Learning to Rank with Multi-Criteria LLM-Judge Annotations

Naghmeh Farzi
University of New Hampshire
Durham, NH, USA
Naghmeh.Farzi@unh.edu

Laura Dietz
University of New Hampshire
Durham, NH, USA
dietz@cs.unh.edu

Abstract

Large Language Models (LLMs) are increasingly used as automated judges (LLM judges) to evaluate Information Retrieval (IR) systems, offering a cost-effective complement to human assessments. However, most prior work treats evaluation mainly as a tool for comparison rather than as a signal for improving the retrieval system. We study whether criterion grades from Multi-Criteria LLM-Judge relevance labeling, which decomposes relevance into Exactness, Coverage, Topicality, and Contextual Fit, can serve as effective ranking features for learning-to-rank (L2R). We evaluate this approach using manual relevance labels from TREC DL 2019, DL 2020, and DL 2023. We then analyze how Multi-Criteria LLM-Judge feature importance varies relative to retrieval scores across IR system performance levels.

The results show that although weaker IR systems benefit the most from this treatment, we see improvements ranging from 14% to 36% in NDCG@20 across diverse IR systems. Contextual Fit often becomes the primary discriminator, exceeding the importance of the retrieval scores. As system quality increases, retrieval scores become the dominant feature, with Multi-Criteria LLM-Judge grades providing complementary, dataset-dependent signals. These findings show that LLM-based relevance labeling can go beyond comparison by turning Multi-Criteria LLM-Judge grades into effective ranking features for reranking.

CCS Concepts

• **Information systems** → **Information retrieval**; **Retrieval models and ranking**; **Relevance assessment**.

Keywords

Information Retrieval, Learning-to-Rank, LLM-as-a-Judge

ACM Reference Format:
Naghmeh Farzi and Laura Dietz. 2026. Learning to Rank with Multi-Criteria LLM-Judge Annotations. In *Proceedings of the 49th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '26), July 20–24, 2026, Melbourne, VIC, Australia.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3805712.3809580

SIGIR '26, Melbourne, VIC, Australia

ACM ISBN 979-8-4007-2599-9/2026/07
https://doi.org/10.1145/3805712.3809580

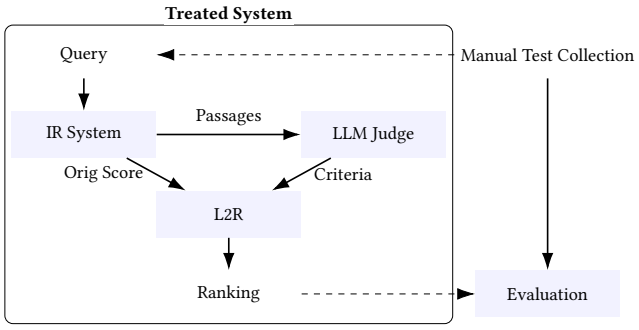

Figure 1: Multi-Criteria-L2R can improve any IR system with Learning-To-Rank features derived from an LLM judge. The evaluation uses only manual relevance judgments, unknown to the LLM judge. The compound system is trained and evaluated via cross-validation.

1 Introduction

A recent trend in information retrieval experimentation is the use of Large Language Models (LLMs) for automatic evaluation, commonly referred to as LLM judges. By providing scalable and cost-effective relevance assessments, LLM judges become feasible options with which to evaluate modern IR systems at a scale that would be impractical with human assessors alone, particularly for retrieval-augmented generation (RAG) systems. As a result, a growing body of work studies automated relevance assessment. Some approaches directly prompt LLMs to assign relevance labels [18, 27, 39, 41, 43], others use pairwise comparisons to rank documents relative to each other [2, 34], while another line of work decomposes relevance labeling into topical or criteria-based rubrics that an LLM judge grades individually before aggregating them into a final label [12, 19, 20, 33].

Although these approaches enable large-scale relevance labeling, they are typically positioned as complements to human evaluation and remain subject to skepticism regarding reliability and transparency [6, 16, 18]. Rather than asking whether LLM judges can replace human assessors, a more consequential question is whether their judgments can be used to improve retrieval systems. Clarke and Dietz [6] use LLM-based relevance judgments to perform final-stage reranking of top-k results. Using human relevance judgments from TREC RAG 24 [42] for validation, they report that reranking retrieval output using the UMBRELA [43] prompt improves the

performance of nearly all systems. This result suggests that LLM-based relevance labeling can provide useful signals for ranking and motivates further study of its utility for reranking.

Task Statement: We consider the task of ad hoc retrieval, where given a query, the goal is to rank passages or documents by their relevance. Given a query and passage, the relevance is specified as a relevance label (0–3), where higher values refer to stronger relevance.

With this perspective, we study how structured Multi-Criteria LLM-Judge grades can be integrated into IR reranking to improve the performance of the original system. We build on the Multi-Criteria LLM-Judge framework [20], which generates more nuanced judgments with grading Exactness, Coverage, Topicality, and Contextual Fit for relevance labeling and evaluation. We repurpose these criterion-level grades as ranking features, along with the system's original retrieval score to rerank retrieved passages/documents.

We use manual relevance judgments to demonstrate that our approach yields a genuine improvement. This avoids circularity, since once LLM judge signals are integrated into the IR system, the same approach used in LLM judge cannot be used for evaluation.

Contributions. Our key contribution is demonstrating that Multi-Criteria LLM-Judge grades, originally designed for evaluation, can be repurposed as effective ranking features to improve retrieval performance. We show that these grades, built for assessment rather than ranking, can be incorporated into a learning-to-rank framework as ranking features yield measurable system improvements, establishing a new use case for LLM-based relevance judgments beyond evaluation.

Concretely, we propose a system-agnostic reranking approach applicable to any first-stage retrieval system. Given the ranked list of candidate passages/documents, we derive Multi-Criteria LLM-Judge grades for each candidate. These grades, used as ranking features alongside the original retrieval scores, form the input features to a listwise L2R model. The model is trained to optimize ranking quality against human relevance judgments.

We demonstrate that our approach significantly improves a wide range of IR systems on standard datasets TREC DL 2019, DL 2020, and DL 2023 using manual relevance judgments. We analyze when and how Multi-Criteria LLM-Judge grades contribute to L2R reranking. Results show that our approach improves reranking performance, yielding relative gains of 14 to 36% in NDCG@20 across a wide range of IR systems and providing complementary ranking signals beyond the retrieval scores.

2 Related Work

LLM-based relevance assessment in IR. The use of LLMs to automate relevance assessments, often termed LLM judges, has gained significant traction as a scalable complement to costly human annotations.

Direct prompting. Early point-wise approaches prompt an LLM to directly assign a single relevance label to a query–document pair [27, 41, 43]. Thomas et al. [41] first demonstrated that LLMs can produce relevance labels that match real searcher preferences as accurately as human labelers, using a zero-shot prompting approach that asks the model to consider aspects such as topicality and trustworthiness before assigning an overall relevance label. Inspired by that, UMBRELA [43] reproduced and open-sourced this approach, assigning 0–3 graded relevance labels across TREC DL 2019–2023 with high correlation to human judgments and later used it in TREC 2024 RAG Track to aid in relevance assessment. These methods demonstrate promising correlations with human judgments, suggesting that LLMs can be a complement to human annotations. However, single-prompt point-wise LLM judges are inherently opaque and collapsing multiple quality dimensions into a single label results in information loss and inconsistencies [45]. Faggioli et al. [18] find reasonable correlation with human judgments in pilot experiments but caution against fully automated evaluation due to concerns around bias, consistency, and circularity. Furthermore, these approaches often require the use of very large models for adequate performance. Farzi et al. [21] demonstrate that UMBRELA [43] requires GPT-4o-scale models to achieve reliable relevance labels, illustrating the limitations of single-prompt point-wise relevance labeling with smaller LLMs.

Preference Judges. Other approaches use pairwise LLM relevance labeling, where the LLM compares two documents and determines which is more relevant to a query. Arabzadeh et al. [2] show that preference-based comparison yields higher agreement with human labels than point-wise scoring.

Nuggets and Rubrics. To improve the transparency and granularity of LLM-based evaluation, recent work decomposes relevance into interpretable components rather than producing a single opaque label. Multi-Criteria judges perform relevance judgments along criteria such as Exactness, Coverage, Topicality, and Contextual Fit [20] (See Section 3.2 for more details). Other work decomposes the user information need into rubrics expressed as subqueries [19], human-authored nuggets [28], or automatically induced nuggets using LLMs [14, 33, 44], before aggregating them into the final relevance labels using different approaches such as taking a maximum or using another prompt.

LLM judges relevance labels as pseudo-ground truth. A separate line of work uses LLM judges to generate relevance labels as substitutes for human supervision. Rahmani et al. [35] create fully synthetic test collections (both queries and judgments) for retrieval evaluation. MacAvaney and Soldaini [27] propose one-shot labeling to fill judgment holes in evaluation pools, finding that LLM predictions can improve system ranking correlations yet cannot reliably identify all relevant documents. Abbasiantaeb et al. [1] find that LLM-generated relevance labels distort rankings when used as ground truth, highlighting risks in treating LLM outputs as substitutes for human judgments. A related direction uses LLM annotations as training supervision rather than evaluation labels. Yu et al. [46] investigate whether LLM-based annotations can substitute user clicks for training L2R models.

These approaches use LLM outputs as pseudo-ground truth, risking distortion and circularity. Our work differs fundamentally in that we use Multi-Criteria LLM-Judge grades, within an L2R model, as ranking features alongside the retrieval score, while human relevance judgments serve exclusively as training supervision and evaluation gold standard.

Learning-to-rank. Learning-to-rank is a supervised machine learning technique that trains a model to optimally order a list of passages/documents for a given query. L2R models are typically categorized as point-wise, pairwise, or listwise, depending on how they formulate the loss function [25]. Several supervised ranking models have been suggested, such as SVMRank [5, 23], ListNet [4], or LambdaMART [3]. For small feature sets, a simple linear model that is directly optimized for a rank evaluation metric such as (mean-)average precision with Coordinate-Ascent, often shows very reliable results [29, 40].

Ranking paradigms. Point-wise approaches score the relevance of each query-document pair independently and rank documents by their scores. Some prompt an LLM to directly assign a relevance label to each pair [39], while others compute the likelihood that a language model would generate the query given each passage, with higher likelihood indicating greater relevance [17, 38].

Pairwise methods prompt the LLM to compare two documents and determine which is more relevant to a query [34]. A final ranking is then derived by aggregating these pairwise preferences across document pairs. Traditionally, listwise approaches refer to directly optimizing for a ranking evaluation metric such as MAP, rather than using a pointwise or pairwise surrogate loss. More recently, listwise LLM rerankers provide an entire list of passages in a single prompt, instructing the model to return a ranked ordering of documents [26, 31, 32, 39]. To handle context length limitations, these approaches employ sliding window strategies that progressively rerank subsets of documents [26, 39]. Setwise methods [47] provide an alternative approach to managing long candidate lists, offering a middle ground between pairwise and listwise approaches.

While many recent approaches replace traditional learning-to-rank with LLM-based rerankers [37, 47], supervised L2R baselines can still outperform zero-shot LLM rerankers. Movin and Hauff [30] find that find that supervised L2R (LambdaMART) outperforms zero-shot LLM reranking on average when both operate on precomputed ranking features, though LLM rerankers excel on challenging queries where traditional L2R struggles. This suggests potential complementary strengths between the two approaches. However, limited work has explored incorporating LLM-generated relevance signals into classical L2R frameworks.

Motivation and positioning. LLM-as-a-judge protocols have primarily been used for offline evaluation. However, Clarke and Dietz [6] demonstrated that when systems are sorted by UMBRELA's LLM-based relevance labels and evaluated against human judgments from TREC RAG 24 [42], nearly all systems show improved performance. This suggests that LLM-based relevance signals contain useful ranking information beyond their role as evaluation metrics. However, Clarke and Dietz raise critical concerns about circularity: once LLM-based signals are integrated into IR systems, the same LLM cannot be used for evaluation, as this creates an invalid feedback loop. Farzi and Dietz [21] show that Multi-Criteria LLM-Judge framework consistently match or outperform UMBRELA [43] across datasets and metrics, demonstrating the potential of structured multi-criteria judgments.

Motivated by these findings, we propose integrating Multi-Criteria LLM-Judge grades as ranking features within a classical learning-to-rank framework. This approach differs from direct LLM reranking by combining structured LLM annotations with retrieval scores in a supervised L2R model. Critically, we evaluate all improvements using human relevance judgments to avoid circularity concerns, allowing LLM-based features to enhance retrieval while maintaining human judgments as the evaluation gold standard.

3 Approach: Learning-to-Rank with Multi-Criteria LLM Annotations

Our approach is a general-purpose reranking method applicable to any IR system, using structured LLM-Judge grades as ranking features in a traditional L2R reranking pipeline. The first-stage IR system assigns each retrieved passage/document a retrieval score (*OrigScore*). The LLM judge assigns a criterion grade (0–3) per relevance criteria (Exactness, Coverage, Topicality, Contextual Fit) for each query–passage pair. We refer to this set of four grades as Multi-Criteria LLM-Judge grades, and, when used in reranking, as Multi-Criteria LLM-Judge features. These features, together with the original retrieval score, form the input ranking features to the L2R model, which is trained against human relevance labels from TREC. The resulting compound system is referred to as Multi-Criteria-L2R.

The pipeline consists of three stages:

3.1 Stage 1: Candidates:

For a given query and IR system, we begin with the candidate documents retrieved by that system. Each retrieved document is associated with the system's original retrieval score (*OrigScore*), which we retain as a ranking feature. Our approach is applied independently to each IR system, without pooling candidate documents across different systems.

3.2 Stage 2: Features

In the second stage, we decompose relevance into four complementary dimensions following the Multi-Criteria framework [20]: *Exactness*, *Coverage*, *Topicality*, and *Contextual Fit*, with the following descriptions:

> **Exactness**: How precisely does the passage answer the query?
> **Topicality**: Is the passage about the same subject as the whole query (not only a single word of it)?
> **Coverage**: How much of the passage is dedicated to discussing the query and its related topics?
> **Contextual Fit**: Does the passage provide relevant background or context?

For each query–passage pair, we prompt an LLM with criterion-specific instructions (Table 1) to assign a criterion grade on a 0–3 scale, aligned with the datasets' grading scale. The prompts include the criterion name, its description, the query, and the passage. Explicitly decomposing relevance improves performance for smaller models [20] and produces interpretable relevance grades. Each candidate document, already associated with a retrieval score, is now additionally represented by four more ranking features.

For a given passage, each criterion grade is generated once in a zero-shot, point-wise manner using the prompt in Table 1. During this process, no access to human relevance judgments is permitted.

Table 1: System messages and prompts for criterion-specific grading. Variables denoted in blue, such as Criterion Name or Query are replaced with respective content. The question mark from the Criterion Description is removed to fit the grammatical form. Prompts are designed for chat completion models. For models that do not support chat completion both messages can be concatenated.

Criterion-Specific Grading
System Message: Please assess how well the provided passage meets specific criteria in relation to the query. Use the following scoring scale (0-3) for evaluation: 0: Not relevant at all / No information provided. 1: Marginally relevant / Partially addresses the criterion. 2: Fairly relevant / Adequately addresses the criterion. 3: Highly relevant / Fully satisfies the criterion.
Prompt: Please rate how well the given passage meets the {Criterion Name} criterion in relation to the query. The output should be a single score (0-3) indicating {Criterion Description}. Query: {Query} Passage: {Passage} Score:

Prior work shows that aggregating these Multi-Criteria LLM-Judge grades into relevance labels achieves very high agreement with human judgments at the system level (Spearman $\rho \approx 0.99$, Kendall's $\tau \approx 0.92$–0.95) across different LLMs on TREC DL 2019, DL 2020, and DL 2023 [20]. These criterion grades serve as ranking features in the downstream learning-to-rank model.

3.3 Stage 3: Learning-To-Rank

L2R training. In this stage, candidate passages are reranked using a listwise L2R model that integrates the Multi-Criteria LLM-Judge grades produced by an LLM in Stage 2. The feature set consists of the four criterion-level grades (Exactness, Coverage, Topicality, Contextual Fit) together with the original retrieval score (*OrigScore*). We train a listwise L2R model to learn a weighted combination of these five features. We use Multi-Criteria LLM-Judge grades as input features while training on human relevance judgments. This allows the model to learn how these features relate to human relevance. Training is performed only on queries in the training split, and the learned model is applied to rerank held-out queries of the same IR system run.

L2R prediction. For each test query, the trained model uses the feature vector of each candidate document to produce a reranked list of documents, applying the learned weights to reorder documents.

4 Experimental Evaluation

4.1 Experimental Setup

Datasets. We evaluate the effectiveness of our reranking approach on multiple benchmark IR system runs from TREC DL 2019, DL 2020, and DL 2023, spanning diverse query types and systems with human-annotated relevance labels. These IR systems cover a range of retrieval paradigms, including lexical baselines, neural rankers, and prompt-based methods, enabling principled comparison across systems.

- **DL 2019:** Passage retrieval systems from the TREC Deep Learning Track 2019 [10].
- **DL 2020:** Passage retrieval systems from the TREC Deep Learning Track 2020 [7].
- **DL 2023:** Systems submitted to TREC DL 2023 with duplicates removed from corpus [36].[1]

Statistics about these datasets, including the number of submitted systems, queries, and the distribution of label statistics, are provided in Table 3. We omit TREC DL 2021 and TREC DL 2022 due to concerns about incomplete relevance judgments arising from the substantial increase in collection size without a corresponding increase in assessment effort, which may affect evaluation reliability [8]. We primarily report NDCG@20 to evaluate graded relevance at top ranks, following the use of NDCG as the primary metric in TREC Deep Learning evaluations (typically reported at @10) [7, 9–11] and prior reranking studies [6]. For completeness, we also report MAP@20 results in the Appendix (Table 5), to confirm consistent improvements across both metrics.

LLM models. We generate Multi-Criteria LLM-Judge grades using three openly available instruction-tuned LLMs spanning different model families and sizes, allowing us to study the impact of model scale on LLM-based annotation quality. Multi-Criteria LLM-Judge grades are generated once per LLM for all query–passage pairs and treated as fixed ranking features. These annotations are reused across reranking experiments for different IR system runs, while the L2R model is trained separately for each run using only features from its training queries.

- **LLaMA-3-8B:**[2] An 8B-parameter instruction-tuned model from the LLaMA 3 family, representing a compact but strong general-purpose LLM suitable for efficient annotation.
- **LLaMA-3.3-70B:**[3] A 70B-parameter instruction-tuned model from the same LLaMA 3 family, providing a high-capacity variant to assess the effect of larger models on Multi-Criteria grading quality.
- **FLAN-T5-large:**[4] A 780M-parameter encoder–decoder model from the FLAN-T5 family, used as a smaller, non-LLaMA baseline to contrast lightweight sequence-to-sequence models with modern decoder-only LLMs.

Learning-to-rank training. We train the learning-to-rank model using RankLips [13],[5] a listwise L2R framework that directly optimizes for the training evaluation metric with coordinate ascent and smart caching. Training optimizes for (mean-) average precision which, due to its smooth objective surface, has been found to obtain the best downstream results, even when the target evaluation metric is NDCG.

We use five-fold cross-validation (80% train, 20% held-out per fold) over queries. This prevents test-data contamination by ensuring no query is seen during both training and evaluation. This approach ensures the learned weights are never optimized with

[1] Available at https://llm4eval.github.io/LLMJudge-benchmark/
[2] https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct
[3] https://api.together.xyz/models/meta-llama/Llama-3.3-70B-Instruct-Turbo
[4] https://huggingface.co/google/flan-t5-large
[5] https://www.cs.unh.edu/~dietz/rank-lips/

Table 2: Mean absolute gain ($\overline{\Delta}_{\text{NDCG@20}}$) and geometric mean gain ratios ($\overline{G}_{\text{NDCG@20}}$) of NDCG@20 for systems in which the L2R model combines all Multi-Criteria LLM-Judge grades as ranking features (exactness, coverage, topicality, contextual fit) with the original system retrieval score. Results are across three datasets and grouped by base performance percentiles. For $\overline{\Delta}_{\text{NDCG@20}}$, higher values indicate greater improvement. For $\overline{G}_{\text{NDCG@20}}$, values above 1.0 indicate improvement, 1.0 indicates no change, and values below 1.0 indicate a decrease in performance.

System / Model	$\overline{\Delta}_{\text{NDCG@20}}$ / $\overline{G}_{\text{NDCG@20}}$						
	0–5%	5–25%	25–50%	50–75%	75–95%	95–100%	0–100%
TREC DL 2019							
FLAN-T5-large	0.234/ 2.739	0.185/ 1.413	0.104/ 1.200	0.050/ 1.082	0.010/ 1.015	0.016/ 1.022	0.090/ 1.209
LLaMA-3-8B	0.209/ 2.566	0.172/ 1.384	0.095/ 1.183	0.056/ 1.091	0.020/ 1.028	0.033/ 1.047	0.088/ 1.203
LLaMA-3.3-70B	0.275/ 2.949	0.253/ 1.567	0.146/ 1.276	0.089/ 1.144	0.042/ 1.061	0.058/ 1.081	0.134/ 1.290
TREC DL 2020							
FLAN-T5-large	0.009/ 1.293	0.159/ 1.464	0.073/ 1.149	0.006/ 1.009	0.001/ 1.001	0.001/ 1.002	0.055/ 1.137
LLaMA-3-8B	0.006/ 1.293	0.196/ 1.464	0.094/ 1.149	0.094/ 1.014	*-0.004/ 0.994*	0.004/ 1.005	0.068/ 1.157
LLaMA-3.3-70B	0.008/ 1.253	0.308/ 1.855	0.196/ 1.385	0.080/ 1.123	0.042/ 1.060	0.026/ 1.035	0.145/ 1.303
TREC DL 2023							
FLAN-T5-large	0.087/ 1.604	0.104/ 1.502	0.098/ 1.273	0.079/ 1.179	0.032/ 1.062	0.032/ 1.054	0.078/ 1.247
LLaMA-3-8B	0.094/ 1.656	0.100/ 1.467	0.132/ 1.368	0.094/ 1.213	0.045/ 1.087	0.022/ 1.038	0.091/ 1.277
LLaMA-3.3-70B	0.111/ 1.761	0.147/ 1.696	0.144/ 1.401	0.127/ 1.286	0.060/ 1.116	0.033/ 1.056	0.118/ 1.359

Table 3: Dataset Statistics

			Manual Relevance Label Statistics			
	Systems	Queries	0	1	2	3
DL 2019	36	43	5158	1601	1804	697
DL 2020	59	54	7780	1940	1020	646
DL 2023	35	25	2005	1233	808	377

knowledge of held-out queries, preventing test-data contamination. At test time, fold-specific models rerank only their respective held-out queries; results are concatenated to produce a single cross-validated reranked list for unbiased evaluation.

Features are Z-score normalized using mean and standard deviation computed solely from the training queries of each fold (with the same statistics applied to held-out queries during prediction), accommodating varying score scales across retrieval systems and LLM models. We employ mini-batches of size 100 along with 10 random restarts for robust convergence.

The graded TREC relevance judgments (0–3) are binarized prior to training, as RankLips optimizes MAP and therefore requires binary labels. For each system run, the training set consists of all retrieved query–document pairs from the training queries. The original retrieval score (`OrigScore`) is included as a feature alongside the four Multi-Criteria LLM-Judge features. Documents without manual relevance labels are assigned a default value of −99 for all Multi-Criteria LLM-Judge features. During training, they are assigned a *NotRelevant* label. The resulting ranked lists are evaluated with `trec_eval`, which restricts metric computation to documents with manual relevance judgments.

Multi-Criteria-L2R treatment evaluation. We measure the ranking quality improvements introduced by our treatment, applying Multi-Criteria-L2R reranking to each existing IR system run, considering both absolute and relative improvements in the evaluation metric NDCG@20. For each system run $s \in S$, we compare the original IR system's ranking quality, denoted $\text{NDCG@20}_s^{\text{orig}}$, with the ranking quality after applying Multi-Criteria-L2R, denoted $\text{NDCG@20}_s^{\text{Multi-Criteria-L2R}}$.

The absolute gain is defined as

$$\Delta_s = \text{NDCG@20}_s^{\text{Multi-Criteria-L2R}} - \text{NDCG@20}_s^{\text{orig}},$$

and the gain ratio as

$$G_s = \frac{\text{NDCG@20}_s^{\text{Multi-Criteria-L2R}}}{\text{NDCG@20}_s^{\text{orig}}}.$$

To aggregate relative improvements across heterogeneous IR systems, we report the geometric mean gain ratio:

$$\overline{G}_{\text{NDCG@20}} = \exp\left(\frac{1}{|S|}\sum_{s\in S}\log G_s\right),$$

together with the mean absolute gain

$$\overline{\Delta}_{\text{NDCG@20}} = \frac{1}{|S|}\sum_{s\in S}\Delta_s.$$

Compared methods. We compare each original IR system run against its Multi-Criteria-L2R reranked counterpart produced by our proposed treatment. The goal is to quantify how reranking with Multi-Criteria LLM-Judge features improves the original system.

We expect some systems to have already included an LLM-Judge reranking approach as part of their system (e.g., run `Naverloo` from `h2oloo` [24]). Such systems will likely not benefit from an additional LLM-Judge re-ranking as much as other systems.

To gain insights into which systems benefit most, e.g., low-ranked vs high-ranked, we categorize IR systems into percentile ranges based on their original performance: 0–5%, 5–25%, 25–50% (lower inter-quartile range), 50–75% (upper inter-quartile range), 75–95%, 95–100%, and 0–100% (full range).

4.2 Reranking Performance

We evaluate the effectiveness of Multi-Criteria-L2R across three dimensions. First, we assess overall reranking improvements (Section 4.2.1). Second, we analyze how gains vary across different IR system performance levels and datasets (Section 4.2.2). Finally, we examine the impact of LLM model choice and the relative importance of different features (Section 4.2.3).

4.2.1 Reranking Improvement with Multi-Criteria LLM-Judge Annotations. Table 2 reports NDCG@20 improvements on three TREC datasets, including both the absolute gain ($\Delta = A - B$) and the gain ratio ($G = \frac{A}{B}$) under our treatment for each IR system. Relative improvements are aggregated using the geometric mean gain ratio ($\overline{G}_{\text{NDCG@20}}$).

The results show that treating input IR systems with our Multi-Criteria-L2R consistently improves ranking performance. As shown in the last column of Table 2 and in Figure 2, nearly all models yield positive gains across datasets, with $\overline{G}_{\text{NDCG@20}}$ values ranging from 1.14 to 1.36, corresponding to 14% to 36% relative improvement in NDCG@20.

When the original system retrieves a high-recall set of documents, Multi-Criteria-L2R can substantially refine their ordering, including for already strong IR systems. A small number of mid-tier systems in DL 2019 and weak systems in DL 2020 exhibit little to no change after reranking. In these cases, the original systems did not retrieve many of the relevant documents, and the re-ranker could not reconcile this shortcoming. Even already strong systems (75% to 100% range) improve with our approach. Except for one case, no system performs worse when our treatment is applied, demonstrating that Multi-Criteria LLM-Judge grades are valuable ranking features for improving systems' ranking performance.

4.2.2 Effectiveness of Multi-Criteria Reranking Across IR System Performance Levels and Datasets.

System Performance improvements. Ranking quality improvements vary significantly with the original performance of the IR system across datasets. Table 2 shows systems in the bottom quartile (0% to 25% range) obtain the largest relative improvements, with geometric mean gain ratios ($\overline{G}_{\text{NDCG@20}}$) corresponding to roughly +25% to +195% across collections. The weaker models consistently exhibit greater headroom for improvement at lower performance levels: IR systems in the 25% to 50% range typically see noticeable improvements (about +15% to +40%), while higher-performing IR systems (50% to 95% range) experience modest gains (up to +12%). For the strongest IR systems (95% to 100% range), improvements are naturally small but consistent (up to +8%). These trends arise because Multi-Criteria-L2R can only reorder documents retrieved by the original IR system. If the first-stage retrieval fails to retrieve enough relevant documents, reranking has a limited opportunity to improve the ranking. In contrast, systems with reasonable recall but weak initial ordering benefit most, since relevant documents already present at lower ranks can be moved upward.

Overall, Multi-Criteria LLM-Judge features provide information that is complementary to the original system's retrieval scores, and hence manifests in significant across-the-board system improvements via our treatment.

Dataset differences. To compare datasets under comparable conditions, we focus on mid-tier IR systems, where rankings are neither dominated by weak initial retrieval nor saturated by near-optimal performance, leaving meaningful headroom for additional gains from reranking.

On DL 2019, Multi-Criteria-L2R delivers moderate, steady gains (+20 to +30% for lower mid-tier systems) that taper smoothly but remain positive even for stronger systems. The treatment on DL 2020 shows the most dramatic improvements; lower mid-tier systems benefit substantially. After applying Multi-Criteria-L2R on DL 2023 IR systems, there are sustained gains of +20-40% across all mid-tier systems with noticeable benefits even for the strongest IR systems, proving Multi-Criteria-L2R's effectiveness across varying performance levels on this challenging collection.

4.2.3 LLM Model Impact and Feature Importance in Multi-Criteria-L2R.

Which LLM model produces the most effective grades for reranking? Figure 2 shows that grades derived from LLaMA-3.3-70B consistently yield higher gains across all three datasets. This indicates that larger models tend to produce more informative Multi-Criteria LLM-Judge grades for reranking. Compared to the smaller models FLAN-T5-large and LLaMA-3-8B, LLaMA-3.3-70B achieves the highest geometric mean gain ratios ($\overline{G}_{\text{NDCG@20}}$), suggesting that its annotations better complement the retrieval scores of input IR systems.

How important are Multi-Criteria LLM-Judge grades compared to the retrieval score? According to Figure 4, the original system's retrieval score (OrigScore) dominates in the high performance bracket (75% to 100% range). Despite OrigScore being the most influential feature for already strong systems, incorporating the Multi-Criteria LLM-Judge grades as ranking features still improves NDCG@20 by up to 12% for strong systems (75% to 100% range) across datasets and models.

At lower performance levels (0% to 50% range), Multi-Criteria features become most influential. This suggests that weaker IR systems can be much improved with Multi-Criteria-L2R treatment.

Even high-performing systems (Figure 4) improve noticeably with the Multi-Criteria-L2R treatment, revealing that these features provide valuable ranking information that strong systems do not fully exploit on their own. This pattern is consistent across model architectures and sizes, from FLAN-T5-large through LLaMA-3-8B to LLaMA-3.3-70B. This demonstrates that the effect of these features in ranking is robust across models.

How does the choice of LLM model affect which criterion grades matter most? Figure 3 illustrates distinct feature-importance patterns across the three LLM models. With FLAN-T5-large, the original system retrieval score (OrigScore) dominates across nearly all performance percentiles, especially for top-performing systems

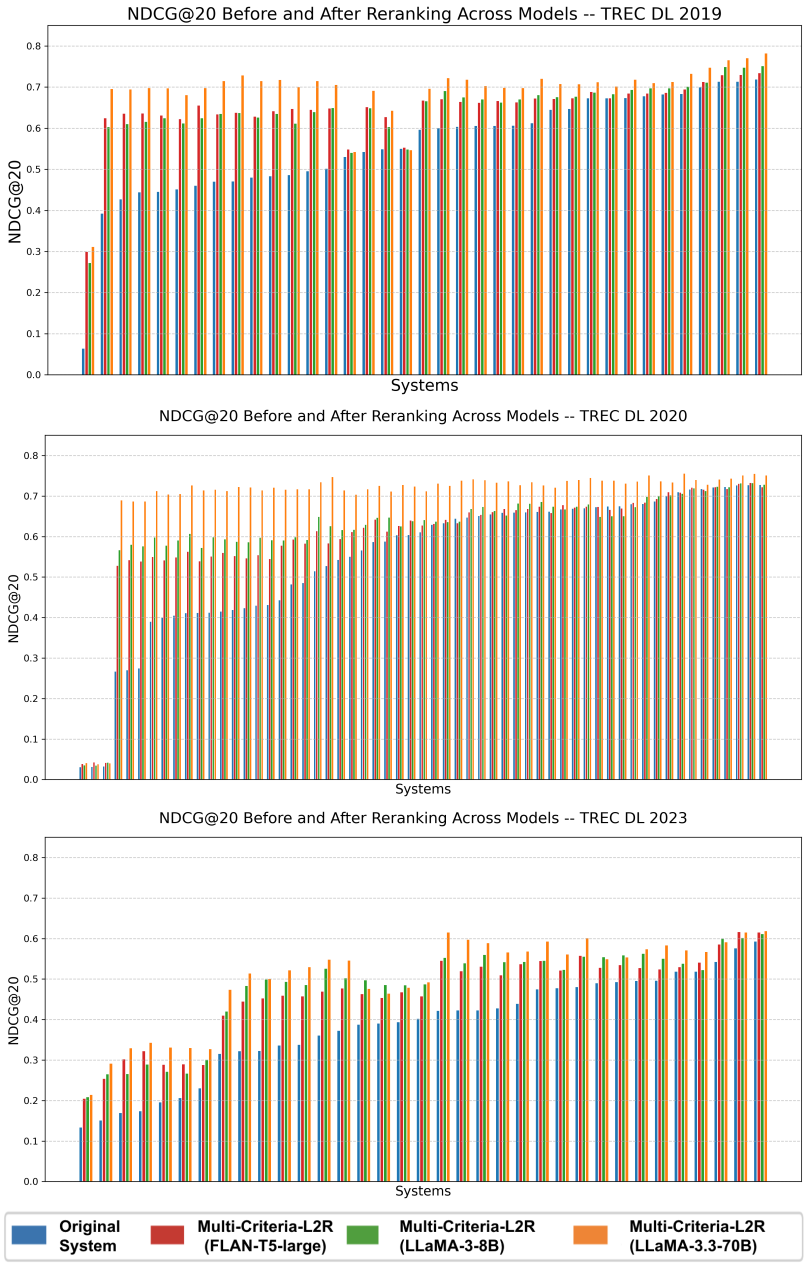

Figure 2: NDCG@20 before and after Multi-Criteria-L2R reranking across IR systems on TREC DL 2019, DL 2020, and DL 2023. Blue bars show the original systems' ranking performances, while red, green, and orange bars show the NDCG@20 of reranked results using Multi-Criteria LLM-Judge grades as ranking features derived from FLAN-T5-large, LLaMA-3-8B, and LLaMA-3.3-70B. Systems are ordered by original performance. Reranking improves ranking quality for the vast majority of IR systems across performance levels.

(95% to 100% range). The Multi-Criteria LLM-Judge features (Exactness, Coverage, Topicality, Contextual Fit) play supplementary but limited roles, demonstrating moderate importance primarily in mid-tier systems.

LLaMA-3-8B yields more balanced feature utilization. In particular, the Contextual Fit criterion becomes the most influential feature for low- to mid-performing systems. Although the original system's retrieval score (OrigScore) remains dominant for strong systems, the Multi-Criteria LLM-Judge features obtained with LLaMA-3-8B are more helpful than those obtained with FLAN-T5-large.

LLaMA-3.3-70B exhibits the most varied feature importance distribution. Exactness plays a strong role for weak and mid-tier

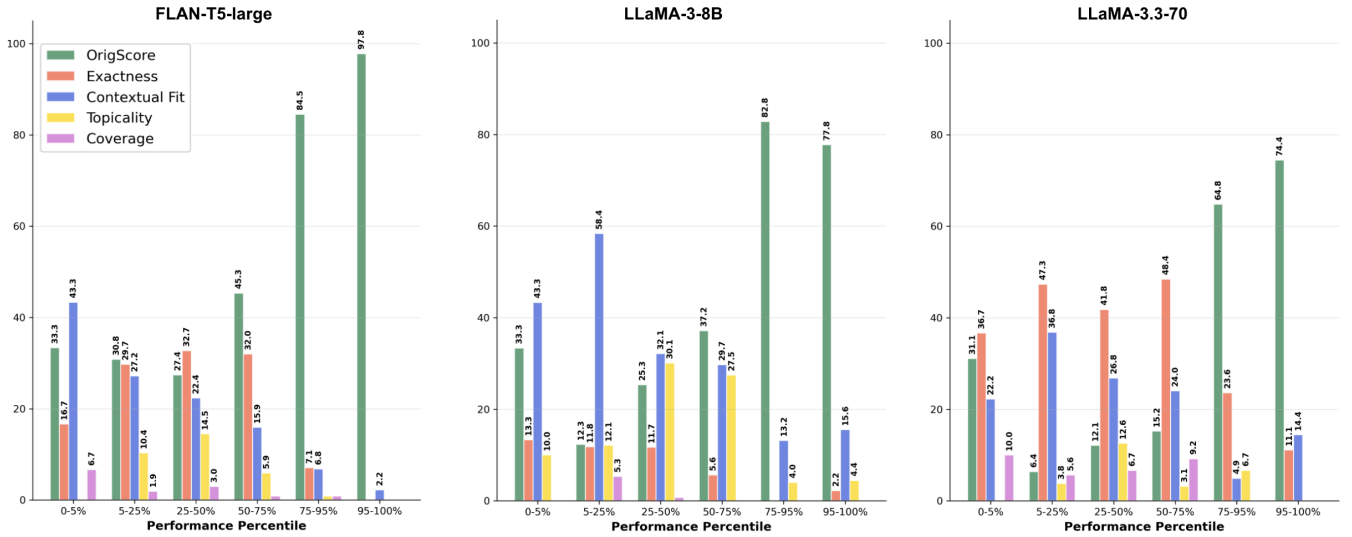

Figure 3: Frequency of features being the most important across performance percentiles, averaged for each model. Each subplot represents a single model, and bars show the percentage of fold-level models (from 5-fold cross-validation) in which a feature was the top contributor within systems of that model. Systems are grouped by performance percentile based on NDCG@20, and values are averaged across all datasets for that model.

systems, while the original system's retrieval score (OrigScore) is less dominant than with FLAN-T5-large. Criterion grades contribute varyingly as ranking features across performance ranges, indicating richer and less redundant signals.

Overall, these patterns suggest that larger and more capable LLM judges generate Multi-Criteria annotations that provide essential information complementary to the system's retrieval score. These model-specific patterns align with the aggregate performance metrics in Table 2, where LLaMA-3.3-70B consistently achieves the highest geometric mean gain ratios across all datasets and performance ranges.

4.3 Ablation Study: Contribution of Individual Multi-Criteria LLM-Judge Grades

The ablation results in Table 4 confirm that using all four Multi-Criteria LLM-Judge features along with the retrieval score is the most robust configuration, remaining strictly positive across all nine dataset–model combinations and achieving the highest $\overline{\Delta}_{\text{NDCG@20}}$ almost all cases. Leave-one-out configurations are generally robust. Removing any single criterion grade from features causes only marginal degradation, typically within 0.01–0.02 in $\overline{\Delta}_{\text{NDCG@20}}$, demonstrating that the four Multi-Criteria LLM-Judge features provide complementary signals. Real performance drops occur only when the feature set is reduced to one or two criteria, particularly with smaller models on DL 2020 (T_only with LLaMA-3-8B: $\overline{\Delta}$=-0.134), confirming that considering all four Multi-Criteria LLM-Judge features is especially critical when the LLM judge model is smaller or less capable. Model size mediates feature robustness. With LLaMA-3.3-70B, even single-criterion configurations mostly yield positive gains, whereas FLAN-T5-large and LLaMA-3-8B require the full combination to avoid degradation, confirming that smaller judges produce noisier individual criterion-level grades. This further motivates the all-four setting as the safest practical choice across model sizes.

Performance sensitivity is strongest on DL 2020, where many reduced-feature settings become negative, whereas DL 2019 and DL 2023 remain mostly positive. Model size also mediates robustness, with LLaMA-3.3-70B, even single-criterion settings are usually beneficial, while FLAN-T5-large and LLaMA-3-8B depend more heavily on combining all four Multi-Criteria LLM-Judge features.

Among individual features, single-criterion settings are generally unreliable, with Topicality in isolation being particularly inconsistent across datasets and models. Minor reversals between $\overline{\Delta}_{\text{NDCG@20}}$ and $\overline{G}_{\text{NDCG@20}}$ occur because they capture absolute versus proportional gains, respectively. A configuration with higher mean score gains can still have a slightly lower geometric gain ratio when those gains occur mainly on already strong baseline systems, where equal absolute improvements correspond to smaller relative changes. An example is `T_CF` with LLaMA-3-8B on DL 2023 (0.088/1.278 vs. all-4's 0.091/1.277). The all-4 leads in absolute gain while `T_CF` leads marginally in geometric gain ratio, reflecting that $\overline{G}_{\text{NDCG@20}}$ amplifies proportional gains on weaker baseline systems while $\overline{\Delta}_{\text{NDCG@20}}$ captures absolute improvement across all systems.

4.4 Efficiency and Runtime

In addition to effectiveness, we report on runtime to assess practical feasibility, separating the cost of learning-to-rank (L2R) training and the inference required for criterion-grade generation. Across DL 2023 systems, L2R training required on average 12.1 seconds

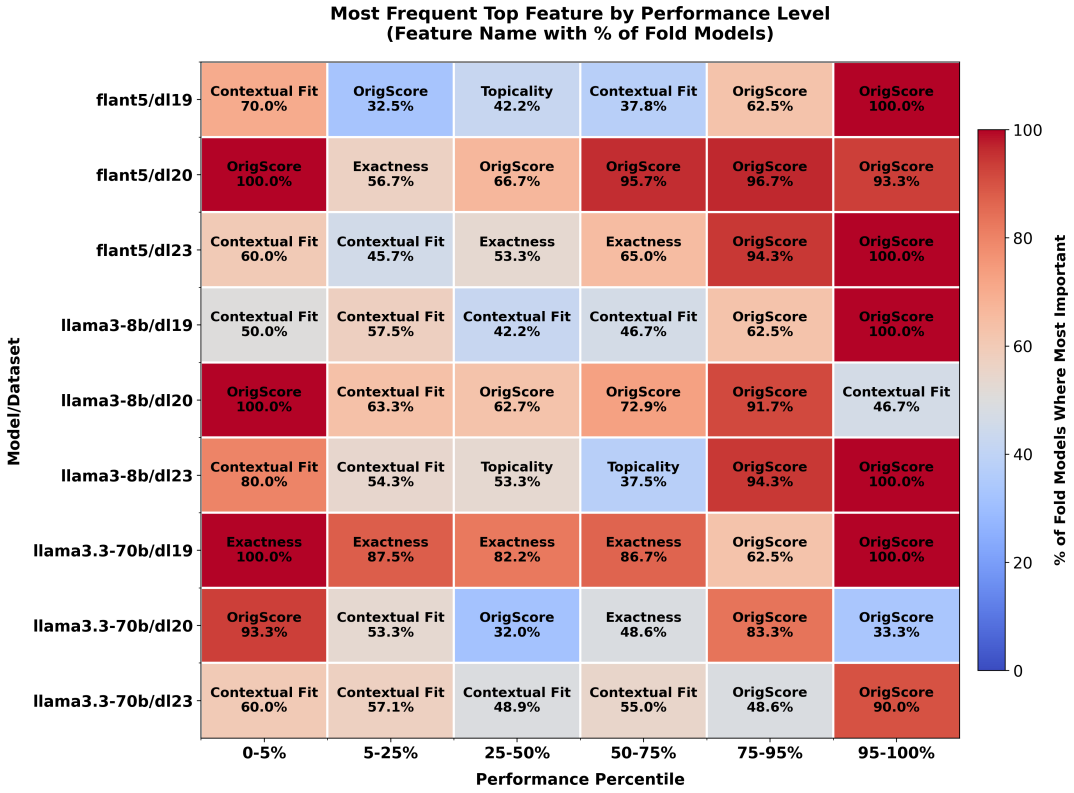

Figure 4: This heatmap shows the most frequently dominant feature for each model–dataset combination across performance percentiles. Each cell lists the feature name and the percentage of cross-validation folds (out of five) in which it was the most important, providing a summary of feature dominance trends. Colors indicate the percentage (cooler = lower, warmer = higher). Percentiles are on the x-axis, and model–dataset pairs are on the y-axis. Dominance shifts from criteria (low perf.) to OrigScore (high perf.), consistent across models/datasets.

(min: 5.0 s, max: 18.0 s), and reranking took 11.7 seconds on average (min: 6.0 s, max: 17.0 s). Overall, the L2R process (training + reranking) averaged 23.8 seconds per system (range: 11 to 35 s), making reranking itself computationally lightweight with the high-performance RankLips implementation.

In our current LLM implementation that uses LLM-prompts on a commercial endpoint, Multi-Criteria annotation takes approximately 0.24 seconds per query–document pair on average. Importantly, annotation is performed once and cached for reuse across folds and input systems, amortizing the cost over all experiments. We emphasize that the goal of this work is not to minimize annotation latency, but to demonstrate the extent to which richer LLM-based relevance signals can improve original IR systems when integrated into listwise L2R model. We note the increased prevalence of massive LLM-based applications, with LLM-controlled agentic pipelines that utilize large context windows. Hence, we anticipate that the costs/time for prompting will only improve with the advent of inference accelerators. This would make our approach viable for deployment, at least for top-k re-ranking.

5 Constraints on Evaluation with LLM Judges

An important consideration in our design is that folding an LLM-judge into the system pipeline introduces a degree of circularity, a vulnerability discussed in prior work on LLM-based evaluation [6, 15, 16, 18]. Once judge-derived signals become part of the ranking pipeline, that same judge family should not be reused as an external evaluator, as this would compromise independence and could overstate gains. We view this as a methodological constraint rather than a limitation of the approach itself: evaluation components repurposed as system components must be paired with external, independently constructed test protocols.

For this reason, all results in this study are reported against manual TREC relevance judgments, ensuring that our empirical findings remain independent of the judge signals used inside the reranking model. More broadly, this setting exemplifies a form of judge–system entanglement, in which the boundary between evaluation machinery and system logic becomes porous. This highlights the need for evaluation frameworks that explicitly account for such entanglement [22], and for benchmarks that remain robust even when judge-style components are internalized into retrieval and generation systems.

Table 4: Mean absolute gain ($\overline{\Delta}_{\text{NDCG@20}}$) and geometric mean gain ratios ($\overline{G}_{\text{NDCG@20}}$) for the ablation study. Each configuration uses the original retrieval score (`OrigScore`) along with the indicated subset of Multi-Criteria LLM-Judge features: *All 4* uses all criteria; *no_X* removes feature X; *X_only* uses only X; pair labels (e.g., E + CF) use only those two features. Values are reported across TREC DL 2019, DL 2020, and DL 2023 for three judge models. Higher $\overline{\Delta}_{\text{NDCG@20}}$ is better; $\overline{G}_{\text{NDCG@20}} > 1$ indicates improvement. E = Exactness, C = Coverage, T = Topicality, CF = Contextual Fit.

Features	$\overline{\Delta}_{\text{NDCG@20}}$ / $\overline{G}_{\text{NDCG@20}}$								
	DL 2019			DL 2020			DL 2023		
	FLAN-T5-large	LLaMA-3-8B	LLaMA-3.3-70B	FLAN-T5-large	LLaMA-3-8B	LLaMA-3.3-70B	FLAN-T5-large	LLaMA-3-8B	LLaMA-3.3-70B
All 4	**0.090/1.209**	**0.088/1.203**	**0.134/1.290**	**0.055/1.137**	**0.068/1.157**	**0.145/1.303**	**0.078/1.247**	**0.091/1.277**	0.118/1.359
no_E	0.028/1.097	0.021/1.083	0.093/1.216	−0.068/0.913	−0.017/1.005	0.074/1.185	0.046/1.168	0.083/1.266	0.113/1.348
no_C	0.043/1.125	0.039/1.116	0.106/1.240	−0.057/0.935	−0.013/1.017	0.116/1.251	0.059/1.202	0.086/1.272	0.118/1.360
no_T	0.008/1.060	0.030/1.100	0.114/1.255	−0.079/0.892	−0.012/1.014	0.116/1.252	0.060/1.204	0.082/1.261	**0.125/1.378**
no_CF	0.029/1.099	0.024/1.090	0.114/1.255	−0.056/0.937	−0.025/0.997	0.107/1.243	0.061/1.210	0.082/1.260	0.114/1.352
E_only	−0.020/1.008	−0.010/1.025	0.084/1.199	−0.089/0.877	−0.070/0.912	0.089/1.210	0.057/1.200	0.070/1.231	0.105/1.326
C_only	−0.021/1.006	−0.030/0.990	0.058/1.152	−0.102/0.852	−0.064/0.925	0.045/1.131	0.051/1.185	0.065/1.215	0.102/1.321
T_only	0.006/1.055	−0.060/0.933	0.015/1.073	−0.079/0.896	−0.134/0.792	−0.006/1.036	0.045/1.171	0.052/1.180	0.077/1.251
CF_only	−0.036/0.977	0.007/1.057	0.068/1.169	−0.124/0.810	−0.036/0.977	0.056/1.150	0.045/1.169	0.082/1.259	0.102/1.318
E + C	0.002/1.049	0.012/1.066	0.109/1.246	−0.076/0.901	−0.037/0.976	0.105/1.234	0.065/1.220	0.077/1.249	0.121/1.369
E + T	0.038/1.115	0.011/1.065	0.101/1.231	−0.050/0.949	−0.045/0.959	0.097/1.222	0.059/1.205	0.080/1.255	0.109/1.337
E + CF	−0.005/1.036	0.033/1.104	0.104/1.236	−0.092/0.870	−0.017/1.009	0.114/1.248	0.060/1.205	0.080/1.254	0.119/1.361
C + T	0.017/1.077	−0.008/1.030	0.071/1.175	−0.071/0.910	−0.050/0.950	0.051/1.142	0.047/1.178	0.074/1.241	0.105/1.332
C + CF	0.004/1.053	0.013/1.068	0.089/1.207	−0.090/0.871	−0.020/1.001	0.074/1.184	0.047/1.173	0.080/1.258	0.117/1.361
T + CF	0.033/1.107	0.013/1.067	0.077/1.187	−0.067/0.918	−0.031/0.983	0.063/1.164	0.050/1.183	0.088/1.278	0.100/1.315

Limitations. Our evaluation is conducted on TREC DL 2019, 2022, and 2023 system runs, which reflect the retrieval landscape of that period. These collections include many traditional systems such as BM25 and keyword-based models, as well as early neural rankers, but are missing many more recent LLM-enabled retrieval systems. It is possible that systems which already incorporate LLM-based reranking as part of their pipeline would benefit less from our treatment, since the complementary Multi-Criteria LLM-Judge features may already be partially captured by their existing components. The generalizability of our findings to more recent, LLM-heavy system landscapes remains an open question.

Across three TREC benchmarks, we show that the vast majority of IR systems benefit from this treatment, often with substantial improvements ranging from +14% to +36% relative gains in NDCG@20 as measured against manual relevance judgments.

Our findings show that Multi-Criteria LLM-Judge ideas can directly improve IR systems, not merely to generate training data [35]. This marks a paradigm shift in which evaluation methods are elevated into core system components rather than remaining external scoring tools. By re-purposing evaluation machinery as part of the retrieval and generation pipeline, these methods lend their strengths directly to the system and translate into immediate, tangible benefits for users.

6 Conclusion

We introduce Multi-Criteria-L2R, a treatment for improving the quality of any input IR system. Our approach builds on a successful implementation of the Multi-Criteria LLM-Judge, previously validated in the LLM-Judge Challenge [20]. Rather than confining the LLM-judge to the evaluation side—its original purpose—we incorporate it directly into the IR system itself. Concretely, we propose a re-ranking framework in which criterion grades are used as ranking features alongside the original system's retrieval score, inspired by Clarke and Dietz [6].

References

[1] Zahra Abbasiantaeb, Chuan Meng, Leif Azzopardi, and Mohammad Aliannejadi. 2024. Can We Use Large Language Models to Fill Relevance Judgment Holes? *arXiv preprint arXiv:2405.05600* (2024).

[2] Negar Arabzadeh and Charles L. A. Clarke. 2024. A Comparison of Methods for Evaluating Generative IR. arXiv:2404.04044 [cs.IR] https://arxiv.org/abs/2404.04044

[3] Christopher JC Burges. 2010. From ranknet to lambdarank to lambdamart: An overview. *Learning* 11, 23-581 (2010), 81.

[4] 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.

[5] Olivier Chapelle and S Sathiya Keerthi. 2010. Efficient algorithms for ranking with SVMs. *Information retrieval* 13, 3 (2010), 201–215.

[6] Charles L. A. Clarke and Laura Dietz. 2025. LLM-based relevance assessment still can't replace human relevance assessment. In *EVIA 2025: Proceedings of the Tenth International Workshop on Evaluating Information Access (EVIA 2025), a Satellite Workshop of the NTCIR-18 Conference, June 10-13, 2025, Tokyo, Japan*. 1–5. doi:10.20736/0002002105

[7] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, and Daniel Campos. 2021. Overview of the TREC 2020 deep learning track. *arXiv preprint arXiv:2102.07662* (2021).

[8] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Jimmy Lin. 2025. Overview of the TREC 2021 deep learning track. arXiv:2507.08191 (2025). doi:10.48550/arXiv.2507.08191 arXiv:2507.08191 [cs].

[9] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Jimmy Lin. 2025. Overview of the TREC 2021 deep learning track. arXiv:2507.08191 [cs.IR] https://arxiv.org/abs/2507.08191

[10] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Ellen M Voorhees. 2020. Overview of the TREC 2019 deep learning track. *arXiv preprint arXiv:2003.07820* (2020).

[11] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Hossein A. Rahmani, Daniel Campos, Jimmy Lin, Ellen M. Voorhees, and Ian Soboroff. 2025. Overview of the TREC 2023 deep learning track. arXiv:2507.08890 [cs.IR] https://arxiv.org/abs/2507.08890

[12] Mouly Dewan, Jiqun Liu, and Chirag Shah. 2025. TRUE: A Reproducible Framework for LLM-Driven Relevance Judgment in Information Retrieval. arXiv:2509.25602 [cs.IR] https://arxiv.org/abs/2509.25602

[13] Laura Dietz. 2019. ENT Rank: Retrieving entities for topical information needs through entity-neighbor-text relations. In *Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval*. 215–224.

[14] Laura Dietz, Bryan Li, Gabrielle Liu, Jia-Huei Ju, Eugene Yang, Dawn Lawrie, William Walden, and James Mayfield. 2026. Incorporating Q&A Nuggets into Retrieval-Augmented Generation. arXiv:2601.13222 [cs.IR] https://arxiv.org/abs/2601.13222

[15] Laura Dietz, Bryan Li, Eugene Yang, Dawn Lawrie, William Walden, and James Mayfield. 2026. Insider Knowledge: How Much Can RAG Systems Gain from Evaluation Secrets? arXiv:2601.13227 [cs.IR] https://arxiv.org/abs/2601.13227

[16] Laura Dietz, Oleg Zendel, Peter Bailey, Charles L. A. Clarke, Ellese Cotterill, Jeff Dalton, Faegheh Hasibi, Mark Sanderson, and Nick Craswell. 2025. Principles and Guidelines for the Use of LLM Judges. In *Proceedings of the 2025 International ACM SIGIR Conference on Innovative Concepts and Theories in Information Retrieval (ICTIR)* (Padua, Italy) *(ICTIR '25)*. Association for Computing Machinery, New York, NY, USA, 218–229. doi:10.1145/3731120.3744588

[17] Andrew Drozdov, Honglei Zhuang, Zhuyun Dai, Zhen Qin, Razieh Rahimi, Xuanhui Wang, Dana Alon, Mohit Iyyer, Andrew McCallum, Donald Metzler, and Kai Hui. 2023. PaRaDe: Passage Ranking using Demonstrations with LLMs. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore, 14242–14252. doi:10.18653/v1/2023.findings-emnlp.950

[18] Guglielmo Faggioli, Laura Dietz, Charles L. A. Clarke, Gianluca Demartini, Matthias Hagen, Claudia Hauff, N. Kando, E. Kanoulas, Martin Potthast, Benno Stein, and Henning Wachsmuth. 2023. Perspectives on Large Language Models for Relevance Judgment. *Proceedings of the 2023 ACM SIGIR International Conference on Theory of Information Retrieval* (2023). https://api.semanticscholar.org/CorpusID:258187001

[19] Naghmeh Farzi and Laura Dietz. 2024. Pencils Down! Automatic Rubric-based Evaluation of Retrieve/Generate Systems. In *Proceedings of the 2024 ACM SIGIR International Conference on Theory of Information Retrieval*. ACM, Washington DC USA, 175–184. doi:10.1145/3664190.3672511

[20] Naghmeh Farzi and Laura Dietz. 2025. Criteria-Based LLM Relevance Judgments. In *Proceedings of the 11th ACM SIGIR / The 15th International Conference on Innovative Concepts and Theories in Information Retrieval*.

[21] Naghmeh Farzi and Laura Dietz. 2025. Does UMBRELA work on other LLMs?. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 3214–3222.

[22] Jiaman He, Zikang Leng, Dana McKay, Damiano Spina, and Johanne R Trippas. 2025. Can We Hide Machines in the Crowd? Quantifying Equivalence in LLM-in-the-loop Annotation Tasks. In *Proceedings of the 2025 Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region*. 426–436.

[23] Thorsten Joachims. 2006. Training linear SVMs in linear time. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. 217–226.

[24] Carlos Lassance, Ronak Pradeep, and Jimmy Lin. 2023. Naverloo @ TREC Deep Learning and Neuclir 2023: As Easy as Zero, One, Two, Three - Cascading Dual Encoders, Mono, Duo, and Listo for Ad-Hoc Retrieval. https://openreview.net/forum?id=2S3jl2JkQU

[25] Tie-Yan Liu et al. 2009. Learning to rank for information retrieval. *Foundations and Trends® in Information Retrieval* 3, 3 (2009), 225–331.

[26] Xueguang Ma, Xinyu Zhang, Ronak Pradeep, and Jimmy Lin. 2023. Zero-Shot Listwise Document Reranking with a Large Language Model. arXiv:2305.02156 [cs.IR] https://arxiv.org/abs/2305.02156

[27] Sean MacAvaney and Luca Soldaini. 2023. One-Shot Labeling for Automatic Relevance Estimation. *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval* (2023). https://api.semanticscholar.org/CorpusID:257078657

[28] James Mayfield, Eugene Yang, Dawn Lawrie, Sean MacAvaney, Paul McNamee, Douglas W. Oard, Luca Soldaini, Ian Soboroff, Orion Weller, Efsun Kayi, Kate Sanders, Marc Mason, and Noah Hibbler. 2024. On the Evaluation of Machine-Generated Reports. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, Washington DC USA, 1904–1915. doi:10.1145/3626772.3657846

[29] Donald Metzler and W Bruce Croft. 2007. Linear feature-based models for information retrieval. *Information Retrieval* 10, 3 (2007), 257–274.

[30] Maria Movin and Claudia Hauff. 2025. Zero-Shot Reranking with Large Language Models and Precomputed Ranking Features: Opportunities and Limitations. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Padua, Italy) *(SIGIR '25)*. Association for Computing Machinery, New York, NY, USA, 2276–2286. doi:10.1145/3726302.3730119

[31] Ronak Pradeep, Sahel Sharifymoghaddam, and Jimmy Lin. 2023. RankVicuna: Zero-Shot Listwise Document Reranking with Open-Source Large Language Models. arXiv:2309.15088 [cs.IR] https://arxiv.org/abs/2309.15088

[32] Ronak Pradeep, Sahel Sharifymoghaddam, and Jimmy Lin. 2023. RankZephyr: Effective and Robust Zero-Shot Listwise Reranking is a Breeze! arXiv:2312.02724 [cs.IR] https://arxiv.org/abs/2312.02724

[33] Ronak Pradeep, Nandan Thakur, Shivani Upadhyay, Daniel Campos, Nick Craswell, and Jimmy Lin. 2024. Initial Nugget Evaluation Results for the TREC 2024 RAG Track with the AutoNuggetizer Framework. arXiv:2411.09607 [cs.IR] https://arxiv.org/abs/2411.09607

[34] Zhen Qin, Rolf Jagerman, Kai Hui, Honglei Zhuang, Junru Wu, Le Yan, Jiaming Shen, Tianqi Liu, Jialu Liu, Donald Metzler, Xuanhui Wang, and Michael Bendersky. 2024. Large Language Models are Effective Text Rankers with Pairwise Ranking Prompting. In *Findings of the Association for Computational Linguistics: NAACL 2024*, Kevin Duh, Helena Gomez, and Steven Bethard (Eds.). Association for Computational Linguistics, Mexico City, Mexico, 1504–1518. doi:10.18653/v1/2024.findings-naacl.97

[35] Hossein A Rahmani, Nick Craswell, Emine Yilmaz, Bhaskar Mitra, and Daniel Campos. 2024. Synthetic test collections for retrieval evaluation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2647–2651.

[36] Hossein A. Rahmani, Emine Yilmaz, Nick Craswell, Bhaskar Mitra, Paul Thomas, Charles L. A. Clarke, Mohammad Aliannejadi, Clemencia Siro, and Guglielmo Faggioli. 2024. LLMJudge: LLMs for Relevance Judgments. https://arxiv.org/abs/2408.08896 arXiv:2408.08896.

[37] Mandeep Rathee, Sean MacAvaney, and Avishek Anand. 2025. Guiding Retrieval using LLM-based Listwise Rankers. arXiv:2501.09186 [cs.IR] https://arxiv.org/abs/2501.09186

[38] Devendra Sachan, Mike Lewis, Mandar Joshi, Armen Aghajanyan, Wen-tau Yih, Joelle Pineau, and Luke Zettlemoyer. 2022. Improving Passage Retrieval with Zero-Shot Question Generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (Eds.). Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 3781–3797. doi:10.18653/v1/2022.emnlp-main.249

[39] Weiwei Sun, Lingyong Yan, Xinyu Ma, Pengjie Ren, Dawei Yin, and Zhaochun Ren. 2023. Is ChatGPT Good at Search? Investigating Large Language Models as Re-Ranking Agent. *ArXiv* abs/2304.09542 (2023). https://api.semanticscholar.org/CorpusID:258212638

[40] Ming Tan, Tian Xia, Lily Guo, and Shaojun Wang. 2013. Direct optimization of ranking measures for learning to rank models. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. 856–864.

[41] Paul Thomas, Seth Spielman, Nick Craswell, and Bhaskar Mitra. 2023. Large language models can accurately predict searcher preferences. In *Annual International ACM SIGIR Conference on Research and Development in Information*

Retrieval. https://api.semanticscholar.org/CorpusID:262054847
[42] Shivani Upadhyay, Ronak Pradeep, Nandan Thakur, Daniel Campos, Nick Craswell, Ian Soboroff, Hoa Trang Dang, and Jimmy Lin. 2024. A Large-Scale Study of Relevance Assessments with Large Language Models: An Initial Look. arXiv:2411.08275 [cs.IR] https://arxiv.org/abs/2411.08275
[43] Shivani Upadhyay, Ronak Pradeep, Nandan Thakur, Nick Craswell, and Jimmy Lin. 2024. UMBRELA: UMbrela is the (Open-Source Reproduction of the) Bing RELevance Assessor. arXiv:2406.06519 [cs.IR] https://arxiv.org/abs/2406.06519
[44] William Walden, Marc Mason, Orion Weller, Laura Dietz, John Conroy, Neil Molino, Hannah Recknor, Bryan Li, Gabrielle Kaili-May Liu, Yu Hou, Dawn Lawrie, James Mayfield, and Eugene Yang. 2025. Auto-ARGUE: LLM-Based Report Generation Evaluation. arXiv:2509.26184 [cs.IR] https://arxiv.org/abs/2509.26184
[45] Yidong Wang, Yunze Song, Tingyuan Zhu, Xuanwang Zhang, Zhuohao Yu, Hao Chen, Chiyu Song, Qiufeng Wang, Cunxiang Wang, Zhen Wu, Xinyu Dai, Yue Zhang, Wei Ye, and Shikun Zhang. 2025. TrustJudge: Inconsistencies of LLM-as-a-Judge and How to Alleviate Them. arXiv:2509.21117 [cs.AI] https://arxiv.org/abs/2509.21117
[46] Lulu Yu, Keping Bi, Jiafeng Guo, Shihao Liu, Shuaiqiang Wang, Dawei Yin, and Xueqi Cheng. 2025. Can LLM Annotations Replace User Clicks for Learning to Rank? arXiv:2511.06635 [cs.IR] https://arxiv.org/abs/2511.06635
[47] Shengyao Zhuang, Honglei Zhuang, Bevan Koopman, and Guido Zuccon. 2024. A setwise approach for effective and highly efficient zero-shot ranking with large language models. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 38–47.

A Appendix

A.1 MAP@20 Results

To provide a fair comparison with NDCG@20, we also evaluate improvements using MAP@20 (Mean Average Precision at rank 20). While NDCG@20 uses logarithmic discounting and graded relevance, MAP@20 treats relevance as binary and computes average precision within the top 20 positions. Both metrics evaluate the same ranking depth.

Table 5 presents both metrics side-by-side, using the same evaluation methodology (absolute gain Δ, gain ratio G, and geometric mean gain ratio $\overline{G}$) as defined in Section 4. The patterns are consistent: MAP@20 improvements mirror NDCG@20, with largest gains in lower-performing systems (0% to 25% range) diminishing as system quality increases. Similar geometric mean gain ratios in the 0% to 100% range confirm that Multi-Criteria-L2R improves retrieval systems across both graded relevance (NDCG@20) and binary precision-based (MAP@20) evaluation frameworks.

Table 5: Mean absolute gain ($\overline{\Delta}_{\text{MAP@20}}$) and geometric mean gain ratios ($\overline{G}_{\text{MAP@20}}$) of MAP@20 for systems in which the learning-to-rank model (L2R) combines all Multi-Criteria LLM-Judge grades as ranking features (exactness, coverage, topicality, contextual fit) with the original system retrieval score. This table parallels Table 2 but evaluates using MAP@20 instead of NDCG@20. Results are across three datasets and grouped by base performance percentiles. For $\overline{\Delta}_{\text{MAP@20}}$, higher values indicate greater improvement. For $\overline{G}_{\text{MAP@20}}$, values above 1.0 indicate improvement, equal to 1.0 indicates no change, and values below 1.0 indicate a decrease in performance.

System / Model	$\overline{\Delta}_{\text{MAP@20}}$ / $\overline{G}_{\text{MAP@20}}$						
	0–5%	5–25%	25–50%	50–75%	75–95%	95–100%	0–100%
TREC DL 2019							
FLAN-T5-large	0.108/ 4.124	0.105/ 1.626	0.056/ 1.261	0.028/ 1.104	0.005/ 1.017	0.012/ 1.035	0.051/ 1.306
LLaMA-3-8B	0.102/ 3.907	0.102/ 1.611	0.053/ 1.242	0.036/ 1.134	0.019/ 1.060	0.025/ 1.078	0.054/ 1.316
LLaMA-3.3-70B	0.131/ 4.516	0.152/ 1.906	0.085/ 1.386	0.059/ 1.218	0.035/ 1.111	0.054/ 1.164	0.084/ 1.455
TREC DL 2020							
FLAN-T5-large	0.002/ 1.241	0.107/ 1.585	0.047/ 1.172	*-0.001/ 0.997*	0.002/ 1.004	*-0.004/ 0.992*	0.034/ 1.156
LLaMA-3-8B	0.002/ 1.160	0.133/ 1.720	0.061/ 1.219	0.001/ 1.002	*-0.000/ 1.000*	*-0.001/ 0.997*	0.043/ 1.184
LLaMA-3.3-70B	0.001/ 1.136	0.246/ 2.322	0.151/ 1.531	0.062/ 1.156	0.031/ 1.073	0.009/ 1.019	0.111/ 1.402
TREC DL 2023							
FLAN-T5-large	0.023/ 1.613	0.035/ 1.583	0.033/ 1.286	0.032/ 1.190	0.011/ 1.054	0.011/ 1.044	0.028/ 1.272
LLaMA-3-8B	0.026/ 1.687	0.026/ 1.354	0.055/ 1.469	0.043/ 1.258	0.011/ 1.054	0.005/ 1.019	0.035/ 1.296
LLaMA-3.3-70B	0.033/ 1.871	0.060/ 1.947	0.062/ 1.515	0.063/ 1.370	0.023/ 1.119	0.002/ 1.008	0.051/ 1.462