The Name of the Title Is Hope

Anonymous Author(s)

Abstract

LLM agents with web access are increasingly becoming the default for user-facing question answering. Reliable evaluation in this setting requires tests that remain unseen at train time and cannot be answered via verbatim lookup at inference. However, standard practice still relies on static benchmarks that age quickly and are vulnerable to contamination. We analyze two channels, pretraining leakage where test items appear in the pretraining data, and run-time leakage where the agent finds the exact target answer verbatim during evaluation and reports it rather than demonstrating genuine problem solving. We uncover evidence across five QA datasets for both pretraining and run-time leakage, and we show that static evaluation can overestimate capability. We introduce DynamicWebQA, a dynamic evaluation framework that constructs questions and verifiable answers at evaluation time from relevant web sources. The framework enforces multi-document grounding, records evidence chains, and offers controllable complexity through source set size, number of hops, and reasoning constraints. By generating fresh test instances, DynamicWebQA reduces memorization risk, mitigates run-time leakage, and naturally covers time-sensitive queries.

CCS Concepts

• **Do Not Use This Code → Generate the Correct Terms for Your Paper**; *Generate the Correct Terms for Your Paper*; Generate the Correct Terms for Your Paper; Generate the Correct Terms for Your Paper.

Keywords

Do, Not, Use, This, Code, Put, the, Correct, Terms, for, Your, Paper

ACM Reference Format:

Anonymous Author(s). 2018. The Name of the Title Is Hope. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (Conference acronym 'XX)*. ACM, New York, NY, USA, 5 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

TBD

Conference acronym 'XX, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/2018/06
https://doi.org/XXXXXXX.XXXXXXX

2 Related Work

2.1 Preliminaries

2.1.1 Benchmark Leakage. Benchmark leakage refers to the contamination that occurs when a model has prior exposure to a benchmark's test items during pretraining, leading to inflated evaluation scores and undermining the benchmark's validity [20, 21]. Since most LLMs are pretrained indiscriminately on publicly available web content, there is a high likelihood that they might have encountered benchmark datasets hosted on platforms like HuggingFace, GitHub, Kaggle, and similar sites. If the base LLM used in a WebQA agent has already seen the test examples during pretraining, the evaluation effectively becomes a case of testing on the training set — severely compromising the credibility of the reported results.

2.1.2 Dynamic Benchmarking. Dynamic benchmarking is an emerging evaluation paradigm that generates or adapts test samples at evaluation time, so models are scored on fresh, unseen data and testset contamination is avoided [7, 10, 12, 19, 21, 22]. That means the agent does not see the same test sample twice, effectively addressing issues such as LLM memorization.

2.1.3 Evolving Information. In this work, we refer to information that can change over time, such as the stock price of Tesla, as evolving information. Arguably, this type of content could be labeled as dynamic content or temporal content, but that might cause confusion, since we are already using the term "dynamic" to describe an evaluation paradigm (dynamic benchmarking), and "temporal" might lead readers to associate it with time-series data. To avoid such confusion, we adopt the term evolving information to specifically denote factual content that may shift over time without implying a particular format or evaluation framework.

2.2 Related Work

2.2.1 Open-Domain QA and Retrieval-Augmented Agents. Open-domain question answering traditionally involves retrieving relevant documents and extracting or generating an answer. For instance, consider DrQA [1], which combined a TF-IDF document retriever with a neural reader to answer questions using Wikipedia. Since then, large-scale datasets like Natural Questions [8], TriviaQA [5] and HotPotQA [15] have driven progress in retrieval-based QA, with models achieving impressive in-domain accuracy. The introduction of transformer-based retrievers and readers led to substantial gains – for example, dense passage retrieval and a fusion-in-decoder reader (FiD) improved open QA by aggregating information from multiple passages [3]. More recently, the trend has shifted to retrieval-augmented generation (RAG) systems [17], which integrate retrieval into the generative process. In a RAG model, an LLM conditions on retrieved text chunks when constructing its answer, thereby injecting fresh knowledge and reducing factual errors. RAG-based approaches have become a standard for knowledge-intensive tasks, demonstrating superior factual accuracy and generality [17].

With improvements in LLM reasoning capabilities, research has shifted towards agentic frameworks where a model iteratively interacts with tools, such as search engines and knowledge graphs, to answer queries. For example, WebGPT [11] augments an LLM with the ability to issue web search queries and navigate webpages, guided by human feedback to produce high-quality answers with citations. Other works like ReAct [17] combines reasoning steps with tool use, allowing the model to plan multi-step solutions (e.g. search for a fact, then use a calculator) in a single unified prompting framework. These "LLM-as-agent" approaches are promising because they mimic how humans gather and verify information, and they can handle more complex queries that require multi-hop reasoning or cross-referencing sources [23].

2.2.2 Robustness of QA Systems. Robustness of QA systems has garnered a lot of interest as researchers realized that high IID (in-distribution) test scores do not guarantee real-world reliability [13]. Prior research has examined multiple facets of robustness. One line of work looks at adversarial robustness: for example, adding misleading but irrelevant sentences to a passage can confuse models that lack true understanding [4]. This revealed that many QA models rely on shallow cues and can be tricked by simple perturbations. In open-domain QA, models must also cope with naturally occurring distractions or errors in retrieved text. The HotpotQA dataset [15], which requires multi-hop reasoning across multiple Wikipedia articles and includes some irrelevant paragraphs, was an early attempt to test a model's ability to stay focused on relevant facts. Dense retrievers and readers have been shown to drop in accuracy when deployed on different source distributions or domains – for instance, a model trained on Wikipedia may struggle on biomedical articles. The BEIR benchmark [13] quantified this by evaluating retrieval models on 18 heterogeneous IR tasks: no single model performed uniformly well across all domains, highlighting generalization gaps.

Another crucial aspect is temporal robustness. QA models quickly become outdated as world knowledge changes. To address this, temporal benchmarks have been proposed to evaluate systems on questions about current events or facts that change over time. RealTime QA [6], for example, continually releases new questions (on a weekly basis) about recent news and evaluates systems' ability to answer using up-to-date information. Similarly, FreshQA [14] is a dataset of time-sensitive questions where answers need to be periodically refreshed to remain correct. These benchmarks show that without retrieval augmentation, LLMs fail completely on questions beyond their training cutoff, and even with retrieval, systems must be robust to latency (documents may not yet reflect the latest answers) and potential contradictions between old and new information.

Closely related is the challenge of conflicting evidence. In realistic web search, not all sources agree – some may have incorrect or outdated information. [9] introduced the QACC dataset to study how QA systems handle conflicting contexts: they found that as many as 25% of straightforward factoid questions yield conflicting answers on the web, and current LLM-based QA systems often stumble in these cases, either averaging contradictory statements or choosing incorrectly. Another recent work, RARE (Retrieval-Aware Robustness Evaluation) [18], proposes a unified framework to stress-test RAG models by introducing controlled perturbations at the query and document level. RARE generates variants of questions (e.g. paraphrases or altered facts) and of documents (inserting noise or updating facts) to evaluate if a system remains correct or can recover when its inputs change. Using a time-sensitive fixed set of documents, RARE showed that state-of-the-art RAG systems are brittle: for instance, they are most vulnerable to document perturbations (altered or conflicting content) and degrade significantly on multi-hop questions compared to single-hop ones [18]. These findings reinforce the importance of developing QA agents that maintain high fidelity in the face of distribution shifts — whether those are shifts in language (paraphrasing), content (new or conflicting facts), or context over time.

2.2.3 Dynamic Benchmarking. Most of the benchmarks currently used for evaluating agentic QA systems remain predominantly static, consisting of fixed samples that are publicly accessible, such as HotPotQA [15]. Studies show significant evidence that many widely used static datasets have already been contaminated, rendering them unreliable [20, 21]. In [20], the authors refer to this issue as benchmark leakage – a growing concern as large foundation models are trained on web-scale datasets encompassing vast portions of publicly available internet data. Studies indicate that even simple paraphrasing can degrade performance, emphasizing the brittle nature of these systems and their reliance on memorization [21].

To overcome these limitations, dynamic benchmarks have emerged as a promising alternative [7, 10, 19, 21, 22]. Unlike their static counterparts, dynamic benchmarks are designed to resist memorization and provide a more accurate assessment of adaptive, context-aware performance in evolving scenarios [12, 16, 21, 22]. This shift is particularly important for ensuring that reported evaluations reflect genuine advancements in capabilities rather than superficial performance gains.

Early attempts at dynamic benchmarking rely on crowdsourcing for data collection [7, 10], making them costly and difficult to scale. More recent approaches leverage graph-based methods to generate test samples [19, 21, 22], offering advantages such as controllable complexity and adaptability to evolving requirements — an essential feature given the rapid advancements in foundation models. While these approaches were proven to be effective in reasoning tasks [19, 21, 22], dynamic benchmarking for open-domain QA systems remains largely unexplored. A recent study, Dynamic-KGQA [2], represents an early attempt at dynamic evaluation of QA systems but relies on knowledge graphs (KGs), which limits the benchmark to information stored in KGs. While KGs are updated periodically, the changes are not instantaneous, and supporting fast-evolving QA pairs becomes infeasible.

3 Method

We propose a method that utilizes *seed graphs* to anchor future question generation instead of relying on static datapoints that are susceptible to memorization or data contamination. Because each evaluation round renders a fresh QA pair from each graph, no fixed test item can be leaked ahead of time, yet all rounds remain semantically comparable via the common anchors.

3.1 Seed Graphs

At evaluation round t, we maintain a set $\mathcal{S}_t = \{G_1^{(t)}, \ldots, G_{n_t}^{(t)}\}$ of seed graphs. Each graph encodes the evidence trail (e.g., queries, retrieved documents, answer node) for a distinct information need. To support evolving information, graphs are constructed at evaluation time.

3.2 QA Generation

The generation function Gen takes a seed graph $G_i^{(t)}$, a random seed $r_{i,t} \sim \mathcal{R}$, and a fixed generation configuration to produce one QA pair:

$$(q_{i,t}, a_{i,t}) = \text{Gen}(G_i^{(t)}; r_{i,t}, \theta) \tag{1}$$

Here, Gen is an LLM-based generator that bundles a frozen prompt template and fixed decoding settings (e.g., temperature, top-p, stop tokens). $r_{i,t}$ is an i.i.d. sample from randomness source $\mathcal{R}$, introducing per-sample stochasticity. Holding θ constant and varying only $r_{i,t}$ ensures that each system sees independent but identically distributed QA samples.

3.3 Distributional Consistency

Sampling an independent seed $r_{i,t} \sim \mathcal{R}$ for each graph yields the per-graph distribution:

$$\mathcal{P}_{i,t} = \text{Dist}(\text{Gen}(G_i^{(t)}; r, \theta)) \tag{2}$$

Because seeds are independent across graphs, the round-t dataset $D_t = \{(q_{1,t}, a_{1,t}), \ldots, (q_{n_t,t}, a_{n_t,t})\}$ is an i.i.d. sample from the product measure:

$$\mathcal{P}_t = \prod_{i=1}^{n_t} \mathcal{P}_{i,t} \tag{3}$$

For a model M under evaluation, let $\text{score}((q,a), M)$ be any per-question metric (e.g., exact match, F1, log-loss). The aggregate dataset score is defined as:

$$\text{Agg}(D_t, M) = \frac{1}{n_t} \sum_{i=1}^{n_t} \text{score}((q_{i,t}, a_{i,t}), M) \tag{4}$$

By linearity of expectation and independence of $(q_{i,t}, a_{i,t}) \sim \mathcal{P}_{i,t}$:

$$\mathbb{E}_{D_t \sim \mathcal{P}_t}[\text{Agg}(D_t, M)] = \frac{1}{n_t} \sum_{i=1}^{n_t} \mathbb{E}_{(q,a)\sim\mathcal{P}_{i,t}}[\text{score}((q,a), M)] \tag{5}$$

This shows that the expected accuracy is simply the average of the per-graph expectations—so every seed graph contributes equally.

3.4 Collision Bound

Assume for every seed graph $G_i^{(t)}$, the generator can produce K distinct question–answer pairs:

$$C_{i,t} = \{(q,a) \in \mathcal{P}_{i,t} \mid \Pr((q,a)) > 0\}, \quad |C_{i,t}| = K$$

To characterize overlap across rounds $u < v$, define:

$$J_{i,u,v} = |C_{i,u} \cap C_{i,v}|, \quad J_{\max} = \max_{u<v} J_{i,u,v}$$

Because answers and evidence evolve over time, $J_{\max}$ tends to remain small for fast-changing topics.

Collision Probability. Sampling one QA pair per round, the probability that seed graph i produces the same QA pair twice within t rounds satisfies:

$$\Pr[\text{collision within } t] \le \frac{t(t-1)}{2K^2} J_{\max} \tag{6}$$

Collision Control. To guarantee that this probability is at most $\delta \in (0,1)$, it suffices to choose K such that:

$$K \ge \sqrt{\frac{t(t-1)}{2\delta} J_{\max}} \tag{7}$$

Equation (7) captures key trade-offs:

- Larger K (richer candidate pool) reduces collision probability for fixed t.
- Fewer evaluation rounds t allow a smaller K for the same error tolerance δ.
- Smaller $J_{\max}$ (i.e., greater semantic drift) reduces required K.

Because each seed graph is rebuilt from fresh evidence each round, its candidate set changes, helping keep $J_{\max}$ small. Equation (6) quantifies the collision risk, while Equation (7) provides a tunable design guideline for selecting K.

3.5 Cross-Round Reporting and Compatibility

Because each round rebuilds the seed graph set $\mathcal{S}_t$, the underlying distribution $\mathcal{P}_t = \prod_i \mathcal{P}_{i,t}$ may drift over time, especially for evolving topics. Consequently, raw accuracies from different rounds may not be directly comparable. We offer two strategies:

i) Snapshot Evaluation. Fix a reference round t^* and freeze its seed graphs $\mathcal{S}_{t^*}$. All systems are then evaluated on the same dataset D_{t^*}, ensuring identical test conditions.

ii) Macro-Averaged Score. For longitudinal tracking, aggregate performance across a window $\{t_1, \ldots, t_T\}$ (e.g., most recent T rounds) using:

$$\text{Score}_{\text{macro}} = \frac{1}{T} \sum_{j=1}^{T} \text{Agg}(D_{t_j}, M) \tag{8}$$

Each D_{t_j} is i.i.d. from its own $\mathcal{P}_{t_j}$, so Eq. (8) summarizes average effectiveness under the benchmark's natural evolution, without letting any single round dominate.

4 Results

5 Discussion

TBD

6 Conclusion

TBD

References

[1] Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading Wikipedia to Answer Open-Domain Questions. arXiv:1704.00051 [cs] doi:10.48550/arXiv.1704.00051 Comment: ACL2017, 10 pages.

[2] Preetam Prabhu Srikar Dammu, Himanshu Naidu, and Chirag Shah. 2025. Dynamic-KGQA: A Scalable Framework for Generating Adaptive Question Answering Datasets. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '25)*. Association for Computing Machinery, New York, NY, USA, 3498–3508. doi:10.1145/3726302.3730324

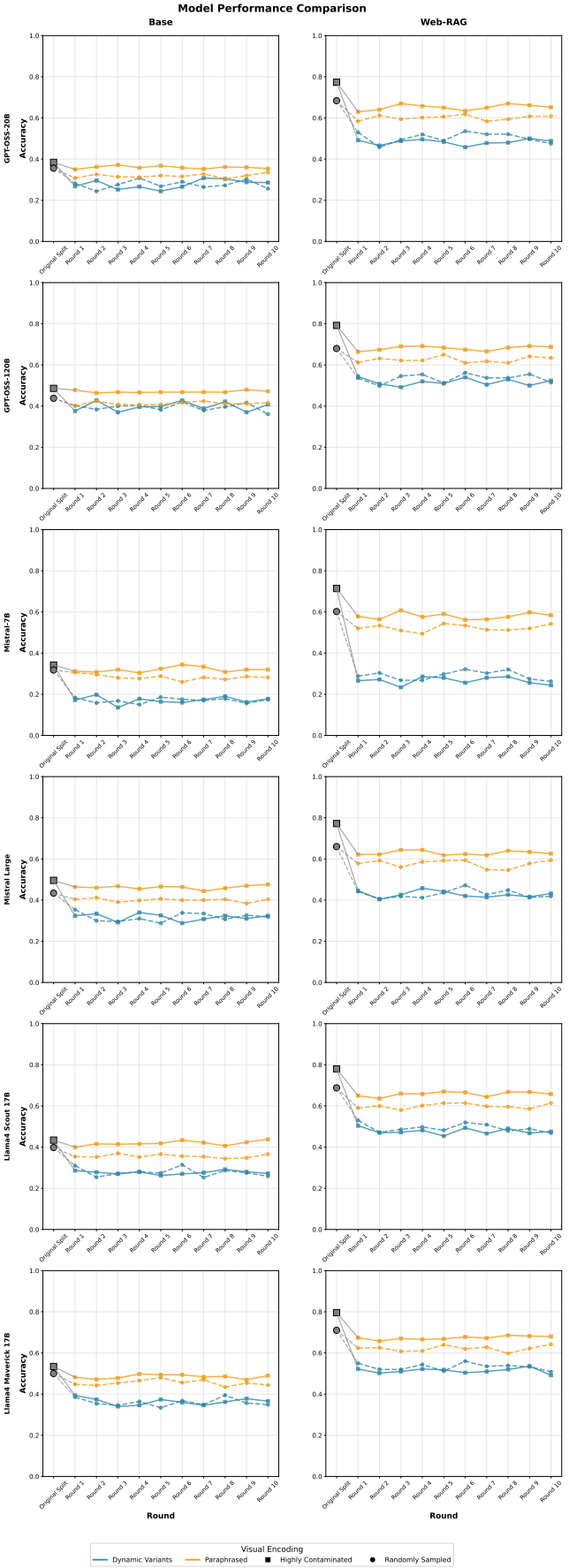

Figure 1: model_performance_collage.pdf

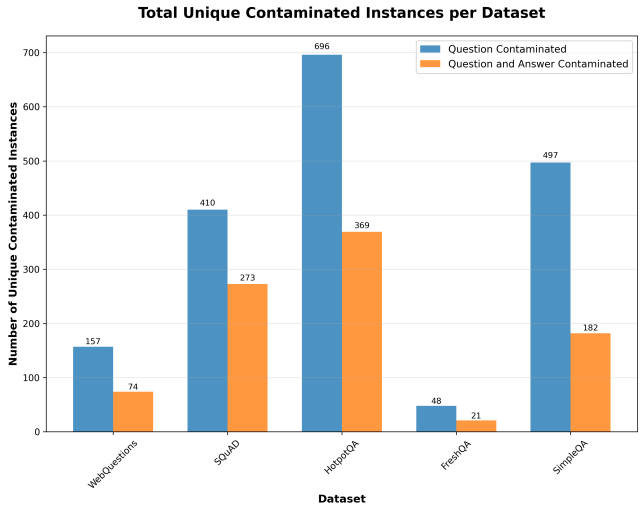

Figure 2: total_unique_contaminated_instances_per_dataset.png

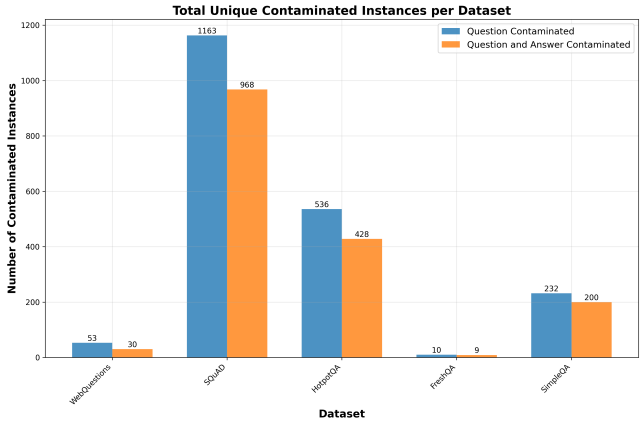

Figure 3: total_unique_retrieval_contaminated_instances_per_dataset.png

[3] Gautier Izacard and Edouard Grave. 2021. Leveraging Passage Retrieval with Generative Models for Open Domain Question Answering. arXiv:2007.01282 [cs] doi:10.48550/arXiv.2007.01282

[4] Robin Jia and Percy Liang. 2017. Adversarial Examples for Evaluating Reading Comprehension Systems. arXiv:1707.07328 [cs] doi:10.48550/arXiv.1707.07328 Comment: EMNLP 2017.

[5] Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. 2017. TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension. arXiv:1705.03551 [cs] doi:10.48550/arXiv.1705.03551 Comment: Added references, fixed typos, minor baseline update.

[6] Jungo Kasai, Keisuke Sakaguchi, Yoichi Takahashi, Ronan Le Bras, Akari Asai, Xinyan Yu, Dragomir Radev, Noah A. Smith, Yejin Choi, and Kentaro Inui. 2024. RealTime QA: What's the Answer Right Now? arXiv:2207.13332 [cs] doi:10.48550/arXiv.2207.13332 Comment: RealTime QA Website: https://realtimeqa.github.io/.

[7] Douwe Kiela, Max Bartolo, Yixin Nie, Divyansh Kaushik, Atticus Geiger, Zhengxuan Wu, Bertie Vidgen, Grusha Prasad, Amanpreet Singh, Pratik Ringshia, Zhiyi Ma, Tristan Thrush, Sebastian Riedel, Zeerak Waseem, Pontus Stenetorp, Robin Jia, Mohit Bansal, Christopher Potts, and Adina Williams. 2021. Dynabench: Rethinking Benchmarking in NLP. arXiv:2104.14337 [cs] doi:10.48550/arXiv.2104.14337 Comment: NAACL 2021.

[8] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural Questions: A

Benchmark for Question Answering Research. *Transactions of the Association for Computational Linguistics* 7 (2019), 452–466. doi:10.1162/tacl_a_00276

[9] Siyi Liu, Qiang Ning, Kishaloy Halder, Wei Xiao, Zheng Qi, Phu Mon Htut, Yi Zhang, Neha Anna John, Bonan Min, Yassine Benajiba, and Dan Roth. 2025. Open Domain Question Answering with Conflicting Contexts. arXiv:2410.12311 [cs] doi:10.48550/arXiv.2410.12311

[10] Zhiyi Ma, Kawin Ethayarajh, Tristan Thrush, Somya Jain, Ledell Wu, Robin Jia, Christopher Potts, Adina Williams, and Douwe Kiela. 2021. Dynaboard: An Evaluation-As-A-Service Platform for Holistic Next-Generation Benchmarking. arXiv:2106.06052 [cs] doi:10.48550/arXiv.2106.06052

[11] Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, Xu Jiang, Karl Cobbe, Tyna Eloundou, Gretchen Krueger, Kevin Button, Matthew Knight, Benjamin Chess, and John Schulman. 2022. WebGPT: Browser-assisted Question-Answering with Human Feedback. arXiv:2112.09332 [cs] doi:10.48550/arXiv.2112.09332 Comment: 32 pages.

[12] Christopher Rawles, Sarah Clinckemaillie, Yifan Chang, Jonathan Waltz, Gabrielle Lau, Marybeth Fair, Alice Li, William Bishop, Wei Li, Folawiyo Campbell-Ajala, Daniel Toyama, Robert Berry, Divya Tyamagundlu, Timothy Lillicrap, and Oriana Riva. 2025. AndroidWorld: A Dynamic Benchmarking Environment for Autonomous Agents. arXiv:2405.14573 [cs] doi:10.48550/arXiv.2405.14573

[13] Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A Heterogenous Benchmark for Zero-shot Evaluation of Information Retrieval Models. arXiv:2104.08663 [cs] doi:10.48550/arXiv.2104.08663 Comment: Accepted at NeurIPS 2021 Dataset and Benchmark Track.

[14] Tu Vu, Mohit Iyyer, Xuezhi Wang, Noah Constant, Jerry Wei, Jason Wei, Chris Tar, Yun-Hsuan Sung, Denny Zhou, Quoc Le, and Thang Luong. 2023. FreshLLMs: Refreshing Large Language Models with Search Engine Augmentation. arXiv:2310.03214 [cs] doi:10.48550/arXiv.2310.03214 Comment: Preprint, 26 pages, 10 figures, 5 tables; Added FreshEval.

[15] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering. arXiv:1809.09600 [cs] doi:10.48550/arXiv.1809.09600 Comment: EMNLP 2018 long paper. The first three authors contribute equally. Data, code, and blog posts available at https://hotpotqa.github.io/.

[16] Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. 2024. τ-Bench: A Benchmark for Tool-Agent-User Interaction in Real-World Domains. arXiv:2406.12045 [cs] doi:10.48550/arXiv.2406.12045

[17] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. ReAct: Synergizing Reasoning and Acting in Language Models. arXiv:2210.03629 [cs] doi:10.48550/arXiv.2210.03629 Comment: v3 is the ICLR camera ready version with some typos fixed. Project site with code: https://react-lm.github.io.

[18] Yixiao Zeng, Tianyu Cao, Danqing Wang, Xinran Zhao, Zimeng Qiu, Morteza Ziyadi, Tongshuang Wu, and Lei Li. 2025. RARE: Retrieval-Aware Robustness Evaluation for Retrieval-Augmented Generation Systems. arXiv:2506.00789 [cs] doi:10.48550/arXiv.2506.00789

[19] Zhehao Zhang, Jiaao Chen, and Diyi Yang. 2024. DARG: Dynamic Evaluation of Large Language Models via Adaptive Reasoning Graph. arXiv:2406.17271 [cs] doi:10.48550/arXiv.2406.17271

[20] Kun Zhou, Yutao Zhu, Zhipeng Chen, Wentong Chen, Wayne Xin Zhao, Xu Chen, Yankai Lin, Ji-Rong Wen, and Jiawei Han. 2023. Don't Make Your LLM an Evaluation Benchmark Cheater. arXiv:2311.01964 [cs] doi:10.48550/arXiv.2311.01964 Comment: 11 pages.

[21] Kaijie Zhu, Jiaao Chen, Jindong Wang, Neil Zhenqiang Gong, Diyi Yang, and Xing Xie. 2024. DyVal: Dynamic Evaluation of Large Language Models for Reasoning Tasks. arXiv:2309.17167 [cs] doi:10.48550/arXiv.2309.17167 Comment: ICLR 2024 spotlight; 38 pages; code is at aka.ms/dyval.

[22] Kaijie Zhu, Jindong Wang, Qinlin Zhao, Ruochen Xu, and Xing Xie. 2024. Dynamic Evaluation of Large Language Models by Meta Probing Agents. arXiv:2402.14865 [cs] doi:10.48550/arXiv.2402.14865 Comment: International Conference on Machine Learning (ICML) 2024.

[23] Yutao Zhu, Huaying Yuan, Shuting Wang, Jiongnan Liu, Wenhan Liu, Chenlong Deng, Haonan Chen, Zheng Liu, Zhicheng Dou, and Ji-Rong Wen. 2025. Large Language Models for Information Retrieval: A Survey. *ACM Transactions on Information Systems* (Sept. 2025), 3748304. arXiv:2308.07107 [cs] doi:10.1145/3748304 Comment: Updated to version 4; Accepted by ACM TOIS.

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009