RankingSHAP – Listwise Feature Attribution Explanations for Ranking Models

Maria Heuss
University of Amsterdam, Amsterdam
The Netherlands
m.c.heuss@uva.nl

Maarten de Rijke
University of Amsterdam, Amsterdam
The Netherlands
m.derijke@uva.nl

Avishek Anand
Delft Institute of Technology, Delft,
The Netherlands, Netherlands
Avishek.Anand@tudelft.nl

ABSTRACT

Feature attributions are a commonly used explanation type, when we want to posthoc explain the prediction of a trained model. Yet, they are not very well explored in IR. Importantly, feature attribution has rarely been rigorously defined, beyond attributing the *most important* feature the highest value. What it means for a feature to be more important than others is often left vague. Consequently, most approaches focus on just selecting the most important features and under utilize or even ignore the relative importance within features. In this work, we rigorously define the notion of feature attribution for ranking models, and list essential properties that a valid attribution should have. We then propose **RankingSHAP** as a concrete instantiation of a list-wise ranking attribution method. Contrary to current explanation evaluation schemes that focus on selections, we propose two novel evaluation paradigms for evaluating attributions over learning-to-rank models. We evaluate RankingSHAP for commonly used learning-to-rank datasets to showcase the more nuanced use of an attribution method while highlighting the limitations of selection-based explanations. In a simulated experiment we design an interpretable model to demonstrate how list-wise ranking attributes can be used to investigate model decisions and evaluate the explanations qualitatively. Because of the contrastive nature of the ranking task, our understanding of ranking model decisions can substantially benefit from feature attribution explanations like RankingSHAP.

KEYWORDS

Explainable ranking systems, Explainability, Explanation evaluation, Feature attribution

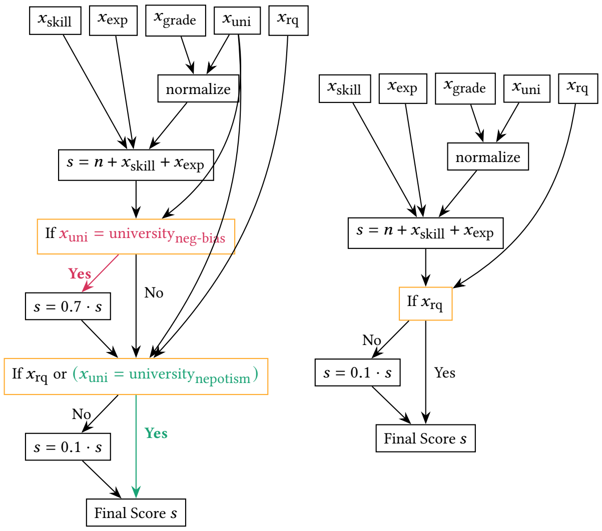
1 INTRODUCTION

Feature attribution explanations are a posthoc family of explainability approaches that assign scores to each feature, indicating their relative contribution to a model decision. Feature attributions are among the most commonly used explanation types, when we want to posthoc explain the prediction of a trained model in general machine learning [16, 23, 29] and also in document ranking tasks in information retrieval [2, 21, 37, 38]. However, feature attribution has rarely been rigorously defined, beyond attributing the *most important* feature the highest value. What it means for a feature to be more important than others is often left vague. Another commonly used type of posthoc explainability approach, *feature selection explanations*, defines explanation as a subset of the input features, usually containing the most important features for a decision. Prior work often indirectly treats feature *attribution* explanations as feature *selection* explanations by only considering the attribution scores to decide which subset of features to select, mostly ignoring the relative importance within features. We argue that, especially for ranking tasks, attributions are more nuanced than the general selection or rejection of features into an explanation.

Typical ML tasks are pointwise prediction tasks, i.e., they explain a single classification or regression decision. However, rankings can be considered as aggregations of multiple pointwise, or pairwise decisions or even a single listwise decision. Consequently, ranking explanations can be cast as explanations of different aspects of the ranking such as pairwise preferences (*pairwise explanations*) or top-k subrankings (*listwise explanations*) apart from the pointwise explanation of why a query-document pair is considered relevant by the model. There has only been limited work on pairwise [24] and listwise explanations [21, 37, 47], none of which focuses on the relative importance of features or feature interactions – they essentially do not distinguish between attribution and selection. Listwise feature attribution for ranking models has been unexplored, except for [6], and has never been rigorously defined.

We argue that because of the contrastive nature of the ranking task, listwise feature attribution can be a powerful and flexible tool for gaining insight into model decisions. The focus on different aspects of model decisions also allows us to compare the same aspects among different queries as well as different aspects of the same ranking decision with each other, giving us a nuanced understanding of the model.

Case study/Use case – Talent Search. In the context of talent search, systems use learning-to-rank to produce a candidate ranking based on features like academic performance, experience, skills, and private attributes like gender, ethnicity, and university attended. The inclusion of certain attributes (like gender, ethnicity, or even university attended) in the decision-making process is debatable, since biases from past decisions can be reflected in the learned model, and best left to a human. Yet, sometimes the use of such attributes is necessary for the model to achieve reasonable performance. Consider the two models in Fig. 1. Both rely on the same set of features, including the skills, experience and graduation grade of the candidate, as well as the university they went to and a feature measuring whether the candidate meets the minimum job requirements. While the right model (Fig. 1b) uses the university in a reasonable way, namely by normalizing the grades that might come from institutions with different grading schemes, the left model (Fig. 1a) has learned to discriminate against candidates from x_{uni} = university$_{\text{neg-bias}}$ and to advantage candidates from x_{uni} = university$_{\text{nepotism}}$. Through the use of explanations we would like to differentiate between two such models with similar performance to identify which model contains less bias and should hence be trusted. Here, feature selection is often not nuanced enough to provide sufficient insights into the model decision



(a) Biased model (b) Unbiased model

Figure 1: Flow chart of a biased and an unbiased model for the talent search task. With the help of explanations we would like to be able to differentiate between those and pick the unbiased ranker for use in production.

as it likely selects the same two features, x_{uni} and x_{rq} as the most important feature for both models, hence we should use feature attribution instead to identify that the biased model puts more weight on the university than on the candidate meeting the job requirements. We will revisit this case study in Section 5, where we will use a simulated experiment setup to evaluate different explanation approaches as well as to demonstrate how listwise feature attribution can be used in practice.

Approach and Contribution. Feature attribution is conventionally understood as the quantification of a feature's marginal contribution to the prediction score. However, this definition encounters complexities in the context of ranking models, where the output is not a singular score but a series of decisions regarding the relative ordering of documents. It necessitates the identification of a specific model decision to focus on and the translation of the ranked output into a singular metric that can capture the impact of input variations on this decision. The challenge lies in the absence of a universal metric for this purpose. Depending on the focal aspect of the model, one might consider various metrics, such as the shift in a document's rank or the number of permutations within the top-k results. These diverse approaches underscore the need for a nuanced definition of feature attribution in ranking models.

To address this gap, we introduce a novel methodology, RankingSHAP, for listwise feature attribution. Unlike prior methodologies that evaluate documents in isolation, thus disregarding the context provided by the ranked list, RankingSHAP preserves this integral aspect. Evaluation of posthoc attribution methods is a challenging problem. We rigorously assess RankingSHAP, both qualitatively and quantitatively, using established learning-to-rank (LtR) benchmark datasets. Our study demonstrates the application of feature attributions in interpreting the outputs of ranking models, offering a deeper insight into their decision-making processes.

- We propose and define the notion of listwise feature attribution and formalize the properties that listwise attribution explanations should have.
- We provide a novel and concrete instantiation of our feature attribution framework called RankingSHAP.
- We propose multiple valid evaluation schemes to evaluate ranking feature attributions, and perform extensive experiments to showcase the performance of RankingSHAP.

2 RELATED WORK

Explainable information retrieval [2, 3] has mostly dealt with constructing models that are explainable by design [18, 48] or with approaches that can posthoc (after model training) explain models [36, 37, 44]. Posthoc approaches for explaining rankers can operate at the global level (model level) or at the local level (per-query). Global explainability approaches have been used to diagnose ad-hoc neural text rankers with well-understood axioms of text ranking [4, 28, 45] or to probe pre-trained transformer-based ranking models for ranking abilities [46]. We focus on posthoc feature and local attributions.

Feature Selection and Attribution for Ranking Models. The earliest works of interpreting ranking models were adapted from the popular paradigms of black-box methods like [20, 30] or white-box methods [20, 33, 34, 42] for explaining relevance of a query-document pair. Singh and Anand [36], Verma and Ganguly [43] modify LIME [30] to generate terms as the explanation for a trained black-box ranker. Choi et al. [5], Fernando et al. [8] applied gradient-based feature attribution methods [20, 42] to interpret the relevance score of a document given a query. Contrary to posthoc feature attribution approaches, instance-wise feature selection or local feature selection [11, 17, 18] approaches select a subset of features without distinction among the importance between features. However, most of the work on instance-wise feature selection for rankings [11, 18] is not posthoc, and has been performed on text features and not on learning-to-rank data. In this work, we work on posthoc approaches for attribution and not selection.

Listwise Explanations for Ranking Models. Typical ML tasks are pointwise prediction tasks, i.e., they explain a single classification or regression decision. In rankings, even for a single query, we also have to deal with pairwise and listwise explanations – that is explaining preference pairs or ordered lists, which might be constructed by an aggregation of decisions instead. However, there has been limited work on pairwise [24] and listwise explanations [21, 37, 47]. LiEGe [47] tackles the task as text generation. In contrast [37] and [21] use simple rankers to approximate the original ranking of a complex black-box model by expanding the query terms by solving a combinatorial optimization problem called *the preference coverage problem*. The probably closest work to ours on RankLIME [6] approaches the problem with the local surrogate approach called LIME, which the authors adapt for ranking models. Again most of the approaches focus on text features and are not directly applicable to learning-to-rank models.

Explainability in Learning-to-Rank. In principle local or instance-wise feature selection approaches can be applied to learning-to-rank [9, 10, 26]. Among the feature-selection approaches, filter methods are model-agnostic [9] while wrapper methods are designed for a particular type of model [10]. In the context of ranking, [26, 35] produce local feature selections. As one of the closest to our work, [39] proposes the notions of validity and completeness based on the information contained in the explanation. While these notions are useful in both conception and evaluation of explanations, they still view the explanation as a selection of features. Feature selection methods, however, lack the capability to differentiate between features of varying importance, thereby avoiding a nuanced understanding of which features are substantially more critical in the decision-making process. Hence, in this work we focus on feature attributions.

Shapley Values and SHAP. Shapley values, which originated from game theory where they define the marginal contribution that a player has to a game [32], have become a popular tool in the field of explainable AI. The development of different approximation techniques has enabled the efficient use to model decisions for AI models [40, 41]. SHAP (SHapley Additive exPlanations) [20], is an efficient approximation technique that determines the expected marginal contribution of a feature to any feature set not containing the feature. A good overview of the method itself as well as recent progress on it can be found in [23]. We base our approach on this prior work, but extend it for use with ranking models. Parallel to our work, Pliatsika et al. [25] propose a similar framework for generating shapley value based explanations for rankings and preferences.

3 DEFINING FEATURE ATTRIBUTION FOR POINTWISE MODELS

While early work on local feature explanations introduce the concept of feature attribution [40], recent work in the field has often worked with an underspecified understanding of what it means for a feature to be *important*. Despite some attempts to formalize feature attribution [1], these formalizations have not found adoption in the broader explainability literature. This has led to ambiguity in the evaluation of faithfulness of feature attributes. Another result of this under-specification is a confusion on why different instance-wise feature explanation methods, which implicitly use different notions of importance, disagree [14]. To ensure clarity and enable a proper evaluation of the faithfulness of our explanations, we define feature attribution in a rigorous manner. In this section we define the *pointwise feature attribution* for black-box models with one dimensional model output such as a pointwise ranking model

$$\tilde{R} : \mathcal{D} \to \mathbb{R}, x_{q,j} \mapsto s_{q,j}, \tag{1}$$

that predicts the ranking scores $s_{q,j} \in \mathbb{R}$, representing probability of relevance, for the feature vectors of each document-query pair, $x_{q,j} \in \mathcal{D}_q \subset \mathcal{D}$. We consider feature attribution explanations that assign each feature i an attribution value $\phi_i(x, \tilde{R})$, directly reflecting the importance of the feature to the total model decision. Feature attribution explanations can hence be understood as dictionaries $\{i \mapsto \phi_i(x, f)\}_{i=1,\dots,n}$ containing exactly one attribution value per feature. We define desirable properties of feature attribution explanations in Section 3.1. They rely on the concept of pointwise feature importance, which we will define in Section 3.2.

3.1 Desirable Properties of Listwise Feature Attribution Explanations

We define the following properties that a well-defined, instance-wise definition of feature attributions should have:

(1) **Feature interactions:** We should be able to find specific combinations of feature values in the input that collectively cause the model to predict a high score.
(2) **Relative importance of attribution values:** Features with higher importance for the prediction should have higher attribution values.
(3) **List-wise attribution:** The attribution values should take into account the context of other candidate feature vectors within the same ranked list.

We have not defined yet what we mean by "importance" of a feature. The next section defines feature attribution and discusses what we understand by *important* features.

3.2 Defining Pointwise Feature Attribution

Based on the first two desired properties listed above, we define *pointwise feature importance* by building on the concept of marginal contributions.[1] Our definition is strongly inspired by the definition of SHAP from [20]. Feature attribution will then be defined as the approximation of this feature importance, with a first example for it being SHAP. In Section 4 we extend this definition of feature attribution to also include the third property, i.e. to *listwise feature attribution* and define RankingSHAP as a way to approximate feature attribution for these kind of models.

We define the *importance* of a feature i in terms of marginal contributions. Let $n = dim(\mathcal{D})$ be the dimension of the input space and let a coalition be a subset $S \subset \{1, \dots, n\} \setminus i$ of the input features that do not contain i. To measure the marginal contribution of feature i to coalition S, we compare the model output when the model is shown only features in S resp. features in $S \cup \{i\}$. Because we can not simply erase features, we instead model this erasure by masking all other features with samples from a set of feature-vectors $B \subset \mathcal{D}$, called *background data*, that ideally presents a good summary of the underlying data distribution. For the masking we use different templates $t \in \{0, 1\}^d$, which define the presence ($t_i = 1$) or absence ($t_i = 0$) of a feature in the mask as well as data-points from the background data $b \in \mathcal{D}$ and define $m_{t,b} : \mathcal{D} \to \mathcal{D}$ as

$$m_{t,b}(x)_i = \begin{cases} x_i, & \text{if } t_i = 0 \\ b_i, & \text{if } t_i = 1. \end{cases} \tag{2}$$

So the marginal contribution of feature i to coalition S for background vector b is given by:

$$f(m_{S\cup\{i\},b}(x)) - f(m_{S,b}(x)). \tag{3}$$

We define the **pointwise importance** of feature i to the model decision of $\tilde{R}$ at input x as the expected marginal contribution of feature i to all possible coalitions of features:

$$\phi_i^{imp}(x, \tilde{R}) = \sum_{S\subset\{1,\dots n\}\setminus i} w_S \cdot \mathbb{E}_{b\sim B}[\tilde{R}(m_{S\cup\{i\},b}(x)) - \tilde{R}(m_{S,b}(x))], \tag{4}$$

[1]For a detailed discussion of the concept of marginal contribution and the intuition behind it, we refer the reader to [23].

with weighting factor $w_S = \frac{1}{n!}|S|!(n - |S| - 1)!$ and uniform sampling from B.

Then **pointwise feature attribution** ϕ_i can be defined as an approximation of the actual feature importance, ϕ_i^{imp}.

4 FEATURE ATTRIBUTION FOR RANKING MODELS

For many ML tasks, SHapley Additive exPlanations (SHAP) [20] have proven effective in approximating feature attribution values for individual model decisions like regression scores or classification class probabilities. What makes this method difficult to use for ranking models is that such models output a ranked list rather than a single score. Within this ranked list, different decisions are made on the order of the individual documents. SHAP, or how we also call it, *pointwise* SHAP, has only been defined for a single one-dimensional model output. While we can use it to explain the model score of an individual document, it is oblivious to the context of the other documents in the list. In this work we extend on this definition of SHAP to an approach that caters for listwise ranking decisions, called RankingSHAP.

So now, instead of looking at pointwise ranking models, as we did in Section 3.2, we consider a listwise ranking model

$$R : \{\mathcal{D}_q\}_q \to S_d, \{x_{q,j}\}_j \mapsto \pi_q \tag{5}$$

that maps a set of candidate feature vectors, $\mathcal{D}_q = \{x_{q,j}\}_j$ to some permutation matrix π_q representing the ranked list. In practice, usually a model is trained with a listwise loss to predict the ranking score for each document, according to which the documents are being ranked:

$$\tilde{R} : \mathcal{D} \to \mathbb{R}, x_{q,i} \mapsto s_{q,i}, \tag{6}$$

with $R = \text{ranked} \circ \prod_{|\mathcal{D}_q|} \tilde{R}$.

We start this section by discussing the two components, *listwise masking* and *listwise explanation objectives*, that allow us to define listwise feature attribution for ranking models. Then we define our method RankingSHAP for the approximation of listwise attribution values in Section 4.2. Finally, we go into more detail on the listwise explanation objectives, with some examples in Section 4.3.

4.1 Feature Importance for Ranking Models

Our definition of feature attribution/feature importance for ranking models consists of two parts: (i) First, we need to define how masking applies to a each of the documents in the ranking $\mathcal{D}_q$ for query q. (ii) Then, we need to be able to measure how much the model decision is impacted by a change in the input. Furthermore, this change in model decision needs to be quantified with a single number.

Masking the Inputs of a Ranking Model. We enforce a listwise mask that applies the same mask $m_{t,b}$ to all documents $\{x_j\}$ in the ranking, i.e., $m_{t,b}(\mathcal{D}_q) = \prod_{|\mathcal{D}_q|} m_{t,b}(x_{q,i})$. By masking the feature vector x_j of each document with the same mask $m_{t,b}$, we disregard the impact of the masked features to the ranking decision. This enables us to identify the partial contributions of only the non-masked features towards the document ordering.

Reducing the Model Prediction to a Single Prediction Value. Since feature attribution is defined by the expected change of the predicted score, we need to reduce the decisions/predictions of the ranking model to a single value that reflects how much the model prediction changes for a perturbed input sample.

One example for such a function could be a rank similarity coefficient like Kendall's tau, which is commonly used in the interpretability literature to measure the rank correlation [21, 37, 39]. By comparing how much the relative order of the document has changed, we can measure how far the prediction deviates from the optimal order of the list π_q as predicted by the model:

$$g_q(\tilde{\pi}) = \tau(\pi_q, \tilde{\pi}), \tag{7}$$

This way, features that are important for the overall order of the documents can be identified. Section 4.3 provides further examples. For any such *listwise objective* g_q we can define the feature importance with respect to this specific objective through the concatenation with the original ranking model, $g_q \circ R$.

We defined both how to "remove" a feature from the model input through masking as well as how to measure the impact of this masking on the model prediction with a single value, which lets us determine the **listwise feature importance** with the help of Eq. 4. We define **listwise feature attribution** as an approximation of listwise feature importance.

4.2 Estimating Listwise Feature Attribution with RankingSHAP

With the definition of feature attribution for ranking models we can now define our solution – RankingSHAP. The definition depends on the choice of listwise objective g. The goal of RankingSHAP is to explain which features are important to achieve a ranked list similar to the original one. What *similar* means can be defined flexibly. That way, RankingSHAP can explain different aspects of the ranked list, making it a contrastive and flexible approach for generating instance-wise explanations for rankers.

The pointwise feature attribution method SHAP aims to approximate the marginal contribution of each feature to the model prediction at a given input. In other words, SHAP aims to approximate pointwise feature importance exactly in the way we defined it in Section 3.2. The way we define listwise feature attribution allows us to incorporate SHAP for the approximation of the attribution values, enabling us to draw from prior work in the field [20, 41]. Here, we show how to combine SHAP with the definition of feature attribution for ranking models to obtain RankingSHAP.

SHAP is based on sampling both coalitions (i.e., templates for creating masks) as well as background data to generate masked perturbations (see Eq. 2) of the input to approximate the marginal contribution of a feature to any coalition. Given a sampled mask $m_{t,b}$, we illustrate how RankingSHAP adjusts the model prediction to be used with SHAP in Algorithm 1. Given a mask $m_{t,b}$, we loop over all documents $x_j \in \mathcal{D}_q$ (line 1–2) and perturb the document features with the mask to get $\tilde{x}_j = m_{t,b}(x_j)$. Then we rank the perturbed feature vectors with the ranking model $\pi = R(\{\tilde{x}_j\}_j)$ in line 3. Finally, we apply the listwise explanation objective $v = g(\pi)$ and output v, measuring the change in the output (lines 4 and 5).

Our approach allows us to use existing implementations of SHAP without adjustments for the ranking use case, which we make use of for our implementation of RankingSHAP.

The Defining Axioms of SHAP and RankingSHAP. Prior work has shown that SHAP is uniquely defined by 4 axioms (Efficiency,

Algorithm 1 Adjusted model prediction (used in combination with SHAP)

Require: ranking-model R, feature-vectors $\mathcal{D}_q$ for query q, listwise explanation objective g,
Input: masking function $m_{t,b}$
1: **for all** $x_j \in \mathcal{D}_q$ **do**
2: $\tilde{x}_j \leftarrow m_{t,b}(x_j)$
3: $\pi \leftarrow R(\{\tilde{x}_j\}_j)$
4: $v \leftarrow g(\pi)$
5: **return** v

Symmetry, Dummy, and Additivity) [20]. With our definition of RankingSHAP, defining a new model with the concatenated function $g_q \circ R$, we can see that RankingSHAP directly inherits this property from SHAP.

On the Desirable Properties of Feature Attribution Explanation. We take one step back and evaluate RankingSHAP based on the desirable properties of listwise feature attribution from Section 3.1. Since RankingSHAP uses coalitions of features to approximate the attribution values, the first property of capturing feature interactions is clearly met. This is in contrast to greedy explanation approaches that we will discuss in Section 6. The third property is also clearly met by RankingSHAP as opposed to a pointwise method like SHAP. The second property on the other hand, we will have to evaluate based on empirical experiments, which we will do in Sections 5 and 6.

4.3 Listwise Explanation Objectives

We give a few examples for listwise explanation objectives to give the reader an idea of what kind of contrastive explanations RankingSHAP can produce.

Explaining the Top-*k* of a Ranked List. Instead of focusing on the whole ranked list, one can of course focus on a top-k to identify features that were especially important for those documents to be ranked on top of the list or that were important for the relative ordering of this top. Practically, we can adjust a similarity coefficient like Kendall's tau to only consider a subset of the document pairs containing the top-k documents.

Explaining the Position of a Singular Document. Similar to explaining the top-k we can explain the position of a single document in the ranked list. Here, the listwise explanation objective could, for example, measure the rank distance of the document within the perturbed example to the position in the original rank.

Explaining the Position of a Group of Documents. RankingSHAP also enables us to compare the ranking decisions for two different groups of documents. Similar to the prior points, the relative ordering or even the absolute distance of members of the different groups could be considered. More intended as a suggestion for future work, one could also try to explain how fair the model considers itself or investigate whether we can actually identify biases with listwise feature attribution.

5 TESTING CORRECTNESS WITH A SYNTHETIC EXAMPLE

To demonstrate how RankingSHAP can be used and to evaluate feature attributes that different explanation approaches generate, we create a synthetic example, for which we return to the case study for talent search from the introduction. We design an interpretable model, to be able to estimate the importance of the features for different model decisions. In the following sections, we start by defining the features that the model is using to rank the candidates and give an intuition behind the model in Section 5.1. After describing the experimental setup in Section 5.2 we go through some queries that aim to model different kinds of model decisions in Section 5.3. We use those queries to demonstrate how to use listwise feature attribution in practice as well as to qualitatively evaluate three different feature explanation approaches. We conclude this section with a detailed discussion in Section 5.4.

Table 1: Candidate evaluation criteria.

Feature	Description
Job requirements	Binary value $x_{\text{rq}} \in \{\text{True}, \text{False}\}$ measuring whether the candidate meets the minimum requirements of the job.
Prior experience	Measuring relevant work experience that the candidate comes with. On a scale $x_{\text{exp}} \in [0, 1]$, 1 indicates a lot of relevant experience and 0 indicates no experience at all.
Skills	Measuring how well the skills that a candidate has fit the job description. On a scale $x_{\text{skill}} \in [0, 1]$, 1 indicates a very good fit, and 0 indicates no relevant skills.
University	The university that the candidate obtained their degree at x_{uni}.
Grades	The mean graduation grade of the candidate, x_{grade}. The range depends on the university the candidate graduated from.

5.1 Model Design

We design a model using the features described in Table 1 to rank candidates for different query scenarios, each of which requires an academic degree. We aim to imitate biases that trained models tend to learn from the data. We define the model to give an advantage to a privileged group of people coming from $\textbf{university}_{\textbf{nepotism}}$. In contrast, candidates from $\textbf{university}_{\textbf{neg-bias}}$ are disadvantaged by the model. A flowchart can be found in Fig. 1a in Section 1. The model determines the ranking score as follows:

- Since we consider candidates from different institutions with different grading schemes, we determine the "normalized grade" $\textbf{norm}(x_{\text{grade}}, x_{\text{uni}})$, i.e., the grade scaled in such a way that the minimum possible grade gets a value of 0, while the maximum possible grade gets a value of 1.
- Calculate the sum of x_{skill} and x_{exp} and $\text{norm}(x_{\text{grade}}, x_{\text{uni}})$.
- For candidates from $\textbf{university}_{\textbf{neg-bias}}$, the model has picked up a negative bias from the data. For these candidates, the score gets multiplied by a factor of 0.7.
- If the candidate does not meet the job requirements, the score gets multiplied by a value of 0.1, basically putting them on the bottom of the list. The only exception of this are candidates that graduated from $\textbf{university}_{\textbf{nepotism}}$, where the model has learned that this feature is not required for obtaining an interview.

Finally, the candidates are ranked according to the ranking scores, with the candidate with the highest score being at the top.

In the remainder of this section, we show how to investigate different queries with RankingSHAP to identify the defined biases and compare their attribution values to those generated by other explanation approaches.

5.2 Experimental Setup

We compare RankingSHAP with the **Greedy** feature selection approach from [38] that iteratively adds features to an originally empty set that have the biggest marginal contribution to the current feature set until either the marginal contribution of each remaining feature is negative or an explanation size of 2 is reached. Furthermore, we compare to the pointwise SHAP explainer, **PointwiseSHAP**. We do not include RankLIME in this experiment, since it cannot easily handle categorical features. The generated attribution values will be presented in Fig. 2, where we will visualize the greedy feature selection explanations as bars of length 1.

The exact feature values for the different candidates, as well as the list of candidates that we consider for each query, can be found in Appendix A. For the background data, we create a set of 100 candidates by sampling values from the uniform distributions over the possible feature values as defined in Section 5.1. We use Kendall's tau as rank similarity coefficient, which focuses on the order of the ranked list as a whole.

5.3 Query Scenarios

Next, we define scenarios that help us to demonstrate how feature attribution can be used for contrastive ranking explanations as well as to evaluate the explanations. We will go through 5 query scenarios, starting by discussing the setup and constellation of candidates, then defining some aspects of feature importance $imp_{feature}$ that we are looking for in the attribution values. Finally, we evaluate the different explanation approaches on these aspects.

5.3.1 Average query. **Description.** The *average query* considers candidates that all come from universities with the same grading scheme, some with and some without meeting the requirements, nobody from $\text{university}_{\text{neg-bias}}$ or from $\text{university}_{\text{nepotism}}$.

Importance. Since neither of the exceptions for candidates from specific universities applies and the grades are all within the same grading scheme, following the model definition, we estimate imp_{rq} to have the highest importance, since by hiding this feature, the ranked list might change a lot, the candidates not meeting the requirements but with high values for the other features suddenly being ranked above candidates meeting the requirements. Furthermore, we expect imp_{uni} to take a not too big but positive value, since a change of university for all candidates causes ambiguity for the evaluation of the grade.

Evaluation of the feature attributes. In Fig. 2, the first bar plot (a) shows the rough expectations on the feature importance (red bars) as well as the attribution values/the selected features (visualized by bars with length 1), that the different approaches output. Both RankingSHAP and PointwiseSHAP identify x_{rq} as the most important feature and assign a positive value to x_{uni}. The greedy feature selection approach on the other hand selects none of these features.

5.3.2 Nepotism query. **Description.** For this query one additional candidate from $\text{university}_{\text{nepotism}}$ with good records for x_{skill}, x_{exp} and x_{grade} is considered, but lacking some of the job requirements.

Importance. As we know, the model has picked up on a bias in the data, advantaging candidates coming from $\text{university}_{\text{nepotism}}$, which coincidentally or not is the same university that some people that made past hiring decisions graduated from. Hence, for this query we estimate imp_{rq} to take a smaller value, and imp_{uni} to take a higher importance value.

Evaluation of the feature attributes. In Fig. 2(b) we see that all approaches correctly pick up on the bias towards $\text{university}_{\text{nepotism}}$ by assigning a high value to x_{uni}, while assigning a low value to/not selecting the usually important x_{req}.

5.3.3 Qualified query. **Description.** This query is similar to the average query, but with only candidates that meet the requirements. Hence, for this query the model can ignore the usually important feature x_{rq} without introducing a bias.

Importance. We estimate the feature importance from the model similarly as before, but with imp_{rq} taking a value close to 0 since for the order of these candidates, x_{rq} is irrelevant.

Evaluation of the feature attributes. Fig. 2(c) shows that Greedy and RankingSHAP correctly assign a value close or equal to 0 to the x_{rq}. PointwiseSHAP is not able to identify that the feature that is most important for attaining a high ranking score for each individual document, x_{rq}, is not important for this specific query.

5.3.4 International query. **Description.** This query considers candidates from universities with different grading schemes, most meeting the job requirements, none from $\text{university}_{\text{nepotism}}$ or $\text{university}_{\text{neg-bias}}$.

Importance. For this query we estimate imp_{uni} to take a higher value than for the average query. Since candidates from universities with different grading schemes are compared, knowing which university the candidate went to is important for the interpretation of the grades.

Evaluation of the feature attributes. By comparing Fig. 2(d), with the plot for the average query (a) we see that RankingSHAP is the only approach assigning x_{uni} a higher value than for the average query.

5.3.5 Negative bias query. **Description.** Again a similar set-up as *average query*, with an additional candidate $\text{university}_{\text{neg-bias}}$ with the best overall score with respect to x_{skill}, x_{exp} and x_{grade}. Recall that the model has learned a negative bias towards candidates of this university.

Importance. Compared to the average query, we estimate imp_{uni} to take a higher importance value, because of the bias.

Evaluation of the feature attributes. In Fig. 2(e), only RankingSHAP is able to identify the negative bias towards one candidate, by correctly assigning a higher attribution value to x_{uni} than for the average query.

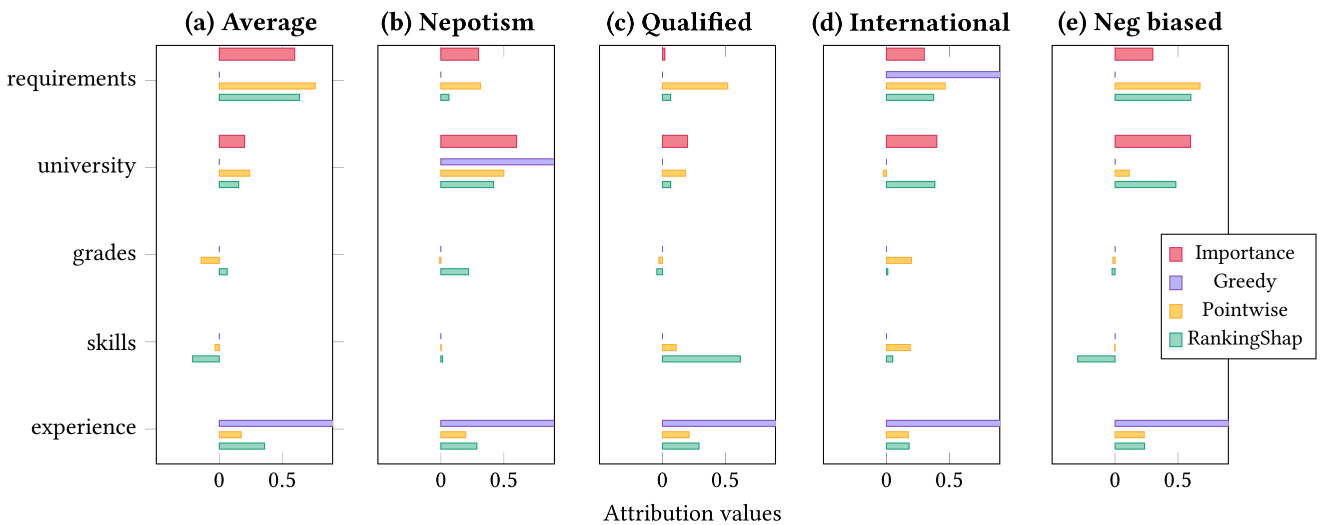


Figure 2: Feature attribution values for different query scenarios from Section 5.2

5.4 Discussion

The Contrastive Use of Feature Attribution. We define aspects of the estimated feature importance in a contrastive way, comparing them to other queries. As prior work [23] has argued, attribution values can be hard to interpret in isolation; putting them in context of other model decisions can help an interpretation of the attribution values and lead to a better understanding of the model. In practice, contrastive comparisons of different explanations – either different aspects of the same ranking, or explanations of different queries – can be useful in understanding ranking explanations. This makes feature attribution especially effective for ranking models: since a model decision is a complex interplay of different model decisions about the relative ordering of the documents, by contrasting different aspects of the decision, we can find nuances that led to a certain model decision.

Using RankingSHAP to Identify Biases. Our synthetic example highlights one way to use feature attribution methods. By comparing attribution values of different queries we can identify queries where an usually important feature, such as x_{rq}, contributes little to the final order of the documents.

We can use RankingSHAP to identify queries, e.g., the **qualified query** and **nepotism query**, for which the feature x_{rq} is not considered important by the model. Upon closer investigation, we find that for the **qualified query**, the rest of the feature attributes behave as expected and we can even identify the reason for the low attribution score, viz. that all candidates meet the requirements. A closer look at the attribution values of the **nepotism query** reveals the strong reliance on the university as a feature, indicating a bias towards candidates from a specific university in that query. A similar approach can be used to differentiate a biased model decision for the **biased query**; for the **international query** the increased importance of x_{uni} can be explained with the reliance of the total prediction on the grading scheme.

Pointwise vs. Listwise Ranking Explanations. From our synthetic example we see that simply using a pointwise explanation approach to explain listwise ranking decisions fails to consider interactions between the feature values of different documents. Features that are important for a high ranking score are assigned a high attribution value, independent of whether they are important for the relative ordering of the list.

Selection is not Attribution. While feature selection can be a useful tool for understanding ranking models, in some situations more nuanced explanations are necessary to interpret model decisions. Next to the overall bad agreement with the feature selection and the importance in this example, even if the selection approach would correctly identify the most important features, for all queries but the qualified query, this feature selection would be the same, making it impossible to identify biases in the model the way we can when considering attribution values.

6 QUANTITATIVE FEATURE ATTRIBUTION EVALUATING

The quantitative evaluation of explanations is a difficult task [19]. In contrast to usual machine learning tasks, where labeled data to benchmark different models can be used for the evaluation, for explanations there is nothing like a *ground truth explanation.* Prior work on evaluating feature attribution often defaults to evaluating the feature selection of the top-k features instead [31]. Especially for the ranking task, selection does not equal attribution. We introduce a evaluation framework that aims to directly evaluate the second property from Section 3.1, i.e., how well the relative importance of the feature importance values is captured by the attribution explanations. We pose the following two research questions: (**RQ1**) Does RankingSHAP order feature attributes more accurately than existing approaches? (**RQ2**) Does RankingSHAP accurately approximate feature importance? We describe our experimental setup in Section 6.1. In Section 6.2 we describe how we estimate feature importance for use as *ground truth feature attributes.* In Section 6.3 we define the evaluation framework. Section 6.4 presents the experimental results. We close with a discussion in Section 6.5.

6.1 Experimental Setup

6.1.1 Datasets. Following [38] we consider two datasets from LETOR4.0 [27]. MQ2008 consists of 800 queries with pre-computed query-document feature vectors of dimension 46. The MSLR data set consists of 10k queries with query-document feature vectors of dimension 136. For both, we use the train-val-test split of fold1 and evaluate the explanations on the test data.

6.1.2 Ranking model. We use the LightGBM [12] to train a listwise ranker with LambdaRank, using NDCG as metric.

6.1.3 Baselines. We consider the following baselines:
Random: Random feature attribution, normalized.
PointwiseSHAP: Previously used as a baseline in [38], we take the mean over the pointwise SHAP values of the top-5 documents.
PointwiseLIME: The mean over the pointwise attribution values generated with LIME of the top-5 documents.
Greedy@k: A greedy feature selection approach from [38]. We iteratively add the features with the biggest marginal contribution to the initially empty explanation set until a set size of k is reached. To attain attribution values we define three approaches: When a feature gets added to the explanation set its marginal contribution to that specific set is used as attribution value for **Greedy@k$_{\mathbf{iter}}$**. For **Greedy@k$_{\mathbf{marg}}$** every feature in the explanation gets assigned the marginal contribution to the explanation set without that specific feature. Both these approaches assign 0 values to all features that are not in the explanation set. **Greedy$_{\mathbf{iter}}$** takes the same approach as Greedy@k$_{\text{iter}}$ but instead adding features to the explanation set until all are included. We only consider Greedy@k$_{\text{iter}}$ for the MQ2008 dataset, since the runtime for data with higher feature dimension become so much higher than the runtime of the other approaches that considering this method in practice would be unreasonable.
RankLIME: A listwise LIME implementation for rankers. We re-implement [6] and define the model as the composition of a pointwise ranking model with the same predicted performance function that RankingSHAP uses and let LIME explain this model. Perturbation is done on each individual feature of each document independently, producing number of documents times n attribution values. We report two versions, RankLIME$_{\text{mean}}$ determines for each feature the mean attribution value over all documents, RankLIME$_{\text{max}}$ reports the max instead.

6.1.4 Predicted performance measure. We focus on the order of the ranked list as a whole by using Kendall's tau [13] as predicted performance function.

6.1.5 Implementation details. Except for Random, each approach that we consider uses background data, either for masking or for perturbing the input features. For MQ2008, we sample 100 random samples from the train data and use it for each approach. For the MSLR10k data we sample 20 background samples instead to compensate for the increased runtime because of the higher feature dimension. For evaluation we sample a different background data set of the same size as for approximation. For our implementation of RankingSHAP and PointwiseSHAP we use the KernelSHAP implementation from the SHAP python library [20] with all default settings. For PointwiseLIME we use the TabularExplainer for regression models with all default settings from the LIME python library [30].

6.2 Estimating Ground Truth Feature Attributes for evaluation

Determining the exact feature importance (Eq. 4) as ground truth is in most cases intractable due to the large number of possible coalitions of features ($\binom{n}{|S|}$ for coalition size $|S|$) and the integration over the background data distribution. Hence, we need a good approximation of the actual feature importance that we can use as *ground truth attributes*, $\phi^{gt} = \{i : \phi_i^{gt}(\mathcal{D}_q, R)\}_{i=1,\dots,n}$.

We use the approximation approach from [40] and measure stability in terms of standard deviation between different runs while increasing the size of the background data and the number of n-samples ((b, t)-pairs) that we use in the estimation; see Section C in the Appendix for a detailed analysis. Our experiments show that we can determine a sufficiently stable approximation of the actual feature attributes by sampling 2^{16} n-samples, which we use to approximate the ground truth for both datasets. Since the amount of background data that we use for generating the masks seems to have little impact on the stability, we use as much as is computationally reasonable.

6.3 Evaluation Metrics

Given the ground truth feature attributes, ϕ^{gt}, we propose two evaluation metrics that compare the feature attribution values $\phi = \{i \rightarrow \phi_i(\{x_{q,j}\}_j, R)\}_{i=1,\dots,n}$ to ϕ^{gt}.

Correct Order of Features.

We want to measure how well the order of the features when ranked with respect to ϕ_i aligns with the ground truth order of features ϕ_i^{gt}. Let $\text{rank}_\phi(i)$ resp. $\text{rank}_{\phi^{gt}}(i)$ be the rank of feature i when ordered according to ϕ, resp. ϕ^{gt}. We define order(ϕ) with Spearman's footrule metric following [7]:

$$\text{order}(\phi) = \sum_{i=1}^{n} |\,\text{rank}_{\phi^{gt}}(i) - \text{rank}_\phi(i)|. \tag{8}$$

We chose this metric, over other rank correlation coefficients like Kendall's tau, because it is easy to interpret.

Attribution Values. While attribution values are hard to interpret in isolation, seeing them in contrast with attribution values of other predictions can help us gain contrastive insights into the model decision. Therefore, with the second evaluation metric we measure how similar the actual attribution values of ϕ, and ϕ^{gt} are with help of the L1-distance:

$$\text{valdis}(\phi) = \frac{1}{n}\sum_{i=1}^{n} |\phi_i^{gt}(x, f) - \phi_i(x, f)| \tag{9}$$

Evaluation at top-k. For both metrics we define top-k evaluation metrics by only summing over features i that are among the top-k features in the ground truth attribution.

6.4 Results

We compare the results of experiments on the MQ2008 and MSLR-10k data presented in Table 2 to answer our research questions.

RQ1: Does RankingSHAP Order Feature Attributes More Accurately than Existing Approaches? We investigate the values of the order evaluation metric in the first three columns of Table 2. The values for MSLR-10k are much higher than for MQ2008, which is unsurprising considering that MSLR-10k has roughly three times as many features. For both datasets, we observe that all approaches

Table 2: Results of the quantitative experiments.

	Method	order↓			valdis↓		
		all	@3	@10	all	@3	@10
MQ2008	Random	15.3	21.1	18.6	0.0029	0.0288	0.0099
	PointwiseSHAP	8.4	2.7	5.8	0.006	0.0213	0.0116
	PointwiseLIME	13.7	10.9	13.9	0.0031	0.0204	0.0084
	Greedy@5_{marg}	14.6	20.6	17.7	0.0048	0.0324	0.0134
	Greedy@5_{iter}	14.9	21.6	18.4	0.0043	0.0326	0.0127
	$\text{Greedy}_{\text{iter}}$	13.0	8.9	11.6	0.0078	0.0099	0.0076
	$\text{RankLIME}_{\text{mean}}$	9.1	1.2	6.2	0.0026	0.0281	0.0107
	$\text{RankLIME}_{\text{max}}$	10.8	2.0	7.2	0.0017	0.0168	0.0062
	RankingSHAP	**3.8**	**0.1**	**0.9**	**0.0001**	**0.0001**	**0.0001**
MSLR-10k	Random	45.3	66.9	62.9	0.0005	0.012	0.0051
	PointwiseSHAP	33.3	24.1	27.1	0.1746	3.1706	1.415
	PointwiseLIME	33.7	13.4	16.6	0.1463	0.7981	0.5399
	Greedy@5_{marg}	42.5	26.8	41.1	0.0004	0.0076	0.0038
	Greedy@5_{iter}	42.5	26.4	41.1	0.0009	0.0232	0.0111
	$\text{RankLIME}_{\text{mean}}$	36.8	13.2	22.3	0.0005	0.0132	0.0059
	$\text{RankLIME}_{\text{max}}$	39.4	23.5	29.9	0.0005	0.0121	0.0053
	RankingSHAP	**26.4**	**0.7**	**2.1**	**0.0001**	**0.0002**	**0.0002**

are much more accurate in determining the rank of a feature in the top-3 and top-10 than for the rest of the features. RankingSHAP has by far the best feature rank approximation with rank differences of less than 1 for the top-3 and less than 3 for the top-10 ground truth features. $\text{RankLIME}_{\text{mean}}$ takes second place, showing the importance of using a listwise explanation objective. PointwiseSHAP, for MQ2008 with just under 3 and 6 ranks mean deviation from the ground truth rank for the top-3 resp. top-10 features, could still provide good enough features to gain some insight into the model decision. Its performance on MSLR-10k, with a deviation of more than 20 ranks for the top-3 features, is considerably worse. For both datasets, the selection-based baselines perform only slightly better than the random baseline, showing the clear need to approach the problem in a listwise manner. We answer RQ1 affirmatively: RankingSHAP approximates the order of the feature attributes more accurately than the baselines.

RQ2: Does RankingSHAP Determine the Attribution Values More Accurately than the Baselines? We evaluate the actual attribution values with the valdis metric. For both datasets, RankingSHAP clearly has the closest approximation of the attribution score values. Overall, the metric values are lower for the MSLR-10k dataset, which has more features and hence lower attribution values on the individual features on average. We answer this research question positively: RankingSHAP does determine the attribution values more accurately than the baselines.

6.5 Reflections

Using Pointwise and Listwise Explanations Combined. Our quantitative experiments show that by using a pointwise approach like PointwiseSHAP we cannot accurately determine listwise feature attributes. Yet, using pointwise explanations of ranking scores in combination with listwise ones, can give even more insights into the model decision. E.g., when used for queries with unusual listwise explanations, pointwise explanations of individual documents might help us understand what documents were responsible for high feature attribution values within a ranking decision.

Selection is not Attribution. Our attempts at extending the greedy selection explanations from [38] to generate feature attributes failed. Using a selection approach to select features and coalitions for which we determine the marginal contribution instead of estimating the expected marginal contribution over all coalitions sounds tempting, to reduce the computational costs. The resulting attribution scores are too inaccurate to be used in practice.

A Critical View on Evaluation. The results of our quantitative experiment should be viewed with reservations regarding the evaluation metrics used, which rely on knowledge of the *ground truth* attribution values. In practice, it may be impossible to determine these exactly, even with a large computational budget, so we have to rely on approximation techniques. We leave it to future work to find improved approximation techniques for this purpose.

7 CONCLUSION

We have rigorously defined the concept of listwise feature attribution for ranking tasks, which, through the use of different measures of the predicted performance of a model, allows ofr a flexible and contrastive examination of ranking decisions. To generate such listwise feature attribution explanations we introduced a method called RankingSHAP.

We found that RankingSHAP can aid in understanding model decisions and potentially detecting biases as we have demonstrated through a toy example. We confirmed the ability of RankingSHAP to approximate feature importance through a quantitative analysis, where RankingSHAP outperforms all point- and listwise baselines.

With the push to increased transparency in automated decision-making, by the general public and law makers, explainability is especially important for ranking systems as they are part of applications that directly impact peoples lives. Feature attribution has long been an important tool for understanding model decisions within other domains, therefore we hope that the listwise definition of feature attribution and RankingSHAP will facilitate transparency of ranking model decisions.

Among limitations of our approach are the high computational costs of determining feature attribution explanations, especially for high-dimensional input spaces. Also, it has been argued that SHAP attribution values can be difficult to interpret and do not necessarily align with human expectations [15], such as the contrastiveness of explanations [22].

The most important step that we believe should be taken next, is to examine whether the use of listwise SHAP attribution values in a contrastive manner can bridge the gap between mathematically well defined explanations and the practical use in real life applications.

Data and code. To facilitate reproducibility of our work, code and parameters are shared at https://github.com/MariaHeuss/RankingShap.

ACKNOWLEDGMENTS

We would like to thank Philipp Hager for his valuable feedback on the code, Joeri Noorthoek for his contributions to the code snippets used in this work, Mathijs Henquet and Jasmin Kareem for their valuable feedback on the manuscript. The research was partially funded by the Hybrid Intelligence Center, a 10-year program funded by the Dutch Ministry of Education, Culture and Science through the Netherlands Organisation for Scientific Research, https://hybrid-intelligence-centre.nl, project LESSEN with project number NWA.1389.20.183 of the research program NWA ORC 2020/21, which is (partly) financed by the Dutch Research Council (NWO), and the FINDHR (Fairness and Intersectional Non-Discrimination in Human Recommendation) project that received funding from the European Union's Horizon Europe research and innovation program under grant agreement No 101070212. All content represents the opinion of the authors, which is not necessarily shared or endorsed by their respective employers and/or sponsors.

REFERENCES

[1] Darius Afchar, Vincent Guigue, and Romain Hennequin. 2021. Towards rigorous interpretations: a formalisation of feature attribution. In *International Conference on Machine Learning*. PMLR, 76–86.

[2] Avishek Anand, Lijun Lyu, Maximilian Idahl, Yumeng Wang, Jonas Wallat, and Zijian Zhang. 2022. Explainable Information Retrieval: A Survey. *arXiv preprint arXiv:2211.02405* (2022).

[3] Avishek Anand, Procheta Sen, Sourav Saha, Manisha Verma, and Mandar Mitra. 2023. Explainable Information Retrieval. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 3448–3451.

[4] Arthur Câmara and Claudia Hauff. 2020. Diagnosing BERT with Retrieval Heuristics. *Advances in Information Retrieval* 12035 (2020), 605.

[5] Jaekeol Choi, Jungin Choi, and Wonjong Rhee. 2020. Interpreting Neural Ranking Models Using Grad-CAM. *arXiv preprint arXiv:2005.05768* (2020).

[6] Tanya Chowdhury, Razieh Rahimi, and James Allan. 2023. Rank-LIME: Local Model-agnostic Feature Attribution for Learning to Rank. In *Proceedings of the 2023 ACM SIGIR International Conference on Theory of Information Retrieval*. 33–37.

[7] Ronald Fagin, Ravi Kumar, and Dakshinamurthi Sivakumar. 2003. Comparing Top k Lists. *SIAM Journal on Discrete Mathematics* 17, 1 (2003), 134–160.

[8] 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* (Paris, France) *(SIGIR'19)*. ACM, New York, NY, USA, 1005–1008.

[9] Xiubo Geng, Tie-Yan Liu, Tao Qin, and Hang Li. 2007. Feature Selection for Ranking. In *SIGIR 2007: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Amsterdam, The Netherlands, July 23-27, 2007*, Wessel Kraaij, Arjen P. de Vries, Charles L. A. Clarke, Norbert Fuhr, and Noriko Kando (Eds.). ACM, 407–414.

[10] Andrea Gigli, Claudio Lucchese, Franco Maria Nardini, and Raffaele Perego. 2016. Fast Feature Selection for Learning to Rank. In *Proceedings of the 2016 ACM on International Conference on the Theory of Information Retrieval, ICTIR 2016, Newark, DE, USA, September 12- 6, 2016*, Ben Carterette, Hui Fang, Mounia Lalmas, and Jian-Yun Nie (Eds.). ACM, 167–170.

[11] Sebastian Hofstätter, Bhaskar Mitra, Hamed Zamani, Nick Craswell, and Allan Hanbury. 2021. 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* (Virtual Event, Canada) *(SIGIR '21)*. Association for Computing Machinery, New York, NY, USA, 1349–1358.

[12] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In *Advances in neural information processing systems*, Vol. 30.

[13] Maurice G. Kendall. 1938. A New Measure of Rank Correlation. *Biometrika* 30, 1/2 (1938), 81–93.

[14] Satyapriya Krishna, Tessa Han, Alex Gu, Javin Pombra, Shahin Jabbari, Steven Wu, and Himabindu Lakkaraju. 2022. The Disagreement Problem in Explainable Machine Learning: A Practitioner's Perspective. *arXiv preprint arXiv:2202.01602* (2022).

[15] I. Elizabeth Kumar, Suresh Venkatasubramanian, Carlos Scheidegger, and Sorelle Friedler. 2020. Problems with Shapley-value-based Explanations as Feature Importance Measures. In *International Conference on Machine Learning*. PMLR, 5491–5500.

[16] Yongchan Kwon and James Y Zou. 2022. WeightedSHAP: Analyzing and Improving Shapley based Feature Attributions. *Advances in Neural Information Processing Systems* 35 (2022), 34363–34376.

[17] Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2016. Rationalizing Neural Predictions. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. 107–117.

[18] Jurek Leonhardt, Koustav Rudra, and Avishek Anand. 2021. Learnt Sparsity for Effective and Interpretable Document Ranking. *arXiv preprint arXiv:2106.12460* (2021).

[19] Ana Lucic, Madhulika Srikumar, Umang Bhatt, Alice Xiang, Ankur Taly, Q. Vera Liao, and Maarten de Rijke. 2021. A Multistakeholder Approach Towards Evaluating AI Transparency Mechanisms. In *ACM CHI Workshop on Operationalizing Human-Centered Perspectives in Explainable AI*. ACM.

[20] Scott M. Lundberg and Su-In Lee. 2017. A Unified Approach to Interpreting Model Predictions. *Advances in Neural Information Processing Systems* 30 (2017).

[21] Lijun Lyu and Avishek Anand. 2023. Listwise Explanations for Ranking Models Using Multiple Explainers. In *European Conference on Information Retrieval*. Springer Nature Switzerland Cham, 653–668.

[22] Tim Miller. 2019. Explanation in Artificial Intelligence: Insights from the Social Sciences. *Artificial intelligence* 267 (2019), 1–38.

[23] Christophe Molnar. 2023. *Interpreting Machine Learning Models with SHAP*. Independently published.

[24] Gustavo Penha, Eyal Krikon, and Vanessa Murdock. 2022. Pairwise Review-based Explanations for Voice Product Search. In *Proceedings of the 2022 Conference on Human Information Interaction and Retrieval*. 300–304.

[25] Venetia Pliatsika, Joao Fonseca, Tilun Wang, and Julia Stoyanovich. 2024. ShaRP: Explaining Rankings with Shapley Values. *arXiv preprint arXiv:2401.16744* (2024).

[26] Alberto Purpura, Karolina Buchner, Gianmaria Silvello, and Gian Antonio Susto. 2021. Neural Feature Selection for Learning to Rank. In *European Conference on Information Retrieval*. Springer, 342–349.

[27] Tao Qin and Tie-Yan Liu. 2013. Introducing LETOR 4.0 Datasets. *arXiv preprint arXiv:1306.2597* (2013).

[28] Daniël Rennings, Felipe Moraes, and Claudia Hauff. 2019. An Axiomatic Approach to Diagnosing Neural IR Models. In *European Conference on Information Retrieval*. Springer, 489–503.

[29] 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*. 1135–1144.

[30] 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*. ACM, 1135–1144.

[31] Yao Rong, Tobias Leemann, Vadim Borisov, Gjergji Kasneci, and Enkelejda Kasneci. 2022. A Consistent and Efficient Evaluation Strategy for Attribution Methods. In *Proceedings of the 39th International Conference on Machine Learning*. 18770–18795.

[32] Lloyd S. Shapley. 1953. A Value for n-Person Games. In *Contributions to the Theory of Games II*. Princeton University Press, Princeton, 307–317.

[33] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. 2017. Learning Important Features Through Propagating Activation Differences. In *International Conference on Machine Learning*. PMLR, 3145–3153.

[34] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2014. Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. In *2nd International Conference on Learning Representations, ICLR 2014, Workshop Track Proceedings*.

[35] Jaspreet Singh and Avishek Anand. 2018. Posthoc Interpretability of Learning to Rank Models using Secondary Training Data. *arXiv preprint arXiv:1806.11330* (2018).

[36] 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* (Melbourne VIC, Australia) *(WSDM '19)*. ACM, New York, NY, USA, 770–773.

[37] 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*. 618–628.

[38] Jaspreet Singh, Megha Khosla, Wang Zhenye, and Avishek Anand. 2021. Extracting per Query Valid Explanations for Blackbox Learning-to-Rank Models. In *Proceedings of the 2021 ACM SIGIR International Conference on Theory of Information Retrieval*. 203–210.

[39] 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* (2021).

[40] Erik Strumbelj and Igor Kononenko. 2010. An Efficient Explanation of Individual Classifications using Game Theory. *The Journal of Machine Learning Research* 11 (2010), 1–18.

[41] Erik Štrumbelj and Igor Kononenko. 2014. Explaining Prediction Models and Individual Predictions with Feature Contributions. *Knowledge and information systems* 41 (2014), 647–665.

[42] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic Attribution for Deep Networks. In *International Conference on Machine Learning*. PMLR, 3319–3328.

[43] Manisha Verma and Debasis Ganguly. 2019. LIRME: Locally Interpretable Ranking Model Explanation. In *Proceedings of the 42nd International ACM SIGIR.*

[44] Tejaswani Verma, Christoph Lingenfelder, and Dietrich Klakow. 2022. Explaining Black-box Predictions by Generating Local Meaningful Perturbations. *International Journal of Semantic Computing* 16, 01 (2022), 47–68.

[45] 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* (2021).

[46] Jonas Wallat, Fabian Beringer, Abhijit Anand, and Avishek Anand. 2023. Probing BERT for ranking abilities. In *European Conference on Information Retrieval*. Springer Nature Switzerland Cham, 255–273.

[47] 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*. 669–680.

[48] Zijian Zhang, Koustav Rudra, and Avishek Anand. 2021. Explain and Predict, and then Predict again. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*. 418–426.

A EXPLICIT EXPERIMENTAL SETUP FOR THE SIMULATED EXPERIMENT

Here we include the explicit set-up of the simulated example from Section 5. In Table 3 we give an overview over all candidates that were used for the different query scenarios.

Table 3: Feature values for the individual candidates.

	experience	skills	grades	university	requirements
non-qualified	0.7	0.7	3.2	$\mathbf{university_{us}}$	False
qualified-1	0.8	0.55	3.5	$\mathbf{university_{us}}$	True
qualified-2	0.7	0.3	3	$\mathbf{university_{us}}$	True
non-qualified-privileged	0.8	0.6	3.6	$\mathbf{university_{nepotism}}$	False
qualified-3	0.9	0.8	3	$\mathbf{university_{us}}$	True
qualified-net	0.7	0.9	8	$\mathbf{university_{net}}$	True
qualified-ger	0.8	0.8	1	$\mathbf{university_{ger}}$	True
qualified-biased	0.8	0.6	3.6	$\mathbf{university_{neg\text{-}bias}}$	True

The different universities have different grading schemes, which the models from Figure 1 depends on. Table 4 shows an overview over the different universities that are used in the query scenarios. We show the best possible and the worst passing grade as well as whether the biased model is biased towards the university in question.

Table 4: Comparison of grading schemes and model bias across universities.

University	Best Possible Grade	Worst Passing Grade	Model Bias
$\mathbf{university_{us}}$	4	1	None
$\mathbf{university_{nepotism}}$	4	1	Positive
$\mathbf{university_{neg\text{-}bias}}$	4	1	Negative
$\mathbf{university_{ger}}$	1	4	None
$\mathbf{university_{net}}$	10	6	None

Those candidates were then used for different queries. Which candidates were used for what queries can be found in Table 5. We show the candidates, as defined in Table 3, in the rows and the 5 different query scenarios from Section 5.3 in the columns. A table entry of 1 indicates that the corresponding candidate was included in the ranking decision, a value 0 indicated that they were not included.

Table 5: Query-candidate matrix - 1 indicates that the candidate was considered, 0 that they were not considered.

	average	nepotism	qualified	international	biased
non-qualified	1	1	0	1	1
qualified-1	1	1	1	0	1
qualified-2	1	1	1	0	1
non-qualified-privileged	0	1	0	0	0
qualified-3	0	0	1	1	0
qualified-net	0	0	0	1	0
qualified-ger	0	0	0	1	0
qualified-biased	0	0	0	0	1

B SIMULATED EXPERIMENT - ADDITIONAL RESULTS.

Here we present additional results for the simulated experiment for the unbiased model from the flowchart in Figure 1b. The bar chart in Figure 3 shows the feature attribution values from the three considered approaches from Section 5.2 for the same query scenarios as defined in Section 5.3. We do not show the estimated importance as we do in Figure 2, since we do not want to use this model for the evaluation of the approaches. Comparing the attribution values of different models for different query scenarios like in Figures 2 and 3 can help us with selecting the least biased model when we have a choice of models of similar performance.

C DETAILS ON ESTIMATING GROUND TRUTH FEATURE ATTRIBUTES FOR EVALUATION

For an estimation of the exact feature importance for the use of ground truth as described in Section 6.2, we measure the stability of the approximation in terms of the standard deviation over several runs for the same amount of n-samples (coalition-background (b, t) pairs) and size of background data. Here we show the results of the MQ2008 dataset.

For the background data we experiment with sets of 5, 20 and 100 background samples. Since by increasing the sample size we could not find any effect on the stability of the feature attributes, we decide to use 100 background samples for the rest of our analysis. We leave the investigation of whether more sophisticated ways to create a background summary, instead of just sampling from the underlying data, can improve approximation accuracy for future work.

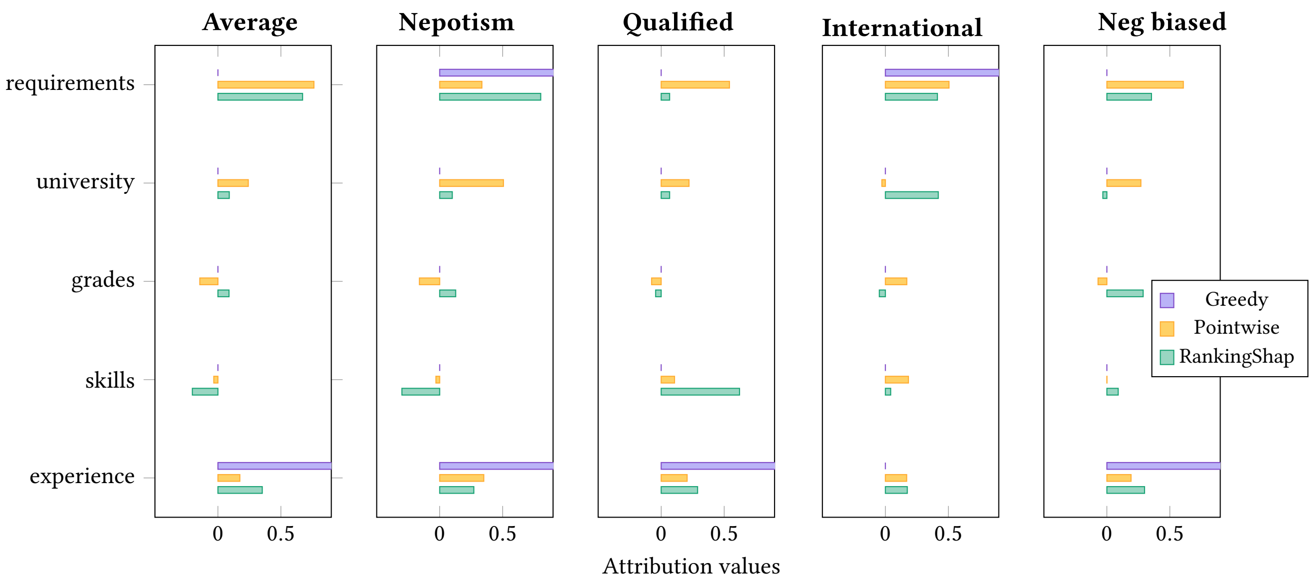
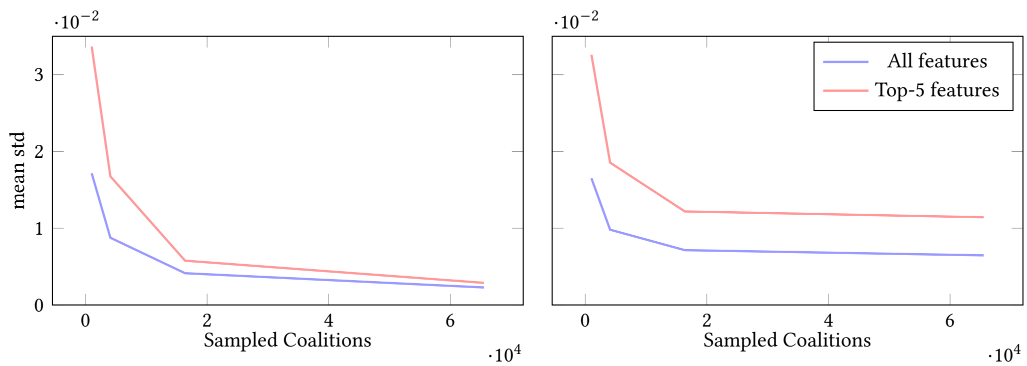


Figure 3: Feature attribution values of the unbiased model for different query scenarios from Section 5.2



(a) Stability wrt same background set. **(b) Stability wrt different background sets.**

Figure 4: Mean standard deviation of the approximated feature importance over 3 disjoint runs with the same (a) resp. different (b) background data.

Figure 4 shows the stability in terms of mean standard deviation over three different runs for different numbers of sampled coalition-background (b, t) pairs which we call n-samples for short. Figure 4a, where each run uses the same background-data, shows a continuing increase in stability by increasing the number of n-samples, both when looking at all as well as only the top-5 features, to values below 0.003. Figure 4b on the other hand, where each run uses a different set of background samples, shows that the increase in stability stagnates at a value of around 0.1. While this potentially could be solved by using a bigger set for the background data or better summarizing techniques, we would likely also have to increase the number of n-samples significantly to get a higher level for stability in combination.

Since the top-5 feature attribution values for this dataset usually take values of more than 0.1, we deem this level of stability with a standard deviation of roughly 0.01 acceptable and proceed by using the mean of those three experiments as ground truth attribution values.