LLM-Augmented Relevance Feedback: Generative Feedback with Automatic LLM Judges for Conversational Search

Anonymous ACL submission

Abstract

Recent research on Large Language Model (LLM) judges in search largely focuses on their role as offline evaluators. Instead, this paper investigates using LLMs closer to simulation, focusing on using them as proxies for human feedback. We present LLM-Augmented Relevance Feedback (LARF), which synthesises the latest LLM Judge methods with *Query Reformulation* and *Query-by-Document* relevance feedback integration approaches to improve the set of candidate documents. We perform experiments on standard conversational search benchmarks, TREC iKAT and CAsT. Our work address three research questions: (1) What is the retrieval benefit when LARF is used with human feedback? (2) How does noise in relevance judgements impact downstream feedback effectiveness? (3) What are issues with the current LLM judges when used with LARF? We find that with human judgements, *Query-by-Document* achieves new state-of-the-art results, significantly outperforming previous work (48% nDCG@3 on CAsT). We study how effectiveness degrades as judgements become noisier. And, when using current automatic LLM judges, we find 18% nDCG@3 gain over previous state-of-the-art on CAsT. We conclude that LARF offers a new and effective mechanism for improving retrieval quality in conversational search and highlight the need for reducing noise, particularly for complex personalised tasks.

1 Introduction

The role of Large Language Models (LLMs) in search quality evaluation is hotly debated. Proponents cite LLMs' ability to approximate human judgments at scale, reduce evaluation costs, and ensure benchmark consistency (Thomas et al., 2024; MacAvaney and Soldaini, 2023; Upadhyay et al., 2024b,c,a). Conversely, LLM evaluation may introduce bias, hallucinate relevance, reinforce training data patterns, or impose artificial performance ceilings, distorting outcomes and hindering system comparison (Takehi et al., 2024; Faggioli et al., 2023; Soboroff, 2025; Dietz et al., 2025; Clarke and Dietz, 2024).

Instead of focusing on LLMs for offline evaluation to replace human judges, we instead focus on how they can be leveraged to improve core retrieval effectiveness during search. Soboroff (2025) argues that asking an LLM to predict document relevance is functionally identical to asking it to rank documents. In this work, we experiment with using the same state-of-the-art LLM judges used offline and leverage them as part of automatic online feedback during the search process.

The concept of simulating user preferences to guide retrieval is well-established, and relevance feedback (RF) is a primary mechanism by which this simulation is realised. Such feedback, aiming to improve effectiveness by acting on user preference signals, can be sourced from dedicated user simulators generating interactive responses (Owoicho et al., 2023; Salle et al., 2022; Sekulić et al., 2022), or through judgements intrinsic to techniques from classical Rocchio (Rocchio, 1971) and RM3 (Abdul-Jaleel et al., 2004) to modern generative RF (GRF) (Mackie et al., 2023). Our work synthesises these threads, proposing LLM judges as active sources of generative RF, dynamically steering search towards improved effectiveness.

We propose a new approach, LLM-Augmented Relevance Feedback (LARF). LARF has two complementary types of feedback policies for integrating LLM-generated feedback into retrieval: a *Query-Reformulation Policy* that integrates the feedback from the top-N candidate documents of an initial retrieval pass and a *Query-By-Document Policy* that utilises the larger set of candidates. Both enrich retrieval by injecting targeted relevance information at key decision points, using LLM judgements to guide the search system.

We systematically evaluate the efficacy and

boundaries of this role for LLM judges by asking:

RQ1 What is the retrieval benefit when LARF is used with human relevance judgements?

RQ2 How does retrieval effectiveness degrade as controlled noise is introduced to human judgements?

RQ3 How do current state-of-the-art LLM judges perform with LARF in the context of RQ2? And what is the impact on end-to-end effectiveness when compared with human feedback?

We perform experiments on multiple standard conversational search benchmarks, TREC iKAT 2023 (Aliannejadi et al., 2024) and TREC CAsT 2022 (Owoicho et al., 2022). We find that human feedback significantly and dramatically improves retrieval effectiveness. We characterise how that effectiveness degrades as noise is added. We find that depending on the policy used, current LLMs are just on the cusp of providing benefits and need further improvement providing gains of up to 18% in nDCG@3 over previous systems on TREC CAsT. The results also highlight that more complex personalised tasks in iKAT result in higher levels of LLM judge noise, showing an important area for future work.

2 Retrieval Pipeline

We address enhancing search system output quality via feedback. Given an initial query q, we use document-level relevance feedback to refine the results presented to the user. Specifically, we aim to improve the *retrieval pool*, documents ultimately reranked and surfaced as results, by leveraging this feedback. Our approach, depicted in Figure 1, operates in a multi-step pipeline:

1. **Candidate Generation:** Given query q, a base search system retrieves an initial candidate set of documents $\mathcal{D}_c$ from the corpus.

2. **Document Feedback:** A judge provides relevance feedback (score or category, e.g., highly relevant) for each $d \in \mathcal{D}_c$ with respect to q, simulating a user's initial assessment.

3. **Pruning:** Documents below a relevance threshold θ are pruned from $\mathcal{D}_c$, yielding $\mathcal{D}_p \subseteq \mathcal{D}_c$. This aligns with information foraging's principle of abandoning low 'information scent' patches (Pirolli and Card, 1999), focusing resources on promising candidates.

4. **Pool Expansion:** $\mathcal{D}_p$ documents seed retrieval of additional related documents $\mathcal{D}_e$ to enrich the candidate set $\mathcal{D}_c$, uncovering items missed by initial retrieval. This mirrors information foraging's exploration of new patches from cues in exploited ones (Pirolli, 2007). Seed selection and expansion use one of two policies:

   - The **Query-Reformulation Policy** (Gupta and Dixit, 2023; Hust et al., 2002; Al-Thani et al., 2023) uses a small number top-ranked documents in $\mathcal{D}_p$ as seeds. This policy emphasises *exploitation* of strong signals of relevance, akin to deeply mining a high-scent information patch.
   - The **Query-By-Document Policy** (Abolghasemi et al., 2022; Yang et al., 2009; Weng et al., 2011) draws seeds from a broader range of documents within $\mathcal{D}_p$, allowing the system to *explore* more diverse or peripheral content areas and promote information patch enrichment.

5. **Reranking:** The updated pool $\mathcal{D}_p \cup \mathcal{D}_e$ is reranked to produce the final ranked list presented to the user.

This framework enables investigation of how feedback type and fidelity impact retrieval effectiveness and allows quantification of system robustness to feedback imperfections. Section 3 details policy instantiations and mechanisms employed for each step in the pipeline.

3 Implementation

3.1 Retrieval Pipeline

Our retrieval pipeline, as outlined in Section 2, is implemented as follows:

3.1.1 Candidate Generation

Initial candidates $\mathcal{D}_c$ are retrieved using a BM25 (Robertson et al., 1995) + MonoT5 (Nogueira et al., 2020) baseline. This choice isolates our feedback mechanism's impact and facilitates rapid experimentation with simpler, faster components. Indexing and BM25 retrieval (parameters $k_1 = 4.46$, $b = 0.82$ based on Castorini (2023) from a similar document retrieval task; up to 1000 docs/query) use Pyserini (Lin et al., 2021). These 1000 documents

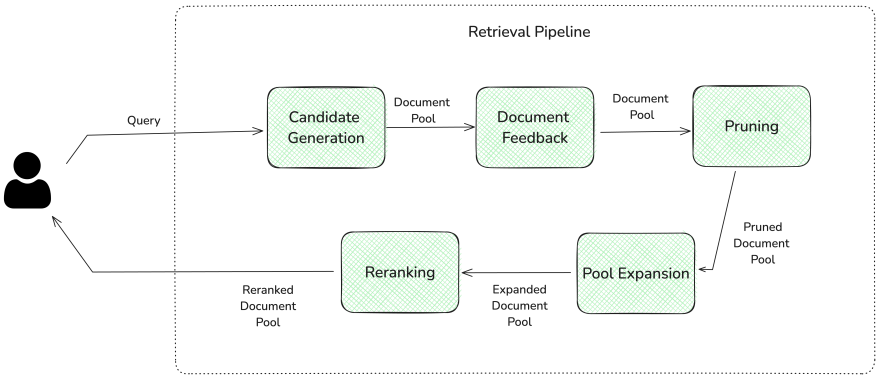


Figure 1: Overview of our feedback driven retrieval pipeline. 1. **Candidate Generation**: An initial document pool ($\mathcal{D}_c$) is retrieved. 2. **Document Feedback:** A judge assesses relevance (information scent) for each document. 3. **Pruning**: Documents with insufficient scent are removed, forming $\mathcal{D}_p$. 4. **Pool Expansion:** $\mathcal{D}_p$ seeds the retrieval of new documents $\mathcal{D}_e$ to enrich the pool, guided by foraging principles. 5. **Reranking**: The final pool ($\mathcal{D}_p \cup \mathcal{D}_e$) is reranked for presentation.

are then reranked by a MONOT5[1] model (trained on MS MARCO passage ranking (Nguyen et al., 2016)) to form the candidate set $\mathcal{D}_c$.

3.1.2 Document Feedback

We explore three distinct types of "judges" to simulate feedback under varying levels of accuracy and realism, matching our research questions:

Human Judge We simulate a human judge based on professional TREC[2] assessments included in our target benchmarks' relevance judgements. Documents without ground-truth labels are assumed to be non-relevant.

Noisy Human Judge For robustness analysis, we design a Noisy Judge that simulates imperfect feedback by injecting Bernoulli noise (Frénay and Verleysen, 2013; Bernoulli, 1713) into the human judgements. With probability p, a document's score is replaced by a random incorrect score. This models a generic probabilistically imperfect annotator to study retrieval robustness at a specified error rate p. Experiments with this judge are averaged over 5 runs with 95% confidence intervals due to the stochasticity introduced.

Automatic LLM Judge We study representative LLM-based judges and one non-LLM baseline judge. The LLM judges were prominent participants in the LLM Judge Challenge at SIGIR 2024 (Rahmani et al., 2025), which evaluated automatic relevance assessment approaches on TREC 2023 Deep Learning track (Craswell et al., 2024) judgements. Judges are designed to predict a relevance score on a 0-3 scale. We select judges based on their reported performance on key inter-rater reliability metrics in the challenge's overview paper and validated our implementations on a 5-fold split of the challenge's dev set, supplemented with 3000 randomly selected relevance judgements from our target IKAT and CAST benchmarks.

- WILLIA-UMBRELA1 (GPT-4O[3]): Achieves strong Cohen's κ via zero-shot prompting with the UMBRELA framework (Upadhyay et al., 2024c) based on the techniques introduced in Thomas et al. (2024).
- OLZ-GPT4O (GPT-4O):, Achieves strong Krippendorff's α via a simple prompt asking for the relevance judgement directly.
- TREMA-4PROMPTS (LLAMA-3-8B-INSTRUCT[4]): Achieves high Kendall's τ and Spearman's Rank Correlation by decomposing relevance into four criteria (exactness, coverage, topicality, contextual fit), assessing for each independently, then combining them to determine overall relevance (Farzi and Dietz, 2024).
- H2OLOO-FEWSELF (GPT-4O): Achieves high Krippendorff's α via prompting techniques from Thomas et al. (2024), similar to WILLIA-UMBRELA1, but with in context examples.

[1] https://huggingface.co/castorini/monot5-base-msmarco-10k

[2] https://trec.nist.gov/

[3] https://platform.openai.com/docs/models/gpt-4o

[4] https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct

- MONOT5-JUDGE (Non-LLM): Based on the MONOT5 reranker. We reinterpret its probability of generating "true" (for relevance) as a 0-3 score by scaling this probability by 3 and rounding to the nearest integer.

3.1.3 Pruning

$\mathcal{D}_c$ documents are pruned via judge feedback using a benchmark-specific relevance threshold θ. Documents scoring below θ are removed, yielding $\mathcal{D}_p$. For CAsT, ≥ 2 is relevant; for iKAT, ≥ 1. We set θ accordingly, after mapping our judge scores to benchmark relevance scales (Section 3.3).

3.1.4 Pool Expansion

To enrich $\mathcal{D}_p$ up to a rerank budget of 1000 unique documents, matching the evaluation depth of our target benchmarks, we apply one of two feedback policies:

Query-Reformulation Policy Expansion seeds are the top N documents in $\mathcal{D}_p$ (by original MONOT5 score; N empirically set to 1 due to diminishing returns observed in preliminary experiments, see Appendix A). For each seed, one of three methods generates a new BM25 query to retrieve $\mathcal{D}_e$: (1) GPT-4O-MINI[5] *summary* of seeds with respect to q (see Appendix C.1 for prompt); (2) GPT-4O-MINI *reformulated query* from seeds (see Appendix C.2 for prompt); or (3) the combined *full text*s of the seeds.

Query-By-Document Policy This policy iterates through documents in $\mathcal{D}_p$ (sorted by descending initial MONOT5 scores). For each seed in $\mathcal{D}_p$, we retrieve a single unique neighbour for $\mathcal{D}_e$ using one of three distinct methods per run: (1) *Summary*: GPT-4O-MINI summary of seed w.r.t. q as BM25 query (Appendix C.3); (2) RM3: seed text with RM3 expansion as BM25 query; or (3) *Full Text*: seed's full text as BM25 query

Unique retrieved neighbours are added to $\mathcal{D}_e$. Expansion stops when $|\mathcal{D}_e| = 1{,}000$ or seeds are exhausted. We use one neighbour per seed to optimise top-rank precision (e.g., nDCG@3), as preliminary tests (Appendix B) showed deeper expansions hurt precision on target benchmarks (Section 3.3).

3.2 Reranking

The final pool $\mathcal{D}_p \cup \mathcal{D}_e$ (up to 1000 documents) is reranked by MONOT5 with respect to query q to produce the system's output.

3.3 Experimental Setup

3.3.1 Benchmarks

We evaluate primarily on TREC CAsT 2022 (Owoicho et al., 2022) and iKAT 2023 (Aliannejadi et al., 2024). These offer (1) challenging, realistic scenarios with evolving needs, a robust testbed for feedback; (2) standardised TREC data/protocols for reproducibility; and (3) a rich conversational context where feedback is conceptually valuable.

Both datasets include information needs that unfold over multiple conversational turns. To isolate our feedback mechanisms from query-reformulation complexity, we use the context-independent "resolved utterance" variants. Note, however, that in the iKAT benchmark some resolved utterances still rely on additional context in so-called Personal Text Knowledge Bases (PTKBs). For example, the query, *"What should I cook for dinner?"* implicitly depends on PTKB details like *"healthy and tasty recipes for my family"*. Although we treat the resolved queries as standalone, this unmodeled PTKB context can introduce noise or ambiguity in relevance judgements, capping performance for both the baseline and our feedback methods.

CAST has 18 topics/conversations with an average of 11.39 turns. It uses a document collection derived from MS MARCO v2 (Craswell et al., 2022), KILT (Petroni et al., 2020), and the Washington Post [6]. IKAT extends CAST's focus to multi-persona conversations, comprising 25 test topics. It draws documents from the ClueWeb22-B (Overwijk et al., 2022) corpus of approximately 117 million documents.

Both benchmarks use a 0-4 relevance scale (0-Fails to meet; 4-Fully meets). As our judges use a 0-3 scale, we map benchmark judgements by collapsing original scores '3' (Highly) and '4' (Fully) into a single '3'. This applies to our Human Judge and for interpreting relevance thresholds (e.g., an original ≥ 2 remains ≥ 2 on our 0-3 scale).

All experiments use the context-independent "resolved utterance" query variants included in the benchmarks that have conversational ambiguity disambiguated by humans. We chose this to focus on the feedback elements instead of the noisy conversational query understanding elements.

[5] https://platform.openai.com/docs/models/gpt-4o-mini

[6] https://trec.nist.gov/data/wapost/

3.3.2 Evaluation Protocol

We evaluate performance using official benchmark measures (primarily nDCG@3; also Recall, MRR, and Precision). Our main baseline is BM25+MONOT5 (Section 3.1.1) without feedback. We include prior SOTA from IKAT/CAST overview papers for context. Paired t-tests ($p < 0.05$) assess significance against our baseline. For LLM judge validation (Section 3.1.2) and alignment with human evaluations, we report Cohen's κ and Krippendorff's α. We run all experiments on a server running NVIDIA RTX 6000 Ada Generation Graphics Card, with each experiment taking between 2 to 8 hours.

4 Results and Discussion

This section presents and discusses experimental results, framed by our research questions: (RQ1) LARF's effectiveness with human feedback, (RQ2) LARF's robustness to noise in human feedback, and (RQ3) LARF's effectiveness with automatic LLM judges.

4.1 RQ1: LARF with Human Feedback

We employed the Human Judge (Section 3.1.2) with Query-Reformulation and Query-By-Document feedback policies on IKAT and CAST. Performance (Table 1) is compared against our BM25+MONOT5 baseline (no feedback) and prior state-of-the-art systems for IKAT (Aliannejadi et al., 2024) and CAST (Owoicho et al., 2022).

The results compellingly show the significant, often transformative, potential of integrating true relevance feedback directly into the retrieval pipeline. On both IKAT and CAST, both policies substantially outperformed the BM25+MONOT5 baseline across most key metrics, validating our hypothesis that repurposing relevance judgments for active, in-pipeline modification yields considerable benefits with high-quality feedback.

4.1.1 Query Reformulation Policy

With the Human Judge, this policy significantly boosted recall. IKAT, R@1000 rose from 0.451 (baseline) to 0.508-0.552; on CAST, from 0.463 to 0.565-0.631 (Table 1). This confirms that human-verified expansion from strong initial candidates effectively brings more relevant documents into the 1000-document pool. However, these recall gains translated to less pronounced top-rank precision improvements (e.g., nDCG@3 on IKAT: 0.288 $\rightarrow$ 0.338-0.355; CAST: 0.508 $\rightarrow$ 0.545-0.583). We attribute this to the policy's exploitative expansion retrieving a pool of mixed quality; while richer in relevant documents, some lower-quality inclusions challenge the final reranker's ability to surface the very best items. Across both benchmarks, the "Full Text" expansion variant performed best or comparably to summary-based methods, suggesting that with perfect feedback, the inherent quality of human-relevance judgements is effective, limiting the added benefit of complex query/summary generation.

4.1.2 Query-By-Document Policy

This policy yielded dramatic improvements in precision-oriented metrics. On IKAT, nDCG@3 surged from a 0.288 baseline (0.412 prior) to 0.622-0.683, and AP@1000 from 0.128 (0.191 prior) to 0.406-0.451. On CAST, gains were similarly striking: nDCG@3 improved from 0.508 (0.513 prior) to 0.750-0.763, and MRR rose from 0.708 (0.717 prior) to 0.967. Critically, under oracle/human conditions, the Query-By-Document policies establish new SOTA effectiveness on both IKAT and CAST across key top-rank metrics, significantly exceeding prior results. This highlights the power of leveraging diverse, high-quality relevance signals for pool enrichment.

Intriguingly, this exceptional precision often came with R@1000 figures at or slightly below baseline levels (iKAT 0.453; CAST 0.463, matching baseline). This occurs because selecting diverse seeds and retrieving only a single neighbour per seed, combined with pruning non-relevant documents, yields a highly refined, though not necessarily larger, unique relevant document set. This higher-quality, higher-precision initial pool enables the final re-ranker (MONOT5) to perform more effectively, leading to superior top-rank outcomes. Conversely, the Query-Reformulation policy, despite higher R@1000, creates a "noisier" pool, diluting reranker effectiveness. Mitigating this R@1000 behaviour in the Query-By-Document policy while preserving precision (e.g., via alternative seed/neighbour selection) is future work.

Consistent with the Query-Reformulation, "Full Text" or "RM3" expansion generally outperformed summary-based approaches for Query-By-Document on both datasets, reinforcing that richer seed representations benefit from high-quality feedback.

System	iKAT							CAsT				
	nDCG@3	nDCG@5	nDCG@1000	P@20	R@20	R@1000	AP@1000	nDCG@3	nDCG@1000	MRR	R@1000	AP@1000
BM25+MONOT5	0.288	0.287	0.333	0.248	0.141	0.451	0.128	0.508	0.426	0.708	0.463	0.223
Best System	0.412	0.426	0.325	0.353	0.206	0.316	0.191	0.513	0.485	0.717	0.557	0.257
Query-Reformulation Policy												
(1) Summary	0.352	0.343	0.406	0.305	0.174	0.549	0.174	0.583	0.531	0.795	0.622	0.321
(2) Reformulated Query	0.338	0.331	0.381	0.289	0.162	0.508	0.160	0.545	0.487	0.751	0.565	0.282
(3) Full Text	0.355	0.349	0.412	0.311	0.185	**0.552**	0.179	0.580	**0.533**	0.802	**0.631**	0.324
Query-By-Document Policy												
(1) Summary	0.622	0.621	0.517	0.620	0.349	0.453*	0.406	0.750	0.462	0.965	0.463*	0.448
(2) RM3	0.664	0.667	0.535	0.651	0.361	0.452*	0.437	0.757	0.464	0.967	0.463*	0.455
(3) Full Text	**0.683**	**0.682**	**0.541**	**0.668**	**0.367**	0.451*	**0.451**	**0.763**	0.465	**0.967**	0.463*	**0.462**

Table 1: Upper bound retrieval performance on the iKAT and CAsT benchmarks using the Human Judge. Compares our Query-Reformulation and Query-By-Document feedback integration policy variants against the BM25+MONOT5 baseline (with no feedback) and the best systems previously reported for IKAT (Aliannejadi et al., 2024) and CAST (Owoicho et al., 2022). Sign * indicates a difference that is **NOT** statistically significant ($p \geq 0.05$) compared to the BM25+MONOT5 baseline. Best results achieved by our systems are shown in bold.

4.2 RQ2: LARF's Robustness to Noisy Human Feedback

Our Noisy Human Judge injects Bernoulli noise (error probability p, 10%-100%; $p = 0\%$ is RQ1's Human Judge) into the Human Judge's labels. We tested RQ1's best "Full Text" Query-Reformulation and Query-By-Document policy variants on iKAT and CAsT. Figure 2 shows nDCG@3 and R@1000 vs. noise; Figure 3 presents Cohen's κ and Krippendorff's α agreement between Noisy Judges and human labels.

As expected, retrieval performance generally degrades with increasing noise p (Figure 2), as do agreement scores (Figure 3). Critically, despite this, noisy feedback can still benefit retrieval over the BM25+MONOT5 baseline up to specific noise thresholds.

4.2.1 Query-Reformulation Policy

On IKAT (Figure 2a) and CAST (Figure 2c), R@1000 remained above baseline even at high noise (e.g., $p \approx 70\%$). This suggests its exploitative expansion maintains a recall advantage despite significant inaccuracies. nDCG@3, though more noise-sensitive, demonstrated a "safe zone" too, staying above baseline until $p \approx 70 - 80\%$ on IKAT and $p \approx 30\%$ on CAST. This indicates a considerable error margin before top-rank quality degrades below baseline. Benefits persist with substantial noise if *some* genuinely relevant seeds are identified.

4.2.2 Query-By-Document Policy

The Query-By-Document policy for nDCG@3 in RQ1, showed a similar robustness profile. While its nDCG@3 gains were susceptible to noise, effectiveness remained above baseline for a significant range; up to $p \approx 75\%$ on IKAT (Figure 2b) and $p \approx 55\%$ on CAST (Figure 2d). This implies its exploration/enrichment precision benefits are somewhat resilient, though the margin shrinks rapidly. As Query-By-Document did not initially improve R@1000 (RQ1), noise generally kept recall at/below baseline. This policy's primary benefit is top rank precision, which noise erodes without compensatory recall gains.

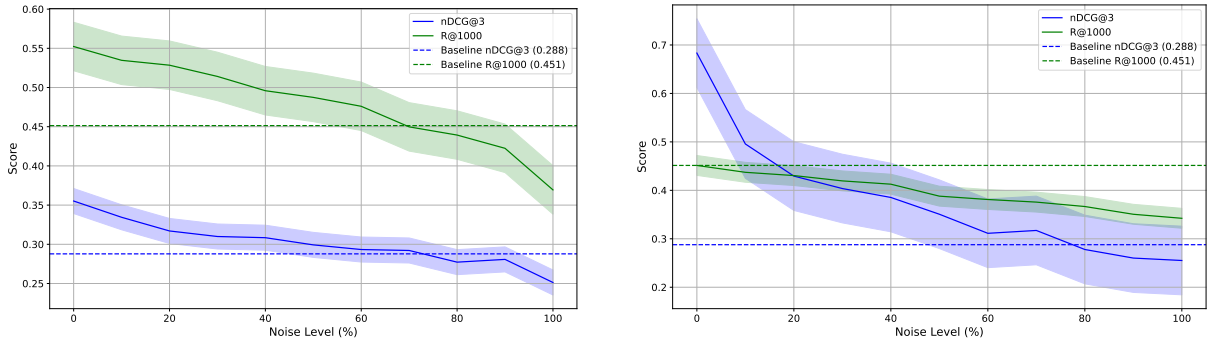
System	LLMJudgeBenchmark		iKAT		CAsT	
	κ	α	κ	α	κ	α
WILLIA-UMBRELA1	0.223	0.389	0.290	0.487	0.245	0.477
OLZ-GPT4O	0.248	0.457	0.356	0.608	0.309	0.567
TREMA-4PROMPTS	0.067	0.113	0.192	0.354	0.134	0.252
H2OLOO-FEWSELF	0.266	0.512	0.328	0.525	0.266	0.550
MONOT5-JUDGE	0.171	0.308	0.267	0.433	0.199	0.369

Table 2: Cohen's κ and Krippendorff's α agreement between various LLM judges (and our MonoT5-Judge) and perfect human judgements across the LLM-JUDGEBENCHMARK dev set, IKAT and CAST. Higher values suggest greater alignment with human evaluators.

4.3 RQ3: LARF with Automatic LLM Judges

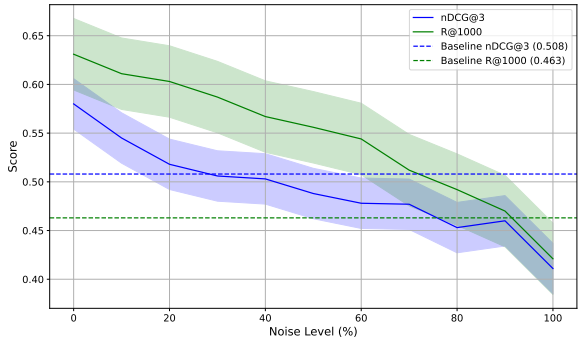
Our central finding is that automatic LLM Judges can significantly improve retrieval over the BM25+MONOT5 baseline, though not reaching Human Judge levels. Several configurations yielded statistically significant gains (Table 3). For example, OLZ-GPT4O with the Query-By-Document policy improved iKAT nDCG@3 from 0.288 to 0.364. On CAsT, this combination achieved nDCG@3 of 0.605 (vs. 0.508 baseline), surpassing the prior work of 0.513 (Table 1).

Policy characteristics from RQ1 largely hold, but with clear degradation. Query-Reformulation with LLM Judges (e.g., OLZ-GPT4O) enhanced R@1000 (iKAT: 0.479 vs. 0.451 baseline), but nDCG@3 gains were modest (iKAT: 0.312 vs.

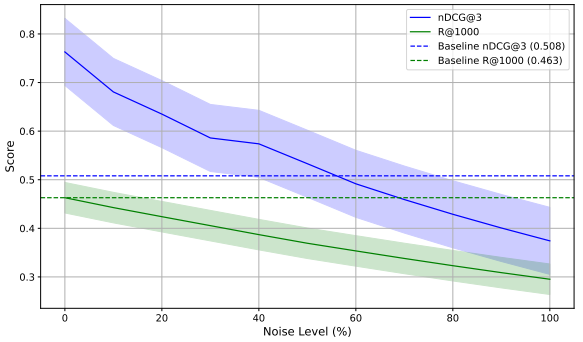


(a) iKAT, Query-Reformulation

(b) iKAT, Query-By-Document



(c) CAsT, Query-Reformulation



(d) CAsT, Query-By-Document

Figure 2: Performance on iKAT (top row) and CAsT (bottom row) benchmarks as a function of increasing noise levels in the Human Judge. Results are shown for the best-performing configurations of the Query-Reformulation (right column) and Query-By-Document (left column) policies identified in RQ1 (see Table 1). Solid lines represent the mean performance over 5 runs; shaded areas indicate 95% confidence intervals. Dashed horizontal lines show the performance of the BM25+MonoT5 baseline (no feedback) for reference.

0.288) and sometimes insignificant. Conversely, Query-By-Document with effective LLMs (e.g., OLZ-GPT4O) delivered stronger nDCG@3 improvements (iKAT: 0.364; CAsT: 0.605), underscoring its exploration strategy's value with reasonably accurate judgments.

A positive correlation emerges between judge agreement (Table 2) and top-rank retrieval impact. For example, OLZ-GPT4O (CAsT $\kappa = 0.309$) produced strong nDCG@3. This κ suggests $p \approx 40 - 50\%$ effective noise (Figure 3). Our RQ2 analysis (Figure 2) predicted Query-By-Document would remain above baseline here; OLZ-GPT4O's strong performance (Table 3) aligns, confirming operation within "safe zones". Conversely, lower-agreement judges like TREMA-4PROMPTS (iKAT $\kappa = 0.192$, implying higher effective noise) yielded top-rank performance near or not significantly above baseline.

Finally, as discussed (Section 3.3), iKAT's implicit PTKB dependencies likely contributed to relatively more modest gains there versus CAsT. LLM judges lacking PTKB context may struggle to align with human judgments reliant on it, limiting effectiveness on iKAT.

5 Conclusion

This paper investigated a novel role for LLM Judges in information retrieval, shifting their application from external evaluators to active, internal components that provide real-time, generative relevance feedback. We introduced two distinct feedback integration policies, Query-Reformulation and Query-By-Document, designed to leverage LLM judgements for pruning and expanding the candidate document pool within a multi-stage retrieval pipeline.

Our systematic evaluation across the iKAT and CAsT benchmarks yielded three key insights. Firstly, under ideal conditions with Human Feedback, our approach demonstrates substantial potential, with the Query-By-Document policy achieving state-of-the-art performance that significantly surpassed baselines and prior results (RQ1). Secondly, we establish that these feedback mechanisms are remarkably robust to noise. Even with considerable

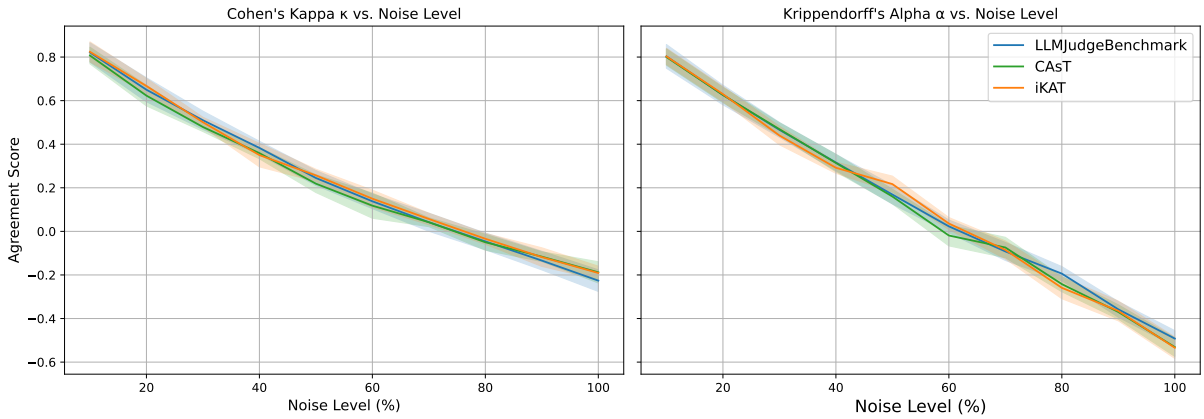


Figure 3: Cohen's κ and Krippendorff's α agreement between a simulated noisy judge (with 0-100% Bernoulli error probability p) and human judgements across the LLMJUDGEBENCHMARK dev set, CAST, and IKAT. Solid/dashed lines represent the mean agreement over 5 simulation runs, with shaded areas indicating 95% confidence intervals.

	iKAT							CAsT				
System	nDCG@3	nDCG@5	nDCG@1000	P@20	R@20	R@1000	AP@1000	nDCG@3	nDCG@1000	MRR	R@1000	AP@1000
BM25+MONOT5	0.288	0.287	0.333	0.248	0.141	0.451	0.128	0.508	0.426	0.708	0.463	0.223
Best System	0.412	0.426	0.325	0.353	0.206	0.316	0.191	0.513	0.485	0.717	0.557	0.257
Query-Reformulation Policy												
WILLIA-UMBRELA1	0.311	0.302	0.351	0.255*	0.144*	0.476	0.138	0.542	0.491	0.742	0.564	0.275
OLZ-GPT4O	0.312	0.304	**0.358**	0.260	0.145*	0.479	0.140	0.545	0.495	0.746	0.569	**0.279**
TREMA-4PROMPTS	0.288*	0.284*	0.350	0.247*	0.140*	0.489	0.133	0.515*	0.485	0.700*	0.556	0.266
H2OLOO-FEWSELF	0.311	0.300	0.352	0.254	0.143*	0.476*	0.137	0.539	0.494	0.740	0.561	0.275
MONOT5-JUDGE	0.285*	0.278	0.348	0.241*	0.139*	**0.490**	0.131*	0.506*	**0.496**	0.701	**0.583**	0.266
Query-By-Document Policy												
WILLIA-UMBRELA1	0.331	0.319	0.316	0.273	0.153	0.370	0.133	0.594	0.315	**0.781**	0.260	0.187
OLZ-GPT4O	0.364	0.349	0.334*	0.300	0.169	0.366	0.148	**0.605**	0.335	**0.781**	0.284	0.208
TREMA-4PROMPTS	0.312	0.307	0.340	0.270	0.154	0.435	0.140	0.529	0.417	0.711*	0.432	0.231
H2OLOO-FEWSELF	0.331	0.320	0.322	0.275	0.155	0.381	0.134	0.586	0.371	0.762	0.329	0.225*
MONOT5-JUDGE	0.288*	0.287*	0.319	0.248*	0.141*	0.413	0.126	0.508*	0.407	0.708*	0.429	0.218

Table 3: Retrieval performance on the iKAT and CAsT benchmarks using the LLM Judges. Compares our Query-Reformulation and Query-By-Document feedback integration policy variants against the BM25+MONOT5 baseline (with no feedback) and the best systems previously reported for IKAT (Aliannejadi et al., 2024) and CAST (Owoicho et al., 2022). Sign * indicates a difference that is **NOT** statistically significant ($p \geq 0.05$) compared to the BM25+MONOT5 baseline. Best results achieved by our systems are shown in bold.

inaccuracies in judgements, Query-Reformulation and Query-By-Document policies often outperformed a system with no feedback, defining "safe operational zones" for practical deployment (RQ2). Thirdly, and most critically, experiments with automatic LLM judges confirm LARF continues to be effective. The best-performing LLM judges, when integrated into our Query-By-Document policy, led to significant improvements over the baseline and even achieved new state-of-the-art results on the CAST benchmark. The effectiveness of these automatic judges correlated with their agreement with human judgements and their operation within the identified robustness thresholds.

Our findings collectively argue for a paradigm shift in how LLM capabilities are used in search. Rather than just relying on them for post-hoc evaluation, their inherent ability to simulate user preference and predict relevance can be constructively embedded within the retrieval process to dynamically enhance search quality. While challenges related to the "effective noise" of current LLM judges and dataset-specific nuances (like the IKAT PTKBs) remain, our work provides strong evidence that LLMs are powerful and practical tools for building more effective search systems when deployed as internal feedback providers.

Future work could explore more sophisticated feedback integration strategies, investigate the cost-benefit trade-offs of different LLM judges, and extend this framework to incorporate richer contextual information, such as user profiles or conversational history, into the feedback generation process. Further, as both feedback policies are complementary, combining them to unlock wholistic retrieval gains is an exciting prospect.

6 Limitations

While our findings demonstrate the promising potential of repurposing LLM judges for in-pipeline feedback, we acknowledge several limitations:

Cost and Latency: Our current framework, particularly the Query-By-Document policy, involves multiple calls to an LLM for feedback on numerous documents and potentially for generating summaries/queries during expansion. This incurs significant cost (if using closed-source LLMs) and latency, making the current instantiation not directly "deployable" in many real-time search scenarios. However, practical systems may not always need to process or rerank up to 1,000 documents; a smaller, more targeted application might be feasible. Furthermore, future advancements in LLM efficiency, smaller specialised models, or caching strategies could mitigate these concerns.

Generalisability: Our experiments were conducted using a BM25 + MonoT5 baseline for retrieval tasks. While this is a standard baseline (Saha et al., 2022; Almeida and Matos, 2024; Rosa et al., 2022b,a), different base retrieval architectures (e.g., dense retrievers, more complex multi-stage systems) might interact differently with our feedback policies, potentially yielding varying magnitudes of improvement or different optimal policy configurations.

Query Type: The CAsT and iKAT benchmarks feature relatively verbose, fully-formed natural language queries. The efficacy of our LLM-feedback approach with shorter, keyword-based queries, or queries from different domains, remains to be explored. LLMs might behave differently in assessing relevance or generating useful expansion material for such query types.

Relevance Scale Mapping: We mapped the original 5-point (0-4) relevance scales of the benchmarks to the 4-point (0-3) scale used by our judges by collapsing the top two original categories. This introduces an assumption about the equivalence of these collapsed levels. While a pragmatic choice, it could subtly influence perceived judge agreement and the precise interpretation of relevance thresholds (θ).

Relevance Threshold (θ) Selection: Our pruning step relies on a predefined relevance threshold θ. In our experiments, this was guided by benchmark definitions of relevance. In practical scenarios without such predefined ground truth or when adapting to new domains/users, dynamically determining or learning an optimal θ would be a necessary and non-trivial challenge.

iKAT PTKB Context: As discussed, our experiments on iKAT did not explicitly incorporate the Personal Text Knowledge Bases (PTKBs) that sometimes provide crucial context for the "resolved utterances." This unmodeled context likely influenced our LLM judge assessments, potentially underestimating the full potential of our approach on this dataset had PTKB information been available to the feedback mechanism.

7 Ethical Considerations

Similarly, we acknowledge the following ethical considerations:

Environmental Impact: The training and inference of large LLMs, as used by some of our tested judges and for expansion query generation, are computationally intensive and consume significant energy resources (Husom et al., 2024). While we used existing models, widespread adoption of such techniques contributes to these broader concerns. We are hopeful that ongoing research into more efficient model architectures and inference methods will alleviate this over time.

Bias Amplification and Fairness: LLMs are known to inherit and potentially amplify biases present in their training data (Li et al., 2025; Navigli et al., 2023; Gallegos et al., 2024). If LLM judges exhibit biases (e.g., demographic, viewpoint), their in-pipeline feedback could systematically skew search results, leading to unfair or unrepresentative information being surfaced to users. This could reinforce societal biases or limit exposure to diverse perspectives. Careful auditing of LLM judges for such biases and developing debiasing techniques for feedback mechanisms are critical before deployment.

Content Moderation and Harmful Content: An LLM judge providing feedback might inadvertently (or if an LLM itself is not well-moderated) assign high relevance to or promote the expansion of problematic content (e.g., misinformation, hate speech) (Vinay et al., 2025; Williams et al., 2024; Zhou et al., 2023; Chen and Shu, 2024; Guo et al., 2025). While the final reranker might also play

a role, the feedback mechanism itself needs safeguards if it's to actively shape the candidate pool.

Transparency and Explainability: The decision-making process of LLMs, especially for relevance assessment, can be opaque (Arabzadeh and Clarke, 2025; Liao and Vaughan, 2023; Singh, 2025; Dietz et al., 2025). If an LLM judge's feedback significantly alters search results, the lack of transparency in why certain documents were favoured or pruned could be problematic for users seeking to understand search behaviour or for system developers trying to debug it.

References

Nasreen Abdul-Jaleel, James Allan, W Bruce Croft, Fernando Diaz, Leah Larkey, Xiaoyan Li, Mark D Smucker, and Courtney Wade. 2004. Umass at trec 2004: Novelty and hard. *Computer Science Department Faculty Publication Series*, page 189.

Amin Abolghasemi, Suzan Verberne, and Leif Azzopardi. 2022. Improving bert-based query-by-document retrieval with multi-task optimization. In *European Conference on Information Retrieval*, pages 3–12. Springer.

Haya Al-Thani, Tamer Elsayed, and Bernard J Jansen. 2023. Improving conversational search with query reformulation using selective contextual history. *Data and Information Management*, 7(2):100025.

Mohammad Aliannejadi, Zahra Abbasiantaeb, Shubham Chatterjee, Jeffrey Dalton, and Leif Azzopardi. 2024. Trec ikat 2023: A test collection for evaluating conversational and interactive knowledge assistants. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 819–829.

Tiago Almeida and Sérgio Matos. 2024. Exploring efficient zero-shot synthetic dataset generation for information retrieval. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 1214–1231.

Negar Arabzadeh and Charles LA Clarke. 2025. Benchmarking llm-based relevance judgment methods. *arXiv preprint arXiv:2504.12558*.

Jakob Bernoulli. 1713. *Ars coniectandi*. Impensis Thurnisiorum, fratrum.

Castorini. 2023. Pyserini/docs/experiments-msmarco-doc.md at master · castorini/pyserini.

Canyu Chen and Kai Shu. 2024. Combating misinformation in the age of llms: Opportunities and challenges. *AI Magazine*, 45(3):354–368.

Charles LA Clarke and Laura Dietz. 2024. Llm-based relevance assessment still can't replace human relevance assessment. *arXiv preprint arXiv:2412.17156*.

Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Jimmy Lin. 2022. Overview of the trec 2021 deep learning track. In *Text REtrieval Conference (TREC)*. NIST, TREC.

Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Hossein A. Rahmani, Daniel Campos, Jimmy Lin, Ellen M. Voorhees, and Ian Soboroff. 2024. Overview of the trec 2023 deep learning track. In *Text REtrieval Conference (TREC)*. NIST, TREC.

Laura Dietz, Oleg Zendel, Peter Bailey, Charles Clarke, Ellese Cotterill, Jeff Dalton, Faegheh Hasibi, Mark Sanderson, and Nick Craswell. 2025. Llm-evaluation tropes: Perspectives on the validity of llm-evaluations. *arXiv preprint arXiv:2504.19076*.

Guglielmo Faggioli, Laura Dietz, Charles LA Clarke, Gianluca Demartini, Matthias Hagen, Claudia Hauff, Noriko Kando, Evangelos Kanoulas, Martin Potthast, Benno Stein, and 1 others. 2023. Perspectives on large language models for relevance judgment. In *Proceedings of the 2023 ACM SIGIR International Conference on Theory of Information Retrieval*, pages 39–50.

Naghmeh Farzi and Laura Dietz. 2024. Best in tau@ llmjudge: Criteria-based relevance evaluation with llama3. *arXiv preprint arXiv:2410.14044*.

Benoît Frénay and Michel Verleysen. 2013. Classification in the presence of label noise: a survey. *IEEE transactions on neural networks and learning systems*, 25(5):845–869.

Isabel O Gallegos, Ryan A Rossi, Joe Barrow, Md Mehrab Tanjim, Sungchul Kim, Franck Dernoncourt, Tong Yu, Ruiyi Zhang, and Nesreen K Ahmed. 2024. Bias and fairness in large language models: A survey. *Computational Linguistics*, 50(3):1097–1179.

Ruohao Guo, Wei Xu, and Alan Ritter. 2025. How to protect yourself from 5g radiation? investigating llm responses to implicit misinformation. *arXiv preprint arXiv:2503.09598*.

Vishal Gupta and Ashutosh Dixit. 2023. Recent query reformulation approaches for information retrieval system-a survey. *Recent Advances in Computer Science and Communications (Formerly: Recent Patents on Computer Science)*, 16(1):94–107.

Erik Johannes Husom, Arda Goknil, Lwin Khin Shar, and Sagar Sen. 2024. The price of prompting: Profiling energy use in large language models inference. *arXiv preprint arXiv:2407.16893*.

Armin Hust, Stefan Klink, Markus Junker, and Andreas Dengel. 2002. Query reformulation in collaborative information retrieval. In *Proceedings of the International Conference on Information and Knowledge Sharing, IKS*, volume 2002.

Miaomiao Li, Hao Chen, Yang Wang, Tingyuan Zhu, Weijia Zhang, Kaijie Zhu, Kam-Fai Wong, and Jindong Wang. 2025. Understanding and mitigating the bias inheritance in llm-based data augmentation on downstream tasks. *arXiv preprint arXiv:2502.04419*.

Q Vera Liao and Jennifer Wortman Vaughan. 2023. Ai transparency in the age of llms: A human-centered research roadmap. *arXiv preprint arXiv:2306.01941*, 10.

Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira. 2021. Pyserini: A python toolkit for reproducible information retrieval research with sparse and dense representations. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2356–2362.

Sean MacAvaney and Luca Soldaini. 2023. One-shot labeling for automatic relevance estimation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2230–2235.

Iain Mackie, Shubham Chatterjee, and Jeffrey Dalton. 2023. Generative relevance feedback with large language models. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2026–2031.

Roberto Navigli, Simone Conia, and Björn Ross. 2023. Biases in large language models: Origins, inventory, and discussion. *J. Data and Information Quality*, 15(2).

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. Ms marco: A human-generated machine reading comprehension dataset.

Rodrigo Nogueira, Zhiying Jiang, and Jimmy Lin. 2020. Document ranking with a pretrained sequence-to-sequence model. *arXiv preprint arXiv:2003.06713*.

Arnold Overwijk, Chenyan Xiong, Xiao Liu, Cameron VandenBerg, and Jamie Callan. 2022. Clueweb22: 10 billion web documents with visual and semantic information. *arXiv preprint arXiv:2211.15848*.

Paul Owoicho, Jeff Dalton, Mohammad Aliannejadi, Leif Azzopardi, Johanne R Trippas, and Svitlana Vakulenko. 2022. Trec cast 2022: Going beyond user ask and system retrieve with initiative and response generation. In *TREC*.

Paul Owoicho, Ivan Sekulic, Mohammad Aliannejadi, Jeffrey Dalton, and Fabio Crestani. 2023. Exploiting simulated user feedback for conversational search: Ranking, rewriting, and beyond. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 632–642.

Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, and 1 others. 2020. Kilt: a benchmark for knowledge intensive language tasks. *arXiv preprint arXiv:2009.02252*.

Peter Pirolli and Stuart Card. 1999. Information foraging. *Psychological review*, 106(4):643.

Peter L. T. Pirolli. 2007. *Information Foraging Theory: Adaptive Interaction with Information*. Oxford University Press.

Hossein A Rahmani, Clemencia Siro, Mohammad Aliannejadi, Nick Craswell, Charles LA Clarke, Guglielmo Faggioli, Bhaskar Mitra, Paul Thomas, and Emine Yilmaz. 2025. Judging the judges: A collection of llm-generated relevance judgements. *arXiv preprint arXiv:2502.13908*.

Stephen E Robertson, Steve Walker, Susan Jones, Micheline M Hancock-Beaulieu, Mike Gatford, and 1 others. 1995. Okapi at trec-3. *Nist Special Publication Sp*, 109:109.

Joseph John Rocchio. 1971. Relevance feedback in information retrieval. *The SMART retrieval system: experiments in automatic document processing*, pages 313–323.

Guilherme Rosa, Luiz Bonifacio, Vitor Jeronymo, Hugo Abonizio, Marzieh Fadaee, Roberto Lotufo, and Rodrigo Nogueira. 2022a. In defense of cross-encoders for zero-shot retrieval. *arXiv preprint arXiv:2212.06121*.

Guilherme Moraes Rosa, Luiz Bonifacio, Vitor Jeronymo, Hugo Abonizio, Marzieh Fadaee, Roberto Lotufo, and Rodrigo Nogueira. 2022b. No parameter left behind: How distillation and model size affect zero-shot retrieval. *arXiv preprint arXiv:2206.02873*.

Sourav Saha, Dwaipayan Roy, B Yuvaraj Goud, Chethan S Reddy, and Tanmay Basu. 2022. Nlp-iiserb@ simpletext2022: To explore the performance of bm25 and transformer based frameworks for automatic simplification of scientific texts. In *CLEF (Working Notes)*, pages 2852–2857.

Alexandre Salle, Shervin Malmasi, Oleg Rokhlenko, and Eugene Agichtein. 2022. Cosearcher: studying the effectiveness of conversational search refinement and clarification through user simulation. *Information Retrieval Journal*, 25(2):209–238.

Ivan Sekulić, Mohammad Aliannejadi, and Fabio Crestani. 2022. Evaluating mixed-initiative conversational search systems via user simulation. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, pages 888–896.

Ajit Singh. 2025. Evaluating the transparency and explainability of llm-based educational systems. *Available at SSRN 5198565*.

Ian Soboroff. 2025. Don't use llms to make relevance judgments. *Information retrieval research journal*, 1(1):10–54195.

Rikiya Takehi, Ellen M Voorhees, Tetsuya Sakai, and Ian Soboroff. 2024. Llm-assisted relevance assessments: When should we ask llms for help? *arXiv preprint arXiv:2411.06877*.

Paul Thomas, Seth Spielman, Nick Craswell, and Bhaskar Mitra. 2024. Large language models can accurately predict searcher preferences. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1930–1940.

Shivani Upadhyay, Ehsan Kamalloo, and Jimmy Lin. 2024a. Llms can patch up missing relevance judgments in evaluation. *arXiv preprint arXiv:2405.04727*.

Shivani Upadhyay, Ronak Pradeep, Nandan Thakur, Daniel Campos, Nick Craswell, Ian Soboroff, Hoa Trang Dang, and Jimmy Lin. 2024b. A large-scale study of relevance assessments with large language models: An initial look. *arXiv preprint arXiv:2411.08275*.

Shivani Upadhyay, Ronak Pradeep, Nandan Thakur, Nick Craswell, and Jimmy Lin. 2024c. Umbrela: Umbrela is the (open-source reproduction of the) bing relevance assessor. *arXiv preprint arXiv:2406.06519*.

Rasita Vinay, Giovanni Spitale, Nikola Biller-Andorno, and Federico Germani. 2025. Emotional prompting amplifies disinformation generation in ai large language models. *Frontiers in Artificial Intelligence*, 8:1543603.

Linkai Weng, Zhiwei Li, Rui Cai, Yaoxue Zhang, Yuezhi Zhou, Laurence T Yang, and Lei Zhang. 2011. Query by document via a decomposition-based two-level retrieval approach. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 505–514.

Angus R Williams, Liam Burke-Moore, Ryan Sze-Yin Chan, Florence E Enock, Federico Nanni, Tvesha Sippy, Yi-Ling Chung, Evelina Gabasova, Kobi Hackenburg, and Jonathan Bright. 2024. Large language models can consistently generate high-quality content for election disinformation operations. *arXiv preprint arXiv:2408.06731*.

Yin Yang, Nilesh Bansal, Wisam Dakka, Panagiotis Ipeirotis, Nick Koudas, and Dimitris Papadias. 2009. Query by document. In *Proceedings of the Second ACM International Conference on Web Search and Data Mining*, pages 34–43.

Jiawei Zhou, Yixuan Zhang, Qianni Luo, Andrea G Parker, and Munmun De Choudhury. 2023. Synthetic lies: Understanding ai-generated misinformation and evaluating algorithmic and human solutions. In *Proceedings of the 2023 CHI conference on human factors in computing systems*, pages 1–20.

A Ablation Study on Number of Seeds for Query-Reformulation Policy Expansion

Section 3 describes our Query-Reformulation feedback policy, where top-N documents from the pruned pool ($\mathcal{D}_p$) are used as seeds to generate new queries for expanding the candidate set. The number of seed documents (N) chosen for this process is a key parameter. To determine an effective value for N, we conducted preliminary experiments on the iKAT benchmark using the Perfect Judge with a relevance threshold of $\theta = 1$ for pruning. We varied N from 1 to 6 seed documents and tested each of our three expansion query generation methods: "Full Text," "Reformulated Query," and "Summary." Table 4 presents these results against the BM25+MONOT5 baseline.

The results in Table 4 reveal that, for the Query-Reformulation policy, top-rank precision metrics such as nDCG@3 generally peak or are strongest when using a very small number of seed documents, typically N=1. For instance, with the "Full Text" expansion method, nDCG@3 is 0.355 with 1 seed document. While there are minor fluctuations, increasing the number of seed documents beyond 1 does not consistently lead to further improvements in nDCG@3 and sometimes results in a slight degradation (e.g., nDCG@3 for "Full Text" with 6 seeds is 0.349). Similar trends are observable for the "Summary" and "Reformulated Query" methods, where N=1 often provides the best or near-best nDCG@3 performance.

Conversely, Recall@1000 tends to exhibit a slight upward trend or plateaus as more seed documents are incorporated, although the gains beyond N=1 are often marginal. For example, with the "Full Text" method, R@1000 increases from 0.552 (1 seed) to 0.583 (6 seeds). This suggests that while using more seeds can bring in a slightly larger pool of relevant documents, the additional documents may not be of sufficiently high quality or distinctiveness to improve top-rank precision, potentially due to increased overlap or the introduction of mildly relevant but not top-tier documents.

Considering the emphasis on top-rank performance in our target benchmarks and the observation that precision benefits diminish or saturate quickly beyond a single seed document, we opted to use N=1 seed document for the Query-Reformulation policy experiments reported in the

Method	nDCG@3	nDCG@5	nDCG@1000	P@20	R@20	R@1000	AP@1000
Baseline	0.288	0.287	0.333	0.248	0.141	0.451	0.128
1 doc							
Full Text	0.355	0.349	0.412	0.311	0.185	0.552	0.179
Reformulated Query	0.338	0.331	0.381	0.289	0.162	0.508	0.160
Summary	0.352	0.343	0.406	0.305	0.174	0.549	0.174
2 docs							
Full Text	0.347	0.338	0.412	0.284	0.166	0.571	0.166
Reformulated Query	0.333	0.330	0.386	0.287	0.168	0.518	0.163
Summary	0.343	0.334	0.405	0.287	0.165	0.565	0.165
3 docs							
Full Text	0.340	0.326	0.410	0.282	0.170	0.579	0.163
Reformulated Query	0.331	0.326	0.383	0.277	0.165	0.514	0.160
Summary	0.341	0.334	0.408	0.287	0.166	0.570	0.166
4 docs							
Full Text	0.340	0.327	0.409	0.278	0.167	0.579	0.160
Reformulated Query	0.334	0.331	0.385	0.283	0.167	0.517	0.162
Summary	0.342	0.332	0.410	0.284	0.164	0.577	0.166
5 docs							
Full Text	0.349	0.333	0.414	0.277	0.168	0.582	0.165
Reformulated Query	0.330	0.325	0.381	0.279	0.165	0.514	0.160
Summary	0.340	0.332	0.407	0.280	0.163	0.573	0.164
6 docs							
Full Text	0.349	0.338	0.415	0.274	0.167	0.583	0.165
Reformulated Query	0.340	0.337	0.386	0.286	0.170	0.511	0.163
Summary	0.337	0.331	0.408	0.280	0.160	0.576	0.164

Table 4: Comparison of retrieval metrics using Query-Reformulation policy across varying numbers of seed documents and methods.

main paper. This choice prioritises optimising for metrics like nDCG@3 while still leveraging the strongest initial signal for exploitation.

B Ablation Study on Number of Seeds for Query-By-Document Policy Expansion

In Section 3, we describe our Query-By-Document feedback policy, where documents from the pruned pool ($\mathcal{D}_p$) seed the retrieval of neighboring documents to enrich the candidate set. A key parameter in this process is the number of neighbors retrieved per seed document. To determine an optimal setting that balances top-rank precision with sufficient pool enrichment, we conducted preliminary experiments on the iKAT benchmark using the Perfect Judge with a relevance threshold of $\theta = 2$ for pruning. We varied the number of neighbors retrieved per seed from 1 to 30. Table 5 presents the results of this ablation study, comparing various retrieval metrics against the BM25+MONOT5 baseline.

The results clearly illustrate a trade-off between top-rank precision and overall recall as the number of retrieved neighbors increases. As seen in Table 5, precision-oriented metrics such as nDCG@3 and P@20 peak when retrieving only 1 neighbour per seed document (nDCG@3 = 0.621). As the number of neighbors increases beyond one, there is a consistent degradation in these top-rank precision metrics. For instance, nDCG@3 drops from 0.621 (1 neighbour) to 0.542 (2 neighbours), and further to 0.393 when retrieving 20 neighbours. This suggests that while retrieving more neighbours might bring in more documents, many of these additional documents are either not as relevant or introduce noise that makes it harder for the final reranker to identify the very best documents for the top positions.

Conversely, Recall@1000 exhibits the opposite trend. It starts relatively low when retrieving only 1 neighbour (R@1000 = 0.205, which is below the baseline's 0.451 in this specific ablation setup with $\theta = 2$) and gradually increases with the number of neighbours retrieved, reaching 0.319 with 30 neighbours. This indicates that retrieving more neighbours per seed does indeed bring more unique

Method	nDCG@3	nDCG@5	nDCG@1000	P@20	R@20	R@1000	AP@1000
Baseline	0.288	0.287	0.333	0.248	0.141	0.451	0.128
1 neighbour	0.621	0.600	0.376	0.412	0.174	0.205	0.205
2 neighbours	0.542	0.524	0.360	0.385	0.172	0.222	0.185
3 neighbours	0.506	0.496	0.356	0.366	0.168	0.235	0.179
4 neighbours	0.485	0.472	0.352	0.347	0.163	0.242	0.173
5 neighbours	0.474	0.462	0.350	0.339	0.161	0.248	0.170
10 neighbours	0.431	0.417	0.345	0.311	0.155	0.274	0.162
15 neighbours	0.407	0.398	0.343	0.299	0.152	0.289	0.159
20 neighbours	0.393	0.382	0.342	0.293	0.151	0.300	0.156
30 neighbours	0.381	0.360	0.342	0.281	0.147	0.319	0.153

Table 5: Comparison of retrieval metrics using the Query-By-Document Policy at varying number of neighbours.

relevant documents into the 1000-document pool overall.

Given that the conversational search benchmarks used in our main experiments (iKAT and CAsT) place a strong emphasis on top-rank performance (e.g., high nDCG@3 is critical for success), and our primary goal is to demonstrate significant improvements in this area, we selected the configuration that maximised these precision metrics. Therefore, based on this ablation, we opted to retrieve 1 neighbour per seed document for all Query-By-Document policy experiments reported in the main paper. While this experiment was conducted with a pruning threshold of $\theta = 2$ on iKAT, we observed similar trends regarding the precision-recall trade-off with varying numbers of neighbours when using a threshold of $\theta = 1$ (the threshold used for iKAT in our main experiments, as detailed in Section 3). This consistent behaviour reinforced our decision to restrict expansion to a single, closest neighbour to optimise for top-rank precision.

C Prompts

C.1 Query-Reformulation Summary Prompt

Query-Reformulation Summary Prompt

You are a Passage Summarizer whose job is to read a set of passages and produce a concise, accurate answer to the user's question using only the information provided.
Your output must satisfy these requirements:

1. **Completeness**: Include all key facts from the passages that directly answer the question.
2. **Fidelity**: Do not add any information or assumptions not present in the passages.
3. **Clarity**: Write in clear, direct language, referencing the same names and terms used in the passages.
4. **Brevity**: Keep the answer as short as possible while fully answering the question.

Question:
`<insert user query here>`
Passages:

- `<passage 1>`
- `<passage 2>`

Answer:

C.2 Query-Reformulation Query Reformulation Prompt

Query-Reformulation Query Reformulation Prompt

You are a Question-Rewriter whose job is to take an arbitrary user question and turn it into a new question that can be answered using the information contained in a given set of passages.
Your output must satisfy these requirements:

1. **Answerability**: The rewritten question must be fully answerable by the facts, names, dates, and relationships explicitly stated in the passages.
2. **Fidelity**: Do not introduce any new facts or assumptions that are not present in the passages.
3. **Clarity**: Make the question as clear and specific as possible, referencing the same concepts used in the passages.
4. **Conciseness**: Keep the question brief; only include what is needed to ensure answerability.

Passages:

- `<passage 1>`
- `<passage 2>`

Original Question:
`<insert original user query here>`
Rewritten Question:

C.3 Query-By-Document Summary Prompt

Query-By-Document Summary Prompt

Generate a concise summary of the Passage so that it completely answers the Question.
Question:
`<insert question here>`
Passage:
`<insert passage here>`
Summary: