Failing Forward: Understanding Query Failure in Retrieval, Generation, and Judgment

Abstract

Modern information retrieval is increasingly implemented as a pipeline in which a retriever selects documents, an LLM synthesizes an answer grounded in the retrieved documents, and an LLM-based judge scores relevance or quality. In this setting, a poor outcome is hard to attribute because failure may originate in retrieval, generation, or judgment, and these failures are typically analyzed in isolation. This paper studies query failure across these three tasks through a unified operationalization of hard queries per task. Using four years of TREC Deep Learning benchmarks from 2019 to 2022, we define *hard-to-retrieve*, *hard-to-generate*, and *hard-to-judge* query sets, and analyze their overlap, their robustness across retrievers, generators, and judging setups, and the query characteristics associated with each failure. We find that hard queries overlap only weakly across tasks, indicating that difficulty does not transfer reliably between retrieval, generation, and judgment. At the same time, while the specific hard queries vary by model, the overlap structure is stable across system choices, suggesting that difficulty is driven more by query characteristics interacting with task-specific constraints than by model-specific effects. We further induce a data-driven typology of roots of failure (*difficulty cues*) and show that they can be used to improve system behavior. We further demonstrate that conditioning generation on task-relevant difficulty cues yields consistent gains in answer quality.

1 Introduction

Modern information retrieval systems no longer operate as single-stage retrieval methods. Instead, they are increasingly structured as multi-stage pipelines in which retrieved documents condition large language model (LLM) generation, and generated content is itself evaluated by automated judges, often LLMs trained to approximate human relevance assessments [22, 46]. While this evolution has expanded system capabilities, it has also introduced new and qualitatively distinct modes of failure. A query may retrieve seemingly relevant documents yet yield an incorrect or misleading generated answer. Conversely, a generated response may appear fluent while relying on weak or misaligned evidence. In addition, automatic judges may diverge from human assessments even when retrieval and generation appear successful. In such pipelines, failure is no longer localized to a single component but emerges across stages, with consequences that propagate to downstream tasks.

Despite decades of research on query difficulty prediction in information retrieval, the field lacks a unified understanding of how and why queries fail across these modern heterogeneous stages of the pipeline. Traditional notions of failure have largely been task-bound. Retrieval research has focused on queries that yield poor rankings and low effectiveness [6, 23]. More recent work on large language models has examined hallucination, factual inconsistency, and prompt sensitivity in generation [25, 47]. In parallel, a growing body of work has investigated the reliability of LLMs as automatic judges of relevance and quality [9, 19]. These lines of work have largely progressed independently, implicitly assuming that failure in each task can be understood and mitigated in isolation. What remains unclear is whether difficulty is a transferable property of the query itself, whether failures in one stage predict failures in others, or whether different tasks fail for fundamentally different reasons even when exposed to the same query.

This gap has become increasingly consequential as IR pipelines incorporate multiple forms of prediction and evaluation. Query Performance Prediction (QPP) methods estimate whether a query is likely to retrieve relevant documents [5, 32]. More recently, Prompt Performance Prediction has emerged to anticipate low-quality generations by LLMs [1, 38]. At the same time, LLM-based judges are now routinely used to replace or augment human relevance assessments in offline evaluation pipelines [30, 41]. However, these approaches address failure at individual points in the pipeline without examining whether the same queries systematically challenge retrieval, generation, and judgment, or whether improvements in one component generalize to others. As a result, current systems risk optimizing isolated components while remaining fragile at the pipeline level. In this work, we argue that query failure in modern information retrieval systems must be understood as a task-conditioned phenomenon rather than a single unified notion of difficulty. Retrieval, generation, and judgment impose distinct demands on queries, and failure in each task reflects different underlying properties of the same input. To operationalize this perspective, we introduce three complementary notions of query failure. **Hard-to-Retrieve** queries are those that consistently yield poor retrieval effectiveness. **Hard-to-Generate** queries are those for which LLMs produce low-quality or misaligned answers. **Hard-to-Judge** queries are those that induce systematic disagreement between LLM-based and human relevance judgments. Rather than treating these categories as heuristics, we use them as analytical instruments to probe how difficulty manifests across tasks.

Using four years of TREC Deep Learning benchmarks from 2019 to 2022 [10, 13, 15], we conduct a large-scale empirical study guided by 3 main *Research Questions (RQs)*. Specifically, we ask:

- **RQ1.** To what extent do hard-to-retrieve, hard-to-generate, and hard-to-judge queries overlap, and does difficulty in one task predict difficulty in others
- **RQ2.** Do observed failure patterns depend on system configuration, including the choice of retriever, generator, or judging setup, or do they remain stable across models
- **RQ3.** What linguistic and semantic properties give rise to failure in retrieval, generation, and judgment, and are these causes shared across tasks or task-specific

Our findings reveal a clear and previously undocumented pattern. Query failures show limited overlap across tasks, indicating that difficulty in retrieval, generation, and judgment is largely task-specific. At the same time, these patterns are stable across systems. Although different models fail on different individual queries, the overall structure of failure overlap remains consistent across retrievers, generators, and judges. This suggests that many failures

Table 1: Top failure reasons for different tasks.

Category	Failure Reasons
Hard to Judge (H2J)	Quantitative data needed , Specific and niche topic , Query unrelated to provided criteria , No relevant passage context
Hard to Generate (H2G)	More recent information needed , Accurate technical knowledge required , Risk of misinformation
Hard to Retrieve (H2R)	Numerical data required , Specific product term , Ambiguous subject reference

stem not from model weakness or architectural choice, but from intrinsic query properties that interact differently with each task's inductive biases. In short, modern pipelines fail not because models are uniformly inadequate, but because each stage demands different information from the same query.

Beyond diagnosis, we show that this analysis has practical consequences. By identifying task-specific causes of failure, we incorporate these signals directly into generation prompts. This difficulty-aware prompting conditions LLMs on anticipated failure modes and yields consistent improvements across all four TREC datasets. Though lightweight, this intervention highlights a broader principle: making failure explicit enables systems to adapt, closing the loop between analysis and performance. We believe this work offers a pipeline-level perspective on query failure in modern information retrieval systems. To our knowledge, it provides the first systematic analysis of how retrieval, generation, and judgment fail on different subsets of queries, introduces an empirically grounded typology of task-specific difficulty causes, and demonstrates that understanding these causes leads to concrete performance gains.

All of our code, data, and evaluation tools used in this study are made available to facilitate reproducibility and future work at https://anonymous.4open.science/r/h2r-h2j-h2g-0268/.

2 Empirical Setup

2.1 Datasets

We use four query sets from the TREC Deep Learning Tracks (DL 2019–2022). DL 2019 and 2020 are based on the MS MARCO v1 corpus [11, 12, 14, 16], while DL 2021 and 2022 use the larger MS MARCO v2. Each dataset includes queries with graded relevance judgments from NIST assessors on a four-point scale from Perfectly relevant to Non-relevant [11–13, 16].

2.2 Hard-to-Retrieve Queries (H2R)

Difficult queries have long been studied in retrieval research, most notably through the lens of Query Performance Prediction (QPP). [4, 6, 17, 23]. Difficult queries are typically queries for which standard retrieval models yield low effectiveness, as measured by ranking metrics. Here, we focus on NDCG@10, the official evaluation metric in TREC DL. To capture different retrieval strategies, we use (1) BM25, a lexical sparse retriever implemented via Anserini [44], and (2) DistilBERT-Base-TAS-B [24], a dense retriever fine-tuned on MS MARCO. For each query, we compute NDCG@10 and identify the bottom quartile (Q1) of queries under each retriever as *Hard-to-Retrieve* (H2R). Across the four datasets, average NDCG@10 for H2R queries is 0.186 for BM25 and 0.297 for DistilBERT-TAS-B, indicating retrieval difficulty for both retrievers.

2.3 Hard-to-Generate Queries (H2G)

To isolate queries that challenge the language model's ability to generate accurate answers, we adopt a zero-shot generation setup. For each query, we generate a single response using a minimal, instruction-style prompt with no additional prompt engineering, following prior work [2, 28]. The prompt and setup are publicly available in our GitHub repository. We use two instruction-tuned models for response generation: `LLaMA3.2:3b` and `Qwen3:8b`. To evaluate generation quality, we use BERTScore [45], which compares generated responses against reference texts using contextual embeddings. BERTScore is particularly suited for this setting because it captures both lexical and semantic similarity, accommodating paraphrased or variational surface forms while preserving meaning. For each query, we compute the F1 variant of BERTScore by comparing the generated output to all passages labeled as Perfectly (3) or Highly relevant (2). Queries are then ranked by their average BERTScore, and the bottom quartile (Q1) are designated as *Hard-to-Generate* (H2G). These queries are those for which the LLMs consistently produce semantically weak or misaligned outputs.

2.4 Hard-to-Judge Queries (H2J)

Recent work shows that LLMs can approximate human relevance judgments with reasonable accuracy [19, 34, 35, 42], making them attractive surrogates for manual annotation. However, their reliability is uneven and may fail under certain query characteristics or judgment conditions [9, 18]. To identify such cases, we analyze disagreement between LLMs and human annotators under two setups: graded relevance based on the UMBRELA reproduction of Bing assessor labels [41, 42], and binary classification [19]. In the graded setting, `Qwen3:8b` assigns relevance scores on a 0–3 scale (with `LLaMa3.2:3b` results available on GitHub). In the binary setting, we follow prior work [11, 16] by mapping levels 0–1 to 'not relevant' (0) and 2–3 to 'relevant' (1). Disagreement is measured differently across the two: for binary, we count cases where the LLM's binary label diverges from the human label across all judged documents, Queries in the top quartile (Q4) with the highest number of such binary misjudgments are labelled as *Hard-to-Judge* (H2J). For graded relevance, we define a misjudgment as a case where the absolute difference between LLM and human relevance score exceeds one point. Queries with the highest frequency of such misjudgments, in the top quartile (Q4), are labelled H2J under the graded criterion. This lets us distinguish between coarse- and fine-grained disagreement and identify queries that challenge automated judges.

3 Findings

3.1 Inter-task Dependency

To address our first research question (RQ1), we evaluate the extent to which failure-prone queries overlap in the three tasks. We ask whether queries that are difficult for retrieval systems also lead to poor generation performance or disagreement with human annotation in relevance judgments. The degree of such alignment is critical for determining whether difficulty can be conceptualized as a general property of the query or must be treated as task-specific.

We compute the pairwise and three-way intersections of the Hard-to-Retrieve (H2R), Hard-to-Generate (H2G), and Hard-to-Judge (H2J) query sets across the TREC DL datasets. The overlap statistics are aggregated across datasets and visualized in Figure 1. The results

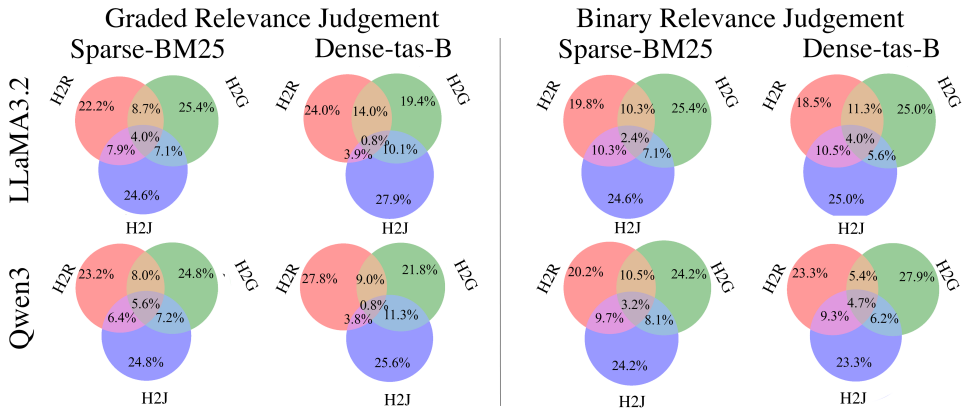

Figure 1: Inter-task dependency of hard queries across different retrievers, generators and judgments.

Algorithm 1 Iterative Reason Extraction for Query Failure

Require: Set of failed queries Q for task $T \in$ {Retrieval, Generation, Judgment}
Ensure: Final list of failure reasons $\mathcal{R}_T$

```
Initialize reason list R_T ← []
for all query q_i ∈ Q do
    Provide q_i and its associated output (answer, retrieved docs, or judgments) to the LLM
    Prompt LLM to generate a candidate reason r_i for failure
    Append r_i to temporary reason list R_T^(i)
    Merge R_T^(i) with R_T and reduce redundancy
    Update R_T ← Merged(R_T, R_T^(i))
end for
Return R_T
```

indicate limited overlap between the three sets. Fewer than 10% of queries are classified as difficult across all tasks. Approximately half of the queries in each category are exclusive to that task, while the remainder overlap with one of the other two. Notably, across all datasets the pairwise overlap between H2R and H2J is consistently the highest among the three task pairs, likely because the same retrieved documents are used for both evaluation and judgment. Although modest in absolute terms, this consistent overlap highlights a stronger connection between retrieval and judgment than between other task pairs. This low degree of inter-task dependency suggests that retrieval, generation, and judgment engage fundamentally distinct LLM capabilities. Retrieval systems rely primarily on lexical or dense similarity matching, and are sensitive to document distributional properties [21, 26, 29]. In contrast, generation tasks are governed by the model's ability to conditionally synthesize accurate responses, which is shaped by its internal knowledge priors, prompt interpretation, and decoding dynamics [27]. Judgment tasks further depend on the LLM's calibration, evaluative alignment, and implicit relevance criteria [20, 40]. As such, a query that fails on one task may not impose the same burden on another. These findings suggest the view that retrieval, generation, and judgment are *essentially* distinct tasks, each characterized by its own inductive biases and operational constraints.

3.2 System Dependency

For RQ2, we examine whether failure patterns depend on system configuration i.e., whether hard queries in retrieval, generation, and judgment vary with the specific models employed. As such, we vary each pipeline component: for retrieval, BM25 [39] vs. DistilBERT-TAS-B [24]; for generation, LLaMA3.2:3b vs. Qwen3:8b; and for judgment, binary vs. graded LLM-based relevance labels. We then compute the intersections between failure sets (H2R, H2G, H2J) across all combinations.

Figure 1 shows that while the identities of hard queries differ across systems, reflecting architectural and training differences, the overlap structure between failure types remains stable. For instance, the proportion of queries overlapping between H2R and H2G is comparable under both BM25 and DistilBERT-TAS-B. Likewise, switching between Qwen and LLaMA has little effect on generation–judgment overlap, and even by replacing binary with graded judgments, the relative pairwise overlaps exhibit only marginal deviation. These findings suggest that system improvements, while potentially beneficial in absolute performance, do not fundamentally disrupt the underlying pattern of failure inter-dependencies. This points to a deeper, architecture-agnostic source of difficulty embedded in the nature of the queries themselves. Put differently, although retrievers, generators, and judges may shift the boundary of which queries are handled successfully, they do not reconfigure the *fundamental separability* of retrieval, generation, and judgment challenges. This result carries significant methodological implications. First, it validates the stability of our failure annotations across system variants, lending credibility to cross-system comparative analyses. Second, it implies that mitigating failures through model substitution alone is unlikely to yield systematic resolution.

3.3 Failure Reasons

For (RQ3), we analyze the underlying properties that contribute to query failure across the three tasks. This section aims to move from descriptive categorization to explanatory interpretation to understand not merely which queries fail, but why they do so. Our approach draws methodological inspiration from *nugget-based evaluation frameworks* [31, 33, 37]. Inspired by AutoNuggetizer pipeline introduced in recent TREC RAG 2024 evaluation [35, 36], we construct a dynamic, data-driven reason discovery process that instead of relying on predefined taxonomies, iteratively refines explanatory

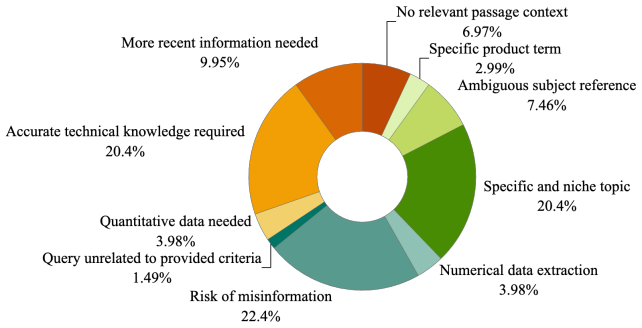

Figure 2: Hard queries intersection reasons distribution

categories for failure. For each query labeled as hard in any of the three tasks, we prompt LLM with the query and its associated context (retrieved passages, generated output, or relevance labels). The model is instructed to generate a concise, high-level explanation for why the system may have failed. These candidate reasons are then aggregated and passed through multiple refinement iterations. In each iteration, the LLM is shown the current list of reasons for the given failure category and is prompted to merge overlapping items, generalize overly narrow cases, and add any novel causes observed in newly sampled queries. The iterative process terminates when no substantive updates are made to the reason set, yielding a converged list of generalizable failure causes. Algorithm 1 outlines the end-to-end workflow for constructing these reasons. Full prompts and examples are available in our GitHub repository. The result of this process is a curated typology of failure causes for each task. Table 1 summarizes the top reasons of failure across the three tasks, as identified through this iterative LLM-guided reasoning process. For instance, in the case of *generation* failures, we observe recurrent issues such as prompts that involve more accurate technical knowledge. In *retrieval*, frequent causes include query ambiguity or queries that rely on numeric data. *Judgment* errors are often caused by the document under evaluation having low topical relevance or addressing a niche subject. Figure 2 shows the overall distribution of reasons across all three tasks where risk of misinformation, specific and niche topic, and accurate technical knowledge required are among the most common failure reasons.

Human Validation. We further validate the LLM-generated reasons with human annotation. Two computer science graduate students each reviewed 50 randomly sampled queries, checking whether the one-hot LLM explanations were correct and tagging any additional applicable causes from a fixed set of ten. Human and LLM judgments show strong alignment, with an accuracy of 0.8120 and a precision rate of 0.7514 across all difficulty reasons categories, which is calculated by averaging the individual accuracy and precision scores across all 10 reasons for each human annotation on one-hot encoding.

In response to RQ3 we find that while some causes recur across tasks, the leading reasons differ, underscoring that retrieval, generation, and judgment face distinct sources of failure.

3.4 Difficulty-Aware Prompting

Having observed that query failures in retrieval, generation, and judgment stem from distinct, task-specific causes, we now examine whether these diagnostic insights can guide system behavior. Our goal is to validate whether the failure typologies identified earlier encode information that is actionable in downstream decision-making. Specifically, we test the hypothesis that conditioning language model responses on an explicit statement of the anticipated reason for query difficulty can lead to higher-quality outputs. To evaluate this, we conduct a targeted experiment comparing two generation conditions (base vs. difficulty-aware). In the baseline condition, the model receives the raw query alone. In the intervention condition, the model is given the query plus a natural-language description of the relevant difficulty factor (e.g., "this query requires numerical data"), selected from the typologies in Table 2. The intervention is limited to the presence of this additional clue. Due to space limitation, we only apply this strategy on the generation task. We assess generation quality using a pairwise comparison framework adjudicated by a state-of-the-art LLM-based judge [7, 46], a method shown to align well with expert opinions [3, 8, 43]. For each query, we identify the passages labeled as highly relevant (2 or 3) and prompt the judge to compare the baseline and difficulty-aware responses in terms of fidelity and informativeness o the expected response. The model is instructed to select the answer that more accurately reflects the reference content, allowing us to quantify generation improvements using minimal conditioning.

Table 2: Preference rate over the relevant answers among base generation vs difficulty-aware generation.

Dataset	Base Generation	Difficulty-aware generation
dl19	35.44%	64.56%
dl20	24.59%	75.40%
dl21	47.23%	52.77%
dl22	39.50%	60.49%

The results in Table 2, indicate that incorporating failure reasons into the generation prompt leads to measurable improvements. Across all four TREC DL datasets, difficulty-aware outputs are preferred by the LLM-based judge in a substantial majority of cases: 64.56% for 2019, 75.40% for 2020, 52.77% for 2021, and 60.49% for 2022. These findings demonstrate that even lightweight conditioning on task-specific failure cues improves response quality. While our current evaluation is limited to generation, this serves as initial evidence that difficulty reasons can be leveraged to make systems more aware of potential pitfalls, a direction we aim to extend to retrieval and judgment in our future work.

4 Concluding Remarks

This paper frames query failure as a pipeline-level reliability problem rather than a single-task deficiency, showing that retrieval, generation, and judgment impose distinct demands on the same query and therefore fail for different reasons. Our findings indicate that difficulty is largely task-conditioned, yet the structure of failure remains consistent across retrievers, generators, and judging setups. This finding explains why improving one component often does not yield end-to-end robustness. We further show that extracting a data-driven typology of failure causes grounds query difficulty in concrete, task-specific properties. Making these causes explicit and incorporating them into difficulty-aware prompting turns failure analysis into a practical mechanism for improving system behavior, pointing toward IR pipelines that detect and adapt to task-specific difficulty rather than relying on a one-size-fits-all notion of failure.

References

[1] Negar Arabzadeh and Ebrahim Bagheri. 2025. VAP3: Variation-Aware Prompt Performance Prediction. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2025, Padua, Italy, July 13-18, 2025*, Nicola Ferro, Maria Maistro, Gabriella Pasi, Omar Alonso, Andrew Trotman, and Suzan Verberne (Eds.). ACM, 2794–2799. doi:10.1145/3726302.3730264

[2] Negar Arabzadeh, Amin Bigdeli, and Charles L. A. Clarke. 2024. Adapting Standard Retrieval Benchmarks to Evaluate Generated Answers. arXiv:2401.04842 [cs.IR]

[3] Negar Arabzadeh and Charles LA Clarke. 2025. Benchmarking LLM-based Relevance Judgment Methods. *arXiv preprint arXiv:2504.12558* (2025).

[4] Negar Arabzadeh, Maryam Khodabakhsh, and Ebrahim Bagheri. 2021. BERT-QPP: Contextualized Pre-trained Transformers for Query Performance Prediction. In *CIKM*. 2857–2861.

[5] Negar Arabzadeh, Chuan Meng, Mohammad Aliannejadi, and Ebrahim Bagheri. 2024. Query Performance Prediction: Techniques and Applications in Modern Information Retrieval. In *Proceedings of the 2024 Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region*. 291–294.

[6] David Carmel and Elad Yom-Tov. 2010. Estimating the Query Difficulty for Information Retrieval. *Synthesis Lectures on Information Concepts, Retrieval, and Services* 2, 1 (2010), 1–89.

[7] Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Banghua Zhu, Hao Zhang, Michael Jordan, Joseph E Gonzalez, et al. 2024. Chatbot arena: An open platform for evaluating llms by human preference. In *Forty-first International Conference on Machine Learning*.

[8] Charles LA Clarke, Fernando Diaz, and Negar Arabzadeh. 2023. Preference-Based Offline Evaluation. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*. 1248–1251.

[9] Charles LA Clarke and Laura Dietz. 2024. LLM-based relevance assessment still can't replace human relevance assessment. *arXiv preprint arXiv:2412.17156* (2024).

[10] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, and Daniel Campos. 2020. Overview of the TREC 2020 Deep Learning Track. In *TREC*.

[11] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, and Daniel Campos. 2021. Overview of the TREC 2020 deep learning track. arXiv:2102.07662 [cs.IR] https://arxiv.org/abs/2102.07662

[12] 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. https://www.microsoft.com/en-us/research/publication/overview-of-the-trec-2021-deep-learning-track/

[13] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, Jimmy Lin, Ellen M. Voorhees, and Ian Soboroff. 2023. Overview of the TREC 2022 deep learning track. In *Text REtrieval Conference (TREC)*. NIST, TREC. https://www.microsoft.com/en-us/research/publication/overview-of-the-trec-2022-deep-learning-track/

[14] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, Jimmy Lin, Ellen M. Voorhees, and Ian Soboroff. 2025. Overview of the TREC 2022 deep learning track. arXiv:2507.10865 [cs.IR] https://arxiv.org/abs/2507.10865

[15] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Ellen M Voorhees. 2019. Overview of the TREC 2019 Deep Learning Track. In *TREC*.

[16] 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).

[17] Suchana Datta, Sean MacAvaney, Debasis Ganguly, and Derek Greene. 2022. A 'Pointwise-Query, Listwise-Document'based Query Performance Prediction Approach. In *SIGIR*. 2148–2153.

[18] 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. (April 2025). arXiv:2504.19076 [cs.IR]

[19] Guglielmo Faggioli, Laura Dietz, Charles L. A. Clarke, Gianluca Demartini, Matthias Hagen, Claudia Hauff, Noriko Kando, Evangelos Kanoulas, Martin Potthast, Benno Stein, and Henning Wachsmuth. 2023. Perspectives on Large Language Models for Relevance Judgment. In *Proceedings of the 2023 ACM SIGIR International Conference on Theory of Information Retrieval, ICTIR 2023, Taipei, Taiwan, 23 July 2023*, Masaharu Yoshioka, Julia Kiseleva, and Mohammad Aliannejadi (Eds.). ACM, 39–50. doi:10.1145/3578337.3605136

[20] Naghmeh Farzi and Laura Dietz. 2024. An Exam-based Evaluation Approach Beyond Traditional Relevance Judgments. arXiv:2402.00309 [cs.IR]

[21] Luyu Gao, Zhuyun Dai, and Jamie Callan. 2021. COIL: Revisit exact lexical match in information retrieval with contextualized inverted list. *arXiv preprint arXiv:2104.07186* (2021).

[22] Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, et al. 2024. A survey on llm-as-a-judge. *arXiv preprint arXiv:2411.15594* (2024).

[23] Claudia Hauff, Djoerd Hiemstra, and Franciska de Jong. 2008. A survey of pre-retrieval query performance predictors. In *CIKM*.

[24] Sebastian Hofstätter, Sheng-Chieh Lin, Jheng-Hong Yang, Jimmy Lin, and Allan Hanbury. 2021. Efficiently teaching an effective dense retriever with balanced topic aware sampling. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 113–122.

[25] Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. 2025. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems* 43, 2 (2025), 1–55.

[26] Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906* (2020).

[27] Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan, Saiful Haq, Ashutosh Sharma, Thomas T Joshi, Hanna Moazam, et al. 2023. Dspy: Compiling declarative language model calls into self-improving pipelines. *arXiv preprint arXiv:2310.03714* (2023).

[28] Satyapriya Krishna, Kalpesh Krishna, Anhad Mohananey, Steven Schwarcz, Adam Stambler, Shyam Upadhyay, and Manaal Faruqui. 2025. Fact, Fetch, and Reason: A Unified Evaluation of Retrieval-Augmented Generation. arXiv:2409.12941 [cs.CL] https://arxiv.org/abs/2409.12941

[29] Robert Krovetz and W Bruce Croft. 1992. Lexical ambiguity and information retrieval. *ACM Transactions on Information Systems (TOIS)* 10, 2 (1992), 115–141.

[30] Sean MacAvaney and Luca Soldaini. 2023. One-Shot Labeling for Automatic Relevance Estimation. In *SIGIR*. 2230–2235.

[31] Gregory Marton. 2006. Nuggeteer: Automatic nugget-based evaluation using descriptions and judgements. (2006).

[32] Chuan Meng, Negar Arabzadeh, Mohammad Aliannejadi, and Maarten de Rijke. 2023. Query Performance Prediction: From Ad-hoc to Conversational Search. In *SIGIR*. 2583–2593.

[33] Virgil Pavlu, Shahzad Rajput, Peter B Golbus, and Javed A Aslam. 2012. IR system evaluation using nugget-based test collections. In *Proceedings of the fifth ACM international conference on Web search and data mining*. 393–402.

[34] Ronak Pradeep, Nandan Thakur, Sahel Sharifymoghaddam, Eric Zhang, Ryan Nguyen, Daniel Campos, Nick Craswell, and Jimmy Lin. 2025. Ragnarök: A reusable RAG framework and baselines for TREC 2024 retrieval-augmented generation track. In *European Conference on Information Retrieval*. Springer, 132–148.

[35] 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 preprint arXiv:2411.09607* (2024).

[36] Ronak Pradeep, Nandan Thakur, Shivani Upadhyay, Daniel Campos, Nick Craswell, and Jimmy Lin. 2025. The Great Nugget Recall: Automating Fact Extraction and RAG Evaluation with Large Language Models. arXiv:2504.15068 [cs.IR] https://arxiv.org/abs/2504.15068

[37] Shahzad Rajput, Virgiliu Pavlu, Peter B. Golbus, and Javed A. Aslam. 2011. A nugget-based test collection construction paradigm. In *Proceedings of the 20th ACM Conference on Information and Knowledge Management, CIKM 2011, Glasgow, United Kingdom, October 24-28, 2011*, Craig Macdonald, Iadh Ounis, and Ian Ruthven (Eds.). ACM, 1945–1948. doi:10.1145/2063576.2063861

[38] Amirhossein Razavi, Mina Soltangheis, Negar Arabzadeh, Sara Salamat, Morteza Zihayat, and Ebrahim Bagheri. 2025. Benchmarking prompt sensitivity in large language models. In *European Conference on Information Retrieval*. Springer, 303–313.

[39] Stephen Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu, and M. Gatford. 1995. Okapi at TREC-3. In *Overview of the Third Text REtrieval Conference (TREC-3)* (overview of the third text retrieval conference (trec–3) ed.). Gaithersburg, MD: NIST, 109–126. https://www.microsoft.com/en-us/research/publication/okapi-at-trec-3/

[40] David P Sander and Laura Dietz. 2021. EXAM: How to Evaluate Retrieve-and-Generate Systems for Users Who Do Not (Yet) Know What They Want.. In *DESIRES*. 136–146.

[41] Paul Thomas, Seth Spielman, Nick Craswell, and Bhaskar Mitra. 2023. Large language models can accurately predict searcher preferences, 2023. *URL https://arxiv.org/abs/2309.10621* (2023).

[42] 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 preprint arXiv:2406.06519* (2024).

[43] Xinyi Yan, Chengxi Luo, Charles LA Clarke, Nick Craswell, Ellen M Voorhees, and Pablo Castells. 2022. Human preferences as dueling bandits. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 567–577.

[44] Peilin Yang, Hui Fang, and Jimmy Lin. 2017. Anserini: Enabling the use of lucene for information retrieval research. In *Proceedings of the 40th international ACM SIGIR conference on research and development in information retrieval*. 1253–1256.

[45] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2019. BERTScore: Evaluating Text Generation with BERT. *CoRR* abs/1904.09675 (2019). arXiv:1904.09675 http://arxiv.org/abs/1904.09675

[46] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems* 36 (2023), 46595–46623.

[47] Jingming Zhuo, Songyang Zhang, Xinyu Fang, Haodong Duan, Dahua Lin, and Kai Chen. 2024. ProSA: Assessing and understanding the prompt sensitivity of LLMs. *arXiv preprint arXiv:2410.12405* (2024).