Explaining Ranking Models using Multiple Explainers

Anonymous ACL submission

Abstract

Current approaches to interpreting complex ranking models are based on local approximations of the ranking model using a simple ranker in the locality of the query. 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 local fidelity. In this paper, we overcome this problem by considering multiple simple rankers for better approximating the 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 post-hoc interpretability of ranking models. Our approach MULTIPLEX uses a linear programming approach to judiciously extract the explanation terms. We conduct extensive experiments on a variety of ranking models and report fidelity improvements of $37\% - 54\%$ over existing baselines and competitors. We finally qualitatively compare modern neural ranking models in terms of their explanations to better understand the differences between them, showcasing our explainers' practical utility.

1 Introduction

Ad-hoc document ranking is a central task in Web search and information retrieval, where the objective is to rank text documents or passages relevant to a user-specified keyword query. Recent approaches for ranking text documents have focused heavily on neural models (McDonald et al., 2018; Karpukhin et al., 2020; Nogueira and Cho, 2019). Neural rankers manage to learn the complex and often non-linear relationships between the query and document terms that are difficult to encode using closed-form analytical ranking functions like BM25 or query likelihood models. However, the superior ranking performance of such text rankers comes at the expense of reduced interpretability, increasing the risk for models encoding undesirable correlations and biases. In parallel to developing better ranking models, there has been an increased focus on interpretability approaches for neural ranking models (Singh and Anand, 2019, 2018; Fernando et al., 2019) that specifically aim at explaining the rationale behind the ranking decisions.

This paper aims to propose post-hoc interpretability algorithms for neural rankers – that is, we intend to explain an already-trained text ranking model. Since post-hoc methods do not compromise the accuracy of the learned model, they have become popular in the emerging landscape of interpretable machine learning. The key idea in post-hoc interpretability is to locally approximate a trained model with a simple and interpretable proxy model where the degree of approximation is called *fidelity*. In this framework, the objective is to maximize the fidelity, where the choice of the proxy model and the notion of fidelity is typically domain-dependent. Adapting this general framework of post-hoc interpretability 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 regarding different relevance factors?*

Rankings as aggregations. Ranking models output a ranked list of documents for a given query. Unlike other learning tasks like regression and classification that deal with a single decision, the ranking task can be viewed as an *aggregation of decisions* – pairwise document preferences combined using approaches such as (Ailon et al., 2008). Any interpretability approach or explainer should therefore explain the reasoning behind multiple-preference pair predictions. For example, for the top-ranked document d_i, the explainer should explain why d_i is more relevant than other documents in the ranked list where each preference pair $d_i \succ d_j$ is a decision to be explained. Therefore

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
(Multiple Explainers)	extinction, deserts, gvwr, north

Table 1: Explaining the query `bobcat` with multiple aspects – (i) "charlotte-bobcat basketball club"; (ii) "learn to hunt bobcat"; (iii) "animal bobcat" and (iv) "bobcat mechanical retailer". MULTIPLEX carefully chooses from multiple aspects to explain a ranking. See Table 3 for more examples.

the first challenge of explaining rankings is that existing explanation methods like (Shrikumar et al., 2017; Choi et al., 2020; Sundararajan et al., 2017; Simonyan et al., 2013) that explain a single decision cannot be seamlessly used for rankings.

Different explanations for different decisions. Secondly, it is well-known that when ranking text, multiple factors determine the relevance of a document to a query, e.g., lexical matching, semantic similarity, query term position in the document, etc. Unlike traditional models that used to model each of these relevance factors, neural rankers automatically learn these 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 factor that explains or satisfies all or most of the preferences $d_i \succ d_j$.

Previous approaches for ranking explanations in the literature fail to address at least one of these limitations. Gradient-based approaches either use simple heuristic aggregation approaches or altogether avoid aggregations. Approaches like Singh and Anand (2020) that consider aggregations try to explain multiple preferences using a single simple explainer that captures only one aspect of relevance, i.e., term matching. Both these limitations in existing systems have resulted in low-fidelity explanations.

In this paper, we propose a more principled approach MULTIPLEX, by considering multiple simple rankers, or explainers, that rely on different notions of relevance and are interpretable by themselves. The output of MULTIPLEX is a set of explanation terms. Table 1 shows an example of explanation terms extracted by different explainers (formally defined in Section 4.3), and our MULTIPLEX is shown to be able to combine terms from multiple explainers, implicitly covering multiple topics for an ambiguous query.

Specifically, we define the GENERALIZED PREFERENCE COVERAGE (GPC) using multiple explainers that intend to maximize our explanation's approximation ability. Unlike previous approaches that rely on greedy heuristic to maximize fidelity, MULTIPLEX is based on convex optimization using the *augmented lagrangian algorithm* to solve the GPC problem. In coming up with the explanation, we hypothesize that an expanded query (the original query along with the explanation terms) combined with a simple and interpretable explainer is an accurate interpretation of the underlying ranking model if it produces a similar ranking.

Experimental Evaluation. We conduct extensive experiments using datasets from the TREC test collections – Trec-DL and Clueweb09 with 3 neural rankers to evaluate MULTIPLEX. We report fidelity improvements of $37\% - 54\%$ over existing competitors. We also present anecdotal case studies that showcase the practical utility of MULTIPLEX in understanding neural rankers.

2 Related Work

Interpretability, in general, of complex learning models is a well researched area (Guidotti et al., 2018). We are interested however in the interpretability of models for text tasks and specifically text ranking . In the following we try to contextualize our contribution in terms of (i) general methodology of post-hoc interpretability (in Section 2.1), and (ii) specific approaches to interpretability for ranking models (in Section 2.2).

2.1 Post-hoc interpretability

Post-hoc methods for interpretability approaches operate on already trained models. Such methods can be categorized into two broad classes: *model introspective* and *model agnostic*. Model introspection refers to approaches that access all the model parameters like gradient-based methods (Shrikumar et al., 2017; Choi et al., 2020; Sundararajan et al., 2017; Simonyan et al., 2013). We operate in the model agnostic regime where we do not assume any access to the ranking model's parameters. Example of such an approach is LIME (Ribeiro et al., 2016) that uses a simple interpretable surrogate model to locally approximate an already trained black-box model. For other notions of inter-

pretability and a more comprehensive description of the approaches, we point the readers to Guidotti et al. (2018).

2.2 Interpreting Ranking Models

In information retrieval (IR) there has been limited work on interpreting rankings. The earliest for explaining rankings was in the context of learning-to-rank (LTR) where Singh and Anand (2018) tried to approximate an already trained LTR model by a subset of (the original) interpretable features using secondary training data from the output of the original model. Singh et al. (2021) also operate on LTR to find a minimal set of relevant features that faithfully explain the ranking. This paper doesn't deal with LTR task but instead focuses on ranking text data.

Gradient-based approaches like Fernando et al. (2019); Choi et al. (2020) have been applied to neural rankers for interpreting the relevance score of a single query document pair. These works do not consider aggregations of decisions for a given query, i.e., preference pairs of documents. We use a gradient-based approach as a competitor in our experiments, showing that aggregation is crucial to explaining ranking models. Rennings et al. (2019); Câmara and Hauff (2020); Völske et al. (2021) explain adhoc ranking models in terms of IR axioms. They argue that neural rankers do not follow IR axioms such as term frequency and semantic similarity. This is also in line with our results based on query terms and a single explainer. With additional explanation terms, we show higher agreements between neural rankers and axioms. Like us, Völske et al. (2021) also focus on optimizing coverage of preference pairs but unlike us using learning approaches. However, they also report lower fidelity values showing the difficulty of the task at hand.

More related to our setup, Verma and Ganguly (2019) modifies LIME to output intent terms by measuring the distance of the explanation terms to the human-created intent description terms. However, their approach does not consider preference pairs induced by rankings. Singh and Anand (2019) propose simplistic aggregations over sampled terms from top-k results, an improved adaptable LIME for rankers. The most related to our work is from Singh and Anand (2020), which proposes a heuristic greedy approach to maximize the coverage of preference pairs.

Unlike above methods (Verma and Ganguly, 2019; Singh and Anand, 2019, 2020), we use multiple explainers and a more principled aggregation approach to maximize fidelity. We also use Singh and Anand (2020) as a baseline to measure the merits of our aggregation techniques with multiple explainers.

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 the ranking function Φ in a model-agnostic manner. Note that the output of a ranking function 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 Explanations and Fidelity

The output of an interpretability procedure is an explanation, which should be *simple*, *human-understandable*, and *faithful* to the behavior of Φ. A natural explanation for keyword-query based ranking tasks are a *set of terms* (words or phrases). Simplicity is ensured by enforcing only a small set of output terms. We adapt the query expansion concept in our setting. Namely, if adding a set of explanation terms $\mathbb{E}$ to the original query, i.e., $\mathcal{Q} \cup \mathbb{E}$, the explainer should result in high agreement with the original ranking preferences. The degree of such agreement is used to evaluate the goodness or *fidelity* of the explanation.

3.2 Problem Statement

Formally, given a query $\mathcal{Q}$, a complex ranking model Φ and a set of simple ranking models which we call explainers $\{\Psi\}$, we aim to select a small set of terms $\mathbb{E} \in \mathcal{V}$ (where $\mathcal{V}$ is the vocabulary), to explain most of preference pairs $(d_i \succ d_j)$ from the original ranking π.

4 Generalized Preference Coverage

As mentioned earlier, a ranking can be considered a set of preference pairs. Therefore, choosing explanation terms to maximize the fidelity can be formulated as a coverage problem of the preference pairs. We briefly describe the preference coverage

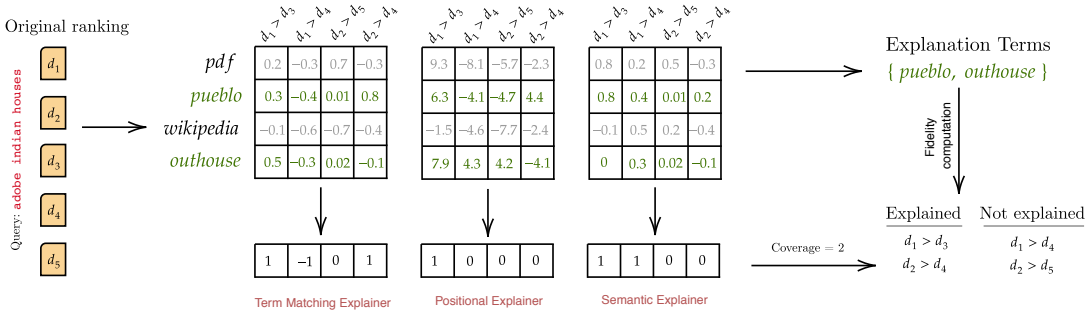


Figure 1: Approach overview of MULTIPLEX using multiple explainers.

(PC) framework as introduced in Singh and Anand (2020), using a single explainer as a precursor to introducing the generalized PC problem.

4.1 The Preference Coverage Framework

Singh and Anand (2020) introduced the PC framework where explaining a ranking is equivalent to maintaining most of the *preferences* derived from the ranking using a single explainer Ψ. First, a set of n candidate terms $\mathcal{X}(\mathcal{X} \subseteq \mathcal{V}, |\mathcal{X}| = n)$ and m preference pairs are sampled (please refer to the original paper for more details) from the original ranking π towards creating a preference matrix $\mathbf{M} \in \mathbb{R}^{n \times m}$. Each cell in $\mathbf{M}$ represents the utility of the term t in explaining the preference $d_{\pi(i)} \succ d_{\pi(j)}$, 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 not. Each t can now be viewed as an m-dimensional vector $\mathbf{f}$ or $|\mathbf{f}| = m$, where each element represents how well it explains a certain 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):

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

$\mathbf{x}$ is a selection vector with boolean values where $x_i = 1$ indicates selecting term $\mathcal{X}_i$, and $x_i = 0$ otherwise. Namely, $\mathbb{E} = \{i | x_i == 1\}$. This equation however is NP-hard and not solvable by the prevalent convex programming solvers supported by CVXPY (Diamond and Boyd, 2016), due to the non-convex `sign` function.

In the 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 more 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 range 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. Taking normalization of such scores into the optimization procedure is crucial to flexibly adding multiple explainers.

We therefore formulate the GENERALIZED PREFERENCE COVERAGE problem that intends to optimize multiple matrices simultaneously as:

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

Like in PC, this equation 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 and p denotes the number of explainers. The current formulation can now be solved efficiently by modern quasi-Newton solvers that handle constraints with the *augmented lagrangian algorithm* like GENO (Laue et al., 2019).

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 1 briefly shows the computing process during optimization.

4.3 Choice of Explainers

In choosing multiple explainers we ensure that each explainer is simple and human-understandable. A widely used explainer is the term-matching-based BM25 model (Singh and Anand, 2019, 2020; Verma and Ganguly, 2019). Another critical aspect in retrieval is term positions in the document. Specifically, in news articles the title and the introductory paragraphs are regarded to be more relevant. Finally, semantic similarity is known to be crucial to address the vocabulary mismatch problem. This is particularly true in neural network models with embedding vectors as input. In summary, we name the combined explainer as Ψ_{multi} which employs the following three explainers while adding more is also allowed:

TERM MATCHING or Ψ_{lm}: the BM25 score of a term t in document d, $sim = \texttt{BM25}(t, d)$.

POSITION AWARE or Ψ_{pa}: a position-aware model (Fetahu et al., 2015) to measure the position importance of a term t in document d, $sim = \frac{1}{|d|}\sum_{p \in d} \texttt{tf}(t,p)^{\frac{1}{p}}$, where p denotes the p_{th} paragraph in d, $\texttt{tf}(t,p)$ denotes the term frequency of t in paragraph p.

SEMANTIC SIMILARITY or Ψ_{emb}: a model to measure the semantic similarity between t and d. We use the pre-trained GloVe embedding (Pennington et al., 2014). Formally, $sim = \frac{1}{|d|}\sum_{w \in d} \texttt{cosine}(t, w)$.

5 Experimental Setup

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) **Trec-DL** 2019 passage ranking testset, from which we randomly select 40 queries. The ranking models are trained on the MsMarco passage ranking dataset.

5.1 Ranking Models

We focus on neural network models, which are inherently hard to interpret due to a large set of parameters. All neural models are trained and evaluated on the top 100 retrieved documents or passages. In detail, we explain three ranking models:

DRMM (Guo et al., 2016) 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[1].

BERT (Devlin et al., 2018) model has achieved the SOTA performances in many language tasks, including text ranking. We fine-tune the pretrained bert-base-uncased model on our datasets.

DPR (Karpukhin et al., 2020) or two-tower bert model encodes the query and document separately. The relevance score is simply the cosine similarity of the pooled representations. We use the pretrained models directly without fine-tuning. All details about data splitting and model training can be found in Appendix A.

5.2 Baseline and Competitors

We compare our approach named MULTIPLEX with the following baseline 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 (Shrikumar et al., 2017) is a popular feature importance method. We compute the importance of a word in a document as $s(w, d_{\pi(i)}) = \texttt{DL}(w, d_{\pi(i)}) \cdot \log(i+2)$. `DL` is the feature importance function. Then we take the average across all documents to extract important terms as 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 token level.

GREEDY-LM (Singh and Anand, 2020) uses a language model explainer to approximate neural

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

		Clueweb09			Trec-DL		
Model	**Method**	*Fidelity*	*Fidelity*†	*Fidelity*‡	*Fidelity*	*Fidelity*†	*Fidelity*‡
BERT	QUERY-TERMS	0.81	0.88	0.76	0.81	0.82	0.63
	DEEPLIFT (Shrikumar et al., 2017)	0.77	0.81	0.67	0.70	0.75	0.62
	GREEDY-LM (Singh and Anand, 2020)	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 (Shrikumar et al., 2017)	0.68	0.71	0.57	0.60	0.63	0.58
	GREEDY-LM (Singh and Anand, 2020)	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 (Shrikumar et al., 2017)	-	-	-	-	-	-
	GREEDY-LM (Singh and Anand, 2020)	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**

Table 2: Fidelity values of all models on both Clueweb09 and Trec-DL datasets. *Fidelity*† refers to fidelity where preference pairs have at least a rank difference $\geq g$. *Fidelity*‡ only considers the preference pairs in the sampled set (500 pairs in our experiments). The best results are in bold.

rankers. It optimizes the preference coverage greedily. Our approach shares a similar pipeline of generating candidate terms and preference matrix. By comparing this baseline, we show the improvements of combining multiple explainers and approximated linear programming optimization.

5.3 Metrics

Similar to Rennings et al. (2019), we measure fidelity by computing the fraction of the satisfied preference pairs when the explanation terms are used in addition to the query terms by the explainers. In other words, the fidelity measures the coverage over the feasible preference pairs.

We consider three variants of fidelity – *Fidelity*, *Fidelity*† and *Fidelity*‡ – depending on what pairs we consider feasible. *Fidelity* considers the coverage over all $\binom{k}{2}$ pairs induced by a k-length ranking. For *Fidelity*†, we disregard pairs with a relevance score difference $< g$, to adjust for rank variations due to small noise. Finally *Fidelity*‡ considers the coverage over the sampled pairs from the matrix construction. Note that we do not consider all pairs from ranking in constructing preference matrix due to the quadratic time complexity. In experiments, we fix 200 candidate terms and sample 500 preference pairs for preference matrix. *Fidelity*‡ is computed on exactly the same 500 pairs for all methods for fair comparison. Since we apply multiple explainers we adopt disjunctive semantics to preference satisfaction for all methods except GREEDY-LM, i.e., a preference pair is explained if any of the explainers explains it. We also fix a maximum of 10 explanation terms for all methods except QUERY-TERMS.

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. Then we show the improvements of adding multiple explainers by an ablation study. Finally, we discuss how our explanations can be used to explain specific preference pairs.

6.1 Effectiveness of Explanations

We present the overall fidelity scores of all methods in Table 2. We observe that MULTIPLEX consistently outperforms all other baselines over all models and datasets. The GREEDY-LM relies on text matching and cannot extract terms that encode semantic similarity, on which the BERT and DPR models heavily rely. This is further justified by the performance of QUERY-TERMS that has access to multiple explainers. It already reaches 81% with only the query terms as an explanation. Interestingly, a feature-attribution-based baseline DEEPLIFT cannot compete with simple QUERY-TERMS. This is mainly because its heuristic aggregation policy results in noisy terms. Therefore, we conclude that focusing on a single document followed by heuristic aggregations is detrimental in explaining ranking models, further advocating a more principled modeling technique like GPC.

We also notice unsurprisingly that *Fidelity*† always has higher scores than *Fidelity* since it's easier to differentiate a pair of prominently different documents. Especially in Clueweb09, there are

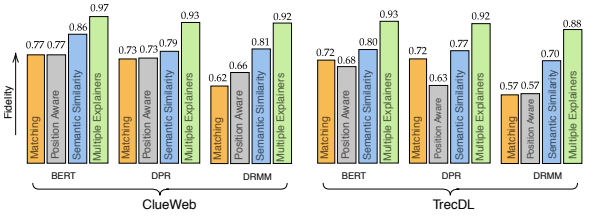


Figure 2: How well can each explainer approximate Φ? *fidelity*† of each single and combined explainer.

many duplicates for which the models generate very close relevance scores, indicating the lower *Fidelity* might be due to noise. From now on we present *Fidelity*† to avoid such noise. Finally, we observe that the explainers on Clueweb09 disagree with each other more often than those on Trec-DL. We attribute this to the fact that documents from Clueweb09 are much lengthier and structurally more complex than passages from Trec-DL.

6.2 The Benefits of Combining Explainers

We also experimented with every single Ψ to extract explanation terms and evaluate the fidelity with each Ψ respectively, to show the benefits of multiple explainers. Figure 2 presents the *Fidelity*† for all models and datasets. We first note that our adapted objective improved the fidelity even when a single explainer was used. Especially when comparing the TERM MATCHING explainer to the greedy baseline across all models and datasets. This proves the effectiveness of our optimization strategy over the greedy algorithm. Besides, Ψ_{emb} in general generated very similar terms as the combined Ψ_{multi}. This suggests that all models pick up the semantic feature over the statistic features. However, there are also occasions when the other two explainers are superior. This is because we use the pre-trained GloVe embeddings as term representation and the case of OOV terms cannot influence Ψ_{emb}. We show some anecdotal examples explaining the BERT model in Table 3 to compare the explanation terms selected by each explainer.

6.3 Explaining Document Preferences

Using MULTIPLEX we can also explain a single preference pair, i.e., why does a model prefer d_i over d_j? We exemplify the explanation terms of query `keyboard reviews` in Figure 3 for all three explainers. Starting from constructing preference scores for each candidate term as described in Section 4.1, and then we present the important terms with significant scores. It suggests the BERT model generally prefers *musical keyboards*, the DPR model has a much stronger preference for the *computer keyboards*.

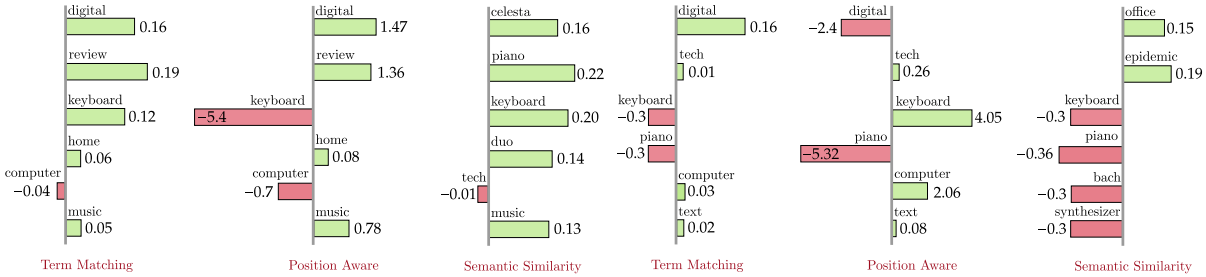
6.4 Rank promotion

We devise an experiment to see if the explanation terms produced by our approach contains any predictive power. This is different from the earlier experiments that measure the fidelity metric that we in some sense implicitly optimize. In this experiment, we simply add the explanation terms to a potentially non-relevant document (the lowest-rank document in our case) and measure the rank improvement as an degree of explanation importance. Figure 4 shows the average rank improvements on addition of the explanation terms using the BERT ranker on the Clueweb09 dataset.

We observe that MULTIPLEX results in the maximum rank increase when explanation terms are added to potentially non-relevant documents. On closer examination we observe that for longer queries like *universal animal cuts reviews*, adding only the query terms can sometimes increase their ranks by over 90 positions. This suggests that BERT rankers heavily rely on exact match for longer queries. However, this is not the case for shorter and ambiguous queries where other baselines show low explanation fidelity and lower rank improvement. On the other hand, MULTIPLEX is able to account for short and ambiguous queries like *voyager* and *titan* in addition to longer queries.

6.5 The curious case of the term "Wikipedia"

One of the primary use cases of explanation methods to find potential problems due to training where the model learns patterns that might be right for the wrong reasons. Towards this, we performed an analysis on the commonly recurring terms in the explanations of the models under investigation. Interestingly, we found the term "wikipedia" appear in many explanations for the DPR model even for unrelated queries. If there are Wikipedia pages (partially) related to a query, the DPR usually chooses them over other types of web pages. We can only hypothesize that the presence of a low IDF term like Wikipedia suggests an artifact of pre-training over a large number of Wikipedia articles. Additionally, to understand how much the Wikipedia keyword contributes to the relevance, we conducted a rank-promotion experiment (as in the previous section) by adding the term "Wikipedia" to the lowest-ranked document for all queries. Unsurpris-



(a) BERT: All explainers identify music-related terms as *pos*. (b) DPR: All explainers identify music-related terms as *neg*.

Figure 3: Comparing explainers for the query: keyboard reviews, 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 intents.

Query	Explainer	Explanation	Fidelity†
adobe indian houses	TEXT 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**
	MULTIPLE EXPLAINERS	pueblo, amarillo, castroville, outhouse, abourezk, pdf	0.91
espn sports	TEXT 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
	MULTIPLE EXPLAINERS	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
	MULTIPLE EXPLAINERS	hp, mini, 2140, 2133, touchpad, overview	0.91

Table 3: Anecdotal examples show each explainer selects terms from different aspects. MULTIPLE EXPLAINERS combines the advantages of all explainers. For ambiguous queries like “adobe Indian houses", TEXT MATCHING and POSITION AWARE focus on popular but ’shallow’ terms indicating “adobe company“. For certain queries like “hp mini 2140", the SEMANTIC SIMILARITY suffers from vocabulary limitation. Position Aware can capture the non-frequent yet important terms based on their position, e.g., the official site for the query “ESPN sports".

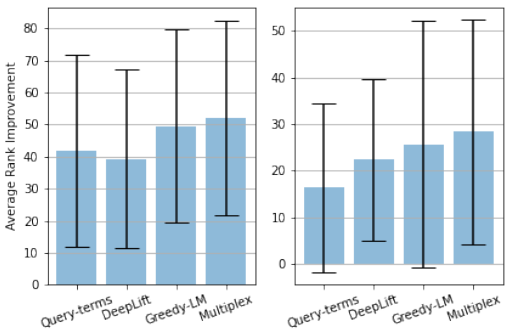


Figure 4: Average rank improvements. Left: on all test queries; Right: on hand-picked ambiguous queries. Note that for each query the document size ≤ 100.

ingly, we observe an average rank improvement of 2, ranging from 0 to 14. On the other hand, when masking “Wikipedia” in the top-ranked document, we observe an average rank drop of 10 positions. We believe that there is a lot of potential for the use of term-based explanation for exposing similar data and model biases for text ranking models.

7 Conclusion and Outlook

This paper proposes a post-hoc model-agnostic framework to explain text ranking models using multiple explainers. MULTIPLEX systematically combines multiple explainers to capture different feature aspects encoded in the ranking decisions. Our solution relies on effectively solving the non-convex generalized preference coverage problem with sparsity constraints. Our extensive experiments show that our method can generate high-fidelity explanations for over-parameterized models like BERT, delivering up to 50% fidelity improvements. We also show anecdotally that the explanations generated by MULTIPLEX help us better understand the underlying model preferences and detect potential biases. For future work, we want to extend our framework to account for n-grams and to make our explanation generation procedure efficient enough to be used during query processing.

References

Nir Ailon, Moses Charikar, and Alantha Newman. 2008. Aggregating inconsistent information: ranking and clustering. *Journal of the ACM (JACM)*, 55(5):23.

Arthur Câmara and Claudia Hauff. 2020. Diagnosing bert with retrieval heuristics. *Advances in Information Retrieval*, 12035:605.

Jaekeol Choi, Jungin Choi, and Wonjong Rhee. 2020. Interpreting neural ranking models using grad-cam. *arXiv preprint arXiv:2005.05768*.

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

Steven Diamond and Stephen Boyd. 2016. CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research*, 17(83):1–5.

Zeon Trevor Fernando, Jaspreet Singh, and Avishek Anand. 2019. 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'19, pages 1005–1008, New York, NY, USA. ACM.

Besnik Fetahu, Katja Markert, and Avishek Anand. 2015. Automated news suggestions for populating wikipedia entity pages. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 323–332.

Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. 2018. A survey of methods for explaining black box models. *ACM computing surveys (CSUR)*, 51(5):93.

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*, pages 55–64. ACM.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*.

Sören Laue, Matthias Mitterreiter, and Joachim Giesen. 2019. Geno–generic optimization for classical machine learning. *Advances in Neural Information Processing Systems (Neurips*.

Ryan McDonald, George Brokos, and Ion Androutsopoulos. 2018. Deep relevance ranking using enhanced document-query interactions. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1849–1860. ACL.

Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage re-ranking with bert. *arXiv preprint arXiv:1901.04085*.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Daniël Rennings, Felipe Moraes, and Claudia Hauff. 2019. An axiomatic approach to diagnosing neural ir models. In *European Conference on Information Retrieval*, pages 489–503. Springer.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1135–1144. ACM.

Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. 2017. Learning important features through propagating activation differences. In *International Conference on Machine Learning*, pages 3145–3153. PMLR.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2013. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*.

Jaspreet Singh and Avishek Anand. 2018. Posthoc interpretability of learning to rank models using secondary training data. *arXiv preprint arXiv:1806.11330*.

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, pages 770–773, New York, NY, USA. ACM.

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.

Jaspreet Singh, Zhenye Wang, Megha Khosla, and Avishek Anand. 2021. Valid explanations for learning to rank models. *International Conference on the Theory of Information Retrieval*.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, pages 3319–3328. PMLR.

Manisha Verma and Debasis Ganguly. 2019. Lirme: Locally interpretable ranking model explanation. In *Proceedings of the 42Nd International ACM SIGIR*.

Michael Völske, Alexander Bondarenko, Maik Fröbe, Matthias Hagen, Benno Stein, Jaspreet Singh, and Avishek Anand. 2021. Towards axiomatic explanations for neural ranking models. *International Conference on the Theory of Information Retrieval*.

A Appendix

Ethical Concerns. Our task is to understand over-parameterized ranking models. This is crucial to the real-world practices, because the complexities of both model and datasets make it extremely challenging to identify potential bugs or biases. Our work does not cause any harm to real-world users. Instead, it can help understand the important terms out of big texts set that affect the model's decisions, thus in the long-term discovering potential biases from the model or datasets.

Ranking Model Details. Unlike Bert and DPR model, we do not truncate documents to meet the 512 token size limit for DRMM model. We fine-tune `bert-base-uncased` by stacking a feed-forward layer on the pooled representation of query and document. For DPR model, we use the `dpr-question_encoder-multiset-base` and `dpr-ctx_encoder-multiset-base` to encode the query and document respectively. The relevance score is computed by the `cosine` similarity of the pooled representations. All three models are trained in pair-wise with five negative samples for each positive instance. Adam optimizer with 3e-5 learning rate and 1000 warm-up steps are applied.

Matrix Building Details. We pre-select 1000 candidate terms by their tf-idf scores. Then for each candidate, we mask all it's appearances in the document and compute the prediction difference by the ranking model. The term results in higher prediction difference is regarded to be more important to the query-document relevance. By doing this, we finally maintain 200 terms out of the 1000 candidates. We also experimented with [100, 300, 500] terms and without masking, all resulting to lower or similar fidelity. For preference pairs, we randomly choose 500 pairs (the random seed is 100.) with $\geq g$ score difference. For Bert model, we set g to [2.0, 4.0] on Clueweb09 and Trec-DL respectively. For DRMM and DPR, we choose [0.01, 0.01] and [0.05, 0.05]. With this setup, we ensure the sampled preference pairs have prominent differences. Consequently, increasing the size of the pairs does not deliver higher fidelity.

Computation Costs. Our experiments are conducted on a GTX 1080Ti GPU with 128G memory. We do not consider the training costs for ranking models. The expenses of our method come from matrix construction and matrix optimization. On Clueweb09, the former procedure takes on average around 30 minutes, and the latter takes around 30 seconds.

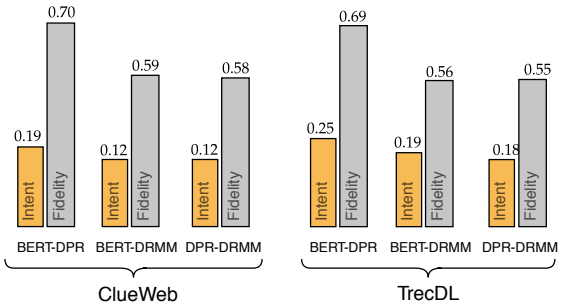


Figure 5: Model correlation: Jaccard similarity of explanation terms and ranking correlation between each 2 models.

Explanation Differences across Models. We also compare the explanation terms for each single query across ranking models. Figure 5 summarizes the Jaccard similarity of explanation terms, along with the ranking correlation measured by *Fidelity*, between each two models. It shows Bert and DPR generate more relevant rankings, resulting in higher overlaps between explanation terms compared to DRMM model. Table 4 and Table 5 present more explanation examples of DRMM and DPR.

Query	Explanation Terms
adobe indian houses	indian, manure, lobby, frenzy, coyote, strain
espn sports	directv, fantasy, nba, brad, golf, berman, walt
hp mini 2140	amazon, handwriting, mini, 2140, qvga, 3.5mm
bobcat	growl, chalkboard, jona, mover, oddity, handy, bobcat

Table 4: Explanation Terms of DRMM by MULTIPLEX.

Query	Explanation Terms
adobe indian houses	oregon, adirondack, style, pueblo, palace, battle
espn sports	espn.com, disney, walt, directv, wikipedia, network, sports
hp mini 2140	nokia, hp, 2140, aircraft, 2133, coating
bobcat	desert, carnivorous, lynx, baffling, prickly, hooded, manatee

Table 5: Explanation Terms of DPR by MULTIPLEX.