A Deep Analysis of an Explainable Retrieval Model for Precision Medicine Literature Search

Jiaming Qu, Jaime Arguello, and Yue Wang

University of North Carolina at Chapel Hill, Chapel Hill NC 27599, USA
jiaming@live.unc.edu, {jarguello, wangyue}@unc.edu

Abstract. Professional search queries are often formulated in a structured manner, where multiple aspects are combined in a logical form. The information need is often fulfilled by an initial retrieval stage followed by a complex reranking algorithm. In this paper, we analyze a simple, explainable reranking model that follows the structured search criterion. Different aspects of the criterion are predicted by machine learning classifiers, which are then combined through the logical form to predict document relevance. On three years of data from the TREC Precision Medicine literature search track (2017-2019), we show that the simple model consistently performs as well as LambdaMART rerankers. Furthermore, many black-box rerankers developed by top-ranked TREC teams can be replaced by this simple model without statistically significant performance change. Finally, we find that the model can achieve remarkably high performance even when manually labeled documents are very limited. Together, these findings suggest that leveraging the structure in professional search queries is a promising direction towards building explainable, label-efficient, and high-performance retrieval models for professional search tasks.

Keywords: Professional search · Precision medicine · Explainable IR

1 Introduction

Professional searchers often formulate complex information needs as a function of various concepts, or *aspects*. For example, in systematic reviews for evidence-based medicine, relevance criteria usually involve four elements, namely *population*, *intervention*, *comparison*, and *outcome*, collectively known as the PICO elements [33]. In the TREC Precision Medicine track, a relevant research article should discuss *cancer treatment* and focus on subjects who had *the same cancer*, *the same genetic variation*, and *the same demographic* as the patient at hand [29]. In legal search, an inquiry often include multiple aspects such as *entity*, *event*, *time*, and *location* [18].

Professional searchers often use big, structured Boolean queries to express their relevance criteria. Each relevance aspect is encoded as a disjunctive clause of synonymous terms, which are then combined in a conjunctive clause to encode the inclusion/exclusion criteria [31]. Boolean queries are popular among professional searchers because they provide an expressive and intuitive way of specifying complex logic. However, composing the "right" Boolean query takes experience and patience. Finding the right terms to include and exclude often requires extensive domain knowledge and many iterations of trial-and-error. As a result, searchers often use Boolean queries to retrieve an initial expansive set of results with high recall but low precision, and then manually sift through these results to identify relevant ones [25].

To refine Boolean search results and reduce a searcher's manual effort, machine learning and text mining approaches are proposed to rerank the initial search results. These include various learning-to-rank [20, 8] and active learning [37, 36] techniques. In these techniques, a ranking function is trained to distinguish relevant results from non-relevant ones, which helps prioritize relevant results for manual review [16]. However, building such a ranking function often requires a substantial amount of training data, which may not be available upfront. To create such training data, a searcher needs to manually assess the relevance of many initial search results. Once trained, the ranking function is often complex (e.g. gradient boosted decision trees or neural networks) and difficult to interpret, as it may not follow the searcher's decision logic expressed in the Boolean query. This lack of transparency is undesirable as many professional searchers value transparency more than pure ranking performance [31].

Our research is motivated by one overarching question: How can we support professional searchers with retrieval systems that leverage machine learning while preserving the transparency and interpretability of Boolean search? In a recent short paper, we explored an explainable retrieval model towards this goal [26]. The task setting is the TREC Precision Medicine (PM) track, where the relevance criteria involve multiple aspects combined in a decision logic [29]. The retrieval model learns separate machine learning classifiers to predict aspect-level relevance, and combines them through the decision logic to produce document-level relevance. Such a model can be easily explained as it closely resembles the searcher's relevance decision process. Preliminary results showed that the model performed as well as complex learning-to-rank models on 2018 PM track topics.

In this paper, we further investigate three research questions in order to gain deeper understanding of the proposed model beyond the preliminary work. Below we state these research questions and summarize the main findings.

1) *Is the proposed model generalizable to a broader range of PM topics?* By cross-validating on three years of data from the TREC PM track (2017-2019), we find that the model consistently performs as well as complex learning-to-rank models, confirming its generalizability beyond a specific year of data.
2) *How does the proposed model compare with those developed by teams in the PM track?* We find that the proposed model can replace competitive black-box models submitted to the TREC PM track without significantly compromising retrieval performance, and sometimes even give performance gains.
3) *Does the proposed model require many labeled documents to learn well?* Through a learning curve analysis, we show that the model is substantially more label-

efficient than a conventional learning-to-rank model. It achieved remarkable retrieval performance even when trained on very few labeled documents.

2 Related Work

Professional search strategies. In professional search tasks, especially systematic literature reviews, searchers formulate structured information needs through complex Boolean queries, where concepts are encoded as disjunctive clauses of synonymous terms, and inclusion/exclusion criteria are built on top of these concepts [2, 31]. Our approach aims to automate and assist in these tasks by replacing manually constructed query components with machine learning classifiers, and by logically explaining predictions using relevance aspects.

Explainable information retrieval. Recent works on explainable search and recommendation systems primarily focus on post-hoc explanation of highly complex ranking algorithms [39, 34, 12], where explanations are usually feature-based (e.g. highlighting query terms in search snippets [12]) and example-based (e.g. showing similar items that the user liked [39]). Our approach differs from these works in two ways. First, instead of explaining black-box models, we design inherently interpretable models. Second, the proposed approach can not only identify important high-level features, but also show intermediate decision steps.

Precision medicine literature search. This work is inspired by the TREC PM track, where the task is to retrieve articles for cancer treatment planning. Most participating teams in this track employ a machine learning model to rerank documents retrieved by a simpler baseline. Some teams use the official relevance judgement criteria to fine-tune search results, e.g., filtering out documents that are not related to cancer treatment [9] or does not match the demographic information in the query [1]. High-performance reranking methods are often black-box models such as boosted decision trees [7, 32], and deep neural networks [11, 21, 40], which means the decision logic is not interpretable. Here the proposed reranking model emulates the structured relevance judgment process in the TREC PM track, which is interpretable by design.

3 Structured Relevance in TREC PM Track

Since 2017, the TREC Precision Medicine (PM) track has been focusing on a specific type of professional search task in which relevance is *structured*, i.e., defined as a function of different aspects [28]. PM track organizers provided *structured* relevance judgements, where each document is assigned a relevance level (*not relevant*, *partially relevant*, *definitely relevant*) based on intermediate judgements on multiple aspects, as illustrated in Figure 1. Each aspect takes a categorical outcome. For example, regarding the *Disease* aspect, a document may take one of four categories: (1) Exact (i.e., mentions the disease in the query), (2) More general (i.e., mentions a more general disease), (3) More specific (i.e., mentions a more specific disease), or (4) No disease (i.e., does not mention a related disease). All aspects and corresponding outcomes are shown in the first

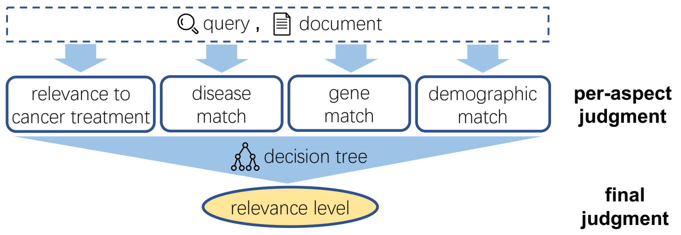


Fig. 1: Structured relevance judgment in the TREC Precision Medicine track.

Table 1: Relevance aspects and classifier features

Aspects	**Outcomes**	**Classifier Features**
Relevance to cancer treatment	Human PM	# "Human PM" keywords (n)
	Animal PM	# "Animal PM" keywords (n)
	Not PM	# "Not PM" keywords (n)
Disease	Exact	# query disease match (n)
	More General	# disease super-category match (n)
	More Specific	# disease sub-category match (n)
	No Disease	
Gene	Exact	# query gene and aliases match (n)
	Missing Gene	is variant in query (b)
	Missing Variant	# query variant match (n)
	Different Variant	# other gene variants match (n)
		is gene modification in query (b)
		# gene modification match (n)
Demographic	Match	is gender mentioned in article (b)
	Exclude	is gender different in article (b)
	Not Discussed	is age mentioned in article (b)
		difference in age (n)
		# age group keywords match (n)

b: binary-valued, #: count of, n: real-valued, PM: $\underline{p}$recision $\underline{m}$edicine
keywords: terms with highest TF-IDF weights from each outcome

two columns Table 1. Given a query, a document's *gold-standard* relevance level is determined by evaluating these intermediate judgements against a pre-defined cascade of rules (i.e., a decision tree). We refer the reader to Roberts et al. [29] for details about the judgment criteria and decision rules in the PM track.

The PM track released 30 queries with 22,642 judged documents in 2017, 50 queries with 22,429 judged documents in 2018, and 40 queries with 18,317 judged documents in 2019 for the subtask of PubMed abstract search. Aspect-level judgments were manually made by oncologists at the University of Texas MD Anderson Cancer Center. Then relevance levels were computed by passing intermediate aspect-level judgments through a pre-defined decision tree. We use these data in this work.

4 An Explainable Retrieval Model

The relevance judgment structure in Figure 1 naturally inspires a new retrieval algorithm. The main idea is as follows. For each aspect, we train a multi-class classifier that predicts the categorical outcome (i.e., the second column in Table 1). Then we feed the predictions to the decision logic to compute document relevance. This approach has the potential to deliver good retrieval performance as it closely resembles the true relevance decision process. It is also highly explainable as its decision steps emulate those of human experts *by design*. Below we describe our implementation of the proposed retrieval algorithm. Its components – aspect classifiers and a decision tree – are learned from data.

4.1 Aspect Classifiers

Each classifier takes aspect-specific features extracted from a query-document pair. The model employs a small set of simple features per aspect (the third column in Table 1). All classifiers are *one-versus-rest* logistic regression models with regularization weight $C = 0.5$. SMOTE algorithm [6] is used to rebalance the severely skewed label distribution in each aspect (e.g., the majority of judged documents are non-relevant to cancer treatment, or *Not PM*).

4.2 Decision Tree

Building the decision tree. Instead of hand-coding the relevance decision logic into a decision tree, we learn the tree from structured relevance judgment data. The manually-assessed aspect outcomes are input features and the relevance level is the target category. We represent all outcomes as binary variables, so that each non-leaf node makes a binary decision on whether an outcome is true or false. Using information gain as the splitting criterion, we can learn a decision tree that achieves nearly 100% accuracy. This is not surprising, since aspect outcomes and relevance levels are known to be related through a simple decision logic. The learned tree structure is illustrated in Figure 2.

Handling predicted outcomes. Such a decision tree assumes *manually-assessed* binary outcomes as inputs. To work as a retrieval component, the same tree should be able to handle classifier-predicted outcomes as inputs. In our context, these are confidence values predicted by our logistic regression models. The original decision process of the tree can be viewed as a ‘walk’ from the root to a leaf, making a binary decision at each non-leaf node. Now given confidence values predicted at each non-leaf node, we propose two ways of ‘taking the walk’:

• *Deterministic walk*: at each node, the walk follows the branch with confidence value of 50% or greater. In the end, the walk will reach a single leaf node, which determines a relevance level.

• *Probabilistic walk*: at each node, the walk will follow either branch with probability equal to the confidence value towards that branch. This (random) walk will reach every leaf node with non-zero probability, *i.e.* the product of all confidence values from the root to the leaf.

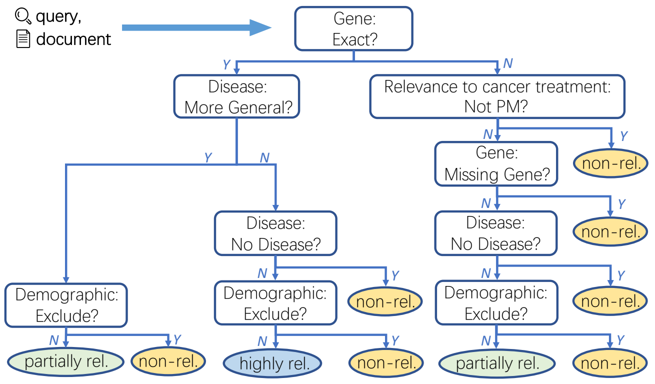


Fig. 2: The learned decision tree structure.

In terms of output, the decision tree predicts a probability distribution $p(r|q,d)$ over relevance levels $r \in \{\textit{not relevant}, \textit{partially relevant}, \textit{definitely relevant}\}$ for a given query-document pair (q,d). The deterministic walk makes a *hard* prediction: $p(r^*|q,d) = 1$ for some r^* and 0 otherwise. We call this approach **Tree-hard**. The probabilistic walk makes a *soft* prediction: it predicts $p(r|q,d)$ as the probability of reaching any leaf associated with relevance level r. We call this approach **Tree-soft**.

Tree-hard and Tree-soft differ in their sensitivity to inaccurate predictions from our aspect classifiers. For Tree-hard, a single prediction error at any node will likely 'sway' the deterministic walk down a wrong path. For Tree-soft, when prediction errors occur, the probabilistic walk will still follow the right path with non-zero probability. In this regard, Tree-soft may have higher tolerance for inaccurate predictions.

These tree-based models offer natural ways of interpreting their decisions. To explain Tree-hard, one can show the single decision path it takes to predict relevance. To explain Tree-soft, one can show k most probable decision paths, each providing an alternative explanation. Upon close inspection, we found that the top-3 most probable paths down the tree account for an average of 80% of the total probability across all paths. In other words, while Tree-soft assigns a non-zero probability to each path, these probabilities tend to be *highly skewed* towards only a few.

Generating a ranking score. To rank documents, we need to generate a score for each (q,d). We use a variant of the approach in Li et al. [19]: $s(q,d) = \left[\sum_{r\in\{0,1,2\}} w_r \cdot p(r|q,d)\right] + b(q,d)$, where the weight w_r should increase with relevance level r. We define $r = 0, 1, 2$ as *not relevant*, *partially relevant*, and *definitely relevant*, respectively, and set $w_0 = 0$, $w_1 = 0.5$ and $w_2 = 1$. The first term $\left[\sum_{r\in\{0,1,2\}} w_r \cdot p(r|q,d)\right]$ is large if $p(r = 2|p,d)$ is large, *i.e.* the decision path unambiguously leads to a *definitely relevant* leaf. $b(q,d) \in [0,1]$ is the min-max scaled score generated from the initial retrieval stage (e.g. BM25). Overall, a large $s(q,d)$ indicates that d is relevant to q in a *clearly interpretable* manner.

5 Leave-One-Year-Out Cross-Validation

In this section, we evaluate the proposed retrieval model on three years of PM track data (2017-2019). We perform leave-one-year-out cross validation, *e.g.*, training on 2017 and 2018 data combined and testing on 2019 data, and so on. Note that although all queries in three years share the same structure as shown in Figure 1, no two queries are identical in all four aspects.

5.1 Initial Retrieval Stage

We first implement a simple initial retrieval stage by concatenating disease and gene terms to generate a search query, and then use the BM25 scoring function implemented in Apache Lucene to retrieve the top 500 documents from a Lucene index of the PubMed corpus. Then, we use tree-based and learning-to-rank models as rerankers after the initial retrieval stage.

5.2 Learning-to-Rank Baselines

To evaluate the reranking performances of the proposed models, we include classical learning-to-rank (LTR) approaches for comparison. LTR models are often highly complex (e.g. neural networks or ensemble models [4]) and make less explainable relevance predictions than the proposed approach.

LTR-high. The first baseline uses the prediction confidence values from our aspect classifiers as features. Again, each aspect has a fixed set of possible outcomes (2nd column in Table 1). Our aspect classifiers output a confidence value per outcome. The LTR-high approach uses a concatenation of these confidence values (14 total) as its input feature vector. We call this approach LTR-high because it uses *high*-level features that *directly* model the outcome of each aspect.

LTR-low. The second baseline uses the raw features used by our four aspect classifiers (3rd column in Table 1). We call this approach LTR-low because it uses *low*-level features that *indirectly* model the outcome of each aspect.

We implement both LTR models with LambdaMART [4] in Lemur RankLib. To obtain the strongest baselines, we perform grid search for hyperparameters of each LTR model to maximize its precision@10 on 5-fold cross validation. These hyperparameters include the number of trees, the number of leaves in each tree, learning rate, and minimum leaf support.

5.3 Results

Three metrics were used to evaluate ranking performance: precision@10 (P@10), which focuses on precision at top ranks; R-precision (R-prec) and mean average precision (MAP), which emphasize both recall and precision. Table 2 shows evaluation results for the above algorithms when training on data from two out of three years and testing on the held-out year. We also included Lucene's BM25 baseline for comparison. When comparing approaches, we tested for statistical significance using Fisher's Randomization Test [35] (α = .05).

Table 2: Evaluation Results in terms of P@10, MAP and R-prec in three years. A ▲(▼), △(▽), and ∧(∨) denotes significantly better(worse) performance when comparing Tree-soft vs. Tree-hard, LTR-low vs. LTR-high, and Tree-soft vs. LTR-low, respectively.

	2017			**2018**			**2019**		
Method	**P@10**	**MAP**	**R-prec**	**P@10**	**MAP**	**R-prec**	**P@10**	**MAP**	**R-prec**
BM25	0.4100	0.1437	0.2374	0.5360	0.2273	0.3122	0.4550	0.1704	0.2394
LTR-high	0.4567	0.1433	0.2156	0.5080	0.2070	0.2864	0.4850	0.1533	0.2215
LTR-low	0.5330△	0.1780△	0.2616△	0.6240△	0.2530△	0.3281△	0.5200	0.1888△	0.2667△
Tree-hard	0.4200	0.1437	0.2321	0.5520	0.2284	0.3098	0.4400	0.1707	0.2422
Tree-soft	0.4333∨	0.1661▲	0.2551▲	0.6220▲	0.2622▲∧	0.3496▲∧	0.5100▲	0.1986▲	0.2736▲

Tree-soft vs. Tree-hard ▲(▼). First, we compare between two tree-based approaches (Section 4.2). Except for P@10 in 2017 ($p = .555$), Tree-soft outperformed Tree-hard across all years and metrics by a significant margin ($p < .001$). This result suggests an important trend—when traversing the "relevance decision tree" using *predicted* (vs. gold-standard) relevance aspects, it is better to traverse the tree *probabilistically* (i.e., using prediction confidence values) than to follow the single most confident path to a leaf node.

LTR-low vs. LTR-high △(▽). Next, we compare between two LTR-based approaches (Section 5.2). Except for P@10 in 2019 ($p = .248$), LTR-low outperformed LTR-high across all years and metrics by a significant margin ($p < .001$). Interestingly, an LTR-based approach performed better with low-level features than high-level relevance aspects predicted by our aspect classifiers (Section 4.1).

Tree-soft vs. LTR-low ∧(∨). Finally, we compare between the better tree-based approach (Tree-soft) and the better LTR-based approach (LTR-low). In terms of P@10, Tree-soft performed significantly *worse* than LTR-low when testing on 2017 ($p < .005$). However, Tree-soft performed at the same level as LTR-low (i.e., no significant differences) when testing on 2018 and 2019. Note that LTR-low was expected to deliver a high P@10 because it was trained to optimize that metric, while Tree-soft was not. In terms of MAP and R-prec, Tree-soft performed at the same level as LTR-low across all years, and significantly better when testing on 2018 (MAP: $p < .005$; R-prec: $p < .005$).

Overall, the Tree-soft approach is consistently better than the Tree-hard approach, and its performance is comparable to LTR-low. This is no small feat considering that the Tree-soft approach is a much simpler (more interpretable) approach than the LTR-low approach which is a ensemble model using 500 decision trees in 2017 and 2018, and 1000 decision trees in 2019.

6 Replacing Black-Box Rerankers in TREC PM Track

A common approach in the TREC PM track has been to employ LTR models to rerank the top results produced by a simpler baseline [5, 7, 11, 21, 27, 32, 40]. While such an approach can effectively improve performance, the reranking model is often complex and not easily interpretable. Our goal in this section is to explore if these complex rerankers can be replaced with the proposed interpretable model without sacrificing performance.

6.1 Selecting Runs From Top-Ranked Teams

We compare the Tree-soft reranker (the better one in Section 5) against competitive TREC PM teams in 2019 (teams on the left side of Table 6 in [29]) that used a complex LTR model to rerank the retrieved results from a simpler baseline *and* submitted a "run" (i.e., result list) using that baseline. The initial retrieval baselines are denoted as **Original Baseline** and the corresponding reranking results are denoted as **Original Reranking**.

We identified six such teams. Table 3 shows the identified runs and reranking techniques. Teams that only submitted reranking results (e.g. BITEM_PM [5]) or did not use LTR reranking (e.g., imi_mug [22], CincyMedIR [38], ims_unipd [23]) were not selected. We also did not select runs from the julie-mug team as their baseline query has been extensively fine-tuned on previous relevance judgment data [10], and further reranking does not help [9].

For each selected team, we use the Tree-soft model (denoted as **Tree-soft Reranking**) to rerank the same initial results produced by the team's baseline. Components in Tree-soft are trained on 2017 and 2018 data combined, as all participants in 2019 had access to these data. This allows us to make head-to-head comparisons between Tree-soft and the rerankers used by these teams. In these comparisons, we vary the reranker but hold all other factors constant.

Table 3: Selected runs from PM 2019 submissions.

Team	**Baseline Run**	**Reranking Run**	**Reranking Technique**
POZNAN [7]	SAsimpleLGD	SA_LGD_letor	LambdaMART in Terrier [24]
CSIROmed [32]	bm25_6801	et_8435	Extremely randomized trees [13]
ECNU [40]	sa_base	sa_base_rr	Doc2Vec + cosine similarity [17]
CCNL [21]	ccnl_sa5	ccnl_sa4	SciBERT [3]
DUTIR [11]	DutirRun1	DutirRun3	Deep semantic matching [14, 15]
UNC_SILS [27]	sils_run1	sils_run3	logistic regression pointwise LTR

6.2 Results

Table 4 summarizes our results in terms of P@10, MAP, and R-precision. We compare the Tree-soft model against each team's original baseline and each team's reranker. When comparing approaches, we used Fisher's Randomization Test [35] ($\alpha = .05$) to test for statistical significance.

Tree-soft Reranking vs. Original Baseline ▲(▼). In terms P@10, the Tree-soft model performed at the same level as all baselines. In terms MAP and

Table 4: Reranking 2019 PM track submissions with Tree-soft. A $\blacktriangle(\blacktriangledown)$, and $\triangle(\triangledown)$ denotes significantly better (worse) performance when comparing Tree-soft vs. Original Baseline and Tree-soft vs. Original Reranking, respectively.

	Original Baseline			**Original Reranking**			**Tree-soft Reranking**		
Team	**P@10**	**MAP**	**R-prec**	**P@10**	**MAP**	**R-prec**	**P@10**	**MAP**	**R-prec**
POZNAN	0.5400	0.2603	0.3092	0.5050	0.2117	0.2714	0.5700$^{\triangle}$	0.2542$^{\triangle}$	0.3240$^{\blacktriangle\triangle}$
CSIROmed	0.5250	0.2499	0.3029	0.5725	0.1279	0.1856	0.5375	0.2470$^{\triangle}$	0.3153$^{\blacktriangle\triangle}$
ECNU	0.5600	0.1767	0.2608	0.5600	0.1769	0.2610	0.5625	0.1848$^{\blacktriangle\triangle}$	0.2690$^{\blacktriangle\triangle}$
CCNL	0.5025	0.2197	0.2770	0.5775	0.2279	0.2886	0.5275	0.2154	0.2944$^{\blacktriangle}$
DUTIR	0.5800	0.2774	0.3266	0.5825	0.2775	0.3227	0.5775	0.2647$^{\blacktriangledown\triangledown}$	0.3261
UNC_SILS	0.5225	0.2216	0.2858	0.5925	0.2216	0.2757	0.5625	0.2278	0.3124$^{\blacktriangle\triangle}$

R-prec, Tree-soft performed significantly better in six cases and significantly worse in one case (DUTIR in terms of MAP). It should be noted that the difference in MAP compared to the DUTIR baseline is significant but small.

Tree-soft Reranking vs. Original Reranking $\triangle(\triangledown)$. In terms of P@10, the Tree-soft model performed significantly better than one reranker (i.e., POZNAN) and at the same level as the other rerankers. In terms of MAP and R-prec, the tree soft models performs significant better than seven cases and significantly worse in one case (DUTIR in terms of MAP). Again, it should be noted that the difference in MAP compared to the DUTIR reranker is significant but small.

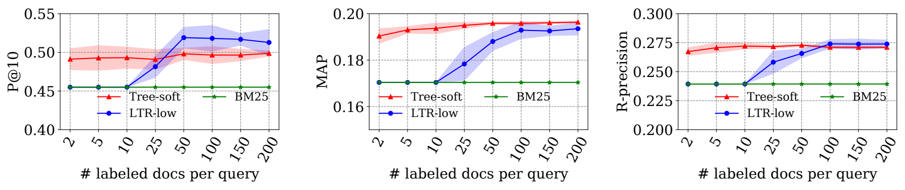
The results in Table 4 show more significant differences in MAP and R-prec than P@10. One possible explanation is that P@10 is a coarser metric (i.e., only 11 possible values). Another explanation is that P@10 considers only the top-10 results while MAP and R-prec consider additional results at lower ranks.

These results and those in Section 5.3 suggest that many black-box LTR models in this domain could be replaced by a simple and interpretable model without significant loss of performance, and in some cases even with performance gain. This finding is encouraging as medical literature search is a high-stake application that warrants an inherently interpretable retrieval model [30].

7 Learning Curve Analysis

The above experiments show that Tree-soft is an effective, explainable, and reusable reranker that performs as well as many black-box rerankers. However, such a model may seem more "data-hungry" than learning-to-rank approaches as it requires human experts to provide aspect-level judgments, which are more elaborate than relevance judgments. To label a document, an expert must first read it carefully to produce aspect-level labels, and then give the final relevance level. This is a time- and effort-consuming task. Therefore it would be ideal if the Tree-soft approach can achieve high performance without requiring a medical expert to label many documents. In this section, we investigate this problem through a learning curve analysis. Here, a learning curve shows the performance of a model when the amount of training data varies from small to large.

We sampled training documents from the 80 topics in 2017 and 2018 datasets combined, and tested on the 40 topics in 2019 dataset. For Tree-soft, we held the tree structure fixed as the decision logic is a piece of prior knowledge. We only re-trained our four aspect-level classifiers with newly sampled data. We include LTR-low for comparison as it is the stronger LTR model in Section 5. Note that Tree-soft and LTR-low may sample different documents for the same training data size. The training data for aspect-level classifiers in Tree-soft were sampled from the official relevance judgments, while those for LTR-low are sampled from initial retrieval results. Every time a new training dataset was sampled, we performed grid search of optimal hyperparameters for LTR-low with the same steps described in Section 5.2. At each data size, We took the mean and standard deviation of performance metrics over 8 random samples. To simulate settings where labeled documents are extremely scarce, we sampled two documents per query (one relevant and one non-relevant).

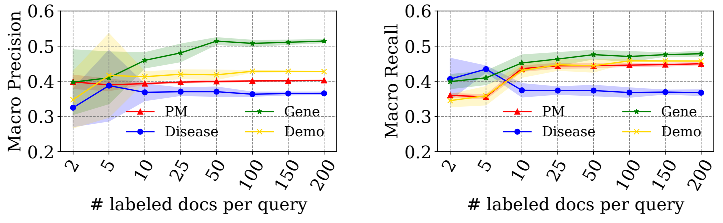


(a) P@10 (b) MAP (c) R-precision

Fig. 3: Learning curves of Tree-soft and LTR-low in terms of P@10, MAP, and R-prec. Color regions correspond to ±1 standard deviation around the mean.

Figure 3 shows the average P@10, MAP, and R-precision of Tree-soft and LTR-low when they are trained on increasing amounts of data. BM25 is included for comparison. Although Tree-soft and LTR-low learns from two kinds of labels (aspect-level labels for Tree-soft and document-level labels for LTR-low), both were generated through the same structured relevance judgment process (shown in Figure 1) for every document. Therefore the number of labeled documents per query can be used as a unified measure of learning cost.

We observe that despite small training data sizes (# labeled documents per query = 2, 5, 10), Tree-soft performs surprisingly well and enjoys a large margin in all metrics compared to BM25 and LTR-low. In contrast, LTR-low does not perform well when the training data is small, due to severe overfitting of the LamdaMART reranker. In recall-oriented metrics (R-precision and MAP), Tree-soft performed on par with (if not better than) LTR-low across all training data sizes. In terms of P@10 (for which LTR-low was optimized), LTR-low outperformed Tree-soft only when more than 25 documents/query (2,000 documents for 80 queries) are labeled, which is a huge burden for manual annotation.



(a) Macro-averaged precision (b) Macro-averaged recall

Fig. 4: Learning curves of aspect classifiers in terms of macro-averaged precision and recall. Color regions correspond to ±1 standard deviation around the mean.

We provide two explanations for this phenomenon. First, our aspect classifiers are simple linear models using a small set of features and therefore can be efficiently trained. The learning curves in Figure 4 show that the classifiers reach stable performance after receiving only 10 labeled documents per query. The large performance variance in the early stage is due to small training data. In particular, the *Disease* classifier suffered a performance drop because its outcome label distribution is severely imbalanced, and increasing the data size exacerbates label imbalance. A second possible explanation for the Tree-soft model being robust with less training data stems from its pre-specified decision tree structure. Importantly, the structure encodes a piece of prior knowledge that LTR-low is completely unaware of: the correct decision-making procedure to combine (possibly noisy) aspect-level relevance into document-level relevance.

Overall, these results highlight the promise of leveraging the structure of professional search queries in a retrieval model – it makes the model not only more interpretable, but also more robust in handling noisy inputs and less hungry for training data.

8 Conclusion

In this paper, we analyzed a recently proposed explainable retrieval model that closely resembles a structured relevance judgment process, where the search query involves multiple aspects and document relevance is decided by a logical function of these aspects. Extensive experiments on TREC precision medicine track data show that the simple, explainable model can perform as well as many complex, black-box learning-to-rank models, and achieve high performance with much fewer labeled documents. These results point to a promising direction towards building effective, explainable, label-efficient retrieval algorithms for professional search tasks. In future work, we will evaluate the interpretability of the proposed retrieval model in prototype systems and user studies.

Acknowledgment. UNC SILS Kilgour Research Grant supported this work.

References

1. Agosti, M., Nunzio, G.M.D., Marchesin, S.: The university of padua ims research group at trec 2018 precision medicine track (2018)
2. Aromataris, E., Riitano, D.: Systematic reviews: constructing a search strategy and searching for evidence. The American Journal of Nursing **114**(5), 49–56 (2014)
3. Beltagy, I., Lo, K., Cohan, A.: Scibert: A pretrained language model for scientific text. arXiv preprint arXiv:1903.10676 (2019)
4. Burges, C.J.: From ranknet to lambdarank to lambdamart: An overview. Learning **11**(23-581), 81 (2010)
5. Caucheteur, D., Pasche, E., Gobeill, J., Mottaz, A., Mottin, L., Ruch, P.: Designing retrieval models to contrast precision-driven ad hoc search vs. recall-driven treatment extraction in precision medicine. In: TREC (2019)
6. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: synthetic minority over-sampling technique. J. of Artificial Intelligence Res. **16**, 321–357 (2002)
7. Cieslewicz, A., Dutkiewicz, J., Jedrzejek, C.: Poznan contribution to TREC-PM 2019. In: TREC (2019)
8. Dang, V., Bendersky, M., Croft, W.B.: Two-stage learning to rank for information retrieval. In: European Conference on Information Retrieval. pp. 423–434 (2013)
9. Faessler, E., Hahn, U., Oleynik, M.: Julie lab & med uni graz@ trec 2019 precision medicine track. In: TREC (2019)
10. Faessler, E., Oleynik, M., Hahn, U.: What makes a top-performing precision medicine search engine? tracing main system features in a systematic way. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval. p. 459–468 (2020)
11. Feng, J., Yang, Z., Liu, Z., Luo, L., Lin, H., Wang, J.: Dutir at trec 2019: Precision medicine track. In: TREC (2019)
12. Fernando, Z.T., Singh, J., Anand, A.: A study on the interpretability of neural retrieval models using deepshap. In: SIGIR 2019. p. 1005–1008. ACM, New York, NY, USA (2019)
13. Geurts, P., Ernst, D., Wehenkel, L.: Extremely randomized trees. Machine learning **63**(1), 3–42 (2006)
14. Guo, J., Fan, Y., Ai, Q., Croft, W.B.: A deep relevance matching model for adhoc retrieval. In: Proceedings of the 25th ACM International on Conference on Information and Knowledge Management. pp. 55–64 (2016)
15. Hui, K., Yates, A., Berberich, K., de Melo, G.: PACRR: A position-aware neural IR model for relevance matching. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. pp. 1049–1058. Association for Computational Linguistics, Copenhagen, Denmark (Sep 2017)
16. Kanoulas, E., Li, D., Azzopardi, L., Spijker, R.: Clef 2019 technology assisted reviews in empirical medicine overview. In: CEUR Workshop Proceedings. vol. 2380
17. Le, Q., Mikolov, T.: Distributed representations of sentences and documents. In: International conference on machine learning. pp. 1188–1196 (2014)
18. LexisNexis: Developing a search with lexisnexis (Accessed September 2020), `http://www.lexisnexis.com/bis-user-information/docs/developingasearch.pdf`
19. Li, P., Wu, Q., Burges, C.J.: Mcrank: Learning to rank using multiple classification and gradient boosting. In: Advances in neural information processing systems. pp. 897–904 (2008)
20. Liu, T.Y.: Learning to rank for information retrieval. Springer Science & Business Media (2011)

21. Liu, X., Li, L., Yang, Z., Dong, S.: Scut-ccnl at trec 2019 precision medicine track. In: TREC (2019)
22. López-Úbeda, P., Vera-Ramos, J.A., López-García, P.: Trec 2019 precision medicine - medical university of graz. In: TREC (2019)
23. Nunzio, G.M.D., Marchesin, S., Agosti, M.: Exploring how to combine query reformulations for precision medicine. In: TREC (2019)
24. Ounis, I., Amati, G., Plachouras, V., He, B., Macdonald, C., Johnson, D.: Terrier information retrieval platform. In: European Conference on Information Retrieval. pp. 517–519. Springer (2005)
25. O'Mara-Eves, A., Thomas, J., McNaught, J., Miwa, M., Ananiadou, S.: Using text mining for study identification in systematic reviews: a systematic review of current approaches. Systematic reviews **4**(1), 5 (2015)
26. Qu, J., Arguello, J., Wang, Y.: Towards explainable retrieval models for precision medicine literature search. In: Proceedings of the 43rd ACM SIGIR Conference on Research and Development in Information Retrieval. p. 1593–1596 (2020)
27. Qu, J., Wang, Y.: UNC SILS at trec 2019 precision medicine track. In: TREC (2019)
28. Roberts, K., Demner-Fushman, D., Voorhees, E.M., Hersh, W.R., Bedrick, S., Lazar, A.J., Pant, S.: Overview of the trec 2017 precision medicine track. (2017)
29. Roberts, K., Demner-Fushman, D., Voorhees, E.M., Hersh, W.R., Bedrick, S., Lazar, A.J., Pant, S., Meric-Bernstam, F.: Overview of the trec 2019 precision medicine track. In: TREC (2019)
30. Rudin, C.: Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. Nature Machine Intelligence **1**(5), 206–215 (2019)
31. Russell-Rose, T., Chamberlain, J., Azzopardi, L.: Information retrieval in the workplace: A comparison of professional search practices. Information Processing & Management **54**(6), 1042–1057 (2018)
32. Rybinski, M., Karimi, S., Paris, C.: Csiro at 2019 trec precision medicine track. In: TREC (2019)
33. Schardt, C., Adams, M.B., Owens, T., Keitz, S., Fontelo, P.: Utilization of the pico framework to improve searching pubmed for clinical questions. BMC medical informatics and decision making **7**(1), 16 (2007)
34. Singh, J., Anand, A.: Posthoc interpretability of learning to rank models using secondary training data. arXiv:1806.11330 (2018)
35. Smucker, M.D., Allan, J., Carterette, B.: A comparison of statistical significance tests for information retrieval evaluation. In: CIKM. p. 623–632 (2007)
36. Tian, A., Lease, M.: Active learning to maximize accuracy vs. effort in interactive information retrieval. In: Proceedings of the 34th ACM SIGIR conference on Research and development in Information Retrieval. pp. 145–154 (2011)
37. Wallace, B.C., Trikalinos, T.A., Lau, J., Brodley, C., Schmid, C.H.: Semi-automated screening of biomedical citations for systematic reviews. BMC bioinformatics **11**(1), 1–11 (2010)
38. Wu, D.T.Y., Su, W., Lee, J.J.: Retrieving scientific abstracts using venue- and concept-based approaches: Cincymedir at TREC 2019 precision medicine track. In: TREC (2019)
39. Zhang, Y., Chen, X.: Explainable recommendation: A survey and new perspectives. arXiv preprint: 1804.11192 (2018)
40. Zheng, Q., Li, Y., Hu, J., Yang, Y., He, L., Xue, Y.: ECNU-ICA team at TREC 2019 precision medicine track. In: TREC (2019)