Democratizing LLM Benchmarking via Automated Dynamic Knowledge Evaluation

Anonymous ACL submission

Abstract

Knowledge memorization is central to large language models (LLMs) and is typically assessed using static benchmarks derived from sources like Wikipedia and textbooks. However, these benchmarks fail to capture evolving knowledge in a dynamic world, and centralized curation struggles to keep pace with rapid LLM advancements. To address this, we propose a fully automated framework for generating high-quality, dynamic knowledge benchmarks on demand. Focusing on the news domain, where knowledge updates daily, we design an agentic framework to automate the sourcing, creation, validation, and distribution of benchmarks while promoting quality and efficiency. Our approach democratizes benchmark creation and facilitates robust evaluation of retrieval-augmented methods by reducing overlap with pretraining data. We evaluate a range of LLMs, both open-source and proprietary, across various sizes and configurations—with and without retrieval—on freshly generated knowledge. Our results reveal distinct model behaviors when confronted with new information and highlight how retrieval narrows the performance gap between small and large models. These findings underscore the importance of evaluating LLMs on evolving benchmarks to more accurately estimate their knowledge capabilities and guide future advancements.

1 Introduction

Assessing the knowledge capabilities of large language models (LLMs) is essential for understanding their performance and limitations. However, this task is increasingly challenging as factual knowledge in the real world evolves rapidly. Well-trained models can quickly become outdated (Li et al., 2024), raising the need for continual model updates (Liška et al., 2022) or improved retrieval-augmented generation (RAG) (Lewis et al., 2020). At the same time, the lack of transparency around training data makes it difficult to assess how current a model's knowledge truly is (Cheng et al., 2024). Existing benchmarks also struggle to keep pace: once released, their contents may be absorbed into future training data, leading to benchmark saturation and weakening their utility. This not only limits our ability to evaluate knowledge retention but also complicates the evaluation of retrieval-based methods, as models may have already memorized the relevant facts. These challenges underscore the need for fast, automated curation of dynamic knowledge benchmarks that can track LLM development in real time and offer a clean testbed for evaluating retrieval augmentation.

Despite the rapid advancement of LLMs and the growing need for accurate knowledge assessment, most standard benchmarks remain *static* after creation. Widely used datasets such as Natural Questions (Kwiatkowski et al., 2019), TriviaQA (Joshi et al., 2017), and HotpotQA (Yang et al., 2018) primarily draw from Wikipedia or curated text snapshots from a fixed time period. While instrumental in advancing open-domain question answering (QA) research, these benchmarks quickly become outdated and are often included in model pretraining corpora, leading to data contamination and inflated performance estimates (Li et al., 2024). More recent efforts—such as StreamingQA (Liška et al., 2022), RealTimeQA (Kasai et al., 2024), FreshQA (Vu et al., 2023), and Daily Oracle (Dai et al., 2024)—have begun incorporating newly emerging facts. However, these dynamic benchmarks still rely on partial human curation, infrequent updates, or focus on narrow domains like forecasting. As a result, they fall short of enabling continuous, decentralized, and user-driven evaluation of dynamic novel knowledge.

To address these challenges and democratize dynamic knowledge benchmarking, we introduce a fully automated framework for generating knowledge benchmarks and evaluating on them. Our goal is to *decentralize* the assessment of LLMs by aligning it with the evolving nature of both model development and real-world information. Focusing on the news domain—where new knowledge emerges daily—our system automates the pipeline from information extraction to benchmark construction in a multiple-choice QA format. We design an agentic framework built on state-of-the-art LLMs, in which specialized agents for QA generation, validation, and revision collaborate to promote quality and consistency.

Since benchmark generation can happen at any time, we introduce a distribution and version control protocol that assigns each benchmark a unique signature, enabling consistent tracking and fair comparison across models and evaluations. These benchmarks serve as snapshots of world knowledge at specific moments—conceptually functioning as *knowledge checkpoints* or *data checkpoints*—supporting longitudinal tracking and temporal comparisons. The framework is fully *open-source* and accessible, empowering *any user* to generate up-to-date benchmarks *at any time*. We refer to our framework as **KODE** (**K**nowledge **O**n-**D**emand **E**valuation). This enables diverse use cases such as monitoring LLM knowledge freshness or evaluating retrieval-augmented models on clean, non-memorized data. By decentralizing benchmark creation, our approach makes knowledge evaluation truly dynamic and ensures it keeps pace with both LLM development and real-world information change.

We present preliminary results using benchmarks recently generated by our framework. Each benchmark includes a ground-truth knowledge source and well-formed multiple-choice QA pairs, facilitating straightforward and reliable evaluation. To assess the quality of the automatically generated benchmarks, we conduct manual validation and find them relatively high quality.[1] To demonstrate the utility of our framework and provide a faithful assessment of current model capabilities, we evaluate a range of LLMs—both open-source and proprietary—across different model sizes, with and without retrieval augmentation. Our results reveal a notable drop in performance when models are tested on newly introduced knowledge, highlighting their limitations in staying current. Interestingly, when retrieval is introduced, the performance gap between smaller and larger models narrows significantly on knowledge not seen during training. We also benchmark different retrieval strategies, showcasing how our dataset can support in-depth evaluation of retrieval-augmented generation.

In summary, we make the following contributions:

- We democratize knowledge evaluation by introducing a dynamic, on-demand benchmarking framework that can be generated at any time, keeping pace with evolving world knowledge and avoiding overlap with model training data.
- We develop an agentic, fully automated pipeline for benchmark generation using LLMs for QA creation, evaluation, and revision—producing high-quality, versioned benchmarks grounded in source documents and openly available for diverse use cases.
- We conduct a comprehensive evaluation of state-of-the-art open-source and proprietary LLMs, both with and without retrieval, demonstrating performance gaps on newly introduced knowledge and showing how retrieval reduces disparities between small and large models.

2 Related Work

Dynamic QA Benchmarks While most QA benchmarks remain static—quickly becoming outdated as world knowledge evolves—recent work has introduced *dynamic* benchmarks to address temporal shifts of knowledge.[2] StreamingQA (Liška et al., 2022) simulates knowledge accumulation over time by organizing questions chronologically across years of news data, but it does not support continuous updates. RealTime QA (Kasai et al., 2024) offers a weekly quiz based on current news headlines, though its scope is limited by the availability and coverage of its external news feeds. FreshQA (Vu et al., 2023) refreshes the answers to a fixed set of time-sensitive questions, but it relies heavily on manual updates, resulting in a centralized and labor-intensive curation process. Daily Oracle (Dai et al., 2024) is fully automated and updated daily, but it centers on forecasting near-future events rather than assessing factual knowledge that

[1] One potential drawback of the automated approach is a compromise in quality. We tolerate certain noise levels as a tradeoff for full automation and large-scale benchmark generation, and we monitor quality through separate manual inspection.

[2] For detailed descriptions of each benchmark, see Appendix A.

Benchmark	Human Involvement	Automation	Update Freq. & Scale
StreamingQA	Partial (curated + synthetic)	Partial	Static
RealTime QA	Yes (media-sourced quizzes)	Partial	Weekly ($\sim$ 30 QA pairs)
FreshQA	Yes (human-written)	Low	Weekly (answers only)
Daily Oracle	No (auto-generated)	Full	Daily ($\sim$ 17.3 QA pairs)
Ours	No (auto-generated)	Full	Any time ($\sim$ 2000 QA pairs)

Table 1: Comparison of dynamic QA benchmarks in terms of human involvement, automation, update frequency, and scale.

has already been established. As summarized in Table 1, none of these approaches combine complete automation and large-scale daily updates:

- **Automation.** RealTime QA and FreshQA still rely on human inputs (e.g., curated quizzes or hand-written questions), and StreamingQA is only partially synthetic. Daily Oracle is fully automated but narrowly focused on event forecasting. In contrast, our pipeline is *fully automated* and operates without human curation, enabling *decentralized* benchmarking of dynamic world knowledge at scale.
- **Frequency and scale.** RealTime QA releases approximately 30 QA pairs weekly, and FreshQA does not only tracks the answer changes for a fixed set of questions. Daily Oracle provides around 17.3 per day. In contrast, our framework generates around 2,000 QA pairs *each time it is invoked*, and can be called *at any time*, enabling scalable and real-time evaluation of LLMs on dynamic knowledge.

RAG Evaluations Existing benchmarks for retrieval-augmented generation (RAG) often suffer from data contamination, where evaluation examples significantly overlap with a model's pretraining corpus—allowing models to bypass retrieval and simply regurgitate memorized content (Li et al., 2024). Many widely used QA datasets, such as Natural Questions (Kwiatkowski et al., 2019), TriviaQA (Joshi et al., 2017), and HotpotQA (Yang et al., 2018), are derived from common sources such as Wikipedia or open web text, making it likely that models already "know" the answers. This reduces the necessity of retrieval and undermines the evaluation of knowledge-seeking behavior. Moreover, including training data in the prompt can further inflate performance by triggering memorized responses (Wang et al., 2022). As a result, current benchmarks fall short in testing whether models can effectively retrieve and reason over genuinely novel information. These limitations underscore the need for a new benchmark paradigm—one that ensures freshness of knowledge and enables accurate assessment of real-time retrieval capabilities.

By emphasizing both automation and high-volume benchmarking data generation at any time, our approach offers a continuous, up-to-date evaluation of factual knowledge without the bottleneck of centralized human curation. It also supports robust assessment of retrieval-augmented methods as models are required to retrieve genuinely *new* information rather than relying on memorized content.

3 Automated Dynamic Benchmarking

3.1 Dynamic Knowledge Source

We focus on the news domain—where new facts are introduced continuously. Specifically, we scrape a diverse set of news outlets, including both mainstream and specialized publications. The categorization and considered sources of news are presented in Table 2. This approach provides broad coverage across geopolitical regions, topical domains, and journalistic styles.

3.2 Benchmark Construction Pipeline

To enable fully automated and democratized benchmark creation, we design an agentic framework for dynamic knowledge benchmarking (Yao et al., 2023; Madaan et al., 2023). The pipeline consists of four key stages: (1) source data extraction, (2) QA pair generation, (3) question validation and revision, and (4) dataset versioning. An overview of the pipeline is shown in Figure 1.

Knowledge Source Extraction We collect and preprocess news articles published within the past 24 hours from a diverse set of outlets (Section 3.1). Articles are retrieved via RSS feeds, parsed, and organized by topic. For each article, we retain a structured representation that includes metadata such as the title, publication date, author, content body, and

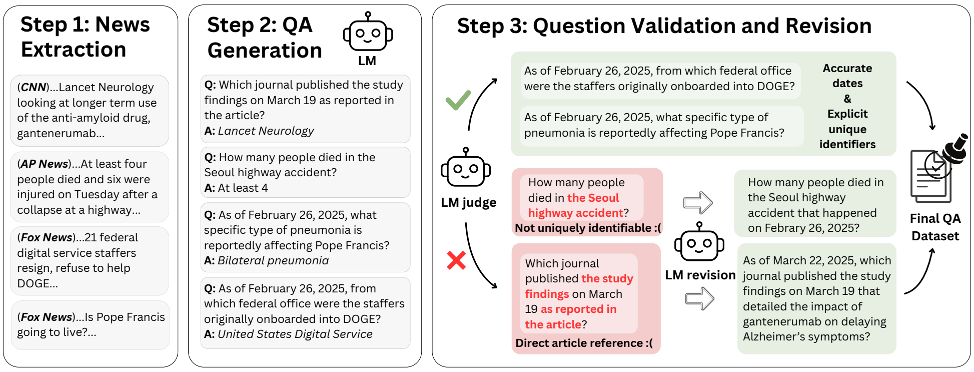

Figure 1: Automated dynamic knowledge benchmark construction pipeline.

Category	Sources
General / Mainstream News	CNN, BBC, Reuters, The Guardian, Fox News, NBC News, USA Today, HuffPost, CBS News
International Coverage	Al Jazeera, DW, RT, Channel News Asia (CNA), Times of India, South China Morning Post (SCMP)
Political Focus	Politico, The Hill, NPR
Technology and Science	TechCrunch, The Verge, Engadget, Ars Technica, Gizmodo, PC Gamer, TechRadar
Business / Finance	Bloomberg
Lifestyle / Culture	GQ, Vanity Fair
Open-Source Community News	WikiNews

Table 2: News sources used for dynamic knowledge extraction.

source URL. The output of this step is a curated, timestamped feed of news articles, which serves as the raw knowledge base for dynamic benchmark construction in subsequent stages.

QA Generation We employ an LLM-based agent to generate initial multiple-choice QA pairs from the curated news articles. The agent is instantiated using an LLM[3] guided by a specialized prompt designed to elicit high-quality, time-sensitive questions (see Appendix B). The generation process involves identifying salient facts from each article, drafting a corresponding question, and producing one correct answer along with plausible distractor options. The agent is instructed to prioritize recent and unique facts—particularly entities, events, and developments that are unlikely to appear in older training data. Our prompt design encourages questions that are factually grounded, require minimal external context, and emphasize up-to-date knowledge.

Question Validation and Revision Despite detailed prompting, LLM-generated questions may not always be well suited for reliable model evaluation. In particular, some questions may rely heavily on context from the source article, making them unclear or unanswerable in isolation. To address this, we introduce a dedicated question validation agent (see validation prompt in Appendix B) that assesses the quality and clarity of each question.

The agent is tasked with verifying whether each question can be answered uniquely and unambiguously, without requiring access to the original article. Specifically, it checks whether the question: (1) avoids direct references to the source article, (2) includes accurate and clear date references, (3) uses explicit identifiers for entities such as people, organizations, or events, and (4) avoids vague or ambiguous phrasing. Questions that fail any of these criteria are automatically routed to a revision agent for correction.

A dedicated revision agent refines any QA pairs that do not meet the specified quality criteria, ensuring that each question is clear, unambiguous, and context-independent. The final evaluation dataset consists of both the validated questions that passed the initial checks and the revised questions corrected by the agent. Note that the validation and revision steps can be applied iteratively for further refinement. We adopt a single round of revision in our current pipeline to balance quality and computational efficiency. This setting is configurable, allowing for greater strictness or flexibility depending on downstream evaluation needs. Some example QA pairs dynamically created in the datasets are shown in Table 3.

Dataset Versioning To support reproducibility and fair comparison, each benchmark release is assigned a unique *signature* serving as its version

[3]We use `o3-mini-2025-01-31` (and also for other LLM agents in our pipeline).

identifier. Because dataset content can shift—due to changes in daily news and the inherent stochasticity of LLM generation—we adopt a principled versioning approach inspired by SacreBLEU's reproducibility framework (Post, 2018). Each signature encodes the agent LLM model name and version (e.g., "GPT-4o" with revision), the decoding hyperparameters (temperature, top-p, etc.), the dataset generation date and timestamp, and a randomly generated hash (e.g., MD5) as a unique identifier.

Users reporting results on our benchmarks should explicitly cite the full dataset signature and share the corresponding dataset snapshot. This enables precise reproduction and fair evaluation by others. By versioning each dataset and requiring explicit references, future work can reliably evaluate on the same benchmark instance—an essential safeguard in our decentralized benchmarking protocol, where potentially numerous, independently generated datasets may exist.

3.3 Human Validation

We randomly sample 400 QA pairs and check them for clarity, answerability, and distractor plausibility, ensuring direct language, exclusive reliance on the article, correct use of dates and names, four plausible choices with only one correct answer, and no explicit references to the article. Following Appendix D, each QA pair is labeled pass or fail. Because we aim for fully automated, decentralized usage, a small level of noise is acceptable to maintain scalability, freshness, and real-time evaluation. We also release a daily version of the benchmark, enabling on-demand dataset generation under evolving knowledge conditions. As proprietary LLMs change over time, we recommend periodic audits and updates to maintain consistent quality. By keeping human validation separate from the core pipeline, our framework remains cost-effective and adaptive, while still supporting quality control when needed.

3.4 Dataset Statistics

When generating the dataset, our pipeline collects the latest 24 hours of news articles and typically produces around 2,000 questions each time it is invoked. Here, we present an analysis of a dataset snapshot generated on March 22, which contains 2,350 questions after initial processing.

4 Experimental Setup

In the following experiments, we evaluate our models on the *March 22 snapshot* of the dataset (Section 3.4). This final QA set contains 470 news articles and 2,350 validated QA pairs, with an average of 773.89 words per article and 18.01 words per question.[4] We evaluate a variety of open-source and proprietary LLMs. For the full list of models, please see Table 6.

Evaluation Settings We test each LLM under three information-access paradigms:

(i) **No context**: The model sees only the question. We simply provide the prompt: *"Question: {*`Q`*}. Provide the most accurate answer."* This reflects a purely parametric recall scenario, where the model must rely solely on its memorized knowledge.

(ii) **Oracle context**: The model is given the exact ground-truth article (i.e. the document originally used to generate the question) as additional context. Here, the model input is of the form: *"Context: {*`Article`*}. Question: {*`Q`*}."* This setting assesses an upper bound of performance when the necessary information is guaranteed to be available and relevant.

(iii) **Retrieval.** We simulate a scenario where the model queries a recent news corpus and must retrieve relevant passages before answering. We provide the top-k passages (where $k \in \{1, 3, 5, 10\}$) returned by a retrieval system, concatenated into the prompt. The corpus is drawn from the last 24 hours (1-Day), the preceding 5 days (5-Day), or the preceding 10 days (10-Day). As the corpus grows, more outdated or irrelevant content is introduced, increasing retrieval difficulty.

Retrieval Methods We implement a variety of retrievers to supply context in the Retrieval Setting. Each daily snapshot of news is indexed using **BM25 (lexical)**, a classic inverted-index-based method leveraging term frequency and inverse document frequency; **ColBERT v2 (dense)**, which encodes both queries and documents into token-level embeddings, using a late-interaction mechanism to preserve fine-grained matching; and **DPR (dense)**, a dual-encoder approach producing a single embedding per document and question, scored via dot

[4] We focus on this single-day snapshot to provide a concrete, up-to-date evaluation, though our framework can generate new benchmarks daily.

Table 3: Example generated QA Pairs. The date of dataset generation is February 26, 2025.

Question	Choices	Ground Truth
As of February 26, 2025, what percentage of GDP has UK Prime Minister Keir Starmer announced the country will spend on defense?	A. 2.3% of its GDP B. 3% of its GDP C. 2.5% of its GDP D. 7% of its GDP	C. 2.5% of its GDP
On February 14, 2025, at which hospital was Pope Francis hospitalized for a respiratory infection?	A. St. Peter's Hospital B. Vatican Medical Center C. Gemelli Hospital D. Apostolic Palace Clinic	C. Gemelli Hospital
In which year did Pope Francis have a piece of one lung removed?	A. 1967 B. 1955 C. 1947 D. 1957	D. 1957
On February 26, 2025, which individual from the Department of Psychiatry at the University of Cambridge emphasized the urgent need for new dementia treatments?	A. Dr. Marc Siegel B. Dr. Ben Underwood C. Dr. Chris Vercammen D. Melissa Rudy	B. Dr. Ben Underwood
As of March 22, 2025, which journal published the study findings on March 19 that detailed the impact of gantenerumab on delaying Alzheimer's symptoms?	A. The Lancet Psychiatry B. JAMA Neurology C. Neurology D. The Lancet Neurology	D. The Lancet Neurology

Statistic	Initial Gen	Validation (Pass)	Validation (Fail)	Revision of Fail	Final Set
Number of questions	2350	2161	189	189	2350
Avg. words in articles	773.89	773.89	773.89	773.89	773.89
Avg. words in queries	17.83	17.95	16.56	18.69	18.01
Avg. QA/article	5.00	4.60	0.40	0.40	5.00

Table 4: Key statistics of the QA dataset at each phase of the pipeline. The table reflects data generated on March 22.

product. For all dense retrievers, we use FAISS (Douze et al., 2025) with a flat index for approximate nearest neighbor search. We measure top-1, top-3, top-5, and top-10 retrieval accuracy (the fraction of queries where the ground-truth article is among the top-k retrieved documents), as well as final QA performance after the model consumes those retrieved contents.[5]

5 Evaluation Results

5.1 LLM Knowledge vs. Oracle Context

Figure 2 summarizes the performance of three representative model families (Gemma, Llama, Qwen) on our time-sensitive QA task in both No context and Oracle context settings. Table 6 then provides a more complete set of results for all open-sourced models.

Observation 1: Impact of Fresh Knowledge. When models must rely solely on parametric memory (No context), their performance is far from perfect across all sizes. This reflects the challenge of truly new facts that arise after the model's pretraining cutoff. Nevertheless, larger models do retain a slight edge. For instance, `gemma-3-1b-it` only achieves 31.1% accuracy in No context mode, whereas `gemma-3-27b-it` reaches 54.0%. The same trend appears in other families like Llama (26.6% vs. 57.2%) and Qwen (28.2% vs. 56.3%) when comparing the smallest and largest variants. Some events in the news may be connected to prior context (e.g., ongoing political debates) that even a smaller model has partially encountered, while larger models have even more background knowledge, allowing them to guess more accurately than random chance (i.e. 25%) in No context mode.

Observation 2: Oracle Context and a "Cutoff" for Reading Comprehension. Once the ground-truth article is given (*Oracle context* setting), we see a pronounced improvement in accuracy. However, contrary to the idea that *all* models do well with the article, Table 6 shows a sharp performance

[5] More implementation details are in Appendix E.

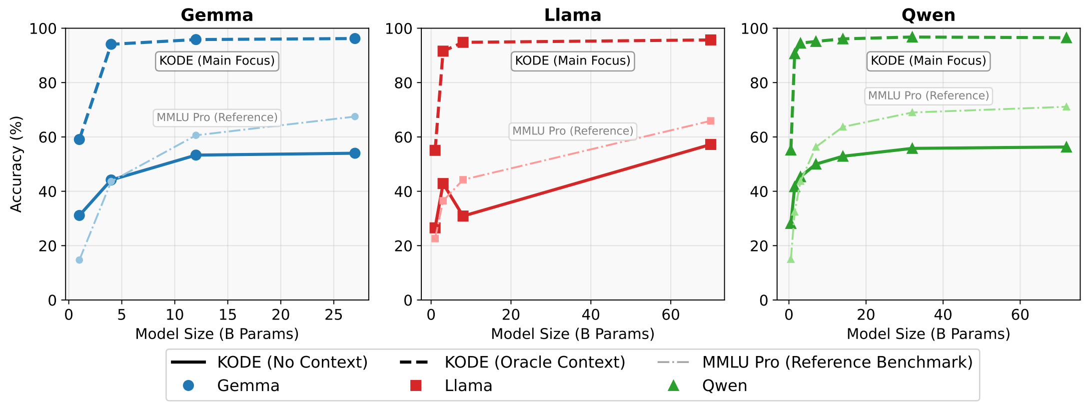

Figure 2: **No context vs. Oracle context QA Accuracy on KODE**, plotted alongside each model's performance on MMLU Pro (lighter lines) as a reference for memorized knowledge. We show three representative model families (Gemma, Llama, Qwen) at various parameter scales (Billion Parameters). Solid lines denote *No context* accuracy (fresh knowledge), and dashed lines denote *Oracle context* accuracy when the ground-truth article is provided.

cutoff. Models around or above roughly 3–4 B parameters can read and understand the article sufficiently to push their Oracle accuracy toward 90–95%. Yet *very small* LLMs (e.g., 1 B parameters) only achieve around 55–60% even with the ground-truth article. This indicates a lower bound on reading comprehension capacity for extremely small models: they simply lack the representational power to parse the passage and correctly pinpoint the answer.

Observation 3: Smaller vs. Larger Models on Fresh Data vs. Memorized Knowledge. Notably, the gap between smaller and larger models in the *No context* setting is smaller than one might expect from standard benchmarks that rely heavily on memorized knowledge. To illustrate this point, we also measured each model's performance on **MMLU Pro**, a knowledge-intensive benchmark widely used for assessing factual recall from pretraining. Table 7 in Appendix G shows that on MMLU Pro, scaling from a 1B to a 27B (or 70B) model often yields improvements exceeding 40–50 percentage points; in contrast, for our newly generated QA data, the improvement over the same size range is closer to 20–25 points. For instance, Gemma 3 (1B) only attains 14.7% on MMLU Pro while Gemma 3 (27B) jumps to 67.5%—a gap of more than 50 points. On *fresh* news QA, that same model scaling moves from 31.1% to 54.0%. This underscores that while model scale is critical for memorizing facts during pretraining, its benefits are comparatively limited for *emergent* knowledge. Consequently, even modestly sized models can hold their own when faced with entirely novel events that arise after training.

Observation 4: Robustness of Oracle Context. Once the ground-truth article is appended to the query, most models (above a certain size threshold) quickly climb to high accuracy ($\sim 95\%$). Even a 4–7 B parameter model can answer correctly given the right passage, suggesting that *timely, precise* context is the main determinant of success. These findings underscore that for fresh or real-time information, building robust retrieval pipelines may be more critical than simply scaling up model size.

5.2 Retrieval Performance

We experiment with three retrievers: **BM25**, **DPR**, and **ColBERT v2**. **Figure 3** shows their top-k accuracy on daily news, while the detailed numerical results (e.g., top-1, top-3, etc.) are presented in **Appendix H** (Tables 8 and 9). Overall, BM25 achieves the highest top-k accuracy in most settings, outperforming both DPR and ColBERT v2. In the 1-day corpus (Figure 3), BM25 yields about 59% top-1 accuracy, whereas DPR and ColBERT v2 follow at 41% and 53%, respectively. As the corpus size grows (e.g., going from 1-day to 5-day or 10-day), retrieval accuracy drops for all methods, reflecting the increased difficulty of searching a larger pool of articles.

Interestingly, even though dense retrievers like DPR and ColBERT v2 often excel on standard benchmarks (Bajaj et al., 2018; Thakur et al., 2021), BM25 proves more robust for this dynamic news scenario. The strong lexical cues (e.g., named entities, event-specific phrasing) may favor exact term matching. Meanwhile, dense retrievers show

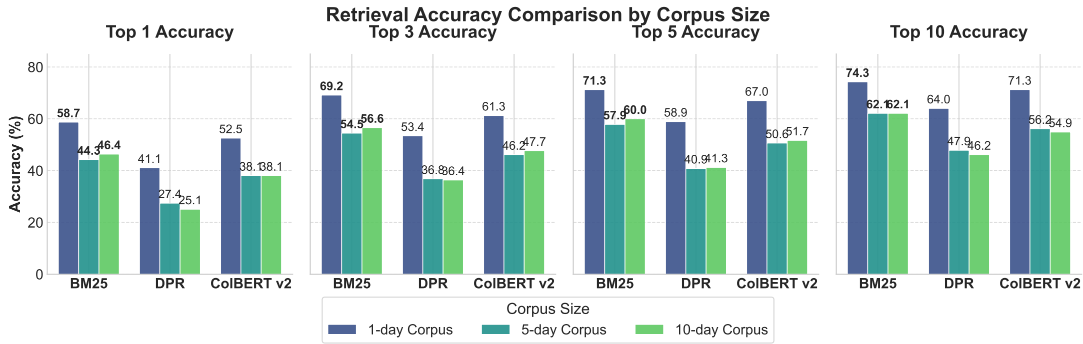

Figure 3: **Top-k Retrieval Accuracy** for BM25, DPR, and ColBERT v2 across news corpora of different time windows (1-day, 5-day, and 10-day).

Table 5: Final QA accuracy (%) of LLMs under Retrieval settings, using `Llama-3.1-8B-Instruct` as the QA backbone. Retrieval is performed over 1-day, 5-day, and 10-day news corpora, returning top-k passages ($k \in \{1, 3, 5, 10\}$).

Retriever	**1-Day Corpus**				**5-Day Corpus**				**10-Day Corpus**			
	Top-1	**Top-3**	**Top-5**	**Top-10**	**Top-1**	**Top-3**	**Top-5**	**Top-10**	**Top-1**	**Top-3**	**Top-5**	**Top-10**
BM25	90.47	93.49	93.40	92.60	88.43	91.79	92.89	92.04	88.30	91.15	92.26	92.09
DPR	66.26	77.66	81.28	84.21	59.49	70.89	74.34	78.13	57.53	68.60	71.57	75.96
ColBERT v2	80.09	86.13	87.79	89.32	74.17	82.55	85.02	86.43	73.06	80.72	83.49	85.45

more pronounced drops in accuracy when the corpus expands, suggesting that domain shift or near-duplicate news articles can degrade dense matching without further adaptation.

5.3 Final QA Accuracy with Retrieved Passages

Beyond simple top-k retrieval accuracy, we also measure how these retrieval methods impact final question answering. Specifically, we feed the top-k passages from each retriever (BM25, DPR, ColBERT v2) into a moderate-scale `Llama-3.1-8B-Instruct` model and evaluate its QA accuracy.

Table Table 5 shows the final QA accuracy (%) across three corpus sizes (1-day, 5-day, 10-day) and various k values. In line with the earlier retrieval results (cf. Figure 3), BM25-based retrieval also yields the highest end-to-end QA performance. For instance, in the 1-day corpus with $k = 1$, BM25 reaches 90.47% whereas DPR and ColBERT v2 yield 66.26% and 80.09%, respectively. When the corpus grows to 10 days, the accuracy drops for all three retrievers, reflecting the increased difficulty of pinpointing the exact relevant article among more documents. Nonetheless, BM25's advantage remains. These findings suggest that in rapidly evolving news scenarios, the strong lexical clues (e.g., named entities, timestamps) may favor exact matching over purely dense retrieval methods, unless the latter are carefully adapted to the domain.

Overall, these results confirm that *accurate retrieval* is vital for time-sensitive QA, perhaps even more so than having a very large model. Even an 8B-parameter Llama achieves high QA accuracy (above 90%) once the correct article is among the retrieved passages. Thus, for fresh or newly breaking news, robust retrieval pipelines can often compensate for the model's limited parametric memory.

6 Conclusion

We introduce a fully automated framework for dynamic knowledge benchmarking, enabling timely and decentralized evaluation of LLMs. Our agentic pipeline generates high-quality, news-driven QA datasets, supporting robust analysis of model knowledge and retrieval performance. Through experiments on a range of open-source and proprietary models, we demonstrate performance disparities on newly introduced knowledge and the benefits of retrieval augmentation. This work highlights the importance of evaluating LLMs on evolving, non-memorized knowledge to better understand and improve their real-world capabilities.

Limitations

While our framework democratizes the creation of *dynamic* knowledge benchmarks, several caveats remain:

- **Domain & Language Bias.** We currently target English-language online news. This excludes non-English, local, pay-walled, or multimedia sources and limits the benchmark's cultural and topical coverage. Extending the pipeline to multilingual or domain-specific corpora (e.g., biomedical literature) will require tailored scraping, prompting, and validation strategies.
- **Dependence on Proprietary LLMs.** Generation, validation, and revision agents rely on proprietary frontier models. Model drift, API quota changes, or access restrictions may affect future reproducibility despite our version-signature protocol. Moreover, researchers without paid API access may face a cost barrier.
- **Legal and Ethical Considerations.** We scrape full-text news articles that remain under copyright. Our release distributes only short excerpts for research under fair-use assumptions, but downstream users bear responsibility for local licensing compliance. Automated harvesting also risks propagating misinformation if upstream outlets publish retracted or false content.

Addressing these limitations remains important future work for making dynamic knowledge evaluation truly global, robust, and sustainable.

References

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. 2018. Ms marco: A human generated machine reading comprehension dataset. *Preprint*, arXiv:1611.09268.

Jeffrey Cheng, Marc Marone, Orion Weller, Dawn Lawrie, Daniel Khashabi, and Benjamin Van Durme. 2024. Dated data: Tracing knowledge cutoffs in large language models. In *First Conference on Language Modeling*.

Hui Dai, Ryan Teehan, and Mengye Ren. 2024. Are llms prescient? a continuous evaluation using daily news as the oracle. *Preprint*, arXiv:2411.08324.

Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. 2025. The faiss library. *Preprint*, arXiv:2401.08281.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *Preprint*, arXiv:1705.03551.

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? *Preprint*, arXiv:2207.13332.

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*, 7:452–466.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. *Preprint*, arXiv:2309.06180.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.

Yucheng Li, Frank Guerin, and Chenghua Lin. 2024. An open source data contamination report for large language models. *Preprint*, arXiv:2310.17589.

Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira. 2021. Pyserini: An easy-to-use python toolkit to support replicable ir research with sparse and dense representations. *Preprint*, arXiv:2102.10073.

Adam Liška, Tomáš Kočiský, Elena Gribovskaya, Tayfun Terzi, Eren Sezener, Devang Agrawal, Cyprien de Masson d'Autume, Tim Scholtes, Manzil Zaheer, Susannah Young, Ellen Gilsenan-McMahon, Sophia Austin, Phil Blunsom, and Angeliki Lazaridou. 2022. Streamingqa: A benchmark for adaptation to new knowledge over time in question answering models. *Preprint*, arXiv:2205.11388.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. Self-refine: Iterative refinement with self-feedback. *Preprint*, arXiv:2303.17651.

Matt Post. 2018. A call for clarity in reporting bleu scores. *Preprint*, arXiv:1804.08771.

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

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. *Preprint*, arXiv:2310.03214.

Shuohang Wang, Yichong Xu, Yuwei Fang, Yang Liu, Siqi Sun, Ruochen Xu, Chenguang Zhu, and Michael Zeng. 2022. Training data is more valuable than you think: A simple and effective method by retrieving from training data. *Preprint*, arXiv:2203.08773.

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. *Preprint*, arXiv:1809.09600.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. *Preprint*, arXiv:2210.03629.

Table of Contents

A Additional Benchmark Details

StreamingQA. Builds a time-indexed dataset from a large news corpus (14 years), enabling retrospective testing of how QA models adapt to new information at specific points in history. Once published, it is no longer updated.

RealTime QA. Scrapes around 30 weekly questions from news quizzes (e.g., CNN, The Week). Offers a rolling evaluation but is constrained by external quiz sources and weekly time slots, rather than daily updates.

FreshQA. Uses a fixed set of around 600 human-written questions whose answers evolve (often involving false premises or rapidly changing facts). Relies on regular human intervention for quality control and updating answers.

Daily Oracle. Automatically generates daily forecasting questions (T/F or multiple-choice) from current news, evaluating models' abilities to predict near-future outcomes. Fully automated, but does not focus on post-event factual retrieval or user-driven updates.

B Prompt for Generating MCQs

```
# News article

**ARTICLE TITLE**:
{article_title}

**ARTICLE TEXT**:
{article_text}

**ARTICLE RELEASE DATE**:
{article_release_date}

# Your task

Generate 5 exceptionally challenging multiple-choice questions based on the article. Follow these requirements:

1. **Question Style**
   - Use a simple, direct tone. For example:
     - "Who was elected president of France in 2022?"
     - "Which country hosted the 2023 Climate Summit?"

2. **Question Content**
   - Each question must focus on factual information about the events or details within the article.
   - Formulate every question so it can be answered exclusively from the provided content.
   - Avoid referencing the article directly (do not use phrases like "According to the article..." or "The text indicates...").
   - For time-sensitive information, incorporate the article's release date. Use as of {article_release_date} when referring to ongoing or current information, or on {article_release_date} when indicating that an event occurred on that specific day.
   - Use explicit identifiers for individuals and organizations (e.g., InfoWars reporter Jamie White), never ambiguous references like the official or his statement.
   - Ensure the question is only answerable if one has access to the article (low no-context accuracy).

3. **Answer Choices**
   - Provide four (4) plausible choices, each of which is the same entity type (person, organization, place, date, number, etc.).
   - The correct answer must be an entity present or derivable from the article.
   - Include distractors that are contextually plausible (either mentioned in the article or logically related).
   - At least one distractor should closely resemble the correct answer to increase difficulty (e.g., a similar name or date).
   - Use partial truths or common misconceptions for other distractors, ensuring all choices appear equally plausible without thorough reading.

4. **Answer Format**
   - Each question must have a single correct answer (entity) that is taken verbatim from the article.
   - The answer must not be open-ended: it should be a specific entity (person, organization, place, time, date, number, etc.).

5. **Question Diversity**
```

```
   - Cover different significant elements or events in the article (avoid repeating the same fact).
   - Use a variety of question types (who, what, when, where, why, how) and difficulty levels, from moderate to very challenging.
   - Aim to require different levels of reasoning (recall, inference, analysis).

6. **Article Release Date** [IMPORTANT]
   - The article includes a release date provided as `{article_release_date}`. Ensure that this date is incorporated appropriately in questions, using  as of {article_release_date} for current or ongoing contexts and  on  {article_release_date} when referencing a specific event or fact that happened that day.

7. **Response Format**
   - Return your final output as a JSON array of exactly 5 objects.
   - Each object must contain the following keys:
     - `"question_idx"`: An integer from 1 to 5.
     - `"question"`: A string containing the question text.
     - `"choices"`: An array of 4 strings, each a distinct answer option.
     - `"ground_truth"`: A string identical to the correct answer choice from `"choices"`.
     - `"rationale"`: A string explaining why the correct choice is correct and why the others are incorrect.

Now generate the JSON array with the specified structure:
```

C Prompt for MCQ Quality Check

```
You are given a multiple-choice
   question in this format:

{qa_pair}

Check if it meets **all** of the
   following requirements:

1. **No direct reference to the
   article**
   - The question does not begin
      or contain phrases like
        According to the
       article or As
      reported in the
       article .

2. **Date references are accurate
    and clear**
   - If the question references
      an event or information
      that took place on a
      specific date, it can
      mention that date directly
      (e.g., on February 25,
      2025 ).
   - If the question references a
       continuing/ongoing
      situation relative to the
       articles publication, it
       should use as of {
      article_release_date} or
        on {
      article_release_date}.
   - The question should not give
       ambiguous timing (e.g.,
         recently without any
      date).

3. **Explicit identifiers for
   individuals or organizations**
   - Any person or group
      mentioned must be named
      clearly (e.g., The
      Transportation Ministry
      instead of They or
        That ministry ).
   - Avoid vague references like
        the company or the
        government if a
      specific entity is known.

4. **No ambiguous references**
   - If referencing a particular
      event, location, or study,
      the question must include
      all critical details known
      (e.g., event date, location
      , or official event name)
      so that its clear which
      event or study is being
      discussed.
   - General phrases like the
      collapse, the
      incident, or the
       study are not acceptable
      . They must include
      identifying details such as
       the location, date, or
      name.

**Output exactly 1 if *all*
   the requirements above are
   met, and 0 otherwise. No
   further explanation or
   commentary.**

\end{Verbatim}

\clearpage
\section{Prompt for MCQ Revision}
\label{app:mcq-revision}
\begin{Verbatim}[breaklines=true]
# The Instruction

Generate 5 exceptionally
   challenging multiple-choice
   questions based on the article
   . Follow these requirements:

1. **Question Content**
   - Each question must focus on
      factual information about
      the events or details
      within the article.
   - Formulate every question so
      it can be answered
      exclusively from the
      provided content.
   - Avoid referencing the
      article directly (do not
      use phrases like "According
```

```
  to the article..." or "The text indicates...").
  - Use explicit identifiers for individuals and organizations (e.g., InfoWars reporter Jamie White), never ambiguous references like the official or his statement.
  - Ensure the question is only answerable if one has access to the article (low no-context accuracy).

2. **Answer Choices**
  - Provide four (4) plausible choices, each of which is the same entity type (person, organization, place, date, number, etc.).
  - The correct answer must be an entity present or derivable from the article.
  - At least one distractor should closely resemble the correct answer to increase difficulty (e.g., a similar name or date).
  - Use partial truths or common misconceptions for other distractors, ensuring all choices appear equally plausible without thorough reading.

3. **Answer Format**
  - Each question must have a single correct answer (entity) that is taken verbatim from the article.
  - The answer must not be open-ended: it should be a specific entity (person, organization, place, time, date, number, etc.).

4. **Article Release Date**
  - The article includes a release date provided as {article_release_date}. Ensure that this date is incorporated appropriately in questions, using as of {article_release_date} for current or ongoing contexts and on {article_release_date} when referencing a specific event or fact that happened that day.


**ARTICLE TITLE**:
{article_title}

**ARTICLE TEXT**:
{article_text}

**ARTICLE RELEASE DATE**:
{article_release_date}

Now generate the JSON array with the specified structure:



# Your generation

{qa_pair}

# Your task

I provide you with one of your generations (one QA pair out of five). Please reflect on this QA pair and evaluate whether it fulfills all the requirements in the instruction. Make the necessary adjustments accordingly, and then send me the revised generation in the same JSON format. Send only the JSON block.
\end{Verbatim}


\clearpage
\section{A Model Generated Q\&A Pair and Revision Task}
\label{app:model-gen-qapair}
\begin{Verbatim}[breaklines=true]
# A model generated Q&A pair
```

```
[
  {
    "question_idx": 4,
    "question": "What was being
       installed on the highway
       bridge on February 25,
       2025, when it collapsed?",
    "choices": [
      "A deck",
      "Concrete pillars",
      "Steel beams",
      "Safety nets"
    ],
    "ground_truth": "A deck",
    "rationale": "Workers were
       installing a deck at the
       time of the collapse. The
       other options are commonly
        used in construction but
       were not mentioned as
       being installed during the
        incident."
  }
]

---

### Your Task
1. Review the generated Q&A pair
   above.
2. Adjust it if it does not
   fulfill all instructions (e.g
   ., date usage, clarity, or
   diversity).
3. Send back the revised Q&A in
   **JSON** format, **and only
   the JSON block**.
```

D Human Annotation Guidelines

D.1 Step 1: Review the Generated Question

Carefully examine the question and its choices.

D.2 Step 2: Check Against Each Requirement

Compare the generated question against all the criteria below. If any criterion is not satisfied, note its requirement number.

1. **Simple, Direct Tone**
   - The question should be concise, clear, and free of convoluted language or indirect phrasing.
2. **No Explicit Article References**
   - Must not contain phrases like "According to the article..." or "The text states...".
3. **Proper Use of Dates**
   - For current/ongoing info: "as of February 26, 2025."
   - For an event that happened on that day: "on February 26, 2025."
   - If the question involves time-sensitive info but omits or misuses these phrases, it fails this requirement.
4. **Explicit Identifiers**
   - Must use specific names (e.g., "Acting President Choi Sang-mok," "National Fire Agency") instead of vague references ("the official," "their statement").

D.3 Step 3: Decide Pass/Fail

1. If all requirements above are satisfied, output: **1**.
2. If one or more requirements are not met, output: **0**.

E Hyperparameters and Implementation Details

We follow standard implementations and use pre-trained checkpoints for each retriever. We use Pyserini's (Lin et al., 2021) implementation of BM25, DPR, and ColBERT v2. We run open-sourced LLMs via vLLM (Kwon et al., 2023). For LLM inference, we use greedy decoding. In the retrieval setting, we concatenate the top-k passages in ascending order of relevance. We do not truncate any retrieved document when feeding it to the LLM. We run all evaluations on a cluster of A6000 GPUs for open-source models, and via the respective hosted APIs for proprietary models.

F Complete Model Benchmarking Results

Table 6 shows the final QA accuracy (%) for a broad range of open-sourced and closed-sourced LLMs under both *No-Context* and *Oracle* settings. As discussed in the main paper, these results highlight the importance of timely context for questions involving fresh, real-world information and illustrate a performance "cutoff" phenomenon for smaller model sizes (e.g., 1B parameters) versus larger ones (e.g., 7B or more). "Oracle" accuracy steadily approaches near-ceiling for models above roughly 3–4B parameters, indicating a scaling threshold for effective reading comprehension on time-sensitive content.

Table 6: Final QA accuracy (%) of open-sourced and closed-sourced LLMs under No-Context and Oracle (Context) settings.

Model	No-Context Acc	Oracle Acc
Open-Sourced Models		
gemma-3-1b-it	31.11	59.06
gemma-3-4b-it	44.17	94.09
gemma-3-12b-it	53.32	95.83
gemma-3-27b-it	54.00	96.21
Llama-3.2-1B-Instruct	26.55	55.06
Llama-3.2-3B-Instruct	42.85	91.57
Llama-3.1-8B-Instruct	30.89	94.81
Llama-3.3-70B-Instruct	57.23	95.70
Phi-3-mini-128k-instruct	44.38	94.30
Phi-4-mini-instruct	43.57	93.62
Qwen2.5-0.5B-Instruct	28.17	55.19
Qwen2.5-1.5B-Instruct	41.70	90.64
Qwen2.5-3B-Instruct	45.36	94.51
Qwen2.5-7B-Instruct	50.00	95.15
Qwen2.5-14B-Instruct	52.89	96.09
Qwen2.5-32B-Instruct	55.79	96.77
Qwen2.5-72B-Instruct	56.30	96.51
Mistral-7B-Instruct-v0.2	35.96	90.21
Mistral-Small-24B-Instruct-2501	53.23	96.43
Mixtral-8x7B-Instruct-v0.1	33.36	93.40
Closed-Sourced Models		
GPT-4o	59.96	96.60
GPT-o1-mini	32.38	96.34
GPT-o3-mini	55.36	97.28
Gemini-1.5-pro	55.36	97.28

G MMLU Pro: Memorized Knowledge Assessment

In Table 7, we report the accuracy of various models on the MMLU Pro benchmark, a knowledge-intensive QA dataset aimed at evaluating factual recall from pre-training. These results offer insight into how well each model retains *static* domain knowledge, in contrast to the *dynamic*, newly emerging facts tested by our daily-updated QA benchmark. We observe that scaling model size often brings significant improvements in MMLU Pro accuracy, reflecting the growing capacity for memorizing factual content. Notably, the performance gains on MMLU Pro can be substantially larger than the gains observed on our fresh-news dataset under No-Context conditions, underscoring the difference between learned "long-term" knowledge and newly introduced facts.

Table 7: **MMLU Pro Results** (% accuracy). We report performance on a knowledge-intensive QA benchmark, reflecting memorized or static knowledge from pre-training.

Model	**Size**	**Accuracy (%)**
`Llama-3.2-1B-Instruct`	1B	22.6
`Llama-3.2-3B-Instruct`	3B	36.5
`Llama-3.1-8B-Instruct`	8B	44.25
`Llama-3.3-70B-Instruct`	70B	65.92
`Gemma-3-1B`	1B	14.7
`Gemma-3-4B`	4B	43.6
`Gemma-3-12B`	12B	60.6
`Gemma-3-27B`	27B	67.5
`Qwen-2.5-0.5B`	0.5B	15.0
`Qwen-2.5-1.5B`	1.5B	32.4
`Qwen-2.5-3B`	3B	43.7
`Qwen-2.5-7B`	7B	56.3
`Qwen-2.5-14B`	14B	63.7
`Qwen-2.5-32B`	32B	69.0
`Qwen-2.5-72B`	72B	71.1

H Additional Retrieval Results

Table 8: Top-k hits accuracy (%) for different retrieval methods across 1-day, 5-day, and 10-day corpora. Each cell represents the fraction of questions for which the ground-truth article is ranked within the top k results.

Retriever	**1-Day Corpus**				**5-Day Corpus**				**10-Day Corpus**			
	Top-1	**Top-3**	**Top-5**	**Top-10**	**Top-1**	**Top-3**	**Top-5**	**Top-10**	**Top-1**	**Top-3**	**Top-5**	**Top-10**
BM25	58.72	69.15	71.28	74.26	44.26	54.47	57.87	62.13	46.38	56.60	60.00	62.13
DPR	41.06	53.40	58.94	64.04	27.45	36.81	40.85	47.87	25.11	36.38	41.28	46.17
ColBERT v2	52.55	61.28	67.02	71.28	38.09	46.17	50.64	56.17	38.09	47.66	51.70	54.89

Table 9: Top-k Mean Reciprocal Rank (MRR) for different retrieval methods across 1-day, 5-day, and 10-day corpora. Each cell represents the average reciprocal rank of the ground-truth article.

Retriever	**1-Day Corpus**				**5-Day Corpus**				**10-Day Corpus**			
	Top-1	**Top-3**	**Top-5**	**Top-10**	**Top-1**	**Top-3**	**Top-5**	**Top-10**	**Top-1**	**Top-3**	**Top-5**	**Top-10**
BM25	0.59	0.63	0.64	0.64	0.44	0.49	0.50	0.50	0.46	0.51	0.52	0.52
DPR	0.41	0.47	0.48	0.49	0.27	0.32	0.32	0.33	0.25	0.30	0.31	0.32
ColBERT v2	0.53	0.56	0.58	0.58	0.38	0.42	0.43	0.43	0.38	0.43	0.43	0.44