Towards Explainable Retrieval Models for Precision Medicine Literature Search

Jiaming Qu, Jaime Arguello, Yue Wang
School of Information and Library Science, University of North Carolina at Chapel Hill
jiaming@live.unc.edu,{jarguello,wangyue}@email.unc.edu

Abstract

In professional search tasks such as precision medicine literature search, queries often involve multiple aspects. To assess the relevance of a document, a searcher often painstakingly validates each aspect in the query and follows a task-specific logic to make a relevance decision. In such scenarios, we say the searcher makes a *structured* relevance judgment, as opposed to the traditional univariate (binary or graded) relevance judgment. Ideally, a search engine can support searcher's workflow and follow the same steps to predict document relevance. This approach may not only yield highly effective retrieval models, but also open up opportunities for the model to explain its decision in the same 'lingo' as the searcher. Using structured relevance judgment data from the TREC Precision Medicine track, we propose novel retrieval models that emulate how medical experts make structured relevance judgments. Our experiments demonstrate that these simple, explainable models can outperform complex, black-box learning-to-rank models.

ACM Reference Format:
Jiaming Qu, Jaime Arguello, Yue Wang. 2020. Towards Explainable Retrieval Models for Precision Medicine Literature Search. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '20), July 25–30, 2020, Virtual Event, China.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3397271.3401277

1 Introduction

Consider a clinician searching the literature to plan on personalized treatments for a patient, a 50-year-old man diagnosed with leukemia with genetic variation on BRAF (V600R). Ideally, the clinician could locate an article on leukemia treatment where the studied subjects have the same characteristics as the patient. In reality, however, the search would more likely return articles where the subjects match the patient on some but not all characteristics (e.g., same disease and genetic variation but different gender and age group). Such articles may still be relevant if the matched characteristics are clinically more important than the unmatched, as judged by the clinician. In such scenarios, the relevance judgment criteria can be described as a cascade of rules [7]. If we represent these rules as a decision tree (Figure 1), then we can represent the relevance judgments as a path from the root to a leaf.

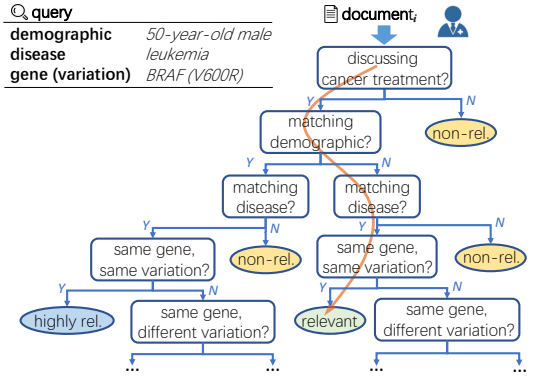

Figure 1: An illustration of relevance judgment decision process in precision medicine literature search.

The above scenario is an example of professional search tasks [9]. These also include tasks like patent search, recruitment search, and systematic review. In these tasks, the search criteria are expressed as a logical function of various high-level aspects. When assessing the relevance of a document, the searcher would carefully check each aspect and deliberate on the overall relevance of a document.

Ideally, a search system can understand the searcher's information need in terms of high-level aspects and reason about result relevance using the same logic. This would allow the system to explain its decision using the same 'lingo' as the searcher. Indeed, previous work has shown that professional searchers value transparency more than pure ranking performance [9]. However, current search systems barely provide transparency in support of these tasks. The simplistic approach of highlighting matched terms in result summaries is viable only for explaining ad-hoc search results. Modern ranking algorithms are highly complex and do not necessarily follow the reasoning process of professionals. Though it is possible to generate post-hoc explanations for these 'black boxes' [10], such explanations may still be unreliable and misleading [8].

In this paper, we explore retrieval models that closely follow the work process of professional searchers. In this respect, our proposed models are *inherently* explainable. Empirical experiments show that they can outperform complex learning-to-rank approaches. Our result suggests a promising direction towards building retrieval models that can better support professional search tasks.

2 Structured Relevance Judgment

Above, we describe search scenarios in which relevance is defined as a function of different aspects or criteria. To develop systems for such search scenarios, the ideal data should include not only relevance judgements for query-document pairs, but also intermediate judgements about different relevance *aspects* (or criteria), and explanations about how these were combined to derive a final relevance judgement. Such data are commonly seen in user studies

SIGIR '20, July 25–30, 2020, Virtual Event, China

ACM ISBN 978-1-4503-8016-4/20/07…$15.00
https://doi.org/10.1145/3397271.3401277

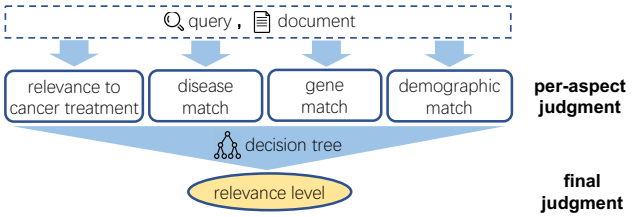

Figure 2: Structured relevance judgment employed in TREC Precision Medicine track.

(e.g., from comments gathered using think-aloud protocols [12]). However, these are rarely considered in batch evaluation set-ups. As one exception, the TREC Precision Medicine (PM) track has been releasing such data since 2017 [7]. This TREC track considers search scenarios such as the one in Section 1, in which relevance is defined logically based on different aspects or criteria.

PM track organizers provide *structured* relevance judgements, where each document is assigned a relevance level (i.e., *not relevant*, *partially relevant*, *definitely relevant*) based on intermediate judgements on multiple aspects, as illustrated in Figure 2. Each aspect takes a categorical outcome. For example, regarding the *Disease* aspect, a document may take one of these 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 left two columns of 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. [7] for details about the judgments criteria and decision rules in the PM track.

The PM track released 30 queries with 22,642 judged documents in 2017, and 50 queries with 22,429 judged documents in 2018, for the subtask of PubMed abstract search. Intermediate judgments are manually made by oncologists from the University of Texas MD Anderson Cancer Center. Then relevance levels are computed by passing intermediate judgments through a pre-defined decision tree. We use these data in this paper.

3 Proposed Retrieval Algorithm

The relevance judgment structure in Figure 2 naturally inspires a new retrieval algorithm as follows. For each aspect, we train a multi-class classifier that predicts categorical outcomes (i.e., middle column in Table 1). Then we feed the predictions to the pre-defined decision tree to compute a relevance level. This approach has the potential to deliver good retrieval performance as it closely follows 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. We use the 30 queries in 2017 PM track to train these components, and the 50 queries in 2018 PM track for experimental evaluation (Section 4).

3.1 Aspect classifiers

Input features. Each classifier takes aspect-specific features extracted from a query-document pair. In this preliminary work, we employ a small set of features per aspect (right column in Table 1).

Table 1: Relevance aspects and classifier features

Aspects	Outcomes	Classifier Features
Relevance to cancer treatment	Human PM Animal PM Not PM	# Human PM keywords (n) # Animal PM keywords (n) # Not PM keywords (n)
Disease	Exact More General More Specific No Disease	# query disease match (n) # disease descendants match (n) # disease ancestors match (n)
Gene	Exact Missing Gene Missing Variant Different Variant	# query gene & aliases match (n) is variation in query (b) # query variation match (n) # other variations match (n) is other info in query (b) # other info match (n)
Demographic	Match Excludes Not Discussed	is gender mentioned in article (b) is gender different in article (b) is age mentioned in article (b) difference in age (n)

b: binary-valued, #: count of, n: real-valued, PM: precision medicine

Cancer Treatment Relevance classifier has three feature categories, each counting terms that indicate an outcome. These are selected by taking top 20 terms with highest TF-IDF weights in documents associated with each outcome, as well as terms used in Oleynik et al. [6]. *Disease* classifier has three feature categories, each counting terms that correspond to synonyms, descendants, and ancestors of the disease in the query. We use the Lexigram API to map disease relations. For *Gene* classifier, we use the NCBI gene database to expand aliases and the PMKB database to expand variations, and include counts of both original and expanded gene and variation terms as features. Not every query gene comes with a variation, which we indicate using a binary feature. We also check whether the query gene has other information like amplification or deletion, and count the match. For *Demographic* classifier, we first detect whether a document mentions any gender or age information, then check whether the information matches that in the query.

Classification Models. All classifiers are *one-versus-rest* logistic regression models with regularization weight $C = 0.5$. During the manual assessment process, documents unrelated to cancer treatment were considered *not relevant* and all other aspects were not further assessed. To distinguish between "missing judgments" and "negative examples", these documents were excluded when training *Disease*, *Gene*, and *Demographic* classifiers. Table 2 summarizes classifier performance. These performance numbers are not high but still reasonable, considering severely skewed label distributions (e.g., the majority of judged documents are non-relevant to cancer treatment, or *Not PM*) and relatively simple feature sets.

Table 2: Aspect classifier performance

Aspect	Macro-F1	Accuracy
Relevance to cancer treatment	0.45	0.58
Disease	0.46	0.59
Gene	0.41	0.46
Demographic	0.48	0.74

3.2 Decision tree

Building the decision tree. Instead of hand-coding the pre-defined cascade of rules into a decision tree, we (re)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 learned a decision tree that achieved nearly 100% accuracy. This is not surprising, since aspect outcomes and relevance levels are known to be related through a simple decision logic. The tree encodes this decision logic with 13 non-leaf nodes and 14 leaf nodes, i.e. 14 root-to-leaf decision paths (the longest has 5 internal decisions). Notably, it learns that if a document is not about cancer treatment, then it is *not relevant* regardless of other aspect outcomes.

Handling predicted outcomes. The above decision tree assumes *manually assessed* binary outcomes as inputs. To work as a retrieval component, the tree should be able to handle classifier-predicted outcomes as inputs. In our context, these are confidence values (i.e., $p(y = 1|x)$) predicted by 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 reaches a single leaf node, which determines a relevance level.
- *Probabilistic walk*: at each node, the walk follows either branch with probability equal to the confidence value towards that branch. This random walk reaches every leaf node with non-zero probability, *i.e.* the product of all confidence values from the root to a leaf.

In terms of output, the decision tree predicts a probability distribution $p(r|q, d)$ over relevance levels $r \in \{$*not relevant*, *partially relevant*, *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, especially if these correspond to low-confidence misclassifications (i.e., $p(y = 1|x) \approx 0.5$).

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. [5]: $s(q, d) = \left[\sum_{r\in\{0,1,2\}} w_r \cdot p(r|q, d)\right] + b(q, d)$, 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 by the initial retrieval algorithm (e.g. BM25). Overall, a large $s(q, d)$ expresses that d is relevant in a clearly interpretable manner.

4 Experimental Evaluation

4.1 Initial retrieval stage

All compared methods take query-document pairs as input and predict ranking scores as output, working as rerankers after an initial retrieval stage. We implement a simple initial retrieval stage as it is orthogonal to the comparison of rerankers. For each topic, we concatenate disease and gene terms to generate a search query, and then use the BM25 scoring function to retrieve the top 500 documents for reranking. We used Lucene to index TREC 2017/18 PM track corpus (26.7M medical abstracts) and perform BM25 scoring.

4.2 Learning-to-rank baselines

To put the performance of the proposed approach in perspective, we compare it with classical learning-to-rank (LTR) approaches described below. In terms of explainability, LTR models are often highly complex (e.g. neural networks or large ensembles of base models[3]). Due to their complexity, LTR models make less explainable relevance predictions than the proposed approach.

LTR-high. The first baseline replaces the simple decision tree in the proposed approach by a more expressive LTR model. It takes classifier-predicted outcomes (second column in Table 1) and BM25 score as its features and predicts a ranking score. In other words, aspect classifiers extract high-level query-document features.

LTR-low. The second baseline takes aspect features (third column in Table 1) and BM25 score as LTR features and predicts a ranking score. Instead of using aspect classifiers as feature extractors, this monolithic LTR model directly works with low-level features.

LTR-high needs classifier-predicted outcomes as features for both training and evaluation. We generate these predictions on training data using 5-fold cross validation, and generate them on test data by applying the classifiers trained on all training data.

Both LTR models were trained using the implementation of LambdaMART [3] available in the RankLib toolkit. To obtain the strongest baselines, we set the hyperparameters of each LTR model to those that maximize its mean average precision on 5-fold cross validation. We performed grid search for the following hyperparameters: number of trees, number of leaves in each tree, learning rate, and minimum leaf support. The resultant optimal models are highly complex (1,900 trees for LTR-high; 700 trees for LTR-low).

4.3 Results

We use three metrics 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 3 shows results for the above algorithms in terms of P@10, R-prec, MAP. For comparison, we also show results of BM25. When comparing approaches, we tested for statistical significance using Fisher's Randomization Test [11] ($\alpha = .05$).

Tree-hard vs. Tree-soft. First, we compare between tree-based approaches (Section 3.2). The Tree-soft approach outperformed the Tree-hard approach by a significant margin in all three metrics ($p < .001$). 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.

Table 3: Evaluation Results of P@10, R-prec, and MAP. Statistically significant differences discussed in the text.

Method	P@10	R-prec	MAP
BM25	0.5360	0.3122	0.2273
LTR-high	0.5440	0.2979	0.2202
LTR-low	0.5880	0.3310	0.2419
Tree-hard	0.5460	0.3232	0.2378
Tree-soft	0.6220	0.3463	0.2605

LTR-low vs. LTR-high. Next, we compare between LTR-based approaches (Section 4.2). LTR-low outperformed LTR-high in terms of all three metrics. However, the differences were significant only in R-prec and MAP ($p < .01$) and not significant in P@10 ($p = .121$). Interestingly, an LTR-based approach (using LambdaMART) performed better with low-level features than the high-level relevance aspects predicted by our aspect classifiers (Section 3.1).

Tree-soft vs. LTR-low. Finally, we compare between the best tree-based approach (Tree-soft) and the best LTR-based approach (LTR-low). Here, the Tree-soft approach outperformed the LTR-low approach with a significant margin in all three metrics ($p < .01$). It is important to note that the Tree-soft approach is a much simpler (more interpretable) approach than the LTR-low approach.

4.4 Discussion

The comparison between the best tree-based model (Tree-soft) and the best LTR model (LTR-low) shows that a simple, inherently interpretable model can outperform a complex black-box model, in both precision- and recall-oriented metrics. Notably, Tree-soft achieves comparable P@10 and R-prec as high-ranking teams in the 2018 PM track, most of which used sophisticated reranking strategies [7]. This is an encouraging result. It implies that retrieval models do not need to sacrifice performance in exchange for interpretability in structured relevance retrieval tasks.

The tree-based approaches 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 account for 62% 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.

5 Related Work

Explainable information retrieval. Recent works on explainable search and recommendation systems primarily focus on post-hoc explanation of highly complex ranking algorithms [4, 10, 13], where explanations are usually feature-based (e.g. highlighting query terms in search snippets [4]) and example-based (e.g. showing similar items that the user liked [13]). This work differs from previous works in two ways. First, instead of explaining black-box models, we design inherently interpretable models. Second, our 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 Precision Medicine track, where the task is to retrieve articles for cancer treatment planning. Most participating teams in this track used the classical IR approach with query expansion to improve recall and a reranker to refine precision. 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 [6] or do not match the demographic information in the query [1]. Also, high-performance reranking methods are black-box models (e.g., deep neural networks [14]), which means the decision logic is not interpretable. To the best of our knowledge, we are the first to propose a retrieval model that emulates the structured relevance judgment process in the PM track, which is interpretable by design.

Professional search strategies. In professional search tasks, especially systematic literature reviews, searchers formulate structured information need 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, 9]. Our approach aims to automate and assist in these tasks by replacing manually constructed query components with classifiers trained using machine learning, and by logically explaining predictions using relevance aspects.

6 Conclusion and Future Work

In this preliminary work, we demonstrated that simple retrieval models that resemble a *structured* relevance judgment process can outperform strong learning-to-rank baselines, while allowing intuitive explanation. This work inspires many future directions. In future work, we will 1) evaluate the interpretability of the proposed approach both empirically and in user studies, 2) improve aspect classifiers with more features and model options, and 3) explore more powerful implementations of the initial retrieval stage.

Acknowledgment. We thank the anonymous reviewers for their constructive comments. This work is supported by UNC SILS Kilgour Research Grant.

References

[1] Maristella Agosti, Giorgio Maria Di Nunzio, and Stefano Marchesin. 2018. The University of Padua IMS Research Group at TREC 2018 Precision Medicine Track.
[2] Edoardo Aromataris and Dagmara Riitano. 2014. Systematic reviews: constructing a search strategy and searching for evidence. *Am. J. Nurs.* 114, 5 (2014), 49–56.
[3] Chris J.C. Burges. 2010. *From RankNet to LambdaRank to LambdaMART: An Overview.* Technical Report MSR-TR-2010-82.
[4] Zeon Trevor Fernando, Jaspreet Singh, and Avishek Anand. 2019. A Study on the Interpretability of Neural Retrieval Models Using DeepSHAP. In *SIGIR 2019*. ACM, New York, NY, USA, 1005âĂŞ1008.
[5] Ping Li, Qiang Wu, and Chris J.C. Burges. 2008. McRank: Learning to rank using multiple classification and gradient boosting. In *Advances in neural information processing systems*. 897–904.
[6] Michel Oleynik, Erik Faessler, Ariane Morassi Sasso, Arpita Kappattanavar, Benjamin Bergner, Harry Freitas Da Cruz, Jan-Philipp Sachs, Suparno Datta, and Erwin Böttinger. 2018. HPI-DHC at TREC 2018 Precision Medicine Track.
[7] Kirk Roberts, Dina Demner-Fushman, Ellen M Voorhees, William R Hersh, Steven Bedrick, and Alexander J Lazar. 2018. Overview of the TREC 2018 Precision Medicine Track.
[8] Cynthia Rudin. 2019. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence* 1, 5 (2019), 206–215.
[9] Tony Russell-Rose, Jon Chamberlain, and Leif Azzopardi. 2018. Information retrieval in the workplace: A comparison of professional search practices. *Information Processing & Management* 54, 6 (2018), 1042–1057.
[10] Jaspreet Singh and Avishek Anand. 2018. Posthoc interpretability of learning to rank models using secondary training data. *arXiv:1806.11330* (2018).
[11] Mark D. Smucker, James Allan, and Ben Carterette. 2007. A Comparison of Statistical Significance Tests for Information Retrieval Evaluation. In *CIKM*. 623âĂŞ632.
[12] Anastasios Tombros, Ian Ruthven, and Joemon M Jose. 2005. How users assess web pages for information seeking. *JASIST* 56, 4 (2005), 327–344.
[13] Yongfeng Zhang and Xu Chen. 2018. Explainable recommendation: A survey and new perspectives. *arXiv preprint: 1804.11192* (2018).
[14] Xuesi Zhou, Xin Chen, Jian Song, Gang Zhao, and Ji Wu. 2018. Team Cat-Garfield at TREC 2018 Precision Medicine Track.