A Counterfactual Explanation Framework for Retrieval Models

Anonymous ACL submission

Abstract

Explainability has become a crucial concern in today's world, aiming to enhance transparency in machine learning and deep learning models. Information retrieval is no exception to this trend. In existing literature on explainability of information retrieval, the emphasis has predominantly been on illustrating the concept of relevance concerning a retrieval model. The questions addressed include why a document is relevant to a query, why one document exhibits higher relevance than another, or why a specific set of documents is deemed relevant for a query.

However, limited attention has been given to understanding why a particular document is considered non-relevant to a query with respect to a retrieval model. In an effort to address this gap, our work focus on the question of what terms need to be added within a document to improve its ranking. This in turn answers the question of which words played a role in not being favored by a retrieval model for a particular query. We use an optimization framework to solve the above-mentioned research problem. To the best of our knowledge, we mark the first attempt to tackle this specific counterfactual problem (i.e. examining the absence of which words can affect the ranking of a document). Our experiments show the effectiveness of our proposed approach in predicting counterfactuals for both statistical (e.g. BM25) and deep-learning-based models (e.g. DRMM, DSSM, ColBERT). The code implementation of our proposed approach is available in https://anonymous.4open.science/r/CfIR/.

1 Introduction

The requirement of transparency of AI models has made explainability crucial, and this applies to Information Retrieval (IR) models as well (Anand et al., 2022). The target audience plays a significant role in achieving explainability for an information retrieval model, as the units of explanation or questions may differ based on the end user. For instance, a healthcare specialist, who is a domain expert but not necessarily an information retrieval specialist, might want to understand the reasons behind a ranked suggestion produced by a retrieval model in terms of words (Singh and Anand, 2019). On the other hand, an IR practitioner may be more interested in understanding whether different IR axioms are followed by a retrieval model or not (Bondarenko et al., 2022).

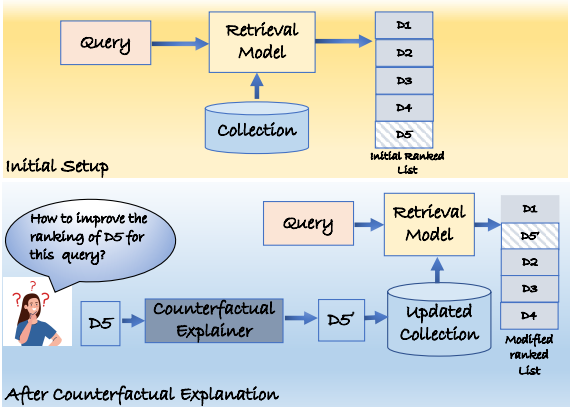


Figure 1: Counterfactual Setup Used in this Work.

This study focus on the perspective of Information Retrieval (IR) practitioners. To be more specific, we introduce a counterfactual framework designed for information retrieval models, catering to the needs of IR practitioners. Existing literature in explainable IR (ExIR) addressed questions like why a particular document is relevant with respect to a query (Singh and Anand, 2019), between a pair of documents why one document is more relevant to the query (Penha et al., 2022) compared to the other and why a list of documents relevant to a query (Lyu and Anand, 2023). Broadly speaking, all the above mentioned questions mainly focus on explaining the relevance of a document or a list of documents from different perspectives.

However, there is limited attention to explain the concept of non-relevance. The question like the absense of which words renders a document unfavorable to a retrieval model (i.e. not within top-K) remains unexplored. The above mentioned explanation can give an idea to an IR practitioner about how to modify a retrieval model. For example, if it is observed that a retrieval model (e.g. specially neural network based retrieval models) does not favour documents because of not having words which are not so related to query topic then the setting of the retrieval model needs to be changed so that it gives more importance to the semantic similarity feature.

With the motivation described above, the fundamental research question which we address in this research work is described as follows.

- **RQ1:** What are the terms that should be added to a document which can push the document to a higher rank with respect to a particular retrieval model?

Figure 1 shows a schematic diagram of **RQ1** discussed in this work. As shown in Figure 1, we would like to note that we framed **RQ1** as a counterfactual setup in our research scope. Similar to existing research in counterfactual explanations in AI (Kanamori et al., 2021; Van Looveren and Klaise, 2021), we also attempt to change the output of model with the provided explanations (i.e. change the rank of a document in IR models). In the counterfactual setup, we primarily used a constrained optimization technique to address **RQ1**. Our experimental results show that on an average in 60% cases the solution provided by the counterfactual setup improves the ranking of a document with respect to a query and a ranking model.

Our Contributions The main contributions of this paper are as follows.

- Propose a model agnostic novel counterfactual framework for retrieval models.
- Estimating the terms that can push the rank of a document. Consequently this work is also the first attempt to explain non-relevance (important for learning relevance in neural retrieval models) in information retrieval framework.
- Provide a comprehensive analysis with existing explanation frameworks in IR.

The rest of the paper is organized as follows. Section 2 describes Related work. Section 3 describes the counterfactual framework used in our work. Section 4 describes the experimental setup. Section 5 discuss about results and ablation study. Section 6 is about conclusion.

2 Related Work

Existing research related to this work can be broadly categorized into three different areas: explainability in general AI, explainability in IR and search engine optimization. Each one of them are described as follows.

2.1 Explainability in AI

The origins of explainable AI (xAI) can be traced back to the early 1960s when pioneering researchers identified the need for AI systems (Association et al., 1966; McCarthy, 2022; Davis, 1989) that could reason and explain their actions, addressing the lack of transparency in decision-making processes. More recently, the development of the Local Interpretable Model-agnostic Explanations (LIME) (Ribeiro et al., 2016) method has provided a way to explain any classification model. One major advantage of LIME is that it requires only access to the model's input and output. Following LIME, a series of posthoc explainers (Lundberg and Lee, 2017; Ribeiro et al., 2018; Selvaraju et al., 2017; Petsiuk et al., 2018; Wang et al., 2020; Jiang et al., 2021; Englebert et al., 2024) were proposed.

Counterfactual Explanations While models like LIME explains why a model predicts a particular output, counterfactual explainers address the question of what changes in input features would be needed to alter the output. Counterfactual xAI was first brought into limelight in early 2010s with seminal works of Judea Pearl (2018). Karimi et al. (2020) provided a practical framework named Model-Agnostic Counterfactual Explanations (MACE) for generating counterfactual explanations for any model. Later series of models (Kanamori et al., 2021; Van Looveren and Klaise, 2021; Parmentier and Vidal, 2021; Carreira-Perpiñán and Hada, 2021; Pawelczyk et al., 2022; Hamman et al., 2023) based on optimization framework was proposed for counterfactual explanation.

2.2 Explainability in IR

Explainability in IR models can be broadly categorized into four areas: a) Pointwise Explanation b)Pairwise Explanation c) Listwise Explanation and d) Generative Explanation.

Pointwise Explanations Here the explainer shows the important features responsible for the relevance score predicted by a retrieval model for a query-document pair. Popular techniques include locally approximating the relevance scores predicted by the retrieval model using a regression model (Singh and Anand, 2019).

Pairwise Explanations Here explainers predict why a particular document was favoured by a ranking model compared to others. Generally explanations are expressed in terms of words. The work in (Xu et al., 2024) proposed a counterfactual explanation method to compare the ranking of a pair of documents with respect to a particular query. A major difference of our proposed work with the study in (Xu et al., 2024) is that we focus on providing counterfactual for a query and a document (i.e. pointwise explanation) instead of a pair of documents.

Listwise Explanations Here the focus is on explaining the key features for a ranked list of documents and a query. Listwise explanations (Yu et al., 2022; Lyu and Anand, 2023) aim to capture a more global perspective compared to pointwise and pairwise explanations. The study in (Lyu and Anand, 2023) proposed an approach which combines the output of different explainers to capture the different aspects of relevance. The study in (Yu et al., 2022) trained a transformer model to generate explanation terms for a query and a ranked list of documents.

Generative Explanation Unlike previously mentioned methods, which focus on analyzing existing features or model internals, generative explanations (Singh and Anand, 2020; Lyu and Anand, 2023) leverage natural language processing to create new text content, like summaries or justifications, that directly address the user's query and information needs. Model-agnostic approaches (Singh and Anand, 2020) were proposed to interpret the intent of the query as understood by a black box ranker.

Search Engine Optimization The study in (Egri and Bayrak, 2014; Erdmann et al., 2022) used different features like commercial cost, links to optimize the performance of the search engine. A major difference of the work in (Egri and Bayrak, 2014; Erdmann et al., 2022) with our work is we only consider the words present in a document as a feature. Our objective is to improve the ranking of a particular document concerning a specific query and a retrieval model rather than improving the ranking of a document concerning any query belonging to a particular topic.

3 Counterfactual Framework for Information Retrieval (CFIR)

In this section, we first outline the general counterfactual setup in explainable AI, followed by a detailed explanation of the counterfactual setup in IR. Within the current literature on xAI, considerable efforts have been dedicated to identifying counterfactuals (Mothilal et al., 2020) in regression and classification scenarios. In the existing counterfactual setup, the problem being addressed is identifying which features in the input instance need to be modified to change the output of a trained model. Generally framed as a constrained optimization problem, the task of discovering counterfactuals involves optimizing various constraints such as minimum edit distance between the generated counterfactual and the input, diversity, and immutability of certain features. In our research scope we particularly used the counterfactual methodology proposed in (Mothilal et al., 2020).

Counterfactual Setup (CF Setup) For the optimization setup described in (Mothilal et al., 2020), the input to the problem is a trained machine learning model f, and an instance, x for which we want to generate counterfactuals. The k number of counterfactuals generated from the optimization problem (denoted as $\{c_1, c_2 \ldots, c_k\}$) is supposed to alter the prediction for x in f. The main assumptions in the above mentioned setup is that the machine learning model should be differentiable and the output of the model should not change over time. The optimization setup takes into account three different criteria while generating counterfactuals. They are

- **Criteria 1:** Minimizing the distance between the desired outcome y' and the prediction of the model f for a counterfactual example $(f(c_i))$.
- **Criteria 2:** Minimizing the distance between the generated counterfactual (c_i) and the orig-

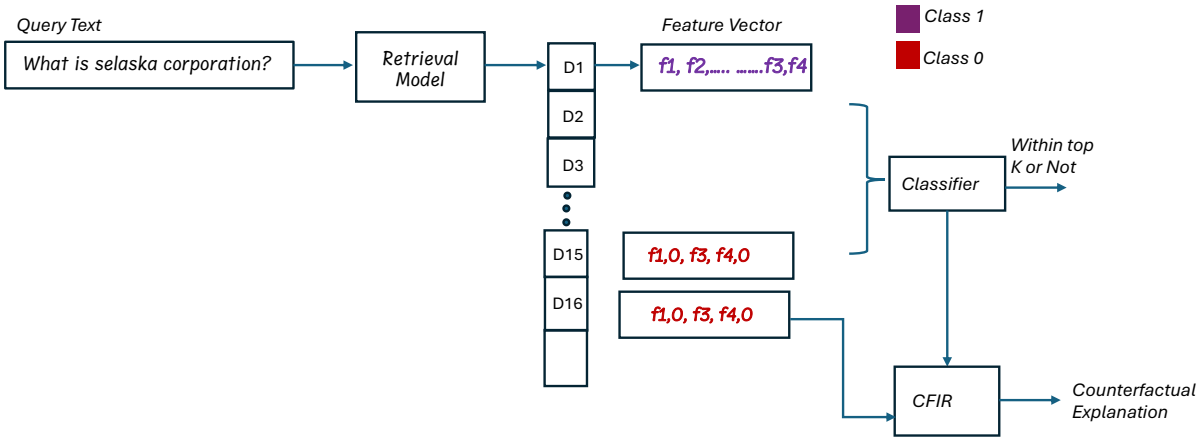


Figure 2: Counterfactual Explanation Model Description.

inal input x. Broadly speaking, a counterfactual example closer to the original input should be more useful for a user.

- **Criteria 3:** Balancing between diversity and feasibility among generated counterfactuals.

Based on the above mentioned criteria the objective function for the optimization function is described as follows.

$$C(x) = \arg\min_{c_1,\ldots c_k} \frac{1}{k}\sum_{i=1}^{k} \text{yloss}(f(c_i), y) + \frac{\lambda_1}{k}\sum_{i=1}^{k} dist(c_i, x) - \lambda_2 div(c_1, \ldots, c_k) \quad (1)$$

In Equation 1, yloss(.) takes care of the first criteria, $dist(c_i, x)$ takes care of the second criteria and *div* takes care of the third criteria as discussed above. λ_1 and λ_2 in Equation 1 are hyperparameters that balance the three parts of the overall objective function. The detailed description of the each part of the Equation is given in Appendix 10. The c_is in Equation 1 comes from the vocabulary set used to construct the feature vector for each document.

3.1 Mapping Retrieval to CF Setup

In IR, the end user is generally interested in the rank of documents within top-K compared to the corresponding relevance score. Hence to align with the optimization setup described for counterfactuals in Section 3, we aim to build on a classifier where we are interested in finding the counterfactuals that can push a document within top-K. The specifics of the classification setup is given below.

Classifier Setup Existing work in XAI developed a simple model which can approximate the decision boundary of the original complex model in a small region (Ribeiro et al., 2016; **?**) to explain the original model. In that direction, the objective of the classifier in our research scope is to locally approximate a retrieval model M, for a query q and a subset of documents retrieved for the query q. More specifically, we are interested in approximating the the behavior of the model M in determining whether a document is retrieved within the top-K results or not. In contrast to (Ribeiro et al., 2016) we build a classification model instead of regression model. We build a binary classifier capable of predicting whether a document d will be ranked within the top-K results or not for a specific query q and retrieval model M.

In the classifier setup, the top-K documents for a query q and model M represents class 1 and any other document not belonging to this class represents class 0. Theoretically, speaking if a corpus had N number of documents, then there will be $N - K$ documents which should have class label zero and $N - K$ is a very large number in general which can cause class imbalance issue. To avoid this issue, for the 0 class, for each document for which we want to generate a counterfactual, we choose a set of closest neighbors in the set of $N - K$ documents and the size of the neighborhood should be similar to K. Consequently, there is no class imbalance issue in our classifier setup. In the classifier setup, K serves as a predefined

threshold, typically set to values such as 10, 20, or 30.

Feature Vector for Classifier Generating the feature vector for the classifier using all the words from documents retrieved for a query can pose challenges. Consequently, we adopted a filtering strategy. Not all words in a document contribute equally to its retrieval. Therefore, we selectively choose the most significant n words from each document. We create a vocabulary set V by taking the union of the top n important words present in each one of the top-K documents of the ranked list. Mathematically,

$$V = \cup_{i=1}^{K} \{ \sum_{j=1, w_j \in d_i}^{n} w_j \} \quad (2)$$

We employ Tf-Idf mechanism to select the top n words from each document. The dimension of the feature vector required for the classifier setup is set to the size of the vocabulary set $|V|$. If a document d contains x number of words from the set $|V|$, then only x number of positions in the corresponding representation will have nonzero values, while the remaining positions will be assigned a value of 0. The non-zero positions within the vector encompass the term frequency value of the word at the i^{th} position within the document d. For example, if the set $|V|$ contains words $w_1, w_2 \ldots, w_5$ and a document D has only w_1, w_2 and w_3 from $|V|$ then the corresponding vector representation for d will be $\{tf_1, tf_2, tf_3, 0, 0\}$ where tf_1, tf_2 and tf_3 represents the term frequency of the words w_1, w_2, w_3 in D respectively.

Equation 3, shows the mathematical representation of the classifier. X in Equation 3 is of $|V|$ dimensional.

$$h : X \rightarrow \{0, 1\} \quad (3)$$

Once the classifier is trained then we use this classifier in the optimization setup described in Equation 1. The output of the optimizer is used to explain what changes needs to be done in the document d to improve its ranking. Informally speaking, the optimizer determines what minimal changes need to be applied to the feature vector to push the document to the higher rank. Figure 2 shows the schematic diagram for counterfactual setup with the workflow between the different components (i.e. classifier and optimizer) within it.

4 Experiment Setup

Here we first describe the dataset used in our experiments, followed by the retrieval models, parameter setup, and evaluation metrics.

Dataset We used two ranking datasets for our experiments: MS MARCO passage ranking dataset for short documents (Bajaj et al., 2016) and MS MARCO document ranking dataset for longer documents (Craswell et al., 2023). The MS MARCO passage ranking dataset contains queries from Bing[1] and a collection of 5 million passages for retrieval. For testing our counterfactual setup, we randomly selected 10 queries from the test set and 5 documents not in the top 10 results for each query, creating 50 query-document pairs similar to existing test sets in IR (Craswell et al., 2020).

For MSMARCO document also we followed a similar approach to construct the test setup. The Equation described in 1, produces x number of countefactuals as an output of the optimization setup. For each query-document pair, we have produced $x = 1$ set of counterfactuals to test. The details of the dataset is given in Table 1.

Classifier Setup			
MS MARCO Passage		**MSMARCO Document**	
# Instances	30	# Instances	30

CFIR Setup					
MS MARCO Passage			**MS MARCO Document**		
Query	Avg Length	5.9	Query	Avg Length	6.9
Document	Avg Length	64.9	Document	Avg Length	1134.2
Query	#Instances	10	Query	#Instances	10
Document	#Instances	50	Document	#Instances	50

Table 1: Dataset Details for Counterfactual Experiments

Retrieval Models The four different retrieval models used in our experiment are described as follows.

BM25: BM25[2] is a statistical retrieval model where the similarity between a query and a document is computed based on the term frequency of the query words present in the document, document frequency of the query words and also the document length.

DRMM: Deep Relevance Matching Model (DRMM) Guo et al. (2016) is a neural retrieval model where the semantic similarity between each pair of tokens corresponding to a query and a document is computed to estimate the final relevance score of a document corresponding to a query.

[1] https://bing.com

[2] https://en.wikipedia.org/wiki/Okapi_BM25

DSSM: Deep semantic similarity model (DSSM) Huang et al. (2013) is another neural retrieval model which uses word hashing techniques to compute the semantic similarity between a query and a document.

ColBERT: Contextualized Late Interaction over BERT (ColBERT) (Khattab and Zaharia, 2020), is an advanced neural retrieval model which exploits late interaction techniques based on BERT (Devlin et al., 2019) based representations of both query and document for retrieval.

Our proposed counterfactual approach is retrieval model agnostic. Hence, it can be applied on any retrieval model. We used BM25, DRMM, DSSM and CoLBERT only to show the contrast between a statistical and neural retrieval model in terms of counterfactual explanations.

Baselines To the best of our knowledge, this is the first work which attempts to provide counterfactual explanations in IR. Consequently, there exists no baseline for our proposed approach. However we have used a query word and top-K word based intuitive baseline to compare with our proposed approach. In query word baseline (QW), we have used query words not originally present in a document to enhance its ranking. For Top-K' ($Top-K'$) baseline we used the top k' words extracted from top 5 documents corresponding to a query as relevance set. Words appearing in the relevance set but not appearing in a document are added to the document to improve its ranking. For different retrieval models we have the corresponding versions of QW and $Top-K'$ baselines.

Evaluation Metrics There exists no standard evaluation framework for exIR approaches. The three different evaluation metrics in our experiment setup are described as follows.

Fidelity (FD): Existing xAI approaches in IR have used Fidelity (Anand et al., 2022) as one of the metrics to evaluate the effectiveness of the proposed explainability approach. Intuitively speaking, Fidelity measures the correctness of the features obtained from a xAI approach. In the context of the CFIR setup described in this work, we define this fidelity score as the number of times the words predicted by the counterfactual algorithm could actually improve the rank of a document. Let n be number of total number of query document pairs in our test case and x be number of query document pairs for which the the rank of the document improved after adding the counterfactuals obtained from the optimization setup described in Equation 1. Then mathematically Fidelity score with respect to a test dataset D and retrieval model M is defined as follows.

$$FD(D, M) = \frac{x}{n} * 100 \quad (4)$$

Avg. New Words: Here we compute the average number of new words added by the counterfactual approach for a set of query document pairs. One of the criteria of the optimization setup described in Equation 1 is the diversity of the explanations generated by the algorithm. Consequently average number of new words will give an approximate idea about how much new content should be added to the documents to improve the ranking.

Avg. Query Overlap: It is intuitive to think that increasing the number of query words in a document is likely to increase the ranking of a document for a particular retrieval model. However, this is not always the case. To address this point we have reported on an average how many of the

Method		Evaluation Metric		
MS MARCO Passage				
Retrieval Model	**Classifier**	**FD(%)**	**Avg. New Words**	**Avg. Query Overlap**
QW_{BM25}	NA	46%	4.78	100%
$Top-K'_{BM25}$	NA	38%	14.28	100%
$CFIR_{BM25}$	RF	62%	11.14	64%
$CFIR_{BM25}$	LR	**67%**	**18.32**	53%
QW_{DRMM}	NA	46%	4.78	100%
$Top-K'_{DRMM}$	NA	40%	**16.21**	100%
$CFIR_{DRMM}$	RF	**68%**	10.21	45%
$CFIR_{DRMM}$	LR	64%	13.47	63%
QW_{DSMM}	NA	46%	4.78	100%
$Top-K'_{DSSM}$	NA	32%	14.53	100%
$CFIR_{DSSM}$	RF	54%	12.42	56%
$CFIR_{DSSM}$	LR	**58%**	**14.77**	55%
$QW_{ColBERT}$	NA	56%	4.78	100%
$Top-K'_{ColBERT}$	NA	48%	**15.63**	100%
$CFIR_{ColBERT}$	RF	62%	12.41	56%
$CFIR_{ColBERT}$	LR	**68%**	14.53	72%
MS MARCO Document				
QW_{BM25}	NA	30%	5.64	100%
$Top-K'_{BM25}$	NA	36%	8.42	100%
$CFIR_{BM25}$	RF	48%	**15.64**	54%
$CFIR_{BM25}$	LR	**54%**	13.45	56%
QW_{DRMM}	NA	44%	5.64	100%
$Top-K'_{DRMM}$	NA	28%	15.00	100%
$CFIR_{DRRM}$	RF	54%	9.42	44%
$CFIR_{DRRM}$	LR	**58%**	**16.53**	44%
QW_{DSMM}	NA	NA	5.64	100%
$Top-K'_{DSSM}$	NA	30%	13.32	100%
$CFIR_{DSSM}$	RF	44%	17.74	56%
$CFIR_{DSSM}$	LR	**50%**	**19.32**	62%
$QW_{ColBERT}$	NA	34%	5.64	100%
$Top-K'_{ColBERT}$	NA	36%	**13.42**	100%
$CFIR_{ColBERT}$	RF	**72%**	11.05	49%
$CFIR_{ColBERT}$	LR	66%	9.42	56%

Table 2: CFIR model Performance for BM25, DRMM, DSSM and ColBERT in MSMARCO Passage and Document Collection. The Best Performing Counterfactual Explanation Method for every retrieval model is boldfaced; the overall best performance explanation across all rows is underlined.

words suggested by the counterfactual algorithm comes from the query words.

Pararameters and Implementation Details The details of implementation about retrieval models are shown in Appendix 9.1. We employed two popular classical machine learning techniques Logistic Regression (LR) and Random Forest (RF) for the classifier described in Section 3.1. For logistic regression the learning rate was set to 0.001. For random forest the number of estimators were set to 100. We train a separate classifier for each query and retrieval model. In total for each retrieval model there are 10 classifiers. As described in Section 3.1, all the words present in a document is not used as input to the classifier. We use top 10 ($n' = 10$) words based on Tf-Idf weights from each document to create the vocabulary ($|V|$) for the classifier. While training the classifier, we put the label of any document appearing within top 10 as 1 ($K = 10$).

5 Results

Table 2 shows the performance of the counterfactual approach with respect to different retrieval models (i.e. BM25, DRMM, DSSM, ColBERT). We did experiment both on MS MARCO passage and document dataset to observe the effectiveness of our proposed explanation approach both on shorter and longer documents. Mainly four different observations can be made from Table 2. ***Firstly***, It can be clearly observed that the CFIR model for each retrieval model has performed better compared to its corresponding query word or top-K words baseline in terms of Fidelity score. The above mentioned observation is consistent for both passages and long documents. ***Secondly***, it can be observed from Table that mostly CFIR approach provided the highest number of new terms (terms not already present in the documents) as part of the explanation to improve ranking. Consequently, we can say the overall set of explanation terms are more diverse for CFIR approach compared to others. ***Thirdly***, it can be also observed from Table 2 that the Fidelity scores are generally better in the MS MARCO passages compared to documents. One likely explanation for this phenomena is that documents are longer in length compared to passages. Consequently, a counterfactual classifier comprising of the top-K features of documents may not always be sufficient to represent the document. ***Fourthly***, another interesting observation from table 2 is that the maximum query word overlap by our proposed approach is 63%. This implies that the counterfactual algorithm is suggesting new words that are not even present in a query.

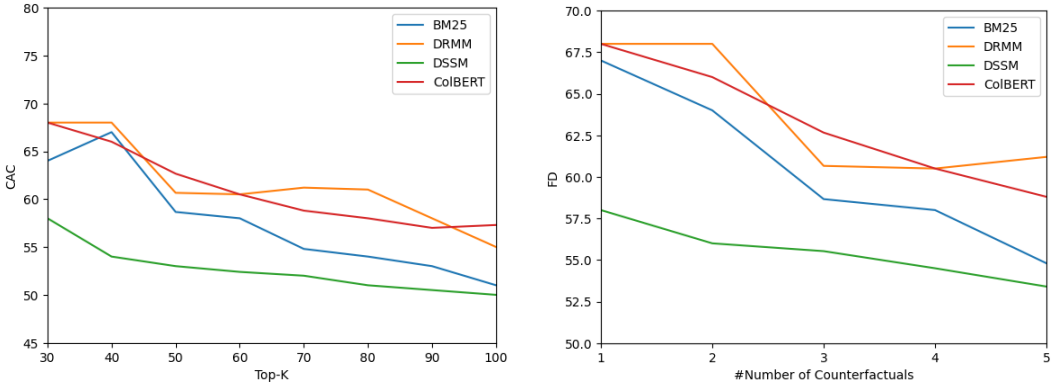
Table 3 shows some example terms extracted by our proposed approach. The words shown in Table 3 have improved the ranking of a docID with respect to the queries shown in .

5.1 Parameter Sensitivity Analysis

In Table 2, we observed that for most of the retrieval models the performance of the counterfactual explainer follows similar trend both in MS-MARCO passage and document dataset (i.e. the best performing model in terms of fidelity score is same in most of the cases). As a result of this we did parameter sensitivity experiments only on MS-MARCO passage dataset. Figure 3 (a) shows the performance of the counterfactual classifier build for the counterfactual setup with respect to the variance of the number of documents with label 0 used to train the classifier. The success of the counterfactual classifier (CAC) is measured by the percentage of cases where the classifier prediction label flipped from 0 to 1 by introducing the explanation terms in the feature vector. The reason for using CAC instead of FD in the Y axis for Figure 3 (a) is that the counterfactual classifier provides a local approximation of the retrieval model and it is computationally very expensive to measure FD (i.e. we need to update the collection index with a the modified document obtained from counterfactual explanation and then we need to execute retrieval) compared to CAC. It is clearly observed from Figure 3 (a) that with increase in the number of documents having label 0, the performance of the classifier decreases. Consequently for all our experiments in Table 2, we have used in total 30 documents to train the classifier. In Figure 3 (b) we show the variance of the counterfactual classifier success rate with respect to the number of counterfactual set provided by the optimization setup in (Mothilal et al., 2020). It is clearly visible from Figure 3 (b) that with increase in the value of number of counterfactuals there is a decrease in the performance of the counterfactual classifier. Intuitively, we can say that with increase in number of counterfactuals there is also increase in noise. This noise eventually affects rank improvement in the corresponding retrieval model. Figure

Retrieval Model	Query Text	Explanation Terms
DRMM	What law repealed prohibition ?	working, strict, Maine, 1929, law, resentment, New York City, Irish, immigrant, prohibition, repeal, fall, Portland, temperance, riot, visit
DSSM	What is the role of lipid in the cell?	phospholipid, fluidity, storage, triglyceride, fatty receptor
ColBERT	what type of wave is electromagnetic?	directly ,oscillations, medium, wave, properties, speed

Table 3: Sample Explanation Terms by CFIR Model for DRMM, DSSM and ColBERT in MS MARCO.



(a) (b)

Figure 3: Counterfactual Classifier Performance Variance with TopK and Counterfactual Performance Variance with variation of number of Counterfactuals.

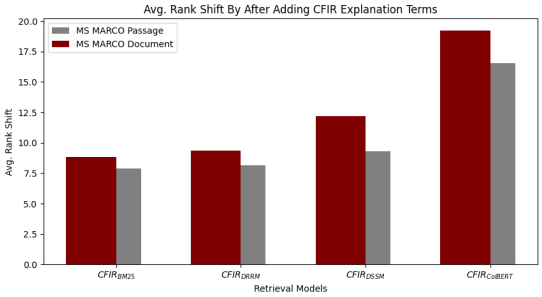


Figure 4: Average Rank shift by CFIR for BM25, DRRM, DSSM and ColBERT

4 shows the average rank change after introducing the explanation terms suggested by the CFIR setup. Figure 4 the actionability introduced by the counterfactual explanation terms. The two things to observe from Figure 4 are Firstly, the average rank shift is greater for documents than for passages. Table 2 shows that ColBERT achieved a significantly higher fidelity score (16^{th} and 31^{st} rows) and a larger average rank shift compared to the other models, is also seen in Figure 4.

6 Conclusion

In this paper we propose a counterfactual setup for a query document pair and a retrieval model. To the best of our knowledge there has not been any work on providing counterfactual explanation for a query-document pair. We did experiments on both MS MARCO passage and document ranking sets. Our experiments show that the proposed approach on an average 65% cases for both in short and long documents could successfully improve the ranking. In future we would like to explore different explanation units for the counterfactual setup.

7 Limitations

One of the limitations of this work is that we assume that top 10 or 20 words (based on tf-idf weights) within a document plays the most important part in improving the rank of a document. However, theoretically speaking we should consider all the words present in a document to determine the most influential words for a retrieval model. We have used top tf-idf words (Similar to statistical retrieval models) to reduce the computational complexity of our experiments and we have seen that increasing the number of top words doesn't affect the performance of the model that much.

8 Ethical Considerations

In this work we have used publicly available search query log and document collection to demonstrate counterfactual explanation. Any kind of sensitive data is not used in this experiment. As

a result of this there is no particular ethical concern associated with this work. If there is any kind of bias present in the search log data that effect can also be observed within our approach. However mitigating that bias was beyond the scope of this work

References

Avishek Anand, Lijun Lyu, Maximilian Idahl, Yumeng Wang, Jonas Wallat, and Zijian Zhang. 2022. Explainable information retrieval: A survey.

American Economic Association, Royal Economic Society, and Herbert A Simon. 1966. *Theories of decision-making in economics and behavioural science*. Springer.

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. 2016. Ms marco: A human generated machine reading comprehension dataset. In *InCoCo@NIPS*.

Alexander Bondarenko, Maik Fröbe, Jan Heinrich Reimer, Benno Stein, Michael Völske, and Matthias Hagen. 2022. Axiomatic retrieval experimentation with ir_axioms. In *Proc. of SIGIR 2022*, pages 3131–3140.

Miguel Á Carreira-Perpiñán and Suryabhan Singh Hada. 2021. Counterfactual explanations for oblique decision trees: Exact, efficient algorithms. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 6903–6911.

Nick Craswell, Bhaskar Mitra, Emine Yilmaz, and Daniel Campos. 2021. Overview of the TREC 2020 deep learning track. *CoRR*, abs/2102.07662.

Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, Jimmy Lin, Ellen M. Voorhees, and Ian Soboroff. 2023. Overview of the trec 2022 deep learning track. In *Text REtrieval Conference (TREC)*. NIST, TREC.

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.

Randall Davis. 1989. Expert systems: How far can they go? *AI Magazine*, 10(2):65–77.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*.

Gokhan Egri and Coskun Bayrak. 2014. The role of search engine optimization on keeping the user on the site. *Procedia Computer Science*, 36:335–342. Complex Adaptive Systems Philadelphia, PA November 3-5, 2014.

Alexandre Englebert, Olivier Cornu, and Christophe De Vleeschouwer. 2024. Poly-cam: High resolution class activation map for convolutional neural networks. *Machine Vision and Applications*, 35(4):89.

Anett Erdmann, Ramón Arilla, and José M. Ponzoa. 2022. Search engine optimization: The long-term strategy of keyword choice. *Journal of Business Research*, 144:650–662.

Jiafeng Guo, Yixing Fan, Qingyao Ai, and W. Bruce Croft. 2016. A deep relevance matching model for ad-hoc retrieval. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, CIKM '16, page 55–64, New York, NY, USA. Association for Computing Machinery.

Jiafeng Guo, Yixing Fan, Xiang Ji, and Xueqi Cheng. 2019. Matchzoo: A learning, practicing, and developing system for neural text matching. In *Proceedings of the 42Nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'19, pages 1297–1300.

Faisal Hamman, Erfaun Noorani, Saumitra Mishra, Daniele Magazzeni, and Sanghamitra Dutta. 2023. Robust counterfactual explanations for neural networks with probabilistic guarantees. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org.

Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM International Conference on Information & Knowledge Management*, CIKM '13, page 2333–2338.

Peng-Tao Jiang, Chang-Bin Zhang, Qibin Hou, Ming-Ming Cheng, and Yunchao Wei. 2021. Layercam: Exploring hierarchical class activation maps for localization. *IEEE Transactions on Image Processing*, 30:5875–5888.

Kentaro Kanamori, Takuya Takagi, Ken Kobayashi, Yuichi Ike, Kento Uemura, and Hiroki Arimura. 2021. Ordered counterfactual explanation by mixed-integer linear optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 11564–11574.

Amir-Hossein Karimi, Gilles Barthe, Borja Balle, and Isabel Valera. 2020. Model-agnostic counterfactual explanations for consequential decisions. In *International Conference on Artificial Intelligence and Statistics*, pages 895–905. PMLR.

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, page 39–48.

Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira. 2021. Pyserini: A python toolkit for reproducible information retrieval research with sparse and dense representations. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 2356–2362.

Scott M. Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 4768–4777.

Lijun Lyu and Avishek Anand. 2023. Listwise explanations for ranking models using multiple explainers. In *Advances in Information Retrieval: 45th European Conference on Information Retrieval, ECIR 2023, Dublin, Ireland, April 2–6, 2023, Proceedings, Part I*, page 653–668, Berlin, Heidelberg. Springer-Verlag.

John McCarthy. 2022. Artificial intelligence, logic, and formalising common sense. *Machine Learning and the City: Applications in Architecture and Urban Design*, pages 69–90.

Ramaravind K Mothilal, Amit Sharma, and Chenhao Tan. 2020. Explaining machine learning classifiers through diverse counterfactual explanations. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 607–617.

Axel Parmentier and Thibaut Vidal. 2021. Optimal counterfactual explanations in tree ensembles. In *International conference on machine learning*, pages 8422–8431. PMLR.

Martin Pawelczyk, Chirag Agarwal, Shalmali Joshi, Sohini Upadhyay, and Himabindu Lakkaraju. 2022. Exploring counterfactual explanations through the lens of adversarial examples: A theoretical and empirical analysis. In *International Conference on Artificial Intelligence and Statistics*, pages 4574–4594. PMLR.

Judea Pearl. 2018. Theoretical impediments to machine learning with seven sparks from the causal revolution. *arXiv preprint arXiv:1801.04016*.

Gustavo Penha, Eyal Krikon, and Vanessa Murdock. 2022. Pairwise review-based explanations for voice product search. In *ACM SIGIR Conference on Human Information Interaction and Retrieval*, pages 300–304.

Vitali Petsiuk, Abir Das, and Kate Saenko. 2018. Rise: Randomized input sampling for explanation of black-box models. *arXiv preprint arXiv:1806.07421*.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "why should i trust you?": Explaining the predictions of any classifier. In *Proc.of SIGKDD 2016*, page 1135–1144.

Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Anchors: High-precision model-agnostic explanations. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 1527–1535. AAAI Press.

Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *ICCV*.

Jaspreet Singh and Avishek Anand. 2019. Exs: Explainable search using local model agnostic interpretability. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, WSDM '19, page 770–773.

Jaspreet Singh and Avishek Anand. 2020. Model agnostic interpretability of rankers via intent modelling. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 618–628.

Arnaud Van Looveren and Janis Klaise. 2021. Interpretable counterfactual explanations guided by prototypes. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 650–665. Springer.

Haofan Wang, Zifan Wang, Mengnan Du, Fan Yang, Zijian Zhang, Sirui Ding, Piotr Mardziel, and Xia Hu. 2020. Score-cam: Score-weighted visual explanations for convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 24–25.

Chen Wu, Ruqing Zhang, Jiafeng Guo, Yixing Fan, and Xueqi Cheng. 2022. Are neural ranking models robust? *ACM Trans. Inf. Syst.*, 41(2).

Zhichao Xu, Hemank Lamba, Qingyao Ai, Joel Tetreault, and Alex Jaimes. 2024. Counterfactual editing for search result explanation. *Preprint*, arXiv:2301.10389.

Puxuan Yu, Razieh Rahimi, and James Allan. 2022. Towards explainable search results: a listwise explanation generator. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 669–680.

9 Appendix

9.1 Retrieval Performance of IR Models

We used Lin et al. (2021) toolkit for implementing BM25. For DRMM and DSSM, we used the implementation released by the study in Guo et al. (2019). For passage ranking we varied the parameters in a grid search and we took the configuration producing best MRR@10 value on TREC DL (Craswell et al., 2021) test set. For both DRMM and DSSM experiments on MSMARCO data, the parameters were set as suggested in (Wu et al., 2022). The MRR@10 values are reported in Table 6 in Appendix 9.1. For DRMM and DSSM, we use randomly chosen $100K$ query pairs from the MSMARCO training dataset to train the model.

Model	MRR@10
MSMARCO Passage	
BM25	0.1874
DRMM	0.1623
DSSM	0.1320
ColBERT	0.3481
MSMARCO Document	
BM25	0.2184
DRMM	0.1168
DSSM	0.1051
ColBERT	0.3469

Table 4: Retrieval Model Performance on MSMARCO passage and document

9.2 Example of Input and Output to Classifier

Given an input query, we employ a LuceneSearcher with MSMARCO Index to retrieve the top 30 documents. The feature vector construction process follows these steps:

For each document, we:

1. Extract the top 10 words based on their tf-idf values
2. Construct a vocabulary V as the union of all top 10 words across documents
3. Note that $|V|$ typically falls in the range of 150-180 words

The feature vector for each document has dimension $|V|$, where each component represents the tf-idf value of the corresponding word from the vocabulary. Formally:

$$\text{feature\_vector} \in R^{|V|}$$

Labels are assigned according to the following criterion:

$$\text{label} = \begin{cases} 1 & \text{for top } K \text{ documents} (K = 10 \text{ by default}) \\ 0 & \text{for remaining documents} \end{cases}$$

Example feature vectors and their corresponding counterfactuals generated using DiCE ML are shown in Figure 5.

Existing Explanation Methods	Word Overlap
PointWise Explanation (Singh and Anand, 2019)	21.46%
ListWise Explanation (Lyu and Anand, 2023)	9.57%

Table 5: Comparison of CFIR with Existing ExIR Approaches.

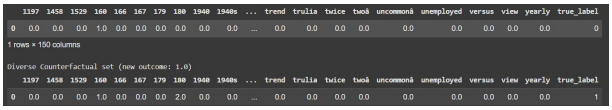


Figure 5: Example input feature vector and one counterfactual produced by DICEML for query 'average rent in california'. Here you can observe $|V| = 150$.

9.3 Existing EXIR approaches VS. CFIR

The existing literature aims to explain the significance of a document, a set of documents, or a pair of documents through various explanation methods. Nonetheless, our proposed approach diverges fundamentally from prior work in that we seek to demonstrate how the absence or frequency of certain tokens impacts document relevance. In this section, we examine whether there is any intersection between the two sets of tokens described earlier.

Pointwise Explanation Approach As outlined in Section 2.2, existing pointwise explanation methods elucidate why a specific document aligns with a given query within a retrieval model. Similarly, our proposed approach operates on individual documents and queries, albeit with a distinct objective. Here, we analyze the overlap between the explanations generated by the pointwise explanation method and those derived from our model, as presented in Table 6. This comparison was conducted across 50 pairs of documents.

Listwise Explanation Approach In Section 2, it is explained that listwise explanations typically

aim to demonstrate the relevance of a list of documents to a given query. In listwise setup, one set of xplanation terms are extracted for a list of documents, a query, and a retrieval model. Conversely, in our approach, we generate distinct explanations for each query-word pair. Therefore, to compare listwise explanations with our method, we aggregate all individual explanations obtained for each document-query pair in the list to create a unified explanation set for the entire list corresponding to a query. The resulting overlap is presented in Table 6.

Existing Explanation Methods	Word Overlap
PointWise Explanation (Singh and Anand, 2019)	21.46%
ListWise Explanation (Lyu and Anand, 2023)	9.57%

Table 6: Comparison of CFIR with Existing ExIR Approaches.

10 Counterfactual Optimization Framework

The different parts of Equation 1 is described here. The $yloss$ in Equation 1 is a hinge loss function as defined in Equation 5. In Equation 5 z is -1 when $y = 0$ otherwise, $z = 1$. $logit(f(c))$ is the logit values obtained form the ML model when the counterfactual c is given as input.

$$yloss = max(0, 1 - z * logit(f(c))) \quad (5)$$

The distance function ($dist(c_i, x)$) in Equation 1 is computed using the formula given in Equation 6. In Equation 6, d_{cat} represents the number of categorical variables used in the counterfactual input. In Equation 6, the value of I is equal to 1 if the corresponding value of the categorical variable is same in both the counterfactual input c and the original input x, otherwise it is set to 0.

$$dist(c, x) = \sum_{p=1}^{d_{cat}} I(c_p \neq x_p) \quad (6)$$

The diversity in Equation is defined by the formula described in Equation 7. In euqation 7, $K_{i,j}$ is equal to $\frac{1}{1+dist(c_i,c_j)}$. $dist(c_i, c_j)$ computes the distance between two counterfactuals c_i and c_j.

$$diversity = \sum_{i,j} det(K_{i,j}) \quad (7)$$