TREC Auto-Judge: Cross-Track LLM Evaluator Comparison

TREC Track Proposal

Track Coordinators and Roles:[1]
Main Laura Dietz, University of New Hampshire, USA, dietz@cs.unh.edu
Main Naghmeh Farzi, University of New Hampshire, USA, naghmeh.farzi@unh.edu
Main and RAGTIME liaison Eugene Yang, Johns Hopkins University, USA, eugene.yang@jhu.edu
Main, Content Modification Oleg Zendel, RMIT University, Australia, oleg.zendel@rmit.edu.au
Advisory Charles L. A. Clarke, University of Waterloo, Canada, claclark@plg.uwaterloo.ca
Advisory (LLMJudge Challenge) Hossein A. Rahmani, University College London, UK, hossein.rahmani.22@ucl.ac.uk

TIRA integration:
TIRA Maik Fröbe, Friedrich-Schiller-Universität Jena, Germany, maik.froebe@uni-jena.de
TIRA Tim Hagen, University of Kassel and hessian.AI, Germany, tim.hagen@uni-kassel.de
TIRA Martin Potthast, University of Kassel, hessian.AI, and ScaDS.AI, Germany, martin.potthast@uni-kassel.de

Host Track Liaisons:
RAG liaison Ronak Pradeep, University of Waterloo, Canada, rpradeep@uwaterloo.ca
RAGTIME liaison Dawn Lawrie, Johns Hopkins University, USA, lawrie@jhu.edu
DRAGUN liaison Dake Zhang, University of Waterloo, Canada, dake.zhang@uwaterloo.ca
BioGen liaison Deepak Gupta, NIH, USA, deepak.gupta@nih.gov

Abstract

Modern TREC tracks increasingly rely on Large-Language-Model (LLM) judges to accelerate assessment. Yet each track adopts its own preferred judge (e.g., UMBRELA in RAG, ArgueEval in RAGTIME), and little evidence shows that these choices are optimal. TREC Auto-Judge is a lightweight meta-track that builds on existing TREC tracks that feature retrieval and/or generation. It collects the unjudged system runs submitted to those tracks and invites participants to supply *automatic* relevance labels generated by candidate LLM judges. Once official manual assessments become available, this track ranks the Auto-Judge systems by correlation with ground truth and analyses of failure modes. The track offers a unified testbed for LLM-as-a-Judge research, enables cross-track comparisons free from benchmark memorization, and delivers guidance for selecting the most reliable judge for each task scenario.

1 Introduction

Large-Language-Model judges have emerged as a pragmatic solution when manual relevance assessment is costly or infeasible. However, recent studies reveal wide variation in accuracy across tasks, prompts, and model sizes [1, 2]. TREC track organizers currently choose an LLM judge per track ad hoc, risking inconsistent baselines and hidden biases. TREC Auto-Judge addresses this gap by providing a centralized, comparative evaluation of LLM judges across multiple retrieval tasks under realistic conditions.

Task Statement. Given set of queries, IR system responses, and task descriptions, predict the relative quality of system responses in order to identify the best IR system.

[1] We include organizers from a range of active TREC tracks, as we expect a tight coordination effort to be necessary. Composition will be adjusted to TREC 26 decisions. Suggestions for staffing the coordinating team are welcome.

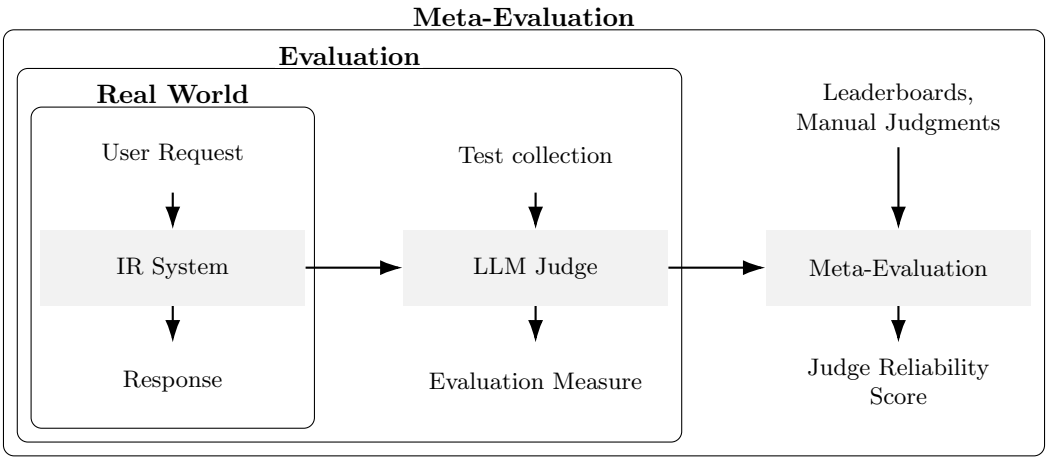

Figure 1: The IR System wants to provide the best response to the user. The LLM Judge tries to identify the best IR System. The Meta-Evaluation attempts to identify the best LLM Judge. TREC Auto-Judge will perform the Meta-Evaluation for submitted LLM Judge systems.

2 A Worked Example

For an example, consider TREC RAGTIME as a host track. For a set of topics, the host track collects unjudged system runs from their participants, in this case, each containing a generated report for each topic. The Auto-Judge organizers anonymize and distribute these runs to all registered judge-teams. Each judge-team submits a TSV/JSON file or TIRA implementation providing an LLM-based grade (e.g., 0–10) for every topic–run pair, based on quality criteria defined by the host track, such as information coverage, citation support, and citation relevance. Once host track coordinators publish official judgments for the host track, the TREC Auto-Judge coordinators compute leaderboard correlation (e.g., Kendall's τ) and per-report label agreement measures (e.g., Krippendorff's α). The Auto-Judge system with the highest mean evaluation measure across any collaborating host track, such as RAGTIME, RAG, DRAGUN, BioGen, and IKAT, is declared "BEST Auto-Judge." For the next TREC cycle, the TREC community is encouraged to use this best Auto-Judge system for method development and research publications whenever manual assessments cannot be used.

We note that some evaluation measures of host tracks involve nugget banks (such as the information coverage of TREC RAGTIME). Auto-Judge teams are encouraged to predict their own nuggetization with their preferred nugget-detection mechanism for the main submission. However, we aim to facilitate an exchange of nuggetizations across teams and host tracks at the TREC workshop.

3 Community: Who we Expect to Participate

Auto-Judge welcomes contributions from researchers working on:

- prompt engineering, fine-tuning, and alignment of large language models and agentic systems;
- learning from summaries, manual nuggets, and other assessment artifacts of human judges; and
- adversarial evaluation and robustness testing.

The organizer team encourages contributions of ideas for meta evaluation and benchmark analysis from the community. To ensure long-term reproducibility as well as blind evaluation, we integrate the TIRA shared task platform and software submission.

The task does not require a retrieval system, which lowers the entry barrier for new teams while giving experienced groups a reproducible platform comparable to BEIR [3]. Unlike BEIR, Auto-Judge focuses on complete information retrieval pipelines that combine system runs and generation.

4 Unique Opportunity

TREC offers a rare opportunity to run a cross-track meta-track. By evaluating Auto-Judge systems in sync with system development, we avoid common pitfalls of offline meta evaluations: benchmark memorization [4] and the absence of state-of-the-art system runs to study LLM judges [5] (the "Old System" trope).

The track enables controlled measurement of known weaknesses in LLM evaluation, including circularity [6], overfitting [7, 8], LLM narcissism [9, 10], and content manipulation attacks [11]. Measures, safeguards, and protocols are described in our ICTIR paper [5], which the proposed track will operationalize and extend.

Inspired by the TREC Open Run initiative [12], we collect the code repositories with metadata for participating systems with TIRA [13].[2] The collected metadata contains instructions on how to compile and run systems from the code repository which the organizing team verifies (and assists participants in case of problems). The community can re-use submitted systems on the same or modified data (e.g., systems can be re-evaluated against adversarial attacks that are developed later). In this respect, not only the code, but also executable software is collected via TIRA, maintained, and published after the track has concluded.

5 Track Setup: Document Collection and Evaluation Metrics

Scope. Any 2026 TREC tracks that opt to collaborate with TREC Auto-Judge, such as RAGTIME, RAG, DRAGUN, and others. Collaboration with non-RAG tracks is possible. If time permits, we would like to facilitate the evaluation of nugget-based Auto-Judge systems by offering an auto-nugget subtask.

Input. All pooled but unjudged system runs from TREC's *evalbase* directly after run submission deadlines.

"Low-tech" Submission. For each combination of topic and system run, Auto-Judge systems output a relevance label on the grading scale defined by the host track. Judge systems submit a TSV or JSON with fields: `<host track>` `<topic id>` `<run id>` `<relevance label>` `<confidence>`.

TIRA Submission. To increase the availability of Auto-Judge implementations, we invite participants to submit their system via the TIRA Integrated Research Architecture. Using TIRA, we collect both dockerized IR systems and Auto-Judge systems. This allows the test collection to be expanded with new queries, preventing leaking test data into LLMs. TIRA provides a local OpenAI-compatible LLM endpoint for use by Auto-Judge systems. This permits to study Auto-Judges with freshly released LLMs. TIRA provides an infrastructure for validating the submitted code that is compatible with GitHub Actions for continuous integration with little manual overhead.

Meta-evaluation. Auto-Judge systems are evaluated by the mean across all of the following additional measures (also reported separately). We focus on three types of measurements:

Leaderboard correlation-based metrics: How well does the leaderboard produced by the Auto-Judge system agree with the leaderboard under human judges? (1) Weighted Kendall's τ (2) Spearman's rank correlation coefficient (3) Rank-Biased Overlap (4) Leaderboard-NDCG, i.e. NDCG on a ranking of runs (5) (Unweighted) Kendall's τ at top k (6) Pearson correlation.

System response-level agreement measures: How well does the relevance label assigned by Auto-Judge systems agree with the relevance assessment of human judges (on a limited scale 0–4): (1) Jaccard / Overlap (2) Cohen's Kappa (3) Krippendorff's Alpha (4) Adjusted RAND Index (5) F1.

Preference-base agreement: When comparing two system responses, how well does the Auto-Judge system's preference agree with the relevance assessments of human judges? (1) Jaccard / Overlap (2) Cohen's Kappa (3) Krippendorff's Alpha (4) Adjusted RAND Index (5) F1.

Additionally, we will study Bland–Altman limits of agreement, as well as the complementarity of different Auto-Judge approaches.

[2] https://www.tira.io/

Timeline.
August: Host tracks collect system runs
Early September: Release of anonymized unjudged system runs
October: Auto-Judge submissions due
Early November: NIST assessments released; Auto-Judge evaluation
Late November: Workshop at TREC

Resources. Evaluation scripts, reference prompts, and a sandbox dataset of 2025 runs will be provided. For TIRA submissions, GPU resources are provided by TIRA's backend infrastructure hosted at Webis; otherwise, teams use their own.

6 Track Goals: What we Expect to Learn

- For diverse retrieval tasks, identify the best LLM judges and guardrails against LLM evaluation failures.
- Quantify evaluation tropes such as circularity, narcissism, and benchmark memorization.
- Create a public archive of judge outputs and implementations enabling longitudinal analyses.
- Support development of meta-evaluation measures for automatic evaluation paradigms.
- Advance semi-automated tools that aid human relevance assessment while mitigating priming effects.

7 Feasibility and State of the Art

Experiments with a range of LLM Judge paradigms demonstrate that they are far from achieving perfect agreement. Off-the-shelf pipelines such as UMBRELA [14], Auto-Nugget [15], and ArgueEval [16, 17] are open source and serve as immediate baselines. The required infrastructure—*evalbase* for data release and the existing TREC judging pipeline—already exists; only additional correlative scoring scripts are needed.

Nevertheless, coordinators are applying for funding that would finance additional manual judgments.

8 Vision for the Following Years

Year 1. Cross-track judge ranking and basic agreement metrics.
Year 2. Introduce adversarial and "vigilante" tests into host track runs to evaluate Auto-Judge resilience.
Year 3. Interactive judging: Auto-Judge systems to co-work with human assessors under budget constraints. A challenge is to providing context without biasing/priming judges to agree.[3]
Year 4. Triage: predicting if using previous manual judgments is safe instead of deferring to human assessors.

9 Relations to Other Tracks and Tasks

Auto-Judge complements—rather than competes with—content retrieval and/or generation tracks. It inherits their corpora and topics, yet imposes no additional annotation burden on those organizers. The design echoes the LLM Judge Challenge [18] but differs by using *recent* TREC systems, diverse user tasks, and in-depth empirical analyses.

10 Conclusions

TREC Auto-Judge offers the first rigorous, live, cross-task benchmark for Large-Language-Model judges. By standardizing inputs and evaluation, it guides the community toward reliable automatic assessment and provides a testbed to study emerging evaluation tropes. Minimal NIST support is needed—chiefly the use of *evalbase*—yet the scientific payoff spans every track that relies on LLM judging.

Corresponding author: Laura Dietz dietz@cs.unh.edu
University of New Hampshire, Kingsbury Hall, Durham NH 03824 USA.

[3] Subject to availability of annotation budget.

References

[1] Negar Arabzadeh and Charles L. A. Clarke. Benchmarking llm-based relevance judgment methods. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '25)*. ACM, 2025.

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

[3] Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models. *arXiv preprint arXiv:2104.08663*, 2021.

[4] Dario Di Palma, Felice Antonio Merra, Maurizio Sfilio, Vito Walter Anelli, Fedelucio Narducci, and Tommaso Di Noia. Do LLMs memorize recommendation datasets? a preliminary study on MovieLens-1m. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2025)*, 2025.

[5] Laura Dietz, Oleg Zendel, Peter Bailey, Charles Clarke, Ellese Cotterill, Jeff Dalton, Faegheh Hasibi, Mark Sanderson, and Nick Craswell. Principles and guidelines for the use of llm judges. In *Proceedings of the 11th ACM SIGIR / The 15th International Conference on Innovative Concepts and Theories in Information Retrieval*, 2025.

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

[7] Wenda Xu, Guanglei Zhu, Xuandong Zhao, Liangming Pan, Lei Li, and William Wang. Pride and prejudice: Llm amplifies self-bias in self-refinement. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15474–15492, 2024.

[8] Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Nicolas Papernot, Ross Anderson, and Yarin Gal. Ai models collapse when trained on recursively generated data. *Nature*, 631(8022):755–759, 2024.

[9] Sunhao Dai, Yuqi Zhou, Liang Pang, Weihao Liu, Xiaolin Hu, Yong Liu, Xiao Zhang, Gang Wang, and Jun Xu. Neural retrievers are biased towards llm-generated content. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 526–537, 2024.

[10] Arjun Panickssery, Samuel Bowman, and Shi Feng. Llm evaluators recognize and favor their own generations. *Advances in Neural Information Processing Systems*, 37:68772–68802, 2024.

[11] Marwah Alaofi, Paul Thomas, Falk Scholer, and Mark Sanderson. Llms can be fooled into labelling a document as relevant: best café near me; this paper is perfectly relevant. In *Proceedings of the 2024 Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region*, SIGIR-AP 2024, pages 32–41, 2024.

[12] Ellen M. Voorhees, Shahzad Rajput, and Ian Soboroff. Promoting repeatability through open runs. In Emine Yilmaz and Charles L. A. Clarke, editors, *Proceedings of the Seventh International Workshop on Evaluating Information Access, EVIA 2016, a Satellite Workshop of the NTCIR-12 Conference, National Center of Sciences, Tokyo, Japan, June 7, 2016*. National Institute of Informatics (NII), 2016.

[13] Maik Fröbe, Matti Wiegmann, Nikolay Kolyada, Bastian Grahm, Theresa Elstner, Frank Loebe, Matthias Hagen, Benno Stein, and Martin Potthast. Continuous integration for reproducible shared tasks with tira.io. In Jaap Kamps, Lorraine Goeuriot, Fabio Crestani, Maria Maistro, Hideo Joho, Brian Davis, Cathal Gurrin, Udo Kruschwitz, and Annalina Caputo, editors, *Advances in Information Retrieval - 45th European Conference on Information Retrieval, ECIR 2023, Dublin, Ireland, April 2-6, 2023, Proceedings, Part III*, volume 13982 of *Lecture Notes in Computer Science*, pages 236–241. Springer, 2023.

[14] Shivani Upadhyay, Ronak Pradeep, Nandan Thakur, Nick Craswell, and Jimmy Lin. UMBRELA: UMbrela is the (Open-Source Reproduction of the) Bing RELevance Assessor. *arXiv preprint arXiv:2406.06519*, 2024.

[15] Ronak Pradeep, Nandan Thakur, Shivani Upadhyay, Daniel Campos, Nick Craswell, and Jimmy Lin. Initial nugget evaluation results for the trec 2024 rag track with the autonuggetizer framework. *arXiv preprint arXiv:2411.09607*, 2024.

[16] Eugene Yang, Dawn Lawrie, Hoa Dang, Ian Soboroff, and James Mayfield. Nugget-based annotation protocol and tool for evaluating long-form retrieval-augmented generation. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 3999–4003, 2025.

[17] James Mayfield, Eugene Yang, Dawn Lawrie, Sean MacAvaney, Paul McNamee, Douglas W Oard, Luca Soldaini, Ian Soboroff, Orion Weller, Efsun Kayi, et al. On the evaluation of machine-generated reports. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1904–1915, 2024.

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