	0.621
CoDIME$_{wmax}$	0.664*	0.666*	0.658*	0.661	0.658	0.662	0.672*	0.677*	0.673*
CoDIME$_{corr}$	0.689*	0.687*	**0.680***	0.723*	0.721*	**0.705***	0.707*	0.686*	0.680*
CoDIME$_{slope}$	0.683*	0.680*	0.674*	0.717*	0.714*	0.702*	0.703*	**0.699***	**0.681***

For what concerns the DL '19 collection (Table 2), we notice that the most effective approaches for nDCG@10 and nDCG@20 are those based on the Linear CoDIMEs (CoDIME$_{corr}$ and CoDIME$_{slope}$). Both approaches have comparable performance, with no clear dominance between the two: in all the cases, the two approaches are statistically equivalent according to the chosen statistical testing procedure. In general, the approaches based on the weighted magnitude of the dimensions (CoDIME$_{wavg}$ and CoDIME$_{wmax}$) tend to

Table 4: Performance comparison on Robust '04. In bold the most effective approach, underlined the runner-up. The top-tier according to an ANOVA with Tukey's HSD post-hoc test at a significance level of 0.05 is marked with *. P, B, and R are the Perfect, Binarized, and Near Random user models.

	Contriever			Dragon			TAS-B		
	P	B	R	P	B	R	P	B	R
	nDCG@10								
retrieval only	0.465	0.465	0.465	0.461	0.461	0.461	0.447	0.447	0.447
VPRF-5	0.474	0.474	0.474	0.447	0.447	0.447	0.470	0.470	0.470
VPRF-20	0.469	0.469	0.469	0.434	0.434	0.434	0.411	0.411	0.411
DIME$_{LLM}$	0.515	0.515	0.515	0.478	0.478	0.478	0.485	0.485	0.485
DIME$_{PRF}$	0.479	0.479	0.479	0.461	0.461	0.461	0.450	0.450	0.450
CoRocchio	0.687	0.639	0.456	0.649	0.605	0.404	0.648	0.599	0.389
CoDIME$_{wavg}$	0.456	0.609	0.474	0.461	0.482	0.461	0.447	0.589	0.443
CoDIME$_{wmax}$	0.535	0.620	0.553	0.491	0.489	0.491	0.524	0.578	0.519
CoDIME$_{corr}$	**0.737***	0.723*	0.643*	0.725*	0.714*	0.625*	0.685*	0.677*	0.604*
CoDIME$_{slope}$	0.734*	**0.725***	**0.648***	**0.734***	**0.726***	**0.639***	**0.699***	**0.689***	**0.618***
	nDCG@20								
retrieval only	0.435	0.435	0.435	0.425	0.425	0.425	0.406	0.406	0.406
VPRF-5	0.446	0.446	0.446	0.422	0.422	0.422	0.435	0.435	0.435
VPRF-20	0.439	0.439	0.439	0.405	0.405	0.405	0.387	0.387	0.387
DIME$_{LLM}$	0.475	0.475	0.475	0.439	0.439	0.439	0.441	0.441	0.441
DIME$_{PRF}$	0.450	0.450	0.450	0.425	0.425	0.425	0.412	0.412	0.412
CoRocchio	0.605	0.568	0.425	0.571	0.541	0.385	0.566	0.528	0.364
CoDIME$_{wavg}$	0.435	0.536	0.435	0.425	0.434	0.425	0.406	0.518	0.406
CoDIME$_{wmax}$	0.475	0.553	0.489	0.443	0.444	0.440	0.459	0.504	0.457
CoDIME$_{corr}$	**0.650***	0.642*	**0.568***	0.635*	0.631*	0.549*	0.601*	0.597*	0.520*
CoDIME$_{slope}$	0.649*	**0.646***	0.567*	**0.642***	**0.641***	**0.554***	**0.614***	**0.607***	**0.539***
	nDCG@50								
retrieval only	0.406	0.406	0.406	0.389	0.389	0.389	0.376	0.376	0.376
VPRF-5	0.415	0.415	0.415	0.392	0.392	0.392	0.397	0.397	0.397
VPRF-20	0.411	0.411	0.411	0.386	0.386	0.386	0.364	0.364	0.364
DIME$_{LLM}$	0.440	0.440	0.440	0.403	0.403	0.403	0.409	0.409	0.409
DIME$_{PRF}$	0.427	0.427	0.427	0.389	0.389	0.389	0.388	0.388	0.388
CoRocchio	0.539	0.505	0.390	0.500	0.471	0.357	0.496	0.462	0.330
CoDIME$_{wavg}$	0.406	0.467	0.406	0.389	0.387	0.389	0.376	0.442	0.376
CoDIME$_{wmax}$	0.423	0.488	0.434	0.400	0.400	0.397	0.402	0.445	0.400
CoDIME$_{corr}$	0.565*	**0.560***	**0.498***	0.536*	0.536*	0.475*	0.514*	0.508*	0.448*
CoDIME$_{slope}$	**0.567***	0.560*	0.496*	**0.553***	**0.554***	**0.483***	**0.528***	**0.523***	**0.465***
	nDCG@100								
retrieval only	0.412	0.412	0.412	0.392	0.392	0.392	0.381	0.381	0.381
VPRF-5	0.420	0.420	0.420	0.393	0.393	0.393	0.400	0.400	0.400
VPRF-20	0.414	0.414	0.414	0.388	0.388	0.388	0.369	0.369	0.369
DIME$_{LLM}$	0.441	0.441	0.441	0.406	0.406	0.406	0.414	0.414	0.414
DIME$_{PRF}$	0.428	0.428	0.428	0.392	0.392	0.392	0.391	0.391	0.391
CoRocchio	0.525	0.495	0.390	0.488	0.460	0.356	0.487	0.453	0.333
CoDIME$_{wavg}$	0.412	0.456	0.412	0.392	0.392	0.392	0.381	0.424	0.381
CoDIME$_{wmax}$	0.424	0.484	0.435	0.401	0.401	0.401	0.399	0.437	0.401
CoDIME$_{corr}$	0.549*	0.543*	0.488*	0.520*	0.516*	0.464*	0.496*	0.491*	0.438*
CoDIME$_{slope}$	**0.550***	**0.545***	**0.489***	**0.535***	**0.535***	**0.474***	**0.508***	**0.503***	**0.448***

be far less effective and are generally surpassed by the CoRocchio baseline. Comparing the Linear CoDIMEs with the most effective baseline, CoRocchio, we notice that the CoDIMEs are almost always more effective than CoRocchio for cutoffs lower than 100 (the only exception is Dragon with nDCG@50 and the binarized model). If we consider nDCG@100, CoRocchio is more effective than the CoDIMEs in the case of TAS-B and Dragon with a perfect or binarized user model but the difference is not statistically significant and small (0.01 or less nDCG@100 points). Compared to CoRocchio, all CoDIMEs are less vulnerable to changes in the user model considered. In fact, for CoRocchio, when moving from the perfect to the near-random user model, we notice a performance drop which is in the range of 10 to 15 points, depending on the considered measure or cutoff. Vice versa, the CoDIMEs tends to be stable, with variations between the perfect and near-random users in the 1-3 points range, up to 5 points in the worst scenarios. This is a desirable property: in a real-world scenario, where the clicks are far more affected by noise than in a simulated environment, a more stable solution as the Linear CoDIMEs offers better guarantees of a good performance.

In line with the tests on DL '19, for DL '20 (Table 3), the CoDIME approaches are particularly effective with the binarized and random user models and measures at small cutoffs. With cutoffs equal to or above 50, the CoRocchio approach is more effective in the case of the perfect user model. Nevertheless, the difference is not statistically significant and generally small (0.015-0.002 nDCG points).

The effectiveness of the CoDIME approach is even more evident when we consider Robust '04 (Table and 4). In this case, the linear CoDIMEs, especially CoDIME$_{slope}$, are always the best. Furthermore, in every case, this difference is statistically significant. This is in line with what was observed by Faggioli et al. [13], who noticed that, in the out-of-domain scenario, the DIMEs are particularly effective, as they allow to denoise the representation. Since the representation model was trained on a different distribution, it is more likely that it contains more noise, and thus, it is easier to debias.

In the remainder of this work, we focus on the CoDIME$_{slope}$ being, together with the CoDIME$_{corr}$, the most effective solution. Similar patterns can be observed with the other approach.

4.5 Effect of the Selection Bias

As a second analysis, we are interested in investigating the impact of selection bias on the performance of the proposed CoDIME framework. In particular, in the experiment described above, the length of the click log simulation was set to 20 (i.e., we considered only the 20 top-ranked documents to simulate the clicks). In this case, we experiment with different sizes of the click logs, considering click logs of length {2, 5, 10, 20, 50}. Figure 2 illustrates the consequences of varying the length of the click log when we consider as backbone model Contriever and TAS-B (blue and orange respectively); we do not consider Dragon, as CoRocchio achieves performances close to 0 due to its asymmetrical nature. The continuous lines represent the performance of CoDIME$_{slope}$, while the dashed lines indicate the performance of CoRocchio. Furthermore, we report only nDCG@10 and nDCG@20 to avoid encumbering — the patterns remain the same (although less marked) also with measures with longer cutoffs. As a first pattern, we notice that if we compare the behaviour with click-logs generated by a perfect user model ('P' columns, on the left) and by a near-random user model ('R', on the right), the difference between the two counterfactual approaches is larger on the latter. This is in line with what was observed until now. If the information is precise, both approaches perform sufficiently well, but as soon as the information gets more noisy, CoDIME$_{slope}$ is superior by far. Interestingly, the plots show that, for very short click-logs (2 documents), the CoRocchio approach tends to be slightly more effective, especially for TAS-B (orange dashed line above at the beginning). Nevertheless, as soon as the click log is 5-10 documents long, the CoDIME$_{slope}$ approach is the most effective in almost all scenarios. When we reach 20 or more documents, CoDIME$_{slope}$ is always the best. Furthermore, for the CoDIME$_{slope}$ in the lines are monotonically non-decreasing — the only exception is represented by nDCG@10 with the near-random user model. This suggests that the more information is available, the better the model performs, or, in the worst case, the performance remains the same. On the contrary, for CoRocchio and the near-random user model, we observe decreasing lines that indicate that CoRocchio

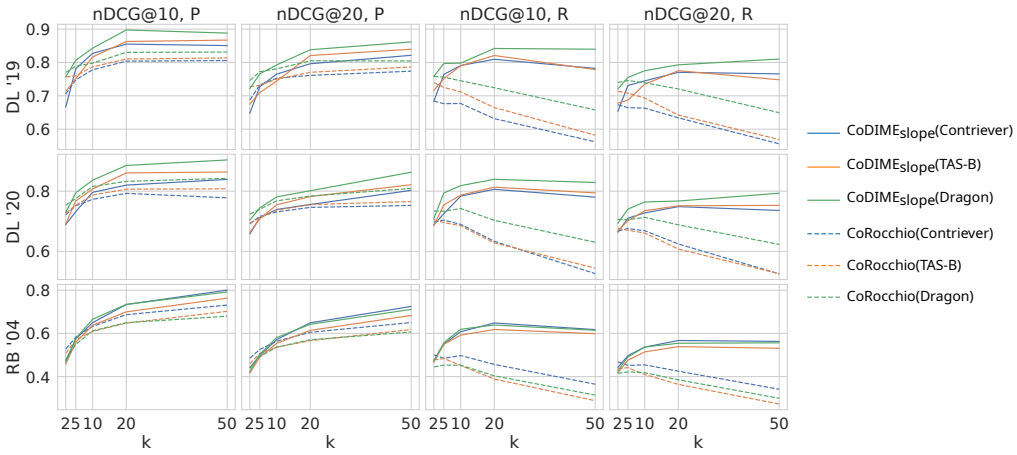


Figure 2: Effect of varying the selection bias (i.e., k) by including more documents on the click log simulations. P and R are the perfect and near-random user models. For $k = 2$, CoDIME$_{slope}$ and CoRocchio have similar effectiveness. The CoDIME$_{slope}$ is more effective in handling the scenario where more feedback is available. This difference is particularly evident for the near-random (R) user model. The same patterns also occur for measures with higher cutoffs.

not only fails to exploit the additional information but is harmed by it. As for the previous experiment, this analysis confirms that the CoDIME framework is on par with the state-of-the-art regarding simulations based on perfect users. Nevertheless, if we consider more realistic and noisy situations, the CoDIME approach shows superior capabilities, remaining stable when highly noisy data is considered.

4.6 Effect of the Position Bias

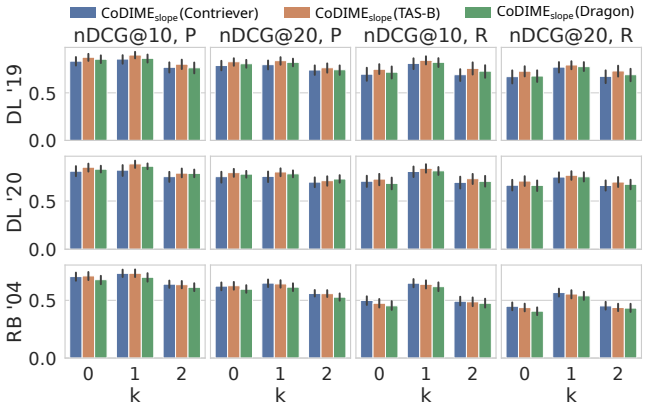


Figure 3: Consequences of overestimating (propensity = 0) or underestimating (propensity = 2) the propensity for TAS-B. P and R are the perfect and near-random user models.

As a final analysis, we illustrate the consequences of overestimating or underestimating the propensity. In detail, we maintain the same debiasing assuming $\eta = 1$ to compute the CoDIME$_{slope}$ query representations, but we change the underlying simulation process, using $\eta = \{0,1,2\}$. This allows us to estimate the approach's effectiveness in an even more noisy situation. Figure 3 reports our results for the three considered encoders. We notice that if we assume a perfect user that perfectly clicks only on relevant documents ('P' columns), regardless of the considered encoder and propensity exponent, the approaches tend to be overall stable, with a slight decrease in effectiveness if we consider an under-estimation error (propensity = 2). The phenomenon is more evident in the case of a near-random user model ('R' columns). In this case, the impact of failing in correctly modeling the η tends to be more severe, especially for the Robust '04 collection. This might be explained by the noise introduced in this scenario, which makes the implicit feedback signal unreliable.

5 Conclusion

In this work, we introduced CoDIME, a novel counterfactual framework for dimension importance estimation in dense text representations, leveraging implicit user feedback to address challenges in existing DIME approaches. By incorporating counterfactual modelling of click probabilities in various dimension importance estimation strategies, our CoDIME approaches achieved state-of-the-art performance in multiple dense IR testbeds. Compared to CoRocchio [44], a state-of-the-art counterfactual approach, the CoDIME framework achieves up to +0.235 nDCG@10 points, moving from 0.404 to 0.639 (+58%) (Dragon and Robust '04) and +0.117 nDCG@100 points, moving from 0.356 to 0.473 (+33%) (Dragon and Robust '04). These findings highlight the efficacy of counterfactual techniques and DIME approaches in adapting dense representations and improving retrieval effectiveness.

Acknowledgments

This work is supported, in part, by the Spoke "FutureHPC & BigData" of the ICSC – Centro Nazionale di Ricerca in High-Performance Computing, Big Data and Quantum Computing, the Spoke "Human-centered AI" of the M4C2 - Investimento 1.3, Partenariato Esteso PE00000013 - "FAIR - Future Artificial Intelligence Research", the "Extreme Food Risk Analytics" (EFRA) project, Grant no. 101093026, funded by European Union – NextGenerationEU, the FoReLab project (Departments of Excellence), the NEREO PRIN project funded by the Italian Ministry of Education and Research Grant no. 2022AEFHAZ and the CAMEO PRIN 2022 Project Grant no. 2022ZLL7MW.

References

[1] Yoshua Bengio, Aaron C. Courville, and Pascal Vincent. 2013. Representation Learning: A Review and New Perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.* 35, 8 (2013), 1798–1828. https://doi.org/10.1109/TPAMI.2013.50

[2] Olivier Chapelle and Ya Zhang. 2009. A dynamic bayesian network click model for web search ranking. In *Proceedings of the 18th International Conference on World Wide Web, WWW 2009, Madrid, Spain, April 20-24, 2009*, Juan Quemada, Gonzalo León, Yoëlle S. Maarek, and Wolfgang Nejdl (Eds.). ACM, 1–10. https://doi.org/10.1145/1526709.1526711

[3] Danqi Chen, Weizhu Chen, Haixun Wang, Zheng Chen, and Qiang Yang. 2012. Beyond ten blue links: enabling user click modeling in federated web search. In *Proceedings of the Fifth International Conference on Web Search and Web Data Mining, WSDM 2012, Seattle, WA, USA, February 8-12, 2012*, Eytan Adar, Jaime Teevan, Eugene Agichtein, and Yoelle Maarek (Eds.). ACM, 463–472. https://doi.org/10.1145/2124295.2124351

[4] Aleksandr Chuklin, Pavel Serdyukov, and Maarten de Rijke. 2013. Using Intent Information to Model User Behavior in Diversified Search. In *Advances in Information Retrieval - 35th European Conference on IR Research, ECIR 2013, Moscow, Russia, March 24-27, 2013. Proceedings (Lecture Notes in Computer Science, Vol. 7814)*, Pavel Serdyukov, Pavel Braslavski, Sergei O. Kuznetsov, Jaap Kamps, Stefan M. Rüger, Eugene Agichtein, Ilya Segalovich, and Emine Yilmaz (Eds.). Springer, 1–13. https://doi.org/10.1007/978-3-642-36973-5_1

[5] Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net. https://openreview.net/forum?id=r1xMH1BtvB

[6] Nick Craswell, Daniel Campos, Bhaskar Mitra, Emine Yilmaz, and Bodo Billerbeck. 2020. ORCAS: 18 Million Clicked Query-Document Pairs for Analyzing Search. *CoRR* abs/2006.05324 (2020). arXiv:2006.05324 https://arxiv.org/abs/2006.05324

[7] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, and Daniel Campos. 2020. Overview of the TREC 2020 Deep Learning Track. In *Proceedings of the Twenty-Ninth Text REtrieval Conference, TREC 2020, Virtual Event [Gaithersburg, Maryland, USA], November 16-20, 2020 (NIST Special Publication, Vol. 1266)*, Ellen M. Voorhees and Angela Ellis (Eds.). National Institute of Standards and Technology (NIST). https://trec.nist.gov/pubs/trec29/papers/OVERVIEW.DL.pdf

[8] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Ellen M. Voorhees. 2020. Overview of the TREC 2019 deep learning track. *CoRR* abs/2003.07820 (2020). arXiv:2003.07820 https://arxiv.org/abs/2003.07820

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, Jill Burstein, Christy Doran, and Thamar Solorio (Eds.). Association for Computational Linguistics, 4171–4186. https://doi.org/10.18653/V1/N19-1423

[10] Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. 2024. The Faiss library. *CoRR* abs/2401.08281 (2024). https://doi.org/10.48550/ARXIV.2401.08281 arXiv:2401.08281

[11] Abhimanyu Dubey, Abhinav Jauhri, and Abhinav Pandey et al. 2024. The Llama 3 Herd of Models. *CoRR* abs/2407.21783 (2024). https://doi.org/10.48550/ARXIV.2407.21783 arXiv:2407.21783

[12] Georges Dupret and Benjamin Piwowarski. 2008. A user browsing model to predict search engine click data from past observations. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2008, Singapore, July 20-24, 2008*, Sung-Hyon Myaeng, Douglas W. Oard, Fabrizio Sebastiani, Tat-Seng Chua, and Mun-Kew Leong (Eds.). ACM, 331–338. https://doi.org/10.1145/1390334.1390392

[13] Guglielmo Faggioli, Nicola Ferro, Raffaele Perego, and Nicola Tonellotto. 2024. Dimension Importance Estimation for Dense Information Retrieval. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2024, Washington DC, USA, July 14-18, 2024*, Grace Hui Yang, Hongning Wang, Sam Han, Claudia Hauff, Guido Zuccon, and Yi Zhang (Eds.). ACM, 1318–1328. https://doi.org/10.1145/3626772.3657691

[14] Fan Guo, Chao Liu, and Yi Min Wang. 2009. Efficient multiple-click models in web search. In *Proceedings of the Second International Conference on Web Search and Web Data Mining, WSDM 2009, Barcelona, Spain, February 9-11, 2009*, Ricardo Baeza-Yates, Paolo Boldi, Berthier A. Ribeiro-Neto, and Berkant Barla Cambazoglu (Eds.). ACM, 124–131. https://doi.org/10.1145/1498759.1498818

[15] Sebastian Hofstätter, Sheng-Chieh Lin, Jheng-Hong Yang, Jimmy Lin, and Allan Hanbury. 2021. Efficiently Teaching an Effective Dense Retriever with Balanced Topic Aware Sampling. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*, Fernando Diaz, Chirag Shah, Torsten Suel, Pablo Castells, Rosie Jones, and Tetsuya Sakai (Eds.). ACM, 113–122. https://doi.org/10.1145/3404835.3462891

[16] D. G. Horvitz and D. J. Thompson. 1952. A Generalization of Sampling Without Replacement From a Finite Universe. *J. Amer. Statist. Assoc.* 47, 260 (1952), 663–685. http://www.jstor.org/stable/2280784

[17] Guido W. Imbens and Donald B. Rubin. 2015. *Causal Inference for Statistics, Social, and Biomedical Sciences: An Introduction.* Cambridge University Press.

[18] Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Towards Unsupervised Dense Information Retrieval with Contrastive Learning. *CoRR* abs/2112.09118 (2021). arXiv:2112.09118 https://arxiv.org/abs/2112.09118

[19] Rolf Jagerman, Harrie Oosterhuis, and Maarten de Rijke. 2019. To Model or to Intervene: A Comparison of Counterfactual and Online Learning to Rank from User Interactions. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, July 21-25, 2019*, Benjamin Piwowarski, Max Chevalier, Éric Gaussier, Yoelle Maarek, Jian-Yun Nie, and Falk Scholer (Eds.). ACM, 15–24. https://doi.org/10.1145/3331184.3331269

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

[21] Thorsten Joachims, Adith Swaminathan, and Tobias Schnabel. 2017. Unbiased Learning-to-Rank with Biased Feedback. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining, WSDM 2017, Cambridge, United Kingdom, February 6-10, 2017*, Maarten de Rijke, Milad Shokouhi, Andrew Tomkins, and Min Zhang (Eds.). ACM, 781–789. https://doi.org/10.1145/3018661.3018699

[22] Karen Spärck Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *J. Documentation* 28, 1 (1972), 11–21. https://doi.org/10.1108/00220410410560573

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

[24] Victor Lavrenko and W. Bruce Croft. 2001. Relevance-Based Language Models. In *SIGIR 2001: Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, September 9-13, 2001, New Orleans, Louisiana, USA*, W. Bruce Croft, David J. Harper, Donald H. Kraft, and Justin Zobel (Eds.). ACM, 120–127. https://doi.org/10.1145/383952.383972

[25] Hang Li, Ahmed Mourad, Shengyao Zhuang, Bevan Koopman, and Guido Zuccon. 2023. Pseudo Relevance Feedback with Deep Language Models and Dense Retrievers: Successes and Pitfalls. *ACM Trans. Inf. Syst.* 41, 3 (2023), 62:1–62:40. https://doi.org/10.1145/3570724

[26] Sheng-Chieh Lin, Akari Asai, Minghan Li, Barlas Oguz, Jimmy Lin, Yashar Mehdad, Wen-tau Yih, and Xilun Chen. 2023. How to Train Your Dragon: Diverse Augmentation Towards Generalizable Dense Retrieval. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, 6385–6400. https://doi.org/10.18653/V1/2023.FINDINGS-EMNLP.423

[27] Iain Mackie, Shubham Chatterjee, and Jeffrey Dalton. 2023. Generative Relevance Feedback with Large Language Models. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2023, Taipei, Taiwan, July 23-27, 2023*, Hsin-Hsi Chen, Wei-Jou (Edward) Duh, Hen-Hsen Huang, Makoto P. Kato, Josiane Mothe, and Barbara Poblete (Eds.). ACM, 2026–2031. https://doi.org/10.1145/3539618.3591992

[28] Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. In *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.). http://arxiv.org/abs/1301.3781

[29] Harrie Oosterhuis and Maarten de Rijke. 2018. Differentiable Unbiased Online Learning to Rank. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, Torino, Italy, October 22-26, 2018*, Alfredo Cuzzocrea, James Allan, Norman W. Paton, Divesh Srivastava, Rakesh Agrawal, Andrei Z. Broder, Mohammed J. Zaki, K. Selçuk Candan, Alexandros Labrinidis, Assaf Schuster, and Haixun Wang (Eds.). ACM, 1293–1302. https://doi.org/10.1145/3269206.3271686

[30] Harrie Oosterhuis and Maarten de Rijke. 2021. Unifying Online and Counterfactual Learning to Rank: A Novel Counterfactual Estimator that Effectively Utilizes Online Interventions. In *WSDM '21, The Fourteenth ACM International Conference on Web Search and Data Mining, Virtual Event, Israel, March 8-12, 2021*, Liane Lewin-Eytan, David Carmel, Elad Yom-Tov, Eugene Agichtein, and Evgeniy Gabrilovich (Eds.). ACM, 463–471. https://doi.org/10.1145/3437963.3441794

[31] Zohreh Ovaisi, Ragib Ahsan, Yifan Zhang, Kathryn Vasilaky, and Elena Zheleva. 2020. Correcting for Selection Bias in Learning-to-rank Systems. In *WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020*, Yennun Huang, Irwin King, Tie-Yan Liu, and Maarten van Steen (Eds.). ACM / IW3C2, 1863–1873. https://doi.org/10.1145/3366423.3380255

[32] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, Alessandro Moschitti, Bo Pang, and Walter Daelemans (Eds.). ACL, 1532–1543. https://doi.org/10.3115/V1/D14-1162

[33] Matthew Richardson, Ewa Dominowska, and Robert Ragno. 2007. Predicting clicks: estimating the click-through rate for new ads. In *Proceedings of the 16th International Conference on World Wide Web, WWW 2007, Banff, Alberta, Canada, May 8-12, 2007*, Carey L. Williamson, Mary Ellen Zurko, Peter F. Patel-Schneider, and Prashant J. Shenoy (Eds.). ACM, 521–530. https://doi.org/10.1145/1242572.1242643

[34] Stephen E. Robertson, Steve Walker, Susan Jones, Micheline Hancock-Beaulieu, and Mike Gatford. 1994. Okapi at TREC-3. In *Proceedings of The Third Text REtrieval Conference, TREC 1994, Gaithersburg, Maryland, USA, November 2-4, 1994 (NIST Special Publication, Vol. 500-225)*, Donna K. Harman (Ed.). National Institute of Standards and Technology (NIST), 109–126. http://trec.nist.gov/pubs/trec3/papers/city.ps.gz

[35] J. J. Rocchio. 1971. Relevance feedback in information retrieval. In *The Smart retrieval system - experiments in automatic document processing*, G. Salton (Ed.). Englewood Cliffs, NJ: Prentice-Hall, 313–323.

[36] Gerard Salton, Edward A. Fox, and Harry Wu. 1983. Extended Boolean Information Retrieval. *Commun. ACM* 26, 11 (1983), 1022–1036. https://doi.org/10.1145/182.358466

[37] Henry Scheffe. 1959. *The analysis of variance.* John Wiley & Sons.

[38] John W. Tukey. 1949. Comparing Individual Means in the Analysis of Variance. *Biometrics* 5, 2 (1949), 99–114.

[39] C. J. van Rijsbergen. 1979. *Information Retrieval.* Butterworth.

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

[41] Ellen M. Voorhees. 2004. Overview of the TREC 2004 Robust Track. In *Proceedings of the Thirteenth Text REtrieval Conference, TREC 2004, Gaithersburg, Maryland, USA, November 16-19, 2004 (NIST Special Publication, Vol. 500-261)*, Ellen M. Voorhees and Lori P. Buckland (Eds.). National Institute of Standards and Technology (NIST). http://trec.nist.gov/pubs/trec13/papers/ROBUST.OVERVIEW.pdf

[42] 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 conference on Research and Development in Information Retrieval, SIGIR 2016, Pisa, Italy, July 17-21, 2016*, Raffaele Perego, Fabrizio Sebastiani, Javed A. Aslam, Ian Ruthven, and Justin Zobel (Eds.). ACM, 115–124. https://doi.org/10.1145/2911451.2911537

[43] Wayne Xin Zhao, Jing Liu, Ruiyang Ren, and Ji-Rong Wen. 2022. Dense Text Retrieval based on Pretrained Language Models: A Survey. *CoRR* abs/2211.14876 (2022). https://doi.org/10.48550/ARXIV.2211.14876 arXiv:2211.14876

[44] Shengyao Zhuang, Hang Li, and Guido Zuccon. 2022. Implicit Feedback for Dense Passage Retrieval: A Counterfactual Approach. In *SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, July 11 - 15, 2022*, Enrique Amigó, Pablo Castells, Julio Gonzalo, Ben Carterette, J. Shane Culpepper, and Gabriella Kazai (Eds.). ACM, 18–28. https://doi.org/10.1145/3477495.3531994

[45] Shengyao Zhuang, Zhihao Qiao, and Guido Zuccon. 2022. Reinforcement online learning to rank with unbiased reward shaping. *Inf. Retr. J.* 25, 4 (2022), 386–413. https://doi.org/10.1007/S10791-022-09413-Y

[46] Shengyao Zhuang and Guido Zuccon. 2020. Counterfactual Online Learning to Rank. In *Advances in Information Retrieval - 42nd European Conference on IR Research, ECIR 2020, Lisbon, Portugal, April 14-17, 2020, Proceedings, Part I (Lecture Notes in Computer Science, Vol. 12035)*, Joemon M. Jose, Emine Yilmaz, João Magalhães, Pablo Castells, Nicola Ferro, Mário J. Silva, and Flávio Martins (Eds.). Springer, 415–430. https://doi.org/10.1007/978-3-030-45439-5_28