Listwise Explanations for Ranking Models Using Multiple Explainers

Lijun Lyu[1,2(✉)] and Avishek Anand[2]

[1] L3S Research Center, Leibniz University Hannover, Hannover, Germany
[2] Delft University of Technology, Delft, The Netherlands
{L.Lyu,Avishek.Anand}@tudelft.nl

Abstract. This paper proposes a novel approach towards better interpretability of a trained text-based ranking model in a post-hoc manner. A popular approach for post-hoc interpretability text ranking models are based on *locally approximating* the model behavior using a simple ranker. Since rankings have multiple relevance factors and are aggregations of predictions, existing approaches that use a single ranker might not be sufficient to approximate a complex model, resulting in low fidelity. In this paper, we overcome this problem by considering multiple simple rankers to better approximate the entire ranking list from a black-box ranking model. We pose the problem of local approximation as a GENERALIZED PREFERENCE COVERAGE (GPC) problem that incorporates multiple simple rankers towards the listwise explanation of ranking models. Our method MULTIPLEX uses a linear programming approach to judiciously extract the explanation terms, so that to explain the entire ranking list. We conduct extensive experiments on a variety of ranking models and report fidelity improvements of 37%–54% over existing competitors. We finally compare explanations in terms of multiple relevance factors and topic aspects to better understand the logic of ranking decisions, showcasing our explainers' practical utility.

Keywords: Explanation · Neural · Ranking · Post-hoc · List-wise

1 Introduction

Recent approaches for ranking text documents have focused heavily on neural models [12,16,17]. Neural rankers learn the complex and often non-linear relationships between the query and document that are difficult to encode using closed-form analytical ranking functions like BM25 [2]. However, the superior ranking performance of such models comes at the expense of reduced interpretability, thus increasing the risk of encoding spurious correlations and undesirable biases [25,32]. In parallel to developing better rankers, there has been an increased focus on interpreting neural ranking models [7,23–25] that specifically aim at explaining the rationale behind the ranking decisions.

This paper aims to propose post-hoc approaches to interpret neural text rankers. Post-hoc methods explain *already-trained* models and do not compromise on the accuracy of the learned model, hence making them popular choices for interpreting machine learning models. One prevalent strategy in post-hoc interpretability is to *locally approximate* a trained model with a *simple and interpretable proxy or a surrogate model*.

J. Kamps et al. (Eds.): ECIR 2023, LNCS 13980, pp. 653–668, 2023.
https://doi.org/10.1007/978-3-031-28244-7_41

Explainers	Explanation Terms
Term Matching	charlotte, north, sales, 2008
Position Aware	basketball, north, states, learn
Semantic Similarity	felidae, carnivorous, boko, extinction, deserts, iucn
MULTIPLEX	felidae, carnivorous, boko, extinction, deserts, gvwr, north

Fig. 1. Explaining the query `bobcat` with multiple relevance factors – (i) "charlotte-bobcat basketball club"; (ii) "learn to hunt bobcat"; (iii) "animal bobcat" and (iv) "bobcat mechanical retailer". MULTIPLEX carefully chooses from multiple relevance factors to explain a ranking. See Fig. 6 for more examples.

The degree of approximation is called *fidelity* and the objective is to maximize the fidelity between the proxy model and the underlying black-box model. Post-hoc methods for rankings entail using simple rankers to locally approximate (on a per-query basis) complex rankers such that the simple ranker has a high rank correlation (or high fidelity) with the complex ranking. Adapting this general post-hoc framework to ranking models has two specific challenges – *how do we aggregate multiple decisions inherent in a single ranking?* And *how do we explain ranking decisions with different inherent relevance factors?*

Rankings as Aggregations of Decisions. Text ranking models output a ranked list of documents for a given query. Unlike other learning tasks (e.g. regression and classification) that deal with a single decision, the ranking task can be viewed as an *aggregation of multiple pointwise or pairwise decisions* [1]. Any interpretability approach or explainer should therefore explain the reasoning behind the ranking list, or multiple-preference pair predictions. Therefore existing explanation techniques such as feature-attribution methods [21,22,28] that explain a single decision (pointwise) cannot be seamlessly used for rankings. Instead, a *listwise explanation* method that intends to cover all individual decisions in the entire ranking list is needed for rankings.

Different Explanations for Different Relevance Factors. Secondly, it is well-known that when ranking text, multiple relevance factors (also called ranking heuristics or axioms) determine the relevance of a document to a query, e.g., *lexical matching*, *semantic similarity*, *term proximity* etc. Unlike traditional models that explicitly encode each of these relevance factors, neural rankers automatically learn them from data. The next challenge in explaining rankings is ascertaining the relevance factor that best explains a given decision. Informally, there might not exist a single relevance factor that explains or satisfies all preferences $d_i \succ d_j$ in a given ranking. Therefore trying to approximate a ranking with a single relevance factor might result in low fidelity. A notable example is the listwise explanation approach [25] that considers covering multiple ranking decisions, but uses a single explainer which captures only one relevance factor (i.e., term matching), resulting in low-fidelity explanations due to the mismatch of exact terms.

In this paper, we define an explanation to be a combination of the underlying *relevance factors* along with the actual *machine intent*. In this paper, we **firstly** consider multiple simple rankers or explainers(formally defined in Sect. 3.1), which rely on different *well-known* and *human-understandable* (to system designers, or IR practitioners) relevance heuristics. **Secondly**, we explain the *machine intent* in terms of *expansion*

terms (in addition to the query terms) such that the simple ranker explains a complex black-box model by inducing a similar ranking list. Thus a combination of *simple rankers* that represents a relevance factor, along with its *expanded query terms* (also called explanation terms) is the listwise explanation of the reasoning behind the ranking.

Approach wise, we carefully select a small set of explanation terms sourced from the documents of the ranked list to maximize the explanation's approximation ability (i.e. fidelity). Specifically, we define the GENERALIZED PREFERENCE COVERAGE (GPC) framework, on which we optimize the preference coverage using approximated integer linear programming. Our method MULTIPLEX is shown to be able to improve the fidelity, and more interestingly combine terms from multiple explainers, implicitly covering multiple topics for an ambiguous query. Figure 1 shows an example of explanation terms extracted by each single explainer and MULTIPLEX can cover terms of multiple aspects. Note the aspects of terms are specified by manual observation.

We conduct extensive experiments using datasets from the TREC test collections – TREC-DL and Clueweb09 with three neural rankers to evaluate MULTIPLEX. We report fidelity improvements of 37%–54% over existing competitors. We also present anecdotal examples that showcase the practical utility of MULTIPLEX in understanding neural rankers. The datasets and source code are publicly available[1].

2 Related Work

Feature Attribution for Ranking Models. The earliest works of interpreting ranking models were simple extensions to existing pointwise explanation techniques – explain a single instance given a trained ML model for general machine learning tasks in vision and language. [24,29] adapted the popular surrogate-based LIME [20] to generate terms as the explanation for a trained black-box ranker. On the other hand, [7] applied a game-theory feature attribution method [15] to interpret the relevance score of a document given a query. Alternatively, other prevalent gradient-based feature attribution methods [21,22,28] can be adapted in the same way to attribute the relevance prediction to the textual input elements. All these methods provide pointwise explanations (why is doc_i relevant?) or pairwise explanations (why is doc_i ranked higher than doc_j?). We instead focus on listwise explanations or explaining the entire ranked list.

Listwise Explanations for Ranking Models. There is limited work on listwise explanations, i.e., explaining the entire ranking list. LiEGe [33] tackles the task as text generation. Specifically, LiEGe employs a Transformer-style model to generate terms for each document in a ranked list, and the explanation contains all generated terms. However, this method presupposes documents with labeled explanation terms, which is unrealistic in most application scenarios. Additionally, the explanation generator is not human-understandable, hindering understanding of the explanation generation process. In contrast, GreedyLM [25] uses a simple ranker to replicate the ranking list of a complex black-box model by expanding the query terms. The *simple ranker* and *expanded query*

[1] https://github.com/GarfieldLyu/RankingExplanation.

terms constitute the explanation for the complex model. We follow the same philosophy that the *explanation terms* along with the explanation generation process should be human interpretable. However, a limitation of [25] is that it assumes that a single relevance factor (modeled by a simple surrogate ranker) is adequate to explain an entire ranking. We challenge this assumption in this paper and use multiple simple explainers instead.

Axioms as Explanations. Another line of work uses IR axioms (or ranking heuristics) to ground the decisions of complex models. Axioms are well-understood, interpretable, and deterministic sets of rules that lay down the fundamental relevance factors of documents given a query. Recent works [4,19] diagnosed a group of ad-hoc neural rankers with a set of axioms and found out that neural models only to a limited extent adhere to the IR axioms. Similarly, [30] also found it hard to characterize BERT models in terms of IR axioms. The hypothesis is axiomatic approaches are limited to using just the query terms, resulting in low fidelity. In this work, we consider a much larger vocabulary of explanation terms to optimize the fidelity of our explanations.

In parallel, there are other works dealing with explaining learning-to-rank (LTR) [23,26], probing contextual ranking models [27,31], and intrinsic methods for extractive explanations [10,14,34]. We point the readers to a recent survey [3] in explainable information retrieval for a more detailed overview. In this work, we operate on text rankers and generate term-based explanations in a post-hoc manner.

3 Background and Preliminaries

We start with the notion of a ranker Φ that takes as input a keyword query $\mathcal{Q}$ to output an ordering π over a set of documents $\pi = (d_1 \succ d_2 \succ \ldots \succ d_n)$ based on the relevance of the documents to the query, i.e., $\Phi(\mathcal{Q}) \rightarrow \pi$. We aim to interpret Φ in a model-agnostic manner, using simple proxy rankers (called explainers Ψ). Note that the output of a ranker can be viewed as a set of preferences over the documents, or w.l.o.g $\pi = \{(d_i \succ d_j)\}$. Therefore explaining a ranking π is akin to explaining all or most of the preference pair decisions in π. An example of a single decision is whether the preference pair $(d_i \succ d_j)$ is `true`/`false`.

3.1 Explainers for Ranking

The explainer Ψ mimicking a black-box ranking model is essentially a simple ranker operating based on human-understandable closed form formulae (i.e. ranking heuristics). A popular example of such interpretable rankers is BM25 [2] model, which ranks documents for a given query by measuring the *term-matching* frequency of query terms in each document. Apart from term matching, there are also other factors or heuristics that might affect the relevance judgment such as the *term position*. Specifically, in news articles, the title and the introductory paragraphs are regarded to be more important. A ranking model should then weigh the term matching that occurred in the earlier paragraphs more than the rest. Additionally, *semantic similarity* is known to be crucial to address the exact mismatch problem. This is particularly true in neural models with embedding vectors as input. However, the semantic meaning of a term is less interpretable as it can vary if the context changes due to different training procedures or

datasets. In this regard, we draw the line of choosing the commonly-used context-free embeddings (i.e. GloVe [18]) as human-understandable input representation, instead of other contextualized embeddings (i.e., generated by BERT language model).

This set of simple ranking heuristics can be large given different granularities [4, 19]. In this work we start from **three** explainers to encode the above three ranking heuristics. Note that our framework allows a flexible amount of explainers, and thus more heuristics can be added if necessary. In summary, the explainers rank a document (d) based on its relevance to a query (q) by:

Term Matching or Ψ_{lm}: $\Psi_{lm}(q,d) = \frac{1}{|d|}\sum_{t\in q} \texttt{tf}(t,d)$, where $\texttt{tf}(t,d)$ denotes the term frequency of t in d.

Position Aware or Ψ_{pa}: a position-aware term-matching model [8], $\Psi_{pa}(q,d) = \sum_{t\in d}\frac{1}{|d|}\sum_{p\in d}\texttt{tf}(t,p)^{\frac{1}{p}}$, where p denotes the p_{th} paragraph in d.

Semantic Similarity or Ψ_{emb}: $\Psi_{emb}(q,d) = \frac{1}{|q|\times|d|}\sum_{t\in q,w\in d}\texttt{cosine}(t,w)$, where t and w are represented by the pre-trained GloVe embedding vectors [18].

3.2 Explanations to a Ranking Model

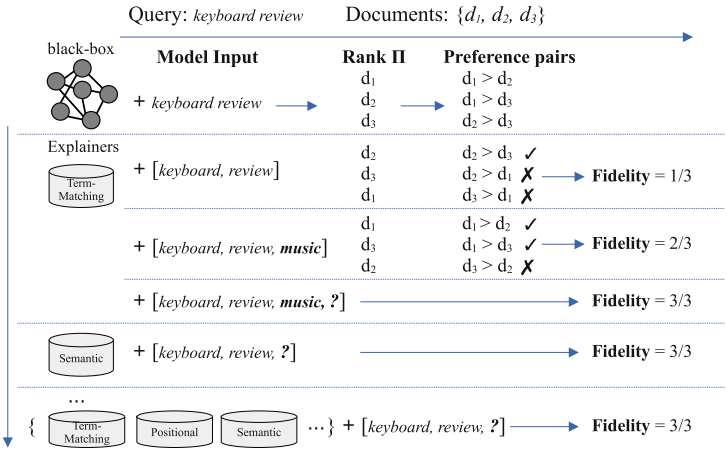


Fig. 2. Explaining black-box model with simple rankers and query terms.

The output of an interpretability procedure is an explanation, which should be *simple*, *human-understandable*, and *faithful* to the behavior of Φ. For the ranking task, the explanation can be decomposed into **two parts**: (1) a simple ranker whose decision-making process is fully transparent; (2) the machine intent of Φ in terms of an expanded query. The quality or fidelity(in XAI parlance) of the explanation can be evaluated by comparing the ranked lists induced by Φ and Ψ by standard rank-correlation metrics, e.g., Kendall's tau or just counting concordant preference pairs.

Take Fig. 2 as an example of interpreting the ranking induced by a black-box model. The simple Term Matching explainer with the input terms ("keyboard" and "review") can be regarded as an explanation, with a fidelity of $1/3$, as only one out of three preference pairs agrees with the original ranking. It is common that the query term is under-specified, and thus the simple ranker fails to extract the exact query intent. One solution is to use *query expansions* (e.g., RM3 [11]) to improve ranking performance. For instance, when adding "music" to the query, the explainer is aware of the musical preference of the black-box ranker and improves the explanation fidelity to $2/3$. The questions we ask are: (1) *which terms can be added to the query to maximize fidelity?*, and if more than one explainer is applied, (2) *how can we combine multiple simple explainers to cover as many pairs as possible?*

Fidelity Variants. Note that rankings can be misleading because they do not show the magnitude of the relevance difference. Sometimes the relevance scores of a preference pair can be very close, and explaining such pair is challenging even to humans. Therefore, to avoid uncertainty due to small score differences, we obtain a set of *important* preference pairs after excluding the similar pairs whose prediction difference is below some threshold. As Fig. 2 shows, suppose the black-box ranker predicts similar scores for d_2 and d_3, then $d_2 \succ d_3$ is not considered for evaluation. As a result, the Term Matching explainer, along with the input terms ("keyboard", "review" and "music"), can faithfully cover all pairs and get 100% fidelity. Given different choices of selecting to-be-explained preference pairs, we introduce different variants of fidelity, which will be further discussed in Sect. 5.3.

3.3 Problem Statement

We solve the explaining task as directly optimizing the fidelity, under the constraints of pre-defined explainers and the associated terms. Formally, given a query $\mathcal{Q}$, a complex ranking model Φ and a set of simple ranking models $\{\Psi\}$, we aim to select a small set of terms $\mathbb{E} \in \mathcal{V}$ (where $\mathcal{V}$ is the vocabulary), to explain most of the preference pairs $\{d_i \succ d_j\}$ from the original ranking π.

4 Generalized Preference Coverage

As mentioned earlier, choosing explanation terms to maximize fidelity can be formulated as a coverage problem of the preference pairs. We briefly describe the preference coverage (PC) framework as introduced in [25], using a single explainer as a precursor to introducing the generalized PC problem.

4.1 The Preference Coverage Framework

Similar to [25], the PC framework operates on a preference matrix constructed with a single Ψ. First, a set of n potentially important candidate terms $\mathcal{X}(\mathcal{X} \subseteq \mathcal{V}, |\mathcal{X}| = n)$ are extracted from the list of documents using simple statistics (e.g., *tf-idf*). Then, m preference pairs are sampled from π to create the preference matrix $\mathbf{M} \in \mathbb{R}^{n \times m}$. Each

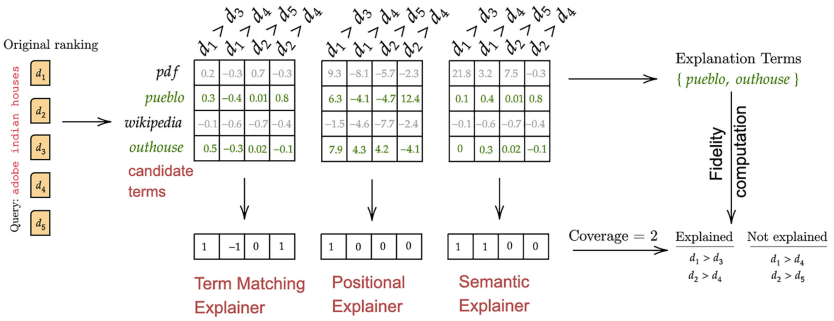


Fig. 3. Approach overview of MULTIPLEX using multiple explainers.

cell in $\mathbf{M}$ represents the utility or degree of Ψ in explaining the preference $d_{\pi(i)} \succ d_{\pi(j)}$ with t as input, by computing a preference score $f_{ij}^t = \Psi(t, d_{\pi(i)}) - \Psi(t, d_{\pi(j)})$. A positive f score means with t, the Ψ can explain or cover this pair, otherwise cannot. Each t can now be viewed as an m-dimensional vector $\mathbf{f}$, where each element represents how well it explains a specific pair. The PC framework using a single Ψ aims to choose a subset of rows $\mathbb{E} \subseteq \mathcal{X}$ (equivalent to selecting terms) from $\mathbf{M}$ so as to maximize the number of non-zero values in the aggregated vector. Since choosing or not choosing the row/term is a boolean decision, we can formulate the PC objective as an Integer Linear Program (ILP):

$$maximize \sum_{i=1}^{m} \big(\texttt{sign}(\mathbf{x}^\top \mathbf{M})\big)_i, \quad \text{s.t. } \mathbf{x} = [x_1, \cdots, x_n]\,; \;\; x_i \in \{0, 1\} \qquad \text{(PC)}$$

$\mathbf{x}$ is a selection vector with boolean values where $x_i = 1$ indicates selecting term $\mathcal{X}_i$, and $x_i = 0$ otherwise. The `sign` is an element-wise operation. Namely, $\mathbb{E} = \{i | x_i == 1\}$. This equation however is NP-hard and not solvable by the prevalent convex programming solvers (e.g., supported by CVXPY [6]) due to the non-differentiable `sign` function. Next, we present an improved formulation of the PC problem followed by a generalization to accommodate multiple explainers called the GENERALIZED PREFERENCE COVERAGE problem.

4.2 Optimizing PC for Multiple Explainers

Compared to PC, our proposal should be (i) practically solvable, (ii) ensuring sparse output $\mathbf{x}$ so that the explanation is human-understandable, and (iii) flexible to combine multiple explainers or $\mathbf{M}$.

Correspondingly, the first change we introduce is using `tanh` to approximate the non-convex `sign` operator. Secondly, we add a ℓ_1-regularization $\|\mathbf{x}\|$ to enforce sparsity constraints on the number of terms to be selected. A straightforward way to combine all explainers is to sum up their scores, i.e., $\Psi_{multi}(t, d) = \sum \Psi(t, d)$. However, different explainers can have different output ranges and exhibit high variance. For instance,

the term-matching score usually lies in $[0, 1]$, whereas the position-aware score typically operates in a much larger range. Normalization these scores in the optimization procedure is central to flexibly adding multiple explainers. We therefore formulate the GENERALIZED PREFERENCE COVERAGE problem that intends to optimize multiple matrices simultaneously as:

$$\begin{aligned} &\textit{minimize} \quad \left(-\sum_{i=1}^{m} \left(\texttt{tanh}(\mathbf{v})\right)_i + \|\mathbf{x}\| \right) \qquad \text{(GPC)} \\ &\text{s.t.} \ \ \mathbf{v} = \sum_{j=1}^{p} \texttt{tanh}(\mathbf{x}^\top \mathbf{M}_j), \quad 0 \le x_i \le 1, \ \ a \le \sum_{i=1}^{m} x_i \le b \end{aligned}$$

Like in PC, GPC also maximizes the number of positive elements in the aggregated vector $\mathbf{v}$, computed by summing up multiple vectors transposed from multiple $\mathbf{M}$. $\mathbf{M}_j$ denotes the matrix constructed by the j^{th} explainer from the total p explainers. Note that `tanh` is also element-wise. The sparsity constraint is ensured by a and b, namely the lower/upper bound of the term-selection budget. The current formulation can now be solved by the latest proposed solver GENO [13] that handles constraints with the *augmented lagrangian algorithm*.

Picking the i^{th} term will choose all i^{th} row vectors simultaneously. Before summing them up, each vector element is already transformed to the same range by `tanh` activation. This accounts for the variable range problem. Figure 3 briefly shows the coverage computing when selecting "pueblo" and "outhouse" during optimization.

5 Experimental Setup

5.1 Datasets and Ranking Models

We choose two datasets: (1) **Clueweb09** collection (category B), for all ranking models, we use 120/40/40 splits for train/dev/test, and the explanation experiments are conducted on the test queries. (2) 40 randomly selected queries from **Trec-DL** 2019 passage ranking test set, and the ranking models are trained on the MS MARCO passage ranking dataset. We focus on the following three neural ranking models:

DRMM [9] computes the term-document similarity histograms beforehand and then jointly learns a matching and a term gate layer from the query and matching histograms. We take the implementation from MatchZoo[2].

BERT [5] takes the query and document separated by [SEP] as input and computes the pooled ([CLS]) representation, on which a feed-forward layer predicts the final relevance score. Both DRMM and BERT models are trained to optimize the margin between the scores of a relevant/non-relevant input pair.

DPR [12] encodes the query and document by two separate BERT models. The relevance is simply measured by the cosine similarity of the two pooled representations. We use the pretrained checkpoints directly without fine-tuning.

[2] https://github.com/NTMC-Community/MatchZoo.

5.2 Baseline and Competitors

We compare our approach named MULTIPLEX with the following methods:

QUERY-TERMS serves as the baseline by feeding only the query terms to our explainers. By comparing this baseline, we argue that only the original query is insufficient to discover the underlying ranking logic.

DEEPLIFT [21] is a popular white-box feature attribution method. To adapt it to ranking, we first compute the importance of a word in a document using Captum[3], then we take the average across all documents and extract important terms as a listwise explanation for a query. Note that we omit this baseline for DRMM since its input is a histogram, thus the importance cannot be attributed to the word level.

GREEDY-LM [25] uses a term-matching explainer to approximate neural rankers. It optimizes the preference coverage greedily. Our approach shares a similar pipeline of generating candidate terms and preference matrices. By comparing this baseline, we show the improvements of combining multiple explainers and approximated linear programming optimization.

5.3 Metrics

Since multiple explainers are applied, a preference pair from the original ranking is counted as explained as long as a single explainer can explain it. This evaluation does not apply to GREEDY-LM as it generates explanation terms based on a single explainer. For both GREEDY-LM and MULTIPLEX, we fix 200 candidate terms and 500 sampled pairs for preference matrix construction. We also fix a maximum of 10 explanation terms for all methods except QUERY-TERMS. For both datasets, we consider a ranking depth (k) of 100.

Similar to [19], we measure fidelity by computing the fraction of the maintained preference pairs by the explainers given the explanation terms. In other words, the fidelity measures the coverage over the *feasible* preference pairs. As mentioned in Sect. 3, depending on the choice of feasible preference pairs, we consider the following three variants of fidelity:

Fidelity-global ($\mathcal{F}_{\text{global}}$) includes all $\binom{k}{2}$ pairs induced by a k-length ranking list.

Fidelity-sampled ($\mathcal{F}_{\text{sampled}}$) considers the sampled pairs from the matrix construction.

Fidelity-diff ($\mathcal{F}_{\text{diff}}$) discards all pairs whose relevance score difference $< g$. The magnitude of g is chosen based on the relevance score distribution of a particular model. For BERT we set $g = 2$ as the prediction margin appears to be larger than the rest two models, for which $g = 0.05$.

6 Evaluation Results

To show the effectiveness of our approach, we first present the quality of our approach in terms of fidelity on all datasets and models compared to other competitors in Table 1.

[3] https://github.com/pytorch/captum.

Table 1. Fidelity ($\mathcal{F}$) results. The best results are in bold.

Model	Clueweb09				Trec-DL		
	Method	$\mathcal{F}_{\text{global}}$	$\mathcal{F}_{\text{diff}}$	$\mathcal{F}_{\text{sampled}}$	$\mathcal{F}_{\text{global}}$	$\mathcal{F}_{\text{diff}}$	$\mathcal{F}_{\text{sampled}}$
BERT	QUERY-TERMS	0.81	0.88	0.76	0.81	0.82	0.63
	DEEPLIFT [21]	0.77	0.81	0.67	0.70	0.75	0.62
	GREEDY-LM [25]	0.63	0.77	0.69	0.59	0.69	0.84
	MULTIPLEX	**0.88**	**0.97**	**0.93**	**0.86**	**0.93**	**0.97**
DPR	QUERY-TERMS	0.81	0.86	0.71	0.82	0.84	0.64
	DEEPLIFT [21]	0.68	0.71	0.57	0.60	0.63	0.58
	GREEDY-LM [25]	0.61	0.68	**0.88**	0.63	0.70	0.75
	MULTIPLEX	0.87	**0.93**	0.87	**0.87**	**0.92**	**0.96**
DRMM	QUERY-TERMS	0.82	0.85	0.72	0.80	0.81	0.59
	DEEPLIFT [21]	–	–	–	–	–	–
	GREEDY-LM [25]	0.57	0.60	0.72	0.53	0.54	0.34
	MULTIPLEX	**0.88**	**0.92**	**0.84**	**0.85**	**0.88**	**0.95**

Then we show the improvements of adding multiple explainers by an ablation study presented in Fig. 4. Finally, we discuss how our explanations can be used to explain a specific preference pair, as well as other potential use cases.

6.1 Effectiveness of Explanations

In terms of fidelity (cf. Table 1), our method consistently outperforms other competitors. Besides, for all methods the global fidelity ($\mathcal{F}_{\text{global}}$) scores are always lower than $\mathcal{F}_{\text{diff}}$ where close, hence potentially noisy pairs are all excluded. This shows that all methods and prominently MULTIPLEX can better explain document pairs with larger differences in relevance scores.

Ranking Heuristics vs. Query Expansion. Though both factors constitute the explanation of ranking, which one is more crucial? Take QUERY-TERMS and GREEDY-LM as a comparison, note that QUERY-TERMS includes the given query terms but three ranking heuristics, while GREEDY-LM on the contrary only relies on one term-matching but richer query information. Their fidelity results show QUERY-TERMS outperforms GREEDY-LM by a large margin, strongly suggesting that ranking heuristics particularly semantic similarity, are more effective in explaining neural models.

The Importance of Explanation Aggregation. Applying simple aggregation strategies (i.e. *average*) on the prevalent pointwise feature attribution methods is shown to be less effective by the results of DEEPLIFT. Compared to QUERY-TERMS, the extra expanded query terms extracted by DEEPLIFT seem unhelpful in enhancing fidelity but introducing noise. On the other hand, methods directly optimizing fidelity (i.e. GREEDY-LM and MULTIPLEX) explicitly include the aggregation in the optimization loop. The $\mathcal{F}_{\text{sampled}}$ results of DEEPLIFT and GREEDY-LM further confirm the importance of aggregation.

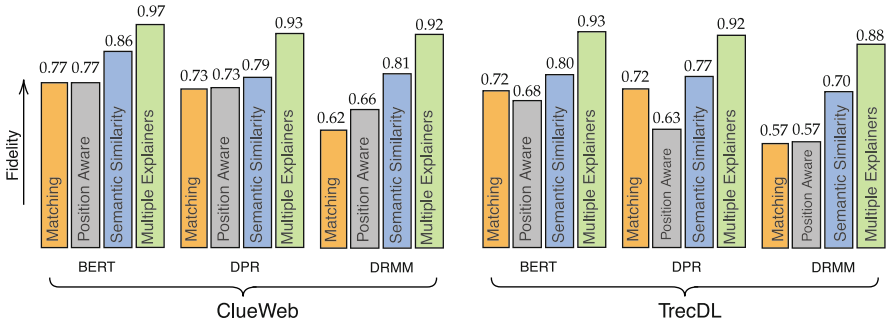
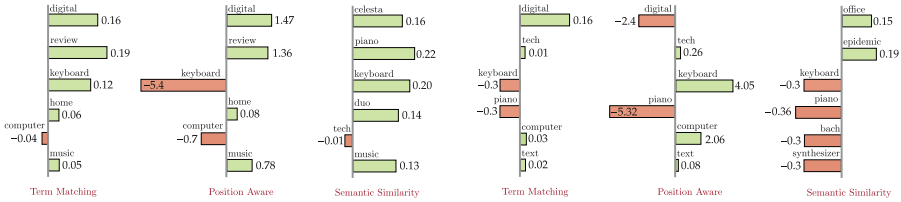


Fig. 4. *Fidelity-diff* results of each single and combined explainer using our method.



(a) BERT: music-related terms are *pos*.

(b) DPR: music-related terms are *neg*.

Fig. 5. Query: keyboard review. Document pair: clueweb09-en0008-49-09140 (musical keyboard) *vs.* clueweb09-en0010-56-37788 (technical keyboard). BERT prefers the former whereas DPR prefers the latter, resulting in opposite explanations.

The Benefits of Our Optimization Solution. We also experimented with every single Ψ to extract explanation terms with our approximated ILP objective shown in Fig. 4. Comparing the fidelity results of term-matching (orange bar) with the $\mathcal{F}_{\text{diff}}$ of GREEDY-LM (using the same explainer) in Table 1, we show the superiority of our optimizing strategy over the greedy-algorithm.

The Benefits of Combining Explainers. As Fig. 4 indicates, semantic explainer overall generates the most faithful explanations than the rest. However, combining all explainers can further improve the preference coverage and in turn increase the fidelity results. When one explainer fails to explain a pair, it is still possible to be covered by other explainers. Moreover, we also notice that combining multiple explainers in optimization can generate explanation terms exhibiting multiple topic aspects, especially for short and ambiguous queries. More examples are presented in Fig. 1 and Fig. 6.

6.2 Utility of Explanations

Explaining Document Preference. We now show how to explain a single preference pair using MULTIPLEX, i.e., why does a model prefer d_i over d_j? We start by constructing preference scores for each candidate term as described in Sect. 4.1. Next, we select

Query	Explainer	Explanation	$\mathcal{F}_{\text{diff}}$
adobe indian houses	Term Matching	pdf, adobe, style, house, first, also	0.85
	Position Aware	pdf, adobe, style, texas, wikipedia, 2009	0.81
	Semantic Similarity	pueblo, amarillo, castroville, outhouse, abourezk, alcove	**0.95**
	Multiplex	pueblo, amarillo, castroville, outhouse, abourezk, pdf	0.91
espn sports	Term Matching	espn, abc, network, company, award, entertainment,	0.86
	Position Aware	espn, sportscenter, abc, company, news, espn.com	**0.99**
	Semantic Similarity	espn, sportscenter, abc, walt, disney, entertainment,	0.93
	Multiplex	espn, sportscenter, abc, walt, disney, news, espn.com	**0.99**
hp mini 2140	Text Matching	hp, mini, 2140, 2133	**0.94**
	Position Aware	hp, mini, 2140, 2133	0.90
	Semantic Similarity	hp, touchpad, overview, hdd,	0.71
	Multiplex	hp, mini, 2140, 2133, touchpad, overview	0.91

Fig. 6. Anecdotal examples show that each explainer selects terms from a different aspect. The color highlights denote the explanation terms in *Multiple* are combined from different explainers. For ambiguous query "adobe Indian houses", *Term Matching* and *Position Aware* focus on popular but 'shallow' terms indicating "adobe company". For certain query "hp mini 2140", the *semantic similarity* suffers from OOV. *Position Aware* can capture the non-frequent yet important terms based on their position, e.g., the official site for the query "ESPN sports".

the important terms with significant scores. Figure 5 illustrates the explanation terms of two opposing decisions by BERT and DPR respectively, for `keyboard review`.

Discovering Model Preference and Spurious Correlations. We believe that explanation terms encode relevance factors that rank relevant documents over others. Based on this assumption, we create a perturbed document by adding explanation terms to a potentially *non-relevant document* (e.g. at the lowest rank). We then feed this modified document to the black-box model and measure the rank improvement. Unsurprisingly, the terms extracted by MULTIPLEX result in the maximum rank increase (cf. Fig. 7), meaning our method can better identify the black-box model's preference. Moreover, we manually selected some ambiguous queries, and our initial observation of their explanation terms suggests the ranking model shows some topic preference when ranking the documents, while the explanation terms representing the preferred topics are also shown dominant quantitively. Thus, it helps understand the model's topic preference more easily by analyzing the explanations instead of going through hundreds of documents.

Another possible usage is model debugging, or finding spurious correlations in models or datasets, by analyzing explanation terms. One simple example is "Wikipedia" which appears as an explanation term for many different queries. This is not surprising as the Wikipedia entity pages are usually labeled as relevant. We leave a more systematic exploration of making use of ranking explanations to future work.

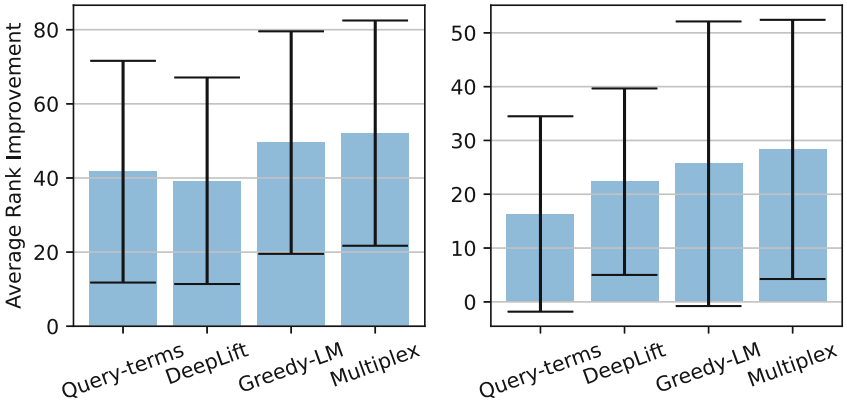


Fig. 7. Average rank improvements. Left: on all test queries; Right: on hand-picked ambiguous queries. Note that for each query the document size ≤ 100.

7 Conclusion and Outlook

This paper proposes a post-hoc model-agnostic framework to explain text ranking models using multiple explainers. Our method MULTIPLEX systematically combines multiple explainers to capture different relevance factors encoded in the ranking decisions. The extensive experiments show that our method can generate high-fidelity explanations for over-parameterized models like BERT, delivering up to 54% fidelity improvements. Our method explains a ranking by a set of terms attributed to a union of multiple explainers. It is interesting to examine which explainer (or ranking heuristic) contributes to which extent using which particular terms for future work. We also plan to extend our framework to account for n-grams and to make our explanation generation procedure efficient enough to be used during query processing. Moreover, it is well known that validating explanations is challenging, especially in the absence of ground-truth data. We measure fidelity in this work, however, the fidelity might not reflect the real underline logic of a complex model. Therefore, incorporating human perspectives into the evaluation and meanwhile, balancing the cost of annotating numerous decisions in a ranking are also worth exploring in future work.

Acknowledgements. This work is partially supported by German Research Foundation (DFG), under the Project IREM with grant No. AN 996/1-1.

References

1. Ailon, N., Charikar, M., Newman, A.: Aggregating inconsistent information: ranking and clustering. J. ACM **55**(5), 23:1–23:27 (2008). https://doi.org/10.1145/1411509.1411513
2. Amati, G., Van Rijsbergen, C.J.: Probabilistic models of information retrieval based on measuring the divergence from randomness. ACM Trans. Inf. Syst. **20**(4), 357–389 (2002). https://doi.org/10.1145/582415.582416, http://doi.acm.org/10.1145/582415.582416

3. Anand, A., Lyu, L., Idahl, M., Wang, Y., Wallat, J., Zhang, Z.: Explainable information retrieval: a survey. CoRR abs/2211.02405 (2022). https://doi.org/10.48550/arXiv.2211.02405
4. Câmara, A., Hauff, C.: Diagnosing BERT with retrieval heuristics. In: Jose, J.M., et al. (eds.) ECIR 2020. LNCS, vol. 12035, pp. 605–618. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-45439-5_40
5. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Burstein, J., Doran, C., Solorio, T. (eds.) 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, 2–7 June 2019, Volume 1 (Long and Short Papers), pp. 4171–4186. Association for Computational Linguistics (2019). https://doi.org/10.18653/v1/n19-1423
6. Diamond, S., Boyd, S.P.: CVXPY: a Python-embedded modeling language for convex optimization. J. Mach. Learn. Res. **17**, 83:1–83:5 (2016). http://jmlr.org/papers/v17/15-408.html
7. Fernando, Z.T., Singh, J., Anand, A.: A study on the interpretability of neural retrieval models using deepshap. In: Proceedings of the 42Nd International ACM SIGIR Conference on Research and Development in Information Retrieval. SIGIR 2019, pp. 1005–1008. ACM, New York (2019). https://doi.org/10.1145/3331184.3331312, http://doi.acm.org/10.1145/3331184.3331312
8. Fetahu, B., Markert, K., Anand, A.: Automated news suggestions for populating Wikipedia entity pages. In: Bailey, J., et al. (eds.) Proceedings of the 24th ACM International Conference on Information and Knowledge Management. CIKM 2015, Melbourne, VIC, Australia, 19–23 October 2015, pp. 323–332. ACM (2015). https://doi.org/10.1145/2806416.2806531
9. Guo, J., Fan, Y., Ai, Q., Croft, W.B.: A deep relevance matching model for ad-hoc retrieval. In: Mukhopadhyay, S., et al. (eds.) Proceedings of the 25th ACM International Conference on Information and Knowledge Management. CIKM 2016, Indianapolis, IN, USA, 24–28 October 2016, pp. 55–64. ACM (2016). https://doi.org/10.1145/2983323.2983769
10. Hofstätter, S., Mitra, B., Zamani, H., Craswell, N., Hanbury, A.: Intra-document cascading: learning to select passages for neural document ranking. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval. SIGIR 2021, pp. 1349–1358. Association for Computing Machinery, New York (2021). https://doi.org/10.1145/3404835.3462889
11. Jaleel, N.A., et al.: Umass at TREC 2004: novelty and HARD. In: Voorhees, E.M., Buckland, L.P. (eds.) Proceedings of the Thirteenth Text REtrieval Conference, TREC 2004, Gaithersburg, Maryland, USA, 16–19 November 2004. NIST Special Publication, vol. 500-261. National Institute of Standards and Technology (NIST) (2004). http://trec.nist.gov/pubs/trec13/papers/umass.novelty.hard.pdf
12. Karpukhin, V., et al.: Dense passage retrieval for open-domain question answering. In: Webber, B., Cohn, T., He, Y., Liu, Y. (eds.) Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, 16–20 November 2020, pp. 6769–6781. Association for Computational Linguistics (2020). https://doi.org/10.18653/v1/2020.emnlp-main.550
13. Laue, S., Mitterreiter, M., Giesen, J.: GENO - generic optimization for classical machine learning. In: Wallach, H.M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E.B., Garnett, R. (eds.) Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019. NeurIPS 2019, 8–14 December 2019. Vancouver, BC, Canada, pp. 2187–2198 (2019). https://proceedings.neurips.cc/paper/2019/hash/84438b7aae55a0638073ef798e50b4ef-Abstract.html
14. Leonhardt, J., Rudra, K., Anand, A.: Extractive explanations for interpretable text ranking. ACM Trans. Inf. Syst. (2022). https://doi.org/10.1145/3576924

15. Lundberg, S.M., Lee, S.: A unified approach to interpreting model predictions. In: Guyon, I., et al. (eds.) Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4–9 December 2017, Long Beach, CA, USA, pp. 4765–4774 (2017). https://proceedings.neurips.cc/paper/2017/hash/8a20a8621978632d76c43dfd28b67767-Abstract.html
16. McDonald, R.T., Brokos, G., Androutsopoulos, I.: Deep relevance ranking using enhanced document-query interactions. In: Riloff, E., Chiang, D., Hockenmaier, J., Tsujii, J. (eds.) Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October–4 November 2018, pp. 1849–1860. Association for Computational Linguistics (2018). https://doi.org/10.18653/v1/d18-1211
17. Nogueira, R.F., Cho, K.: Passage re-ranking with BERT. CoRR abs/1901.04085 (2019). http://arxiv.org/abs/1901.04085
18. Pennington, J., Socher, R., Manning, C.D.: Glove: global vectors for word representation. In: Moschitti, A., Pang, B., Daelemans, W. (eds.) Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, 25–29 October 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL, pp. 1532–1543. ACL (2014). https://doi.org/10.3115/v1/d14-1162
19. Rennings, D., Moraes, F., Hauff, C.: An axiomatic approach to diagnosing neural IR models. In: Azzopardi, L., Stein, B., Fuhr, N., Mayr, P., Hauff, C., Hiemstra, D. (eds.) ECIR 2019. LNCS, vol. 11437, pp. 489–503. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-15712-8_32
20. Ribeiro, M.T., Singh, S., Guestrin, C.: "why should I trust you?": Explaining the predictions of any classifier. In: Krishnapuram, B., Shah, M., Smola, A.J., Aggarwal, C.C., Shen, D., Rastogi, R. (eds.) Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016, pp. 1135–1144. ACM (2016). https://doi.org/10.1145/2939672.2939778
21. Shrikumar, A., Greenside, P., Kundaje, A.: Learning important features through propagating activation differences. In: Precup, D., Teh, Y.W. (eds.) Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6–11 August 2017. Proceedings of Machine Learning Research, vol. 70, pp. 3145–3153. PMLR (2017), http://proceedings.mlr.press/v70/shrikumar17a.html
22. Simonyan, K., Vedaldi, A., Zisserman, A.: Deep inside convolutional networks: visualising image classification models and saliency maps. In: Bengio, Y., LeCun, Y. (eds.) 2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, 14–16 April 2014, Workshop Track Proceedings (2014). http://arxiv.org/abs/1312.6034
23. Singh, J., Anand, A.: Posthoc interpretability of learning to rank models using secondary training data. In: Workshop on ExplainAble Recommendation and Search (EARS 2018) at SIGIR 2018 (2018). https://ears2018.github.io/ears18-singh.pdf
24. Singh, J., Anand, A.: Exs: explainable search using local model agnostic interpretability. In: Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining. WSDM 2019, pp. 770–773. ACM, New York (2019). https://doi.org/10.1145/3289600.3290620, http://doi.acm.org/10.1145/3289600.3290620
25. Singh, J., Anand, A.: Model agnostic interpretability of rankers via intent modelling. In: Hildebrandt, M., Castillo, C., Celis, L.E., Ruggieri, S., Taylor, L., Zanfir-Fortuna, G. (eds.) FAT* '20: Conference on Fairness, Accountability, and Transparency, Barcelona, Spain, 27–30 January 2020, pp. 618–628. ACM (2020). https://doi.org/10.1145/3351095.3375234
26. Singh, J., Khosla, M., Wang, Z., Anand, A.: Extracting per query valid explanations for blackbox learning-to-rank models. In: Hasibi, F., Fang, Y., Aizawa, A. (eds.) ICTIR 2021: The 2021 ACM SIGIR International Conference on the Theory of Information Retrieval, Virtual Event, Canada, 11 July 2021, pp. 203–210. ACM (2021). https://doi.org/10.1145/3471158.3472241

27. Singh, J., Wallat, J., Anand, A.: BERTnesia: investigating the capture and forgetting of knowledge in BERT. In: Alishahi, A., Belinkov, Y., Chrupala, G., Hupkes, D., Pinter, Y., Sajjad, H. (eds.) Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP, BlackboxNLP@EMNLP 2020, Online, November 2020, pp. 174–183. Association for Computational Linguistics (2020). https://doi.org/10.18653/v1/2020.blackboxnlp-1.17
28. Sundararajan, M., Taly, A., Yan, Q.: Axiomatic attribution for deep networks. In: Precup, D., Teh, Y.W. (eds.) Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6–11 August 2017. Proceedings of Machine Learning Research, vol. 70, pp. 3319–3328. PMLR (2017). http://proceedings.mlr.press/v70/sundararajan17a.html
29. Verma, M., Ganguly, D.: LIRME: locally interpretable ranking model explanation. In: Piwowarski, B., Chevalier, M., Gaussier, É., Maarek, Y., Nie, J., Scholer, F. (eds.) Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, 21–25 July 2019, pp. 1281–1284. ACM (2019). https://doi.org/10.1145/3331184.3331377
30. Völske, M., et al.: Towards axiomatic explanations for neural ranking models. In: Hasibi, F., Fang, Y., Aizawa, A. (eds.) ICTIR 2021: The 2021 ACM SIGIR International Conference on the Theory of Information Retrieval, Virtual Event, Canada, 11 July 2021, pp. 13–22. ACM (2021). https://doi.org/10.1145/3471158.3472256
31. Wallat, J., Beringer, F., Anand, A., Anand, A.: Probing BERT for ranking abilities. In: Advances in Information Retrieval - 45th European Conference on IR Research, ECIR 2023, Dublin, Ireland. LNCS. Springer, Cham (2023)
32. Wang, Y., Lyu, L., Anand, A.: BERT rankers are brittle: a study using adversarial document perturbations. In: Crestani, F., Pasi, G., Gaussier, É. (eds.) ICTIR 2022: The 2022 ACM SIGIR International Conference on the Theory of Information Retrieval, Madrid, Spain, 11–12 July 2022, pp. 115–120. ACM (2022). https://doi.org/10.1145/3539813.3545122
33. Yu, P., Rahimi, R., Allan, J.: Towards explainable search results: a listwise explanation generator. In: Amigó, E., Castells, P., Gonzalo, J., Carterette, B., Culpepper, J.S., Kazai, G. (eds.) SIGIR 2022: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, 11–15 July 2022, pp. 669–680. ACM (2022). https://doi.org/10.1145/3477495.3532067
34. Zhang, Z., Rudra, K., Anand, A.: Explain and predict, and then predict again. In: Proceedings of the 14th ACM International Conference on Web Search and Data Mining, pp. 418–426 (2021)