Rank-LIME: Local Model-Agnostic Feature Attribution for Learning to Rank

Tanya Chowdhury
University of Massachusetts Amherst
Amherst, MA, USA
tchowdhury@cs.umass.edu

Razieh Rahimi
University of Massachusetts Amherst
Amherst, MA, USA
rahimi@cs.umass.edu

James Allan
University of Massachusetts Amherst
Amherst, MA, USA
allan@cs.umass.edu

ABSTRACT

Understanding why a model makes certain predictions is crucial when adapting it for real world decision making. LIME is a popular model-agnostic feature attribution method for the tasks of classification and regression. However, the task of learning to rank in information retrieval is more complex in comparison with either classification or regression. In this work, we extend LIME to propose Rank-LIME, a model-agnostic, local, post-hoc linear feature attribution method for the task of learning to rank that generates explanations for ranked lists. We employ novel correlation-based perturbations, differentiable ranking loss functions and introduce new metrics to evaluate ranking based additive feature attribution models. We compare Rank-LIME with a variety of competing systems, with models trained on the MS MARCO datasets and observe that Rank-LIME outperforms existing explanation algorithms in terms of Model Fidelity and Explain-NDCG. With this we propose one of the first algorithms to generate additive feature attributions for explaining ranked lists.

CCS CONCEPTS

• **Information systems → Learning to rank**.

KEYWORDS

ranking interpretability; post-hoc explanations; feature attribution methods; learning to rank; LIME

ACM Reference Format:
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 the Theory of Information Retrieval (ICTIR '23), July 23, 2023, Taipei, Taiwan.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3578337.3605138

1 INTRODUCTION

A large number of explanation methods have been introduced in the last few years [8, 16, 18]. While some are motivated by generating explanations of deep neural models for end users, others are built in order to gain an understanding of unknown properties and mechanisms of the underlying data-centric process. Explanations that belong to the class of additive feature attribution methods (e.g. LIME and SHAP) have played a large role in supporting the above goals, especially for tasks like regression and classification.

Explanation of learning-to-rank models, though, has received little attention. Singh and Anand. [19] extended LIME to explain the relevance of a document to a query. However they (i) do not evaluate the quality of generated explanations, and (ii) do not explain the reason behind the order of documents retrieved for a query by the learning-to-rank model. Singh et al. [20] proposed a greedy approach to identify the top features responsible for generating a ranked list. However, their approach (i) has only been suggested for rankers with human-engineered features as input, and (ii) does not scale well for a large number of features. In contrast, we see great value in a general, post-hoc ranking explanation techniques to understand the behaviour of deep learning-to-rank models.

Existing point-wise explainers can be used to explain a ranked list one document at a time. However, they focus on features that are responsible for a particular document being relevant at a time. In our early experiments, we discovered that those features could not be used to reconstruct the original ordering reliably. Instead, a list-wise approach has the potential to identify exactly which features are important to achieve a particular ordering of documents. We believe such an explanation is thus more useful to a user who wants to understand the rationale behind an ordering (e.g., a product search result).

We describe Rank-LIME, an approach to generate model-agnostic local additive feature attributions for the task of learning to rank. Given a black-box ranker whose architecture is unknown, a query, a set of documents and explanation features, and a small part of the training data, Rank-LIME returns a set of weights w_i for the most important features, measuring the *relative contribution* of those features towards deciding the ordering. We focus on generating Rank-LIME explanations for transformer-based rankers (e.g. BERT and T5), but Rank-LIME can be used to generate explanations for other ranking models as well. We propose metrics based on Kendall's Tau and NDCG to compute the accuracy of the generated explanations for learning-to-rank models and compare them with strong baselines.

Our main contributions are (i) extending LIME to explain List-Wise relevance functions, (ii) introducing correlation-based instance perturbations, (iii) employing ranking reconstruction loss functions in LIME, and (iv) proposing measures to evaluate feature attributions in ranking. To the best of our knowledge this is one of the first works on explaining listwise relevance functions using feature attributions.

ICTIR '23, July 23, 2023, Taipei, Taiwan

ACM ISBN 979-8-4007-0073-6/23/07…$15.00
https://doi.org/10.1145/3578337.3605138

2 RELATED WORK

LIME [16] is one of the first steps towards model explainability in machine learning literature. The authors propose a model-agnostic linear explanation method where they locally approximate a classifier with an interpretable model by perturbing inputs and then generating labels for the perturbed inputs. The output of their model is a bar graph representing contributions of supporting and opposing features. Later, Lundberg and Lee [8] propose KernelSHAP, a framework similar to LIME, which satisfies properties of the classical Shapely values.

Explanations for neural models in Information Retrieval is a relatively unexplored task. One of the first works in this direction was EXS, a local post-hoc explainability technique by Singh and Anand [19], where they extended the general LIME-classifier explainability model to pointwise rankers. They use it to answer three types of IR explainability questions: 1) Why is a given document relevant to a given query? 2) Why is document A ranked higher than document B?, and 3) What is the query intent learnt by the ranking model?. They do so by perturbing the inputs in the locality of the document of interest and then generating binary relevance judgements for these perturbed queries. These perturbed instances are fed to the LIME explainability model and visualized as in the original.

Next, Fernando et al. [5] explore a model-introspective explainability method for Neural Ranking Models (NRMs). They use the DeepSHAP [8] model which in turn extends DEEP LIFT [18] to learn an introspective explainable model for deep neural rankers. They compare with NRM explanations generated by Singh and Anand [19] and find that explanations generated by LIME and DeepSHAP are significantly divergent.

Singh et al. [20] extend IR explainability approaches to human engineered features, and propose two metrics: *validity* and *completeness.* They try to optimize these two metrics by greedily finding a subset of the input model features, capping the set size at K, such that there is a high correlation between the rankings produced by the selected features and the original blackbox model, i.e., high validity. At the same time they try to maximize completeness, which they quantify as the negative Kendall's Tau correlation between non-explanation features and the original ranking.

Verma and Ganguly [22] propose a model-agnostic pointwise approach to compute *explanation vectors*, which can in turn be studied to find positive or negative contributions of a term towards a ranking decision. Sen et al. [17] extend that work to give explanations in terms of three primal IR features: *frequency of a term in a document*, *frequency of a term in a collection*, and *length of a document* as the weights of a linear function. Most recently, Anand et al. [1] wrote a survey paper on explainable approaches for Information retrieval, which discussed other post-hoc explanation mediums like free-text and adverserial examples along with feature attribution methods. Other work [23–25] delves into generating explainable decisions for recommender systems.

3 RANK-LIME

Let f denote a black-box learning-to-rank model that, given a query q and a set of documents $D = \{d_1, \ldots, d_N\}$, returns a ranking R of documents in D, i.e., $f(q, D) = R$. The ranking model f can be uni-variate or multivariate [12], trained with a pointwise, pairwise, or listwise loss function [7]. The dominant approach for learning-to-rank models is a uni-variate scoring function [7, 9] where the ranker scores each document $d_i \in D$ individually and then sorts the documents based on their scores. However, as our goal is explaining a black-box ranker, we do not make any assumption on the ranker and thus want to explain the obtained ranking R from a black-box learning-to-rank model f.

Here, we focus on local explanation of the behavior of a learning-to-rank model for a single ranked list with respect to a query as proposed in LIME and subsequent works [8, 16]. Specifically, the goal is to explain the ranking R_x obtained from $f(.)$ for the single instance $x = (q, D)$. Explanation models usually work on interpretable (or simplified) inputs x' that map to the original inputs through a mapping function as follows.

$$x = h_x(x'). \tag{1}$$

Documents here are represented as a bag of words. For a vocabulary of N words, each document is represented by an N-dimensional vector where the i^{th} element represents the frequency of the i^{th} word in that document.

Extending LIME for the explanation of learning-to-rank models, we sample instances around x' to approximate the local decision boundary of f. The perturbed instances are denoted by z'. Feeding the perturbed instance z' to ranker $f(h_x(z'))$, one obtains the ranking $R_{z'}$.

Following LIME, We define an explanation as a model $g \in \mathcal{G}$ where $\mathcal{G}$ is the class of linear models, such that $g(z') = w_g \cdot z'$. The Rank-LIME explanation is then obtained by minimizing the following objective function.

$$\xi = \arg\min_{g \in \mathcal{G}} \mathcal{L}(f, g, \pi_{x'}) + \Omega(g), \tag{2}$$

where $\Omega(g)$ represents the complexity of the explainable model, and $\pi_{x'}$ measures the locality of perturbed instances. In LIME [16], the loss function $\mathcal{L}(.)$ is defined as the mean squared error, which is not applicable to the output of learning-to-rank models as ranking of documents. We discuss the choice of loss, complexity, and locality functions in the following subsections.

3.1 Locality Function

The local kernel $\pi_{x'}$ in Eq. (2) defines the locality of the instance to be explained to weight the perturbed instances. Following LIME [16], we use an exponential kernel as:

$$\pi_{x'}(z') = \exp(-\Delta(x', z')^2/\sigma^2), \tag{3}$$

where $\Delta(.)$ denotes a distance function between the instance to be explained and a perturbed instance in the space of explanation features, i.e., $x' = (q^{x'}, D^{x'})$ and $z' = (q^{z'}, D^{z'})$. We instantiate the distance function as:

$$\Delta(x', z') = \delta(q^{x'}, q^{z'}) + \sum_{d_i \in D} \delta(d_i^{x'}, d_i^{z'}), \tag{4}$$

where the function δ measures the cosine distance between the vector representations of queries or documents based on the explanation features.

3.2 Feature Perturbations

The LIME algorithm is largely susceptible to assigning improper attributions when its explanation features are large in number and correlated to each other [26]. These become the largest inhibitors in using LIME for explaining ranking decisions, where the documents are often strongly correlated to each other. To counter this, we compute Σ_C, the covariance matrix of the features from a part of the training dataset and incorporate them to generate perturbations in the Rank-LIME algorithm.

Perturbations in the original LIME algorithm are random. Each feature is sampled independently from a normal distribution centred around the instance with $\mu = 0$ and $\sigma = 1$. However in Rank-LIME, we carry out perturbations in the vicinity of an instance x, having $\mu = 0$ and $\Sigma = \Sigma_C$. These include both single feature perturbations as well as group perturbations. This helps maintain the feature correlations from the training set in the perturbed instances and avoids noisy off-manifold or out-of-distribution perturbations [21]. This is especially important in the presence of large documents and correlated features.

3.3 Loss Functions

The Rank-LIME explanation model needs a differentiable loss function quantifying the divergence between two ranked lists, specifically the original ranking R_x and the ranking R_z for a perturbed instance. The metrics generally used to compare or evaluate ranked lists are Kendall's Tau and *normalized discounted cumulative gain* (NDCG). However, neither of these metrics are differentiable, and hence cannot be used to train a regression model to generate explanations. We thus use a proxy of the above non-differentiable metrics in the Rank-LIME framework as follows:

ListNet [2] represents each ranked list with a probability distribution – top-1 probability as an approximation of permutation probability. It then uses cross-entropy to measure the dissimilarity between two ranked lists.

RankNet [3] is a pairwise probabilistic loss function, aggregated to calculate list comparison scores.

ApproxNDCG [14] is a listwise proxy loss to NDCG that replaces the position and truncation functions of NDCG with a smooth function based on the position of documents.

NeuralNDCG [13] is a new proxy to NDCG that uses a differentiable approximate alternative to the sort function in NDCG, which is then plugged into the NDCG formula to compute relevance. Its neural sorting function is able to sort documents with high accuracy and bounded error rates.

We use the allRank[1] implementation for each of these loss functions.

4 EXPERIMENTS

Datasets For experiments on text-based ranking models, we consider BM25, BERT [4], and T5 [15] ranking models as the black-box rankers to be explained. BERT and T5 rankers are fine-tuned using the MS MARCO passage ranking dataset [10] following previous studies [11]. To conduct our experiments, we generate explanations for queries in the test set of the dataset. We generate explanations for the list of ten most relevant documents according to each ranker. We use the PyGaggle implementation of the BM25, BERT, and T5 rankers.[2]

Table 1: Comparing different perturbation methods and loss functions for Rank-LIME for word-based explanation of the BERT-based ranker [11]. Generation Time is measured in seconds.

Perturbation	Loss Function	Fidelity	Gen-Time
Single-Perturbations	ListNet	0.39	**240**
	RankNet	0.35	270
	ApproxNDCG	0.43	340
	NeuralNDCG	0.41	360
Group-Perturbations	ListNet	0.42	420
	RankNet	0.39	440
	ApproxNDCG	**0.49**	500
	NeuralNDCG	0.46	540

Competing Methods. We compare the performance of the following explanation algorithms. We generate explanations by assigning linear attributions to the features chosen by each algorithm.

RANDOM assigns random weights to k of the explanation features, normalized to add up to 1.

Averaged-EXS is proposed by Singh and Anand [19]. This approach addresses explainability for learning-to-rank models which use textual data as their input. They generate LIME attributions for each query-document pair based on various document relevance measures. We then aggregate their attributions of each document, select top k features and normalize them to present as an attribution for the ranked list.

Weighted-EXS: One of the EXS methods uses LIME to attribute why any *document A is more relevant than document B*. We aggregate these explanations for each pair of documents and score them in a weighted manner, such that, the weight of each attribution is based on the difference in their document ranks in the original ranking. High rank difference documents are given more weight than low rank difference ones. The top k features from these weighted aggregated attributions are picked, which are then normalized such that they add up to 1.

TopKFeatures, proposed by Singh et al. [20], introduces two metrics, validity and completeness, to select the top k interaction features that contribute to decision making. This method however does not assign a weight to each feature, based on their contribution. As a result we cannot compare this method to our proposed method as is. Instead we assign a uniform weight value of $\frac{1}{k}$ to each feature in the result of this method, in order to enable comparison.

Evaluation Metrics We propose the following metrics to evaluate the quality of explanation models. We derive intuition from similar metrics for the tasks of classification/regression and adapt them to suit ranking.

Model Faithfulness/Fidelity evaluates how good the explanation models are in reconstructing the black-box model's output for the given query. We construct an *explanation model's ranking* by linearly combining the features multiplied by their weights to

[1] https://github.com/allegro/allRank/tree/master/allrank/models/losses

[2] https://github.com/castorini/pygaggle

Table 2: Comparing performance of different competing systems for word-based explanations of textual rankers based on model fidelity and Explain-NDCG.

Ranker	System	Fidelity	Explain-NDCG
BM25	Random	0.20	0.0014
	Average-EXS	0.45	0.2145
	Weighted-EXS	0.56	0.3419
	Top-K	0.39	0.1998
	Rank-LIME	**0.61**	**0.4019**
BERT	Random	0.19	0.0002
	Average-EXS	0. 39	0.1895
	Weighted-EXS	0.42	0.2109
	TopKFeatures	0.38	0.1034
	Rank-LIME	**0.47**	**0.3210**
T5	Random	0.16	0.0042
	Average-EXS	0.32	0.1989
	Weighted-EXS	0.41	0.2109
	TopKFeatures	0.35	0.1487
	Rank-LIME	**0.48**	**0.3415**

form an ordering. We then compute the Kendall's Tau between the ranking by the black-box model and this obtained ordering.

Explain-NDCG@10 also evaluates how well the explanation model reconstructs the black-box model's output. Unlike Kendall's Tau, this metric is position sensitive and yields higher scores for models which explain top ranked documents better. We use the scores obtained by the learning-to-rank model to assign NDCG relevance to each document. However, the scores of documents might be negative leading to an unbounded NDCG value. As a result, we use min-max normalization on the scores to decide on relevance labels while maintaining statistical significance, as recommended by Gienapp et al. [6]. The higher the score, the more relevant the document. We then compute the NDCG of the reconstructed ranking with these relevance scores. We average the value of NDCG@10 across all queries to report this metric.

Experimental Settings For explanations, we define the *explanation feature set* as words of the set of documents D and the query q in the instance x to be explained. We conduct experiments where ranking models BM25, BERT, and T5 are explained with features derived from the words of the instance to be explained. Perturbations are obtained by (i) modifying the frequency of a single feature at a time or (ii) modifying the frequency of a group of features at a time. The feature covariance matrix is computed using 1,000 randomly sampled documents from the training set. Each training document is represented as a bag of words vector for Σ_C to be computed. We also compare different loss functions for different perturbation settings in Rank-LIME and report fidelity and relative explanation generation time for each scenario. We use the bag-of-words representation of inputs to generate explanations. As a result the generated explanations are not position dependent. For all experiments, we choose 50 instances from the test sets of MS MARCO dataset to generate local explanations. We pick top k ($k = 8$) features for evaluating explanations by different systems in a fair manner.

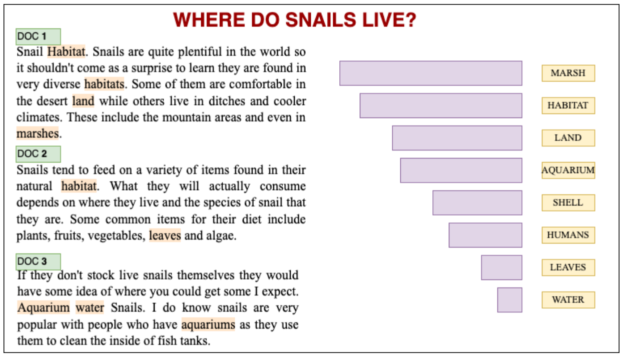
5 RESULTS AND DISCUSSION

Rank-LIME Parameters. Table 1 shows the fidelity of Rank-LIME in the word-based explanation of BERT-based ranker [11] when different perturbations and loss functions are used. Single perturbations refer to perturbations where we perturb a single feature value at a time. Group perturbations refer to perturbations where we perturb groups of features in the instance being explained at a time. We find the ApproxNDCG loss function outperforms other listwise loss functions proposed for generating Rank-LIME explanations by at least 4.8%. The group perturbation setting where we mask a subset of features at each perturbation achieves higher fidelity than the single perturbation scenario by 13.9% but at the cost of generation time (47% overhead). Since there is no time or budget constraint in this scenario, we use the ApproxNDCG loss function and group perturbation scenario for comparisons with later systems. Table 1 also depicts relative generation time for each kind of perturbation. We do not associate any absolute metric to it as generation time is implementation and system dependent. Figure 1 is an example Rank-LIME output corresponding to an instance of the MS MARCO dataset.

Rank-Lime vs Competing systems. Next, we compare Rank-LIME to competing systems in Table 2. The reported results are computed over 50 instances randomly chosen from the test set of the dataset. We observe that Weighted-EXS outperforms Averaged-EXS by 24.4%. However, Rank-LIME outperforms the best baseline by 11.9%. We see an even bigger win for Rank-LIME on Explain-NDCG, 17.6% over the strongest baseline. A bigger win by Rank-LIME on Explain-NDCG as compared to Kendall's Tau (fidelity) intuitively suggests that Rank-LIME pays more attention to explaining high scoring documents as compared to other explanation algorithms. We also observe that explanation algorithms achieve 29.7% higher fidelity while explaining the BM25 ranker as compared to while explaining the BERT ranker. This suggests that LIME-based algorithms that generate linear explanations perform worse at explaining neural models with complex decision boundaries as compared to explaining simple ranking models.

6 CONCLUSION AND FUTURE WORK

The notion of explainability and how we quantify it is largely subjective. However, explanation models which are highly faithful as well as interpretable can be passed off as *good* explanation models. In this work, we presented a model-agnostic algorithm to provide linear feature attributions for results in the task of learning to rank. Our method is general and caters to pointwise, pairwise, and listwise techniques. We proposed novel feature attribution techniques and evaluation metrics suitable for a ranking explanation task and showed that our method outperforms competing baselines. We are aware that there many performance-enhancing modifications to LIME have been proposed in the machine learning community [8], since we ran our experiments. Future work would include seeing which of those modifications are beneficiary for ranking attributions. It would also be good to incorporate causality in explanation generation in place of just considering correlation, since correlation does not always indicate causation. Additionally, it would be interesting to see how explanations from each of the competing



WHERE DO SNAILS LIVE?

DOC 1
Snail Habitat. Snails are quite plentiful in the world so it shouldn't come as a surprise to learn they are found in very diverse habitats. Some of them are comfortable in the desert land while others live in ditches and cooler climates. These include the mountain areas and even in marshes.

DOC 2
Snails tend to feed on a variety of items found in their natural habitat. What they will actually consume depends on where they live and the species of snail that they are. Some common items for their diet include plants, fruits, vegetables, leaves and algae.

DOC 3
If they don't stock live snails themselves they would have some idea of where you could get some I expect. Aquarium water Snails. I do know snails are very popular with people who have aquariums as they use them to clean the inside of fish tanks.

Figure 1: A toy example depicting Rank-LIME results on a query and three documents from the MS MARCO dataset. The documents were initially fed to the BERT Ranker and obtained the relevance scores, based on which they were ordered. Rank-LIME was later used to explain the model ordering and gives the bar chart shown to the right with its relative scores. Here we report the top 8 tokens which Rank-LIME found important in the model decision making.

systems impact human understanding of the blackbox model, via an user study.

7 ACKNOWLEDGEMENTS

This work was supported in part by the Center for Intelligent Information Retrieval, in part by NSF IIS-2039449 , and in part by NSF grant number 1813662. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsor.

REFERENCES

[1] 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).

[2] Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. 2007. Learning to rank: from pairwise approach to listwise approach. In *Proceedings of the 24th international conference on Machine learning*. 129–136.

[3] Wei Chen, Tie-Yan Liu, Yanyan Lan, Zhi-Ming Ma, and Hang Li. 2009. Ranking measures and loss functions in learning to rank. *Advances in Neural Information Processing Systems* 22 (2009), 315–323.

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186. https://doi.org/10.18653/v1/N19-1423

[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*. 1005–1008.

[6] Lukas Gienapp, Maik Fröbe, Matthias Hagen, and Martin Potthast. 2020. The Impact of Negative Relevance Judgments on NDCG. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 2037–2040.

[7] Hang Li. 2011. *Learning to Rank for Information Retrieval and Natural Language Processing*. Morgan & Claypool Publishers.

[8] Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. In *Advances in neural information processing systems*. 4765–4774.

[9] Bhaskar Mitra and Nick Craswell. 2018. An Introduction to Neural Information Retrieval. *Foundations and Trends® in Information Retrieval* 13, 1 (December 2018), 1–126.

[10] Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A human generated machine reading comprehension dataset. In *CoCo@ NIPS*.

[11] Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage Re-ranking with BERT. *arXiv preprint arXiv:1901.04085* (2019).

[12] Liang Pang, Jun Xu, Qingyao Ai, Yanyan Lan, Xueqi Cheng, and Jirong Wen. 2020. Setrank: Learning a permutation-invariant ranking model for information retrieval. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 499–508.

[13] Przemyslaw Pobrotyn and Radoslaw Bialobrzeski. 2021. NeuralNDCG: Direct Optimisation of a Ranking Metric via Differentiable Relaxation of Sorting. *ArXiv* abs/2102.07831 (2021).

[14] Tao Qin, Tie-Yan Liu, and Hang Li. 2010. A general approximation framework for direct optimization of information retrieval measures. *Information retrieval* 13, 4 (2010), 375–397.

[15] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research* 21, 140 (2020), 1–67. http://jmlr.org/papers/v21/20-074.html

[16] 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.

[17] Procheta Sen, Debasis Ganguly, Manisha Verma, and Gareth JF Jones. 2020. The Curious Case of IR Explainability: Explaining Document Scores within and across Ranking Models. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2069–2072.

[18] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. 2017. Learning important features through propagating activation differences. *arXiv preprint arXiv:1704.02685* (2017).

[19] 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*. 770–773.

[20] Jaspreet Singh, Megha Khosla, and Avishek Anand. 2020. Valid Explanations for Learning to Rank Models. *arXiv preprint arXiv:2004.13972* (2020).

[21] Dylan Slack, Sophie Hilgard, Emily Jia, Sameer Singh, and Himabindu Lakkaraju. 2020. Fooling lime and shap: Adversarial attacks on post hoc explanation methods. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*. 180–186.

[22] Manisha Verma and Debasis Ganguly. 2019. LIRME: locally interpretable ranking model explanation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1281–1284.

[23] Nan Wang, Hongning Wang, Yiling Jia, and Yue Yin. 2018. Explainable recommendation via multi-task learning in opinionated text data. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 165–174.

[24] Yongfeng Zhang, Xu Chen, et al. 2020. Explainable recommendation: A survey and new perspectives. *Foundations and Trends® in Information Retrieval* 14, 1 (2020), 1–101.

[25] Yongfeng Zhang, Jiaxin Mao, and Qingyao Ai. 2019. SIGIR 2019 tutorial on explainable recommendation and search. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1417–1418.

[26] Zhengze Zhou, Giles Hooker, and Fei Wang. 2021. S-lime: Stabilized-lime for model explanation. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 2429–2438.