BrowseComp-Plus: A More Fair and Transparent Evaluation Benchmark of Deep-Research Agent

Zijian Chen[*,1], **Xueguang Ma**[*,✉,1], **Shengyao Zhuang**[*,2,5], **Ping Nie**[3], **Kai Zou**[3], **Sahel Sharifymoghaddam**[1], **Andrew Liu**[1], **Joshua Green**[1], **Kshama Patel**[1], **Mingyi Su**[1], **Ruoxi Meng**[1], **Yanxi Li**[1], **Haoran Hong**[1], **Xinyu Shi**[1], **Xuye Liu**[1], **Nandan Thakur**[1], **Crystina Zhang**[1], **Luyu Gao**[4], **Wenhu Chen**[1], **Jimmy Lin**[1]

[1]University of Waterloo, [2]CSIRO, [3]Independent,
[4]Carnegie Mellon University, [5]The University of Queensland,

https://texttron.github.io/BrowseComp-Plus/

Abstract

Deep-Research agents, which integrate large language models (LLMs) with search tools, have shown success in improving the effectiveness of handling complex queries that require iterative search planning and reasoning over search results. Evaluations on current benchmarks like BrowseComp relies on black-box live web search APIs, have notable limitations in (1) ***fairness***: dynamic and opaque web APIs hinder fair comparisons and reproducibility of deep research methods; (2) ***transparency***: lack of control over the document corpus makes it difficult to isolate retriever contributions. To address these challenges, we introduce BROWSECOMP-PLUS, a benchmark derived from BrowseComp, employing a fixed, carefully curated corpus. Each query in BROWSECOMP-PLUS includes human-verified supporting documents and mined challenging negatives, enabling controlled experimentation. The benchmark is shown to be effective in distinguishing the performance of deep research systems. For instance, the open-source model Search-R1, when paired with the BM25 retriever, achieves 3.86% accuracy, whereas the GPT-5 achieves 55.9%. Integrating the GPT-5 with the Qwen3-Embedding-8B retriever further enhances its accuracy to 70.1% with fewer search calls. This benchmark allows comprehensive evaluation and disentangled analysis of deep research agents and retrieval methods, fostering insights into retrieval effectiveness, citation accuracy, and context engineering in Deep-Research system.

1 Introduction

Recent benchmarks for evaluating Deep-Research Agents, such as BrowseComp [1], have showcased the impressive capabilities of combining large language models (LLMs) with web search tools in solving complex, reasoning-intensive queries. These benchmarks typically provide sets of queries paired directly with answers, where agents are employed with live web search APIs to retrieve supporting documents in real time [2, 3]. While this approach effectively assesses the end-to-end performance of Deep-Research agents, it introduces several critical limitations that impede systematic analysis and evaluation of individual system components.

- **Fair Comparison of Deep Research Agents.** Current evaluations of deep-research agents often conflate agent system performance with the effectiveness of their retrieval components, making

[*]Equal Contribution. ✉ Correspondence: x93ma@uwaterloo.ca

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

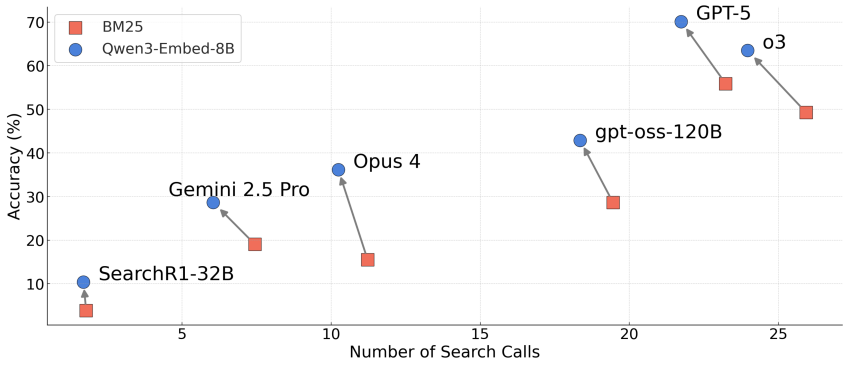

Figure 1: Accuracy vs. number of search calls for Deep-Research agents with different retrievers. GPT-5, o3, gpt-oss are evaluated with high reasoning effort. The figure shows that **Deep-Research agents mostly improve the final accuracy at a cost of more search calls,** whereas **better retrieval systems not only improve the overall accuracy but also reduce the number of search calls**. For reference, GPT-5 achieves 59.9% accuracy when evaluated using the Google Search API.

it difficult to achieve fair and consistent comparisons across systems. This entanglement also severely undermines the reproducibility of experiments, which is a key requirement for rigorous evaluation [4].

- **Transparency of Retrieval Process.** The transparency of the retrieval process comes from two aspects: the retrieval algorithm and the target retrieval corpus. In the current evaluation pipelines, supporting documents are obtained through black-box web search APIs that operate over the entire internet, which are highly dynamic in content and consistently evolving over time. The lack of a controlled retrieval process hinders the evaluation of retrieval models' contribution to deep-research agents.
- **Accessibility**: The dependence on commercial web search APIs introduces substantial practical constraints, including high operational costs and variability in retrieval quality. These issues not only limit accessibility but also introduce unnecessary complexity and uncertainty.

To address these limitations and enable precise, reproducible, transparent, and component-focused evaluation of Deep-Research agents, we introduce BrowseComp-Plus, a new benchmark dataset. BrowseComp-Plus extends the original BrowseComp dataset [1] by providing a fixed and curated corpus of documents specifically selected and verified by human annotators. Each query in BrowseComp-Plus is accompanied by explicitly identified supportive documents and hard negative documents. This carefully collected document corpus allows researchers to evaluate the retrieval and LLM agent components independently, facilitating detailed analysis of each component's impact on the final answer quality. Additionally, by eliminating reliance on dynamic web APIs, BrowseComp-Plus significantly reduces costs, enhances reproducibility, and improves the overall robustness of benchmarking in Deep-Research.

To demonstrate the utility of BrowseComp-Plus, we conduct comprehensive evaluations by pairing various open- and closed-source LLMs with a range of retrieval models on our curated corpus. This setup allows us to systematically analyze how different combinations affect answer quality and to identify where performance bottlenecks lie, whether in the retriever or the language model. We find that even when equipped with state-of-the-art retrievers, Deep-Research agents still face substantial challenges in consistently surfacing all necessary evidence, for reasoning-intensive queries. These findings motivate the need for evaluation frameworks that disentangle retrieval from reasoning, support fine-grained component analysis, and remain fully reproducible.

Furthermore, we extend our evaluation to test retrieval models directly on the original BrowseComp queries, an analysis that was previously infeasible due to the absence of a fixed corpus and grounded relevant document judgments. Our findings reveal that even state-of-the-art retrieval models struggle to retrieve relevant documents for these complex, reasoning-intensive queries.

In summary, our contributions are threefold:

- We present BrowseComp-Plus, a fair and transparent benchmark for Deep-Research Agents, featuring a fixed, human-verified corpus with supporting and challenging negative documents.
- We provide the first systematic analysis of retrieval–agent interactions under controlled conditions, evaluating a broad range of retrievers and LLM-based agents.
- We release all benchmark data, evaluation scripts, and baselines to facilitate reproducible research and foster future advances in various dimensions to improve the deep-research system.

2 Related Works

2.1 Deep-Research Agent

Deep research agents conduct tasks through iterative query reasoning, search planning, and reflection on retrieved results [5], outperforming the traditional single-round retrieval-agumented generation paradigm [6]. Commercial closed-source models such as Gemini [7], Opus [8], and o3 [9], as well as open-source models like GPT-OSS [10], allow access to external retrievers via tool-use APIs or MCP [11]. Recent research works such as Search R1 [12] and WebSailor [13], both based on the Qwen [14] model, leverage reinforcement learning to further enhance search tool capabilities. Fair evaluation of such agents, however, requires a fixed retriever system to make comparisons meaningful.

2.2 Neural Retrieval

Neural retrieval methods, such as Dense Passage Retrieval [15], encode queries and documents into dense vectors using transformer models, and perform retrieval through nearest-neighbor search [16]. These methods have significantly improved retrieval effectiveness compared to traditional lexical-based methods like BM25 [17]. Recent improvements in neural retrievers include advanced training strategies such as continuous pretraining [18, 19], data augmentation [20–22], integration of large language models as backbones [23, 24], and LLM distillation techniques [25, 26]. While retrievers are a critical component of deep research agents, the contribution of different retrievers to the overall performance of these agents remains underexplored.

2.3 Deep Retrieval Benchmarks

Traditional benchmarks such as NaturalQuestions [27] and TriviaQA [28] have significantly contributed to evaluating retrieval and retrieval-augmented generation systems [6, 15, 29]. However, these benchmarks primarily feature single-hop questions, which typically do not require multi-step reasoning or iterative retrieval. Although datasets like HotpotQA [30] offer multi-hop questions, their corpus is limited to Wikipedia. To robustly evaluate deep research systems capable of complex reasoning and strategic search planning, benchmarks requiring sophisticated multi-turn query interactions are essential. BrowseComp [1] stands out as a benchmark explicitly designed for this purpose, offering complex queries paired with verifiable answers. Recent extensions of BrowseComp concepts, such as ZH-BrowseComp [2] and MedBrowseComp [3], further expand to multilingual queries and domain-specific challenges. Existing benchmarks primarily focus on question-answer evaluations of integrated systems without standardized corpora, complicating comparative assessments of retrieval methodologies. BrowseComp-Plus facilitates fair and comprehensive evaluations by providing human-verified corpus, expanding the classic Cranfield paradigm [**?**] to modern deep-research evaluation.

3 BrowseComp-Plus

3.1 Preliminary: BrowseComp

The BrowseComp benchmark contains 1,266 challenging fact-seeking questions specifically designed to assess the capability of Deep-Research AI agents to interactively and creatively navigate the web for complex, hard-to-find information [1]. The questions are deliberately constructed to be difficult for both humans and LLMs, yet they feature verifiable, concise answers, enabling straightforward evaluation through simple answer matching. While effective and widely employed for end-to-end evaluation of integrated deep research systems with web search access, this approach complicates the isolated measurement of retrieval effectiveness within these frameworks.

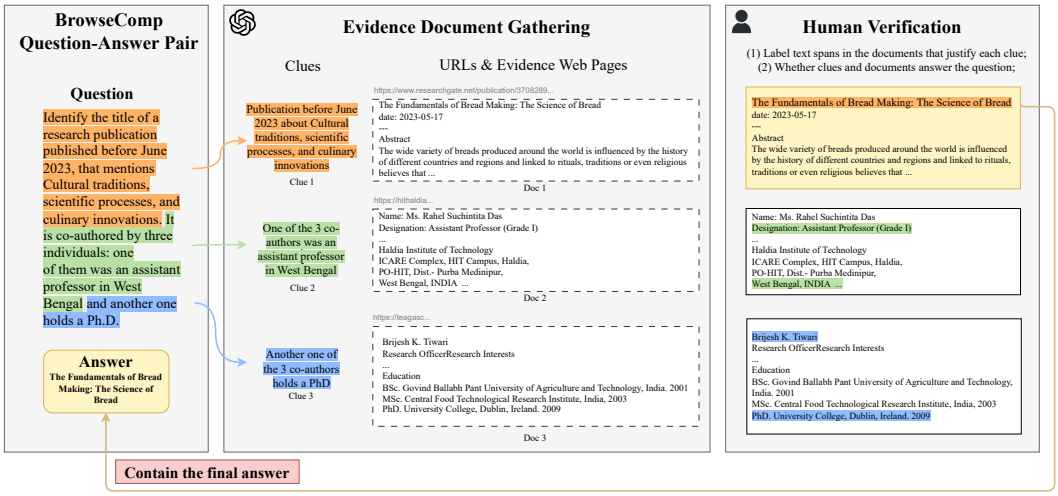

Figure 2: The two-stage pipeline of collecting evidence documents in the corpus (Section 3.2).

3.2 Building the Document Corpus

Constructing a corpus for BrowseComp questions is non-trivial. Three key challenges need to be addressed:

1. **Comprehensive coverage:** The corpus must provide complete evidence to support the entire reasoning chain required to answer each question.
2. **Retrieval difficulty:** The corpus should contain enough distracting negative documents so that search agents and retrievers are challenged in locating the correct evidence.
3. **Practical size:** The corpus should be large enough to yield reliable research insights, while avoiding overly-large computation costs for research purposes.

To meet these criteria, we curate evidence documents through a two-stage pipeline involving automated evidence mining followed by human verification, and perform hard-negative mining via web search to attach challenging, distracting documents to each query. The sections below describe this process in detail and present a 100k-document corpus that effectively supports the study of the Deep Research framework.

3.2.1 Evidence Document Gathering

The original BrowseComp dataset contains only question-answer pairs, without the URLs of the web pages that support these answers. To build a document collection with supporting evidence, the first step involves identifying relevant web pages for each question.

To achieve this, we leverage the OpenAI o3 model with web search enabled. Since the datasets intentionally make direct retrieval of relevant documents difficult, we adopt a *reverse-engineering* strategy: We provide the answer together with the question and instruct the model to search the web for pages that have evidence supporting the answers. We also ask the model to structure the output in a table format with three columns: (1) Clue: the part of the question to address; (2) URL: the web page link containing evidence supporting the clue; and (3) Evidence: the content from the web page that supports the clue. The purpose of this table format is to facilitate human annotators in verifying each clue and its corresponding web page in the next step. An example prompt for this step is provided in Appendix A.

Of the 1,266 original question-answer pairs in BrowseComp, the OpenAI o3 model fails to provide supporting evidence for 124 pairs, either due to output formatting errors or because the model abstains from answering due to low confidence. For the remaining 1,142 pairs, we scrape the URLs cited

as evidence using Selenium,[2] and parse them with Trafilatura [31]. However, a combination of hallucinated URLs and scraping challenges prevents us from successfully scraping all of them. As a result, we exclude 137 question-answer pairs that contain at least one URL where we are unable to scrape, as missing a URL for a clue will make the question incomplete to answer.

This leaves us with 1,005 queries for the next stage: human verification.

3.2.2 Evidence Document Verification

In this stage, we aim to verify that the documents contain sufficient evidence for each clue in the questions. For each question-answer pair, we present human annotators with the output table from OpenAI o3 in the previous stage, with URLs replaced by the corresponding processed documents.

Annotators are asked to:

1. Confirm that each clue is sufficiently justified by the supporting documents. Instead of simply confirming the match, annotators must label the text spans in the documents that justify each clue, as this explicit step encourages high-quality verification.
2. Determine whether the combination of clues and supporting evidence enables a human to answer the *entirety* of the question correctly. For instance, if a query asks for an individual matching five characteristics, all five must be verifiable from the documents.

If the original output from OpenAI o3 fails to meet both criteria, annotators are instructed to revise the clues and search the web for additional supporting documents for at least 20 minutes, before concluding that the desired evidence documents cannot be collected.

In addition to constructing the evidence document set, annotators also label which documents directly contain the final answer; these are designated as *gold documents*. Note that a gold document is not defined merely by containing the ground-truth answer as an exact substring; in some cases, the answer is included in the document in an implicit way. For example, a question might ask for the number of publications by a particular author, with the ground-truth answer being "7". A gold document in this case could be the author's personal webpage listing their publications; while it may not contain the string "7" explicitly, it logically contains the answer. Similarly, there are many cases where the answer appears in the document in a variant form, such as a different date format or a paraphrased phrase, rather than an exact string match. Our goal in constructing the gold document set is to provide a more robust and semantically meaningful alternative to the simple substring-based approach in identifying documents that contain the final answer.

Figure 2 illustrates the complete evidence document collection process. A detailed example, including a screenshot of the labeling interface shown to human annotators, is provided in Appendix B.

For quality control, we sample each annotator's labeled data and cross-validate them among annotators, showing over 80% of agreement on average. Overall, of the 1,005 question-answer pairs from the previous stage, 830 passed human verification. The most common failure mode occurs when the documents provided by OpenAI o3 do not satisfy the two verification criteria, and human annotators are unable to gather sufficient additional evidence within a reasonable effort. In addition to these, we identify and exclude several other categories of problematic cases as detailed in Appendix C.

The entire labeling process involved 14 university student annotators and required over 400 hours of manual effort.

3.3 Hard Negative Mining

To ensure the collected corpus remains a reasonable size while still being challenging enough for search systems to identify correct answers among distracting documents, we mine hard negative documents via web search to form the corpus. This has proven to be effective in evaluating information retrieval systems using a sub-sampled corpus [32, 33].

Specifically, we take each question from BrowseComp and prompt GPT-4o to break it down into simpler, self-contained sub-queries. On average, this results in about seven sub-queries per original query. Each sub-query is then sent to a Google Search API provider (SerpAPI), which returns up to

[2] https://www.selenium.dev/documentation

100 search results. We scrape these results using the same process used for collecting documents during positive example construction. We illustrate this hard negative document collecting process in Figure 4. The prompt used to create these sub-queries is provided in Appendix D.

3.4 Final Corpus Statistics

After deduplicating the positive and negative documents collected as above, we arrive at a corpus of 100,195 documents, along with 830 queries. On average, each query contains 6.1 evidence documents, 76.28 negatives, and 2.9 gold documents. Each document averages 5179.2 words and 32296.2 characters.

4 Experiments

4.1 Experiment Setup

Search Agents We list the agent baseline models in Appendix H.1. To perform agentic search with the LLMs, we provide the LLM with a retriever tool as tool use. We follow the original prompt from BrowseComp [1], which instructs the model to answer a given question along with a confidence estimate (expressed as a percentage). There are two revisions of the original prompts: (1) We explicitly prompt the LLM to use the provided tools to adapt to our custom search tool; (2) We instruct the model to cite the sources when generating the final answer, enabling the evaluation of citation quality. The complete prompt is included in Appendix E. We use this prompt across all models except Search-R1, which uses the prompt aligned with its original fine-tuning.

Retriever We list the retriever baseline models in Appendix H.2. The retriever tool is set to retrieve the top $k = 5$ search results, where each result is truncated to the first 512 token of the corresponding document. This truncation is due to budget constraints, which prevent us from providing full document content. To assess the impact of this design choice, we analyze the distribution of the number of tokens required to include the ground-truth answer for each query. As illustrated in Figure 5 (b), when documents are truncated to the first 512 tokens, 86.5% of queries still contain the ground-truth answer in at least one of their gold documents. Further ablations exploring alternative tool configurations are discussed in Section 4.7.

4.2 Evaluation Metrics

Deep Research Effectiveness We report end-to-end effectiveness of the deep research systems with three metrics: Accuracy, Recall, and Search Calls. Accuracy follows BrowseComp: an LLM-as-judge (GPT-4.1) compares the model's final answer against the ground truth using the evaluation prompt listed in Appendix F. Recall measures how many human-verified evidence documents the agent retrieved during its entire interaction. Search Calls is the average number of search API invocations per query. In addition, following BrowseComp, we compute calibration error using the confidence estimates produced by the search agents, in the same way as Humanity's Last Exam [34], measuring how closely a model's predicted confidence matches the actual accuracy of its predictions. For Search-R1, we do not report calibration error because the input and output format of this model are fixed without a confidence source output.

Retrieval Effectiveness For evaluating retriever effectiveness, our BrowseComp-Plus benchmark provides human-verified evidence documents and gold documents, along with a fixed test document collection, enabling evaluation under the Cranfield paradigm [4]. Specifically, we follow standard TREC practice to create a query-document relevance label file[3] for both evidence documents and gold documents separately, and then compute Recall@k and nDCG@k to assess the effectiveness of retrievers.

4.3 End-to-End Deep-Research Performance

Table 1 summarizes the overall Deep-Research Performance across different LLMs and retrievers. Proprietary models (GPT-4.1, o3, GPT-5, Sonnet-4, Opus-4, Gemini) demonstrate high answer

[3] Known as a qrel file.

Table 1: End-to-end agent accuracy on BrowseComp-Plus across LLMs and retrievers. All agents are prompted with the same tool-use prompt, except for Search-R1, which uses the prompt identical to its training.

LLM	Retriever	Accuracy	Recall	Search Calls	Calibration Error
GPT-4.1	BM25	14.58%	16.42%	10.35	68.96%
	Qwen3-Embed-8B	35.42%	36.89%	8.67	54.67%
o3	BM25	49.28%	56.64%	25.93	12.58%
	Qwen3-Embed-8B	63.49%	73.24%	23.97	16.77%
GPT-5	BM25	55.90%	61.70%	23.23	13.50%
	Qwen3-Embed-8B	70.12%	78.98%	21.74	9.11%
Sonnet 4	BM25	14.34%	21.31%	9.95	29.79%
	Qwen3-Embed-8B	36.75%	47.33%	9.03	24.51%
Opus 4	BM25	15.54%	22.96%	11.22	22.00%
	Qwen3-Embed-8B	36.14%	50.84%	10.24	12.79%
Gemini 2.5 Flash	BM25	15.54%	21.45%	10.56	29.28%
	Qwen3-Embed-8B	33.01%	40.19%	9.77	21.63%
Gemini 2.5 Pro	BM25	19.04%	22.81%	7.44	51.58%
	Qwen3-Embed-8B	28.67%	35.31%	6.04	44.08%
gpt-oss-120B-high	BM25	28.67%	35.50%	19.45	46.48%
	Qwen3-Embed-8B	42.89%	52.63%	18.35	40.34%
Qwen3-32B	BM25	3.49%	3.12%	0.92	57.41%
	Qwen3-Embed-0.6B	4.10%	3.45%	0.91	60.71%
	Qwen3-Embed-4B	7.83%	6.20%	0.89	61.06%
	Qwen3-Embed-8B	10.36%	7.80%	0.94	59.84%
	ReasonIR	9.16%	7.59%	0.91	55.15%
SearchR1-32B	BM25	3.86%	2.61%	1.78	*N/A*
	Qwen3-Embed-0.6B	5.66%	5.30%	1.73	*N/A*
	Qwen3-Embed-4B	9.40%	7.90%	1.68	*N/A*
	Qwen3-Embed-8B	10.36%	10.17%	1.69	*N/A*
	ReasonIR	9.43%	8.37%	1.74	*N/A*

accuracy, with OpenAI's GPT-5 achieving the highest accuracy (70.12%) when paired with the Qwen3-Embedding-8B retriever. Open-source models such as Qwen3-32B and SearchR1-32B lag behind proprietary models. With Qwen3-Embedding-8B as the retriever, Qwen3-32B achieves only 10.36% accuracy, compared to 35.42% for GPT-4.1 and 63.49% for o3. Notably, the only high-performing open-source model we studied is gpt-oss-120B in its high reasoning mode, which achieves 42.89% accuracy, surpassing Opus 4 when both are paired with Qwen3-Embedding-8B.

In general, closed-source agents call the search tool more frequently than open-source models. For instance, OpenAI's GPT-5 and o3 issue an average of more than 20 search calls per query, while Qwen3-32B and SearchR1-32B make fewer than 2, despite being explicitly prompted to use the tool. This reflects a test-time scaling effect: more exhaustive search correlates with better outcomes and aligns with prior findings that reasoning-intensive queries benefit from exploratory retrieval.

4.4 Effect of Retrieval Quality

A consistent trend observed across all models is that stronger retrieval leads to higher final accuracy. First, consider the retriever's effectiveness on our dataset. We evaluate retrieval performance using the original full queries, with results shown in Table 2. Compared to BM25, Qwen3-Embedding-8B and ReasonIR-8B achieve substantially higher recall and nDCG for both evidence document retrieval and gold document retrieval. Notably, we observe a model size scaling law within the Qwen3 embedding family; larger models consistently perform better, with Qwen3-Embedding-8B surpassing ReasonIR-8B at the 8B scale.

Now, as indicated in Table 1, replacing the BM25 retriever with a stronger retriever leads to significant accuracy gains across all LLM agents. For instance, OpenAI's GPT-5's accuracy improves from 55.9% to 70.12%, while Sonnet 4 and Opus 4 both achieve more than double their BM25 accuracy. This suggests a strong positive correlation between retrieval effectiveness and research agent accuracy.

Moreover, stronger retrievers potentially reduce the number of search calls. For most proprietary models, Qwen3-Embedding-8B reduces search calls by approximately 1–3 compared to BM25. This

Table 2: Effectiveness of retrievers. The complete question is used as the query for all retrieval methods for fair comparison.

Retriever	Recall@5	Recall@100	Recall@1000	nDCG@10
Evidence Document Retrieval				
BM25	1.2	4.7	13.7	1.6
jina-colbert-v2	5.7	18.1	35.7	7.9
Qwen3-Embed-0.6B	6.2	26.5	59.7	8.0
Qwen3-Embed-4B	9.8	40.2	71.8	14.0
Qwen3-Embed-8B	14.5	47.7	76.7	20.3
ReasonIR-8B	12.2	43.6	73.9	16.8
Gold Document Retrieval				
BM25	1.4	6.1	17.3	1.7
jina-colbert-v2	6.6	20.4	39.7	6.8
Qwen3-Embed-0.6B	8.5	30.5	66.2	7.4
Qwen3-Embed-4B	13.0	47.3	77.0	13.6
Qwen3-Embed-8B	18.5	55.8	83.5	19.5
ReasonIR-8B	15.3	49.7	78.9	15.5

shows that better retrieval not only improves effectiveness (accuracy) but also efficiency (fewer tool calls). In Appendix K, we also report differences in proprietary agent API costs when using different retrievers. Agents using Qwen3-Embedding-8B incur lower costs due to fewer input and output tokens, further supporting the efficiency gains enabled by stronger retrieval.

In addition, Appendix I reports the coverage, average number, precision, and recall of the document citations attributed by the agent during answer generation, highlighting opportunities for future improvements in long-answer generation with proper evidence attribution via citations. We also assess the role of LLM rerankers as part of the retrieval module in Appendix J, showing the potential of further improving the effectiveness of the deep-research systems through reranking.

4.5 Oracle Retrieval

We evaluate effectiveness in an extreme oracle setting, where search agents are prompted with all labeled positive documents to answer the questions. In this setup, GPT-4.1 achieves an accuracy of 93.49%. This highlights two key points. First, it showcases the importance of the retriever: if the retriever is of perfect quality, search agents can attain substantially high accuracy on complex reasoning tasks in BrowseComp-Plus, in contrast to the 14.58% baseline accuracy of GPT-4.1 when using BM25 as the retriever. Second, it validates the quality of the BrowseComp-Plus corpus itself: GPT-4.1, a non-reasoning model, is able to correctly answer 93.49% of questions using only the evidence documents in the corpus. For the remaining 6.51% of cases, human annotators reviewed each instance and confirmed that the answers are indeed answerable from the positive documents; the errors stem solely from GPT-4.1's failure to reason correctly.

4.6 Impact of Reasoning Effort

We evaluate how the reasoning effort of LLMs influences answer quality and retrieval behavior. To isolate this effect, we focus on the gpt-oss family, which offers three reasoning modes: *low*, *medium*, and *high*. As shown in Table 3, increasing the reasoning effort leads to substantial improvements in accuracy and recall across all model sizes and retrievers. For example, gpt-oss-20B with Qwen3-Embed-8B improves from 13.37% accuracy in *low* mode to 34.58% in *high* mode, along with a recall jump from 17.37% to 49.29%. Similarly, gpt-oss-120B with Qwen3-Embed-8B's accuracy rises from 24.94% to 42.89%. These gains, however, come with a trade-off: higher reasoning modes dramatically increase the average number of search calls (e.g., from $\approx$2 to $\approx$24 for gpt-oss-20B with Qwen3-Embed-8B), implying higher computational and latency costs. Interestingly, calibration error tends to decrease with higher reasoning effort, suggesting that the models become more aligned between confidence and correctness as they reason more extensively.

Table 3: OpenAI gpt-oss models in different reasoning effort settings

LLM	Retriever	Accuracy	Recall	Search Calls	Calibration Error
gpt-oss-20B-low	BM25	4.11%	5.36%	1.89	40.89%
	Qwen3-Embed-8B	13.37%	17.37%	1.87	36.34%
gpt-oss-20B-medium	BM25	16.39%	21.96%	13.72	41.78%
	Qwen3-Embed-8B	29.88%	41.31%	13.64	35.99%
gpt-oss-20B-high	BM25	21.08%	31.98%	26.87	33.42%
	Qwen3-Embed-8B	34.58%	49.29%	23.87	27.81%
gpt-oss-120B-low	BM25	9.52%	8.54%	2.06	43.59%
	Qwen3-Embed-8B	24.94%	22.50%	2.21	40.96%
gpt-oss-120B-medium	BM25	23.73%	27.02%	9.73	45.78%
	Qwen3-Embed-8B	37.59%	43.45%	9.64	41.77%
gpt-oss-120B-high	BM25	28.67%	35.50%	19.45	46.48%
	Qwen3-Embed-8B	42.89%	52.63%	18.35	40.34%

Table 4: Comparison of Qwen3-32B and GPT-4.1 with and without get-document tool, using Qwen3-Embedding-8B as retriever.

Model	Accuracy	Search Calls	Get Document Calls	Calibration Error
GPT-4.1	35.42%	8.67	N/A	54.67%
GPT-4.1 + get-doc	43.61%	10.03	1.85	54.28%
Qwen3-32B	10.36%	0.94	N/A	59.84%
Qwen3-32B + get-doc	11.69%	1.01	0.27	56.47%

4.7 Effect of Document Reading Strategy

In previous experiments, we always presented only the first 512 tokens of each retrieved document as a preview to the LLM during each round of search and reasoning, due to token budget constraints. However, in realistic deep research scenarios, agents often have access to a document reader tool that enables reading the full content of a document. To evaluate the potential benefit of such a tool, we conduct experiments with GPT-4.1 and Qwen3-32B, both with and without access to a whole-document reader (referred to as the get-document tool). Appendix G contains the revised prompt used when the get-document tool is added.

Results are shown in Table 4. For GPT-4.1, enabling the get-document tool improves accuracy from 35.42% to 43.61%, with a modest increase in search calls (from 8.67 to 10.03) and an average of 1.85 full-document reads per query, confirming that full-document access provides additional useful context that enhances decision-making. For Qwen3-32B, which performs worse overall, the benefit is more modest. Accuracy improves slightly from 10.36% to 11.69%, and the number of get-document calls remains low (0.27 per query on average). This suggests that while the tool can help, the model's limited reasoning and tool-use ability constrain its ability to exploit the additional information.

5 Conclusion

We introduced BrowseComp-Plus, a new benchmark designed to address the reproducibility, fairness, and transparency challenges in evaluating Deep-Research Agents. By grounding each query in a fixed, human-verified corpus containing both positive and hard-negative documents, our framework enables the independent and controlled assessment of retrieval and agent components.

Through extensive experiments we demonstrate that retrieval quality substantially impacts both the effectiveness and efficiency of deep research systems. BrowseComp-Plus provides a robust platform for probing these dynamics and paves the way for future research on co-optimizing retrievers and agents, improving out-of-distribution tool-use generalization, and advancing context engineering frameworks. By making our benchmark and baselines publicly available, we aim to catalyze the next generation of Deep-Research systems.

References

[1] Jason Wei, Zhiqing Sun, Spencer Papay, Scott McKinney, Jeffrey Han, Isa Fulford, Hyung Won Chung, Alex Tachard Passos, William Fedus, and Amelia Glaese. Browsecomp: A simple yet challenging benchmark for browsing agents. *arXiv:2504.12516*, 2025. URL https://arxiv.org/abs/2504.12516.

[2] Peilin Zhou, Bruce Leon, Xiang Ying, Can Zhang, Yifan Shao, Qichen Ye, Dading Chong, Zhiling Jin, Chenxuan Xie, Meng Cao, Yuxin Gu, Sixin Hong, Jing Ren, Jian Chen, Chao Liu, and Yining Hua. Browsecomp-zh: Benchmarking web browsing ability of large language models in chinese. *arXiv:2504.19314*, 2025. URL https://arxiv.org/abs/2504.19314.

[3] Shan Chen, Pedro Moreira, Yuxin Xiao, Sam Schmidgall, Jeremy Warner, Hugo Aerts, Thomas Hartvigsen, Jack Gallifant, and Danielle S. Bitterman. Medbrowsecomp: Benchmarking medical deep research and computer use. *arXiv:2505.14963*, 2025. URL https://arxiv.org/abs/2505.14963.

[4] Ellen M. Voorhees. *The Evolution of Cranfield*, pages 45–69. Springer International Publishing, Cham, 2019. ISBN 978-3-030-22948-1. doi: 10.1007/978-3-030-22948-1_2. URL https://doi.org/10.1007/978-3-030-22948-1_2.

[5] Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=hSyW5go0v8.

[6] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

[7] Gemini 2.5 Team. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities, 2025. URL https://arxiv.org/abs/2507.06261.

[8] Anthropic Team. The claude 3 model family: Opus, sonnet, haiku. 2024. URL https://www-cdn.anthropic.com/de8ba9b01c9ab7cbabf5c33b80b7bbc618857627/Model_Card_Claude_3.pdf.

[9] OpenAI Team. OpenAI o3 and o4-mini system card. 2025. URL https://cdn.openai.com/o3-mini-system-card-feb10.pdf.

[10] OpenAI Team. GPT-OSS-120B & 20B model card. 2025. URL https://cdn.openai.com/pdf/419b6906-9da6-406c-a19d-1bb078ac7637/oai_gpt-oss_model_card.pdf.

[11] Anthropic Team. Introducing the model context protocol. November 2024. URL https://www.anthropic.com/news/model-context-protocol.

[12] Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *arXiv:2503.09516*, 2025. URL https://arxiv.org/abs/2503.09516.

[13] Kuan Li, Zhongwang Zhang, Huifeng Yin, Liwen Zhang, Litu Ou, Jialong Wu, Wenbiao Yin, Baixuan Li, Zhengwei Tao, Xinyu Wang, Weizhou Shen, Junkai Zhang, Dingchu Zhang, Xixi Wu, Yong Jiang, Ming Yan, Pengjun Xie, Fei Huang, and Jingren Zhou. Websailor: Navigating super-human reasoning for web agent. *arXiv:2507.02592*, 2025. URL https://arxiv.org/abs/2507.02592.

[14] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin

Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report. *arXiv:2505.09388*, 2025. URL https://arxiv.org/abs/2505.09388.

[15] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.550. URL https://aclanthology.org/2020.emnlp-main.550/.

[16] Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. The faiss library. *arXiv:2401.08281*, 2024.

[17] Stephen E. Robertson. Okapi at trec-3. In *Proceedings of the Third Text REtrieval Conference (TREC-3)*. NIST, 1994.

[18] Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation, 2024.

[19] Luyu Gao and Jamie Callan. Unsupervised corpus aware language model pre-training for dense passage retrieval. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2843–2853, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.203. URL https://aclanthology.org/2022.acl-long.203/.

[20] Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. Towards general text embeddings with multi-stage contrastive learning. *arXiv:2308.03281*, 2023. URL https://arxiv.org/abs/2308.03281.

[21] Xueguang Ma, Xi Victoria Lin, Barlas Oguz, Jimmy Lin, Wen-tau Yih, and Xilun Chen. DRAMA: Diverse augmentation from large language models to smaller dense retrievers. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 30170–30186, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.1457. URL https://aclanthology.org/2025.acl-long.1457/.

[22] Rulin Shao, Rui Qiao, Varsha Kishore, Niklas Muennighoff, Xi Victoria Lin, Daniela Rus, Bryan Kian Hsiang Low, Sewon Min, Wen tau Yih, Pang Wei Koh, and Luke Zettlemoyer. Reasonir: Training retrievers for reasoning tasks. *arXiv:2504.20595*, 2025. URL https://arxiv.org/abs/2504.20595.

[23] Xueguang Ma, Liang Wang, Nan Yang, Furu Wei, and Jimmy Lin. Fine-tuning llama for multi-stage text retrieval. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '24, page 2421–2425, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400704314. doi: 10.1145/3626772.3657951. URL https://doi.org/10.1145/3626772.3657951.

[24] Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. Improving text embeddings with large language models. *arXiv:2401.00368*, 2023.

[25] Jinhyuk Lee, Zhuyun Dai, Xiaoqi Ren, Blair Chen, Daniel Cer, Jeremy R. Cole, Kai Hui, Michael Boratko, Rajvi Kapadia, Wen Ding, Yi Luan, Sai Meher Karthik Duddu, Gustavo Hernandez Abrego, Weiqiang Shi, Nithi Gupta, Aditya Kusupati, Prateek Jain, Siddhartha Reddy Jonnalagadda, Ming-Wei Chang, and Iftekhar Naim. Gecko: Versatile text embeddings distilled from large language models. *arXiv:2403.20327*, 2024. URL https://arxiv.org/abs/2403.20327.

[26] Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv:2506.05176*, 2025. URL https://arxiv.org/abs/2506.05176.

[27] 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. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466, 2019. doi: 10.1162/tacl_a_00276. URL https://aclanthology.org/Q19-1026/.

[28] Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In Regina Barzilay and Min-Yen Kan, editors, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1147. URL https://aclanthology.org/P17-1147/.

[29] Xi Victoria Lin, Xilun Chen, Mingda Chen, Weijia Shi, Maria Lomeli, Richard James, Pedro Rodriguez, Jacob Kahn, Gergely Szilvasy, Mike Lewis, Luke Zettlemoyer, and Wen tau Yih. RA-DIT: Retrieval-augmented dual instruction tuning. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=22OTbutug9.

[30] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1259. URL https://aclanthology.org/D18-1259/.

[31] Adrien Barbaresi. Trafilatura: A Web Scraping Library and Command-Line Tool for Text Discovery and Extraction. In *Proceedings of the Joint Conference of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 122–131. Association for Computational Linguistics, 2021. URL https://aclanthology.org/2021.acl-demo.15.

[32] Maik Fröbe, Andrew Parry, Harrisen Scells, Shuai Wang, Shengyao Zhuang, Guido Zuccon, Martin Potthast, and Matthias Hagen. Corpus subsampling: Estimating the effectiveness of neural retrieval models on large corpora. In *Advances in Information Retrieval: 47th European Conference on Information Retrieval, ECIR 2025, Lucca, Italy, April 6–10, 2025, Proceedings, Part I*, page 453–471, Berlin, Heidelberg, 2025. Springer-Verlag. ISBN 978-3-031-88707-9. doi: 10.1007/978-3-031-88708-6_29. URL https://doi.org/10.1007/978-3-031-88708-6_29.

[33] Shengyao Zhuang and Guido Zuccon. Asyncval: A toolkit for asynchronously validating dense retriever checkpoints during training. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '22, page 3235–3239, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450387323. doi: 10.1145/3477495.3531658. URL https://doi.org/10.1145/3477495.3531658.

[34] Long Phan, Alice Gatti, Ziwen Han, Nathaniel Li, Josephina Hu, Hugh Zhang, Chen Bo Calvin Zhang, Mohamed Shaaban, John Ling, Sean Shi, Michael Choi, Anish Agrawal, Arnav Chopra, Adam Khoja, Ryan Kim, Richard Ren, Jason Hausenloy, Oliver Zhang, Mantas Mazeika, Dmitry Dodonov, Tung Nguyen, Jaeho Lee, Daron Anderson, Mikhail Doroshenko, Alun Cennyth Stokes, Mobeen Mahmood, Oleksandr Pokutnyi, Oleg Iskra, Jessica P. Wang, John-Clark Levin, Mstyslav Kazakov, Fiona Feng, Steven Y. Feng, Haoran Zhao, Michael Yu, Varun Gangal, Chelsea Zou, Zihan Wang, Serguei Popov, Robert Gerbicz, Geoff Galgon, Johannes Schmitt, Will Yeadon, Yongki Lee, Scott Sauers, Alvaro Sanchez, Fabian Giska, Marc Roth, Søren Riis, Saiteja Utpala, Noah Burns, Gashaw M. Goshu, Mohinder Maheshbhai Naiya, Chidozie Agu, Zachary Giboney, Antrell Cheatom, Francesco Fournier-Facio, Sarah-Jane Crowson, Lennart Finke, Zerui Cheng, Jennifer Zampese, Ryan G. Hoerr, Mark Nandor, Hyunwoo Park, Tim

Gehrunger, Jiaqi Cai, Ben McCarty, Alexis C Garretson, Edwin Taylor, Damien Sileo, Qiuyu Ren, Usman Qazi, Lianghui Li, Jungbae Nam, John B. Wydallis, Pavel Arkhipov, Jack Wei Lun Shi, Aras Bacho, Chris G. Willcocks, Hangrui Cao, Sumeet Motwani, Emily de Oliveira Santos, Johannes Veith, Edward Vendrow, Doru Cojoc, Kengo Zenitani, Joshua Robinson, Longke Tang, Yuqi Li, Joshua Vendrow, Natanael Wildner Fraga, Vladyslav Kuchkin, Andrey Pupasov Maksimov, Pierre Marion, Denis Efremov, Jayson Lynch, Kaiqu Liang, Aleksandar Mikov, Andrew Gritsevskiy, Julien Guillod, Gözdenur Demir, Dakotah Martinez, Ben Pageler, Kevin Zhou, Saeed Soori, Ori Press, Henry Tang, Paolo Rissone, Sean R. Green, Lina Brüssel, Moon Twayana, Aymeric Dieuleveut, Joseph Marvin Imperial, Ameya Prabhu, Jinzhou Yang, Nick Crispino, Arun Rao, Dimitri Zvonkine, Gabriel Loiseau, Mikhail Kalinin, Marco Lukas, Ciprian Manolescu, Nate Stambaugh, Subrata Mishra, Tad Hogg, Carlo Bosio, Brian P Coppola, Julian Salazar, Jaehyeok Jin, Rafael Sayous, Stefan Ivanov, Philippe Schwaller, Shaipranesh Senthilkuma, Andres M Bran, Andres Algaba, Kelsey Van den Houte, Lynn Van Der Sypt, Brecht Verbeken, David Noever, Alexei Kopylov, Benjamin Myklebust, Bikun Li, Lisa Schut, Evgenii Zheltonozhskii, Qiaochu Yuan, Derek Lim, Richard Stanley, Tong Yang, John Maar, Julian Wykowski, Martí Oller, Anmol Sahu, Cesare Giulio Ardito, Yuzheng Hu, Ariel Ghislain Kemogne Kamdoum, Alvin Jin, Tobias Garcia Vilchis, Yuexuan Zu, Martin Lackner, James Koppel, Gongbo Sun, Daniil S. Antonenko, Steffi Chern, Bingchen Zhao, Pierrot Arsene, Joseph M Cavanagh, Daofeng Li, Jiawei Shen, Donato Crisostomi, Wenjin Zhang, Ali Dehghan, Sergey Ivanov, David Perrella, Nurdin Kaparov, Allen Zang, Ilia Sucholutsky, Arina Kharlamova, Daniil Orel, Vladislav Poritski, Shalev Ben-David, Zachary Berger, Parker Whitfill, Michael Foster, Daniel Munro, Linh Ho, Shankar Sivarajan, Dan Bar Hava, Aleksey Kuchkin, David Holmes, Alexandra Rodriguez-Romero, Frank Sommerhage, Anji Zhang, Richard Moat, Keith Schneider, Zakayo Kazibwe, Don Clarke, Dae Hyun Kim, Felipe Meneguitti Dias, Sara Fish, Veit Elser, Tobias Kreiman, Victor Efren Guadarrama Vilchis, Immo Klose, Ujjwala Anantheswaran, Adam Zweiger, Kaivalya Rawal, Jeffery Li, Jeremy Nguyen, Nicolas Daans, Haline Heidinger, Maksim Radionov, Václav Rozhoň, Vincent Ginis, Christian Stump, Niv Cohen, Rafał Poświata, Josef Tkadlec, Alan Goldfarb, Chenguang Wang, Piotr Padlewski, Stanislaw Barzowski, Kyle Montgomery, Ryan Stendall, Jamie Tucker-Foltz, Jack Stade, T. Ryan Rogers, Tom Goertzen, Declan Grabb, Abhishek Shukla, Alan Givré, John Arnold Ambay, Archan Sen, Muhammad Fayez Aziz, Mark H Inlow, Hao He, Ling Zhang, Younesse Kaddar, Ivar Ängquist, Yanxu Chen, Harrison K Wang, Kalyan Ramakrishnan, Elliott Thornley, Antonio Terpin, Hailey Schoelkopf, Eric Zheng, Avishy Carmi, Ethan D. L. Brown, Kelin Zhu, Max Bartolo, Richard Wheeler, Martin Stehberger, Peter Bradshaw, JP Heimonen, Kaustubh Sridhar, Ido Akov, Jennifer Sandlin, Yury Makarychev, Joanna Tam, Hieu Hoang, David M. Cunningham, Vladimir Goryachev, Demosthenes Patramanis, Michael Krause, Andrew Redenti, David Aldous, Jesyin Lai, Shannon Coleman, Jiangnan Xu, Sangwon Lee, Ilias Magoulas, Sandy Zhao, Ning Tang, Michael K. Cohen, Orr Paradise, Jan Hendrik Kirchner, Maksym Ovchynnikov, Jason O. Matos, Adithya Shenoy, Michael Wang, Yuzhou Nie, Anna Sztyber-Betley, Paolo Faraboschi, Robin Riblet, Jonathan Crozier, Shiv Halasyamani, Shreyas Verma, Prashant Joshi, Eli Meril, Ziqiao Ma, Jérémy Andréoletti, Raghav Singhal, Jacob Platnick, Volodymyr Nevirkovets, Luke Basler, Alexander Ivanov, Seri Khoury, Nils Gustafsson, Marco Piccardo, Hamid Mostaghimi, Qijia Chen, Virendra Singh, Tran Quoc Khánh, Paul Rosu, Hannah Szlyk, Zachary Brown, Himanshu Narayan, Aline Menezes, Jonathan Roberts, William Alley, Kunyang Sun, Arkil Patel, Max Lamparth, Anka Reuel, Linwei Xin, Hanmeng Xu, Jacob Loader, Freddie Martin, Zixuan Wang, Andrea Achilleos, Thomas Preu, Tomek Korbak, Ida Bosio, Fereshteh Kazemi, Ziye Chen, Biró Bálint, Eve J. Y. Lo, Jiaqi Wang, Maria Inês S. Nunes, Jeremiah Milbauer, M Saiful Bari, Zihao Wang, Behzad Ansarinejad, Yewen Sun, Stephane Durand, Hossam Elgnainy, Guillaume Douville, Daniel Tordera, George Balabanian, Hew Wolff, Lynna Kvistad, Hsiaoyun Milliron, Ahmad Sakor, Murat Eron, Andrew Favre D. O., Shailesh Shah, Xiaoxiang Zhou, Firuz Kamalov, Sherwin Abdoli, Tim Santens, Shaul Barkan, Allison Tee, Robin Zhang, Alessandro Tomasiello, G. Bruno De Luca, Shi-Zhuo Looi, Vinh-Kha Le, Noam Kolt, Jiayi Pan, Emma Rodman, Jacob Drori, Carl J Fossum, Niklas Muennighoff, Milind Jagota, Ronak Pradeep, Honglu Fan, Jonathan Eicher, Michael Chen, Kushal Thaman, William Merrill, Moritz Firsching, Carter Harris, Stefan Ciobâcă, Jason Gross, Rohan Pandey, Ilya Gusev, Adam Jones, Shashank Agnihotri, Pavel Zhelnov, Mohammadreza Mofayezi, Alexander Piperski, David K. Zhang, Kostiantyn Dobarskyi, Roman Leventov, Ignat Soroko, Joshua Duersch, Vage Taamazyan, Andrew Ho, Wenjie Ma, William Held, Ruicheng Xian, Armel Randy Zebaze, Mohanad Mohamed, Julian Noah Leser, Michelle X Yuan, Laila Yacar, Johannes

Lengler, Katarzyna Olszewska, Claudio Di Fratta, Edson Oliveira, Joseph W. Jackson, Andy Zou, Muthu Chidambaram, Timothy Manik, Hector Haffenden, Dashiell Stander, Ali Dasouqi, Alexander Shen, Bita Golshani, David Stap, Egor Kretov, Mikalai Uzhou, Alina Borisovna Zhidkovskaya, Nick Winter, Miguel Orbegozo Rodriguez, Robert Lauff, Dustin Wehr, Colin Tang, Zaki Hossain, Shaun Phillips, Fortuna Samuele, Fredrik Ekström, Angela Hammon, Oam Patel, Faraz Farhidi, George Medley, Forough Mohammadzadeh, Madellene Peñaflor, Haile Kassahun, Alena Friedrich, Rayner Hernandez Perez, Daniel Pyda, Taom Sakal, Omkar Dhamane, Ali Khajegili Mirabadi, Eric Hallman, Kenchi Okutsu, Mike Battaglia, Mohammad Maghsoudimehrabani, Alon Amit, Dave Hulbert, Roberto Pereira, Simon Weber, Handoko, Anton Peristyy, Stephen Malina, Mustafa Mehkary, Rami Aly, Frank Reidegeld, Anna-Katharina Dick, Cary Friday, Mukhwinder Singh, Hassan Shapourian, Wanyoung Kim, Mariana Costa, Hubeyb Gurdogan, Harsh Kumar, Chiara Ceconello, Chao Zhuang, Haon Park, Micah Carroll, Andrew R. Tawfeek, Stefan Steinerberger, Daattavya Aggarwal, Michael Kirchhof, Linjie Dai, Evan Kim, Johan Ferret, Jainam Shah, Yuzhou Wang, Minghao Yan, Krzysztof Burdzy, Lixin Zhang, Antonio Franca, Diana T. Pham, Kang Yong Loh, Joshua Robinson, Abram Jackson, Paolo Giordano, Philipp Petersen, Adrian Cosma, Jesus Colino, Colin White, Jacob Votava, Vladimir Vinnikov, Ethan Delaney, Petr Spelda, Vit Stritecky, Syed M. Shahid, Jean-Christophe Mourrat, Lavr Vetoshkin, Koen Sponselee, Renas Bacho, Zheng-Xin Yong, Florencia de la Rosa, Nathan Cho, Xiuyu Li, Guillaume Malod, Orion Weller, Guglielmo Albani, Leon Lang, Julien Laurendeau, Dmitry Kazakov, Fatimah Adesanya, Julien Portier, Lawrence Hollom, Victor Souza, Yuchen Anna Zhou, Julien Degorre, Yiğit Yalın, Gbenga Daniel Obikoya, Rai, Filippo Bigi, M. C. Boscá, Oleg Shumar, Kaniuar Bacho, Gabriel Recchia, Mara Popescu, Nikita Shulga, Ngefor Mildred Tanwie, Thomas C. H. Lux, Ben Rank, Colin Ni, Matthew Brooks, Alesia Yakimchyk, Huanxu, Liu, Stefano Cavalleri, Olle Häggström, Emil Verkama, Joshua Newbould, Hans Gundlach, Leonor Brito-Santana, Brian Amaro, Vivek Vajipey, Rynaa Grover, Ting Wang, Yosi Kratish, Wen-Ding Li, Sivakanth Gopi, Andrea Caciolai, Christian Schroeder de Witt, Pablo Hernández-Cámara, Emanuele Rodolà, Jules Robins, Dominic Williamson, Vincent Cheng, Brad Raynor, Hao Qi, Ben Segev, Jingxuan Fan, Sarah Martinson, Erik Y. Wang, Kaylie Hausknecht, Michael P. Brenner, Mao Mao, Christoph Demian, Peyman Kassani, Xinyu Zhang, David Avagian, Eshawn Jessica Scipio, Alon Ragoler, Justin Tan, Blake Sims, Rebeka Plecnik, Aaron Kirtland, Omer Faruk Bodur, D. P. Shinde, Yan Carlos Leyva Labrador, Zahra Adoul, Mohamed Zekry, Ali Karakoc, Tania C. B. Santos, Samir Shamseldeen, Loukmane Karim, Anna Liakhovitskaia, Nate Resman, Nicholas Farina, Juan Carlos Gonzalez, Gabe Maayan, Earth Anderson, Rodrigo De Oliveira Pena, Elizabeth Kelley, Hodjat Mariji, Rasoul Pouriamanesh, Wentao Wu, Ross Finocchio, Ismail Alarab, Joshua Cole, Danyelle Ferreira, Bryan Johnson, Mohammad Safdari, Liangti Dai, Siriphan Arthornthurasuk, Isaac C. McAlister, Alejandro José Moyano, Alexey Pronin, Jing Fan, Angel Ramirez-Trinidad, Yana Malysheva, Daphiny Pottmaier, Omid Taheri, Stanley Stepanic, Samuel Perry, Luke Askew, Raúl Adrián Huerta Rodríguez, Ali M. R. Minissi, Ricardo Lorena, Krishnamurthy Iyer, Arshad Anil Fasiludeen, Ronald Clark, Josh Ducey, Matheus Piza, Maja Somrak, Eric Vergo, Juehang Qin, Benjámin Borbás, Eric Chu, Jack Lindsey, Antoine Jallon, I. M. J. McInnis, Evan Chen, Avi Semler, Luk Gloor, Tej Shah, Marc Carauleanu, Pascal Lauer, Tran Đuc Huy, Hossein Shahrtash, Emilien Duc, Lukas Lewark, Assaf Brown, Samuel Albanie, Brian Weber, Warren S. Vaz, Pierre Clavier, Yiyang Fan, Gabriel Poesia Reis e Silva, Long, Lian, Marcus Abramovitch, Xi Jiang, Sandra Mendoza, Murat Islam, Juan Gonzalez, Vasilios Mavroudis, Justin Xu, Pawan Kumar, Laxman Prasad Goswami, Daniel Bugas, Nasser Heydari, Ferenc Jeanplong, Thorben Jansen, Antonella Pinto, Archimedes Apronti, Abdallah Galal, Ng Ze-An, Ankit Singh, Tong Jiang, Joan of Arc Xavier, Kanu Priya Agarwal, Mohammed Berkani, Gang Zhang, Zhehang Du, Benedito Alves de Oliveira Junior, Dmitry Malishev, Nicolas Remy, Taylor D. Hartman, Tim Tarver, Stephen Mensah, Gautier Abou Loume, Wiktor Morak, Farzad Habibi, Sarah Hoback, Will Cai, Javier Gimenez, Roselynn Grace Montecillo, Jakub Łucki, Russell Campbell, Asankhaya Sharma, Khalida Meer, Shreen Gul, Daniel Espinosa Gonzalez, Xavier Alapont, Alex Hoover, Gunjan Chhablani, Freddie Vargus, Arunim Agarwal, Yibo Jiang, Deepakkumar Patil, David Outevsky, Kevin Joseph Scaria, Rajat Maheshwari, Abdelkader Dendane, Priti Shukla, Ashley Cartwright, Sergei Bogdanov, Niels Mündler, Sören Möller, Luca Arnaboldi, Kunvar Thaman, Muhammad Rehan Siddiqi, Prajvi Saxena, Himanshu Gupta, Tony Fruhauff, Glen Sherman, Mátyás Vincze, Siranut Usawasutsakorn, Dylan Ler, Anil Radhakrishnan, Innocent Enyekwe, Sk Md Salauddin, Jiang Muzhen, Aleksandr Maksapetyan, Vivien Rossbach, Chris Harjadi, Mohsen Bahaloohoreh, Claire Sparrow, Jasdeep Sidhu, Sam Ali, Song Bian,

John Lai, Eric Singer, Justine Leon Uro, Greg Bateman, Mohamed Sayed, Ahmed Menshawy, Darling Duclosel, Dario Bezzi, Yashaswini Jain, Ashley Aaron, Murat Tiryakioglu, Sheeshram Siddh, Keith Krenek, Imad Ali Shah, Jun Jin, Scott Creighton, Denis Peskoff, Zienab EL-Wasif, Ragavendran P V, Michael Richmond, Joseph McGowan, Tejal Patwardhan, Hao-Yu Sun, Ting Sun, Nikola Zubić, Samuele Sala, Stephen Ebert, Jean Kaddour, Manuel Schottdorf, Dianzhuo Wang, Gerol Petruzella, Alex Meiburg, Tilen Medved, Ali ElSheikh, S Ashwin Hebbar, Lorenzo Vaquero, Xianjun Yang, Jason Poulos, Vilém Zouhar, Sergey Bogdanik, Mingfang Zhang, Jorge Sanz-Ros, David Anugraha, Yinwei Dai, Anh N. Nhu, Xue Wang, Ali Anil Demircali, Zhibai Jia, Yuyin Zhou, Juncheng Wu, Mike He, Nitin Chandok, Aarush Sinha, Gaoxiang Luo, Long Le, Mickaël Noyé, Michał Perełkiewicz, Ioannis Pantidis, Tianbo Qi, Soham Sachin Purohit, Letitia Parcalabescu, Thai-Hoa Nguyen, Genta Indra Winata, Edoardo M. Ponti, Hanchen Li, Kaustubh Dhole, Jongee Park, Dario Abbondanza, Yuanli Wang, Anupam Nayak, Diogo M. Caetano, Antonio A. W. L. Wong, Maria del Rio-Chanona, Dániel Kondor, Pieter Francois, Ed Chalstrey, Jakob Zsambok, Dan Hoyer, Jenny Reddish, Jakob Hauser, Francisco-Javier Rodrigo-Ginés, Suchandra Datta, Maxwell Shepherd, Thom Kamphuis, Qizheng Zhang, Hyunjun Kim, Ruiji Sun, Jianzhu Yao, Franck Dernoncourt, Satyapriya Krishna, Sina Rismanchian, Bonan Pu, Francesco Pinto, Yingheng Wang, Kumar Shridhar, Kalon J. Overholt, Glib Briia, Hieu Nguyen, David, Soler Bartomeu, Tony CY Pang, Adam Wecker, Yifan Xiong, Fanfei Li, Lukas S. Huber, Joshua Jaeger, Romano De Maddalena, Xing Han Lù, Yuhui Zhang, Claas Beger, Patrick Tser Jern Kon, Sean Li, Vivek Sanker, Ming Yin, Yihao Liang, Xinlu Zhang, Ankit Agrawal, Li S. Yifei, Zechen Zhang, Mu Cai, Yasin Sonmez, Costin Cozianu, Changhao Li, Alex Slen, Shoubin Yu, Hyun Kyu Park, Gabriele Sarti, Marcin Briański, Alessandro Stolfo, Truong An Nguyen, Mike Zhang, Yotam Perlitz, Jose Hernandez-Orallo, Runjia Li, Amin Shabani, Felix Juefei-Xu, Shikhar Dhingra, Orr Zohar, My Chiffon Nguyen, Alexander Pondaven, Abdurrahim Yilmaz, Xuandong Zhao, Chuanyang Jin, Muyan Jiang, Stefan Todoran, Xinyao Han, Jules Kreuer, Brian Rabern, Anna Plassart, Martino Maggetti, Luther Yap, Robert Geirhos, Jonathon Kean, Dingsu Wang, Sina Mollaei, Chenkai Sun, Yifan Yin, Shiqi Wang, Rui Li, Yaowen Chang, Anjiang Wei, Alice Bizeul, Xiaohan Wang, Alexandre Oliveira Arrais, Kushin Mukherjee, Jorge Chamorro-Padial, Jiachen Liu, Xingyu Qu, Junyi Guan, Adam Bouyamourn, Shuyu Wu, Martyna Plomecka, Junda Chen, Mengze Tang, Jiaqi Deng, Shreyas Subramanian, Haocheng Xi, Haoxuan Chen, Weizhi Zhang, Yinuo Ren, Haoqin Tu, Sejong Kim, Yushun Chen, Sara Vera Marjanović, Junwoo Ha, Grzegorz Luczyna, Jeff J. Ma, Zewen Shen, Dawn Song, Cedegao E. Zhang, Zhun Wang, Gaël Gendron, Yunze Xiao, Leo Smucker, Erica Weng, Kwok Hao Lee, Zhe Ye, Stefano Ermon, Ignacio D. Lopez-Miguel, Theo Knights, Anthony Gitter, Namkyu Park, Boyi Wei, Hongzheng Chen, Kunal Pai, Ahmed Elkhanany, Han Lin, Philipp D. Siedler, Jichao Fang, Ritwik Mishra, Károly Zsolnai-Fehér, Xilin Jiang, Shadab Khan, Jun Yuan, Rishab Kumar Jain, Xi Lin, Mike Peterson, Zhe Wang, Aditya Malusare, Maosen Tang, Isha Gupta, Ivan Fosin, Timothy Kang, Barbara Dworakowska, Kazuki Matsumoto, Guangyao Zheng, Gerben Sewuster, Jorge Pretel Villanueva, Ivan Rannev, Igor Chernyavsky, Jiale Chen, Deepayan Banik, Ben Racz, Wenchao Dong, Jianxin Wang, Laila Bashmal, Duarte V. Gonçalves, Wei Hu, Kaushik Bar, Ondrej Bohdal, Atharv Singh Patlan, Shehzaad Dhuliawala, Caroline Geirhos, Julien Wist, Yuval Kansal, Bingsen Chen, Kutay Tire, Atak Talay Yücel, Brandon Christof, Veerupaksh Singla, Zijian Song, Sanxing Chen, Jiaxin Ge, Kaustubh Ponkshe, Isaac Park, Tianneng Shi, Martin Q. Ma, Joshua Mak, Sherwin Lai, Antoine Moulin, Zhuo Cheng, Zhanda Zhu, Ziyi Zhang, Vaidehi Patil, Ketan Jha, Qiutong Men, Jiaxuan Wu, Tianchi Zhang, Bruno Hebling Vieira, Alham Fikri Aji, Jae-Won Chung, Mohammed Mahfoud, Ha Thi Hoang, Marc Sperzel, Wei Hao, Kristof Meding, Sihan Xu, Vassilis Kostakos, Davide Manini, Yueying Liu, Christopher Toukmaji, Jay Paek, Eunmi Yu, Arif Engin Demircali, Zhiyi Sun, Ivan Dewerpe, Hongsen Qin, Roman Pflugfelder, James Bailey, Johnathan Morris, Ville Heilala, Sybille Rosset, Zishun Yu, Peter E. Chen, Woongyeong Yeo, Eeshaan Jain, Ryan Yang, Sreekar Chigurupati, Julia Chernyavsky, Sai Prajwal Reddy, Subhashini Venugopalan, Hunar Batra, Core Francisco Park, Hieu Tran, Guilherme Maximiano, Genghan Zhang, Yizhuo Liang, Hu Shiyu, Rongwu Xu, Rui Pan, Siddharth Suresh, Ziqi Liu, Samaksh Gulati, Songyang Zhang, Peter Turchin, Christopher W. Bartlett, Christopher R. Scotese, Phuong M. Cao, Aakaash Nattanmai, Gordon McKellips, Anish Cheraku, Asim Suhail, Ethan Luo, Marvin Deng, Jason Luo, Ashley Zhang, Kavin Jindel, Jay Paek, Kasper Halevy, Allen Baranov, Michael Liu, Advaith Avadhanam, David Zhang, Vincent Cheng, Brad Ma, Evan Fu, Liam Do, Joshua Lass, Hubert Yang, Surya Sunkari, Vishruth Bharath, Violet Ai, James Leung, Rishit Agrawal, Alan Zhou, Kevin Chen, Tejas Kalpathi, Ziqi Xu, Gavin Wang, Tyler Xiao, Erik Maung, Sam Lee, Ryan Yang, Roy Yue,

Ben Zhao, Julia Yoon, Sunny Sun, Aryan Singh, Ethan Luo, Clark Peng, Tyler Osbey, Taozhi Wang, Daryl Echeazu, Hubert Yang, Timothy Wu, Spandan Patel, Vidhi Kulkarni, Vijaykaarti Sundarapandiyan, Ashley Zhang, Andrew Le, Zafir Nasim, Srikar Yalam, Ritesh Kasamsetty, Soham Samal, Hubert Yang, David Sun, Nihar Shah, Abhijeet Saha, Alex Zhang, Leon Nguyen, Laasya Nagumalli, Kaixin Wang, Alan Zhou, Aidan Wu, Jason Luo, Anwith Telluri, Summer Yue, Alexandr Wang, and Dan Hendrycks. Humanity's last exam. *arXiv:2501.14249*, 2025. URL https://arxiv.org/abs/2501.14249.

[35] Bowen Jin, Jinsung Yoon, Priyanka Kargupta, Sercan O. Arik, and Jiawei Han. An empirical study on reinforcement learning for reasoning-search interleaved llm agents, 2025. URL https://arxiv.org/abs/2505.15117.

[36] Stephen E. Robertson, Steve Walker, Susan Jones, Micheline Hancock-Beaulieu, and Mike Gatford. Okapi at TREC-2. In Donna K. Harman, editor, *Proceedings of The Second Text REtrieval Conference, TREC 1993, Gaithersburg, Maryland, USA, August 31 - September 2, 1993*, volume 500-215 of *NIST Special Publication*, pages 21–34. National Institute of Standards and Technology (NIST), 1993. URL http://trec.nist.gov/pubs/trec2/papers/ps/city.ps.

[37] Niklas Muennighoff, Nouamane Tazi, Loic Magne, and Nils Reimers. MTEB: Massive text embedding benchmark. In Andreas Vlachos and Isabelle Augenstein, editors, *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2014–2037, Dubrovnik, Croatia, May 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.eacl-main.148. URL https://aclanthology.org/2023.eacl-main.148/.

[38] Hongjin SU, Howard Yen, Mengzhou Xia, Weijia Shi, Niklas Muennighoff, Han yu Wang, Liu Haisu, Quan Shi, Zachary S Siegel, Michael Tang, Ruoxi Sun, Jinsung Yoon, Sercan O Arik, Danqi Chen, and Tao Yu. BRIGHT: A realistic and challenging benchmark for reasoning-intensive retrieval. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=ykuc5q381b.

[39] Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira. Pyserini: A python toolkit for reproducible information retrieval research with sparse and dense representations. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, page 2356–2362, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450380379. doi: 10.1145/3404835.3463238. URL https://doi.org/10.1145/3404835.3463238.

[40] Xueguang Ma, Luyu Gao, Shengyao Zhuang, Jiaqi Samantha Zhan, Jamie Callan, and Jimmy Lin. Tevatron 2.0: Unified document retrieval toolkit across scale, language, and modality. SIGIR '25, page 4061–4065, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400715921. doi: 10.1145/3726302.3730135. URL https://doi.org/10.1145/3726302.3730135.

[41] Shuai Wang, Ekaterina Khramtsova, Shengyao Zhuang, and Guido Zuccon. Feb4rag: Evaluating federated search in the context of retrieval augmented generation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '24, page 763–773, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400704314. doi: 10.1145/3626772.3657853. URL https://doi.org/10.1145/3626772.3657853.

[42] Guilherme Penedo, Hynek Kydlíček, Loubna Ben allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. The fineweb datasets: Decanting the web for the finest text data at scale. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024. URL https://openreview.net/forum?id=n6SCkn2QaG.

A OpenAI o3 Evidence Document Gathering Prompt

I will give you a question and a correct answer, and you are to search online for evidence that supports the answer. List the evidence you've used to justify this answer step-by-step, including their urls in your output. Your final list of urls should be in the order such that a human can visit them in order to justify the answer.

Question: {question}

Answer: {answer}

This is all the information you have to work with to produce the final list of urls. Format your answer in a table with 3 columns:
- clue: the clue mentioned in the question
- url: the http web url of the evidence you've found
- evidence: the content in the url page that supports the clue

B Labelling UI Example

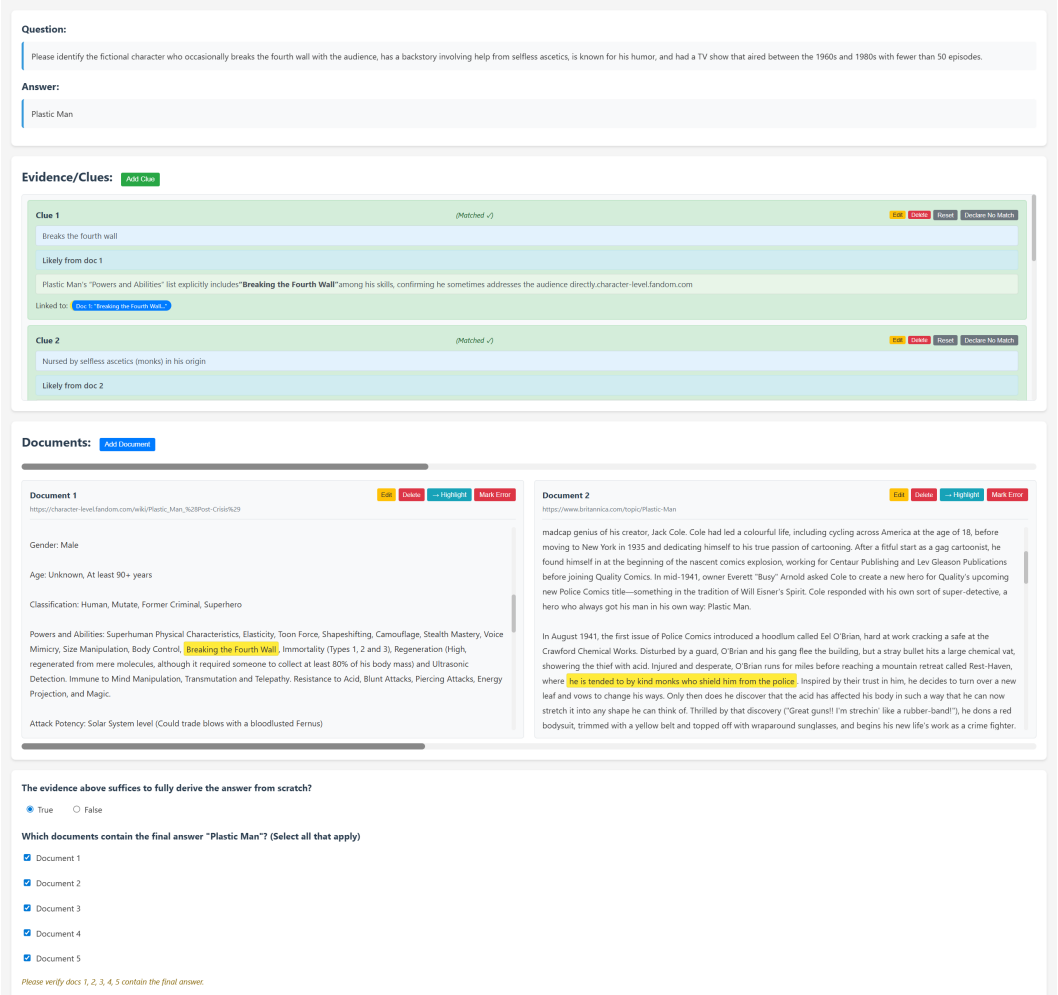

Figure 3: A screenshot of the annotation interface.

C Problematic Cases

- BrowseComp Errors: During the verification process, we discover that some question-answer pairs in BrowseComp are inherently flawed. For example, one question asks for the name of a book whose author later returned to acting. Using the ground-truth answer, we can identify the intended book and its listed author. However, upon further investigation, we find that the individual who wrote the book and the one who returned to acting are two different people who happen to share the same name.

- Extensive Use of Google Maps: 42 queries in BrowseComp require distance-related information that explicitly prompt multiple calls to Google Maps. These are removed because high-quality documents discussing specific Google Maps distances between arbitrary locations are difficult to obtain. Moreover, scraping static snapshots of Google Maps pages to include in the corpus is not a valid substitute; answering such questions as intended should require agents to be augmented with access to the Google Maps API, rather retrieving from a corpus. However, this capability lies outside the scope of our objective to build a static, document-based dataset.

- Ambiguous or Non-Unique Answers: Some question-answer pairs are well-supported by documents, but suffer from ambiguity in the expected answer format or the existence of multiple valid answers. For instance, one question asks for the username of an individual who authored a specific story on an internet forum. While the ground-truth answer is correct, it is only one of three usernames credited as authors. We remove 13 such queries due to this kind of ambiguity.

D Negative Mining Query Decomposition Prompt

You are an expert at breaking down complex, multi-part questions into simpler, self-contained subqueries.

Your task is to analyze the given question and decompose it into a series of smaller, more manageable subqueries that, when answered together, would provide all the information needed to answer the original question.

Guidelines:

1. Each subquery should focus on a single piece of information or concept
2. Subqueries MUST be completely self-contained and answerable independently - do not use pronouns or references like "this person", "the author", "these conditions", "they", "the movie", etc.
3. Each subquery should include all necessary context and constraints from the original query
4. Preserve all important details and constraints from the original query
5. Return only the subqueries as a JSON array of strings

Example:

Original: "Please identify the fictional character who occasionally breaks the fourth wall with the audience, has a backstory involving help from selfless ascetics, is known for his humor, and had a TV show that aired between the 1960s and 1980s with fewer than 50 episodes."

Subqueries: ["Which fictional characters occasionally break the fourth wall with the audience?", "Which fictional characters have a backstory involving help from selfless ascetics?", "Which fictional characters are known for their humor?", "Which TV shows aired between the 1960s and 1980s?", "Which TV shows had fewer than 50 episodes?]

Please decompose this query into subqueries:
{query}

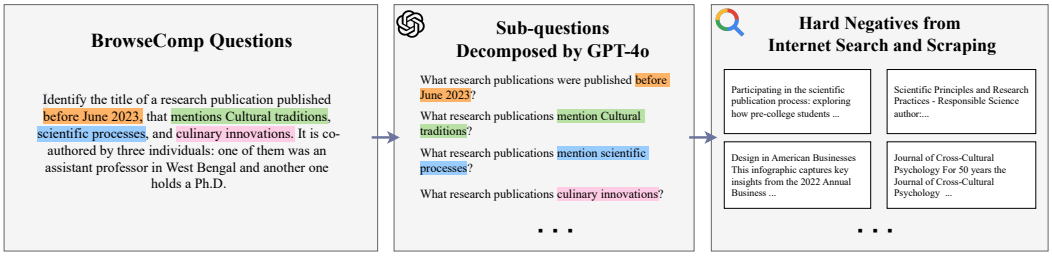

Figure 4: The pipeline of collecting hard negative documents in Section 3.3.

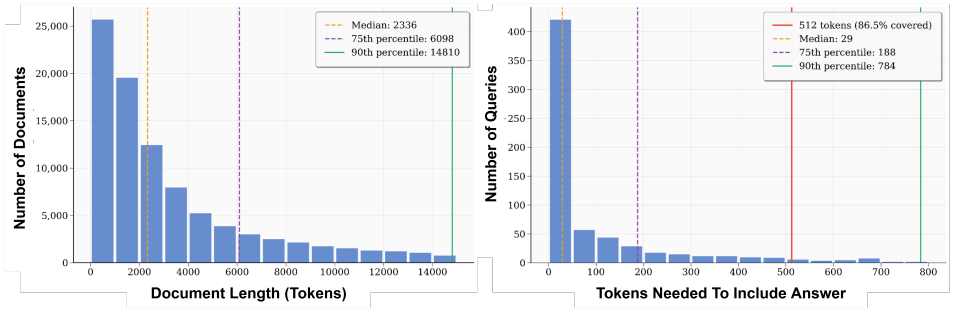

Figure 5: (a) Token distribution of corpus length, showing up to 90th percentile for display; (b) Distribution of tokens needed to include answer in gold documents per query, showing up to 90th percentile for display

E Main Search Prompt

You are a deep research agent. You need to answer the given question by interacting with a search engine, using the search tool provided. Please perform reasoning and use the tool step by step, in an interleaved manner. You may use the search tool multiple times.

Question: {Question}

Your response should be in the following format:
Explanation: {{your explanation for your final answer. For this explanation section only, you should cite your evidence documents inline by enclosing their docids in square brackets [] at the end of sentences. For example, [20].}}
Exact Answer: {{your succinct, final answer}}
Confidence: {{your confidence score between 0% and 100% for your answer}}

F Evaluation Prompt

Judge whether the following [response] to [question] is correct or not based on the precise and unambiguous [correct_answer] below.

[question]: {question}

[response]: {response}

Your judgement must be in the format and criteria specified below:

extracted_final_answer: The final exact answer extracted from the [response]. Put the extracted answer as 'None' if there is no exact, final answer to extract from the response.

[correct_answer]: {correct_answer}

reasoning: Explain why the extracted_final_answer is correct or incorrect based on [correct_answer], focusing only on if there are meaningful differences between [correct_answer] and the extracted_final_answer. Do not comment on any background to the problem, do not attempt to solve the problem, do not argue for any answer different than [correct_answer], focus only on whether the answers match.

correct: Answer 'yes' if extracted_final_answer matches the [correct_answer] given above, or is within a small margin of error for numerical problems. Answer 'no' otherwise, i.e. if there if there is any inconsistency, ambiguity, non-equivalency, or if the extracted answer is incorrect.

confidence: The extracted confidence score between 0|%| and 100|%| from [response]. Put 100 if there is no confidence score available.

G Search Prompt with Get-Doc

You are a deep research agent. You need to answer the given question by interacting with a search engine, using the search and get_document tools provided. Please perform reasoning and use the tools step by step, in an interleaved manner. You may use the search and get_document tools multiple times.

Question: {Question}

Your response should be in the following format:

Explanation: {{your explanation for your final answer. For this explanation section only, you should cite your evidence documents inline by enclosing their docids in square brackets [] at the end of sentences. For example, [20].}}
Exact Answer: {{your succinct, final answer}}
Confidence: {{your confidence score between 0% and 100% for your answer}}

H Baselines

H.1 LLM Search Agents

We evaluate several representative commercial models with strong agentic search capabilities, ranging from the most advanced reasoning models to cost-effective ones: GPT-5, o3, GPT-4.1 [9], claude-opus-4, claude-sonnet-4 [8], gemini-2.5-pro, gemini-2.5-flash [7].

Table 5: Per-query averages of citation coverage, citation count, precision, and recall for labeled evidence documents. Search-R1 is excluded because its fine-tuned outputs do not contain citations.

LLM	Retriever	Coverage	Avg # Citations	Precision	Recall
GPT-4.1	BM25	57.0%	1.92	37.0%	16.1%
	Qwen3-Embedding-8B	79.2%	2.54	58.5%	28.2%
o3	BM25	63.5%	3.27	86.7%	51.0%
	Qwen3-Embedding-8B	78.0%	3.51	91.8%	56.2%
GPT-5	BM25	94.9%	3.89	71.8%	51.3%
	Qwen3-Embedding-8B	98.0%	4.28	83.4%	62.3%
Sonnet 4	BM25	76.1%	3.19	31.9%	21.3%
	Qwen3-Embedding-8B	90.7%	4.19	52.4%	39.9%
Opus 4	BM25	74.9%	3.03	35.1%	22.3%
	Qwen3-Embedding-8B	86.1%	3.82	58.9%	42.6%
Gemini 2.5 Flash	BM25	74.2%	4.89	34.2%	21.7%
	Qwen3-Embedding-8B	89.2%	4.75	51.5%	35.1%
Gemini 2.5 Pro	BM25	53.9%	3.03	52.1%	31.4%
	Qwen3-Embedding-8B	59.4%	3.49	64.9%	41.5%
gpt-oss-120B-high	BM25	62.5%	3.55	50.8%	31.5%
	Qwen3-Embedding-8B	76.9%	3.88	60.8%	38.2%
Qwen3-32B	BM25	87.0%	1.85	8.9%	2.6%
	Qwen3-Embedding-0.6B	90.1%	1.79	8.7%	2.5%
	Qwen3-Embedding-4B	91.7%	1.84	16.1%	4.9%
	Qwen3-Embedding-8B	90.2%	1.78	20.0%	6.6%
	ReasonIR	95.8%	1.74	18.0%	5.7%

We also assess leading open-source efforts. This includes Qwen3-32B [14], a popular open-source reasoning LLM, and Search-R1 [12, 35], a model fine-tuned for agentic search based on the Qwen backbone. Specifically, we use the 32B checkpoint released in [35]. Finally, we evaluate the recent advanced gpt-oss-120B [10], a reasoning LLM optimized for search tool usage that offers multiple reasoning effort settings, ranging from low to high.

H.2 Retrievers

In our study, we compared a range of retrieval methods from a traditional lexical baseline to modern state-of-the-art dense embedding retrievers:

- BM25 [36]: The classic sparse lexical retriever, which matches queries to documents based on term statistics.
- Qwen3-Embedding [26]: A dense embedding retriever, available in sizes 0.6B, 4B, and 8B, built on the Qwen3 foundation model family [14]. It achieves state-of-the-art performance on retrieval benchmarks such as MTEB [37].
- ReasonIR [22]: A dense embedding specifically trained for reasoning-intensive retrieval via synthetic data generation, setting a new state-of-the-art on reasoning-intensive information retrieval benchmark BRIGHT [38].
- Jina-ColBERT-v2 [**?**]: A late-interaction retriever that trains ColBERTv2 [**?**] from a newer BERT backbone to support much longer contexts.

We use the Pyserini IR toolkit [39] to serve the BM25 retriever, the Tevatron dense retrieval toolkit [40] to serve Qwen3-Embedding and ReasonIR, along with PyLate [**?**] to serve Jina-ColBERT-v2.

I Citation Quality

Table 5 reports the coverage, average number, precision, and recall of the document citations attributed by the agent during answer generation. As the results show, although agents using BM25 issue

Table 6: Effectiveness of rerankers with Qwen3-Embed-8B in retriever-only evaluation. The full question is used as the query in both stages. Reranking is applied to the top-20 and top-100 candidates. Scores in parentheses denote improvements over the base retriever (Δ vs. first stage).

Reranker	**Top–20**		**Top–100**	
	Recall@5 (Δ)	**nDCG@10 (Δ)**	**Recall@5 (Δ)**	**nDCG@10 (Δ)**
Qwen3-Embed-8B	14.5 (−)	20.3 (−)	14.5 (−)	20.3 (−)
Evidence Document Retrieval				
ReasonRank-7B	22.9 (+8.4)	29.5 (+9.2)	29.5 (+15.0)	38.0 (+17.7)
Qwen3-8B	23.3 (+8.8)	30.0 (+9.7)	29.6 (+15.1)	37.7 (+17.4)
ReasonRank-32B	24.9 (+10.4)	32.1 (+11.8)	34.4 (+19.9)	43.8 (+23.5)
Qwen3-32B	24.7 (+10.2)	31.8 (+11.5)	35.0 (+20.5)	44.3 (+24.0)
Gold Document Retrieval				
ReasonRank-7B	28.7 (+10.2)	28.9 (+9.4)	36.8 (+18.3)	37.1 (+17.6)
Qwen3-8B	29.2 (+10.7)	29.6 (+10.1)	36.7 (+18.2)	36.6 (+17.1)
ReasonRank-32B	30.7 (+12.2)	31.5 (+12.0)	42.5 (+24.0)	43.5 (+24.0)
Qwen3-32B	30.5 (+12.0)	31.3 (+11.8)	42.2 (+23.7)	43.0 (+23.5)

Table 7: Effect of reranking on end-to-end agent performance. Qwen3-Embed-8B is used as the first-stage retriever and Qwen3-8B is used for reranking top 20 retrieved candidates.

LLM	**Retriever/Reranker**	**Accuracy**	**Recall**	**Search Calls**	**Calibration Error**
GPT-4.1	Qwen3-Embed-8B	35.42%	36.89%	8.67	54.67%
	+Qwen3-8B	47.11%	51.46%	8.77	49.86%
gpt-oss-20B-high	Qwen3-Embed-8B	34.58%	49.29%	23.87	27.81%
	+Qwen3-8B	40.24%	57.98%	21.98	21.47%

more search calls, nearly all metrics are lower than those achieved with Qwen3-Embedding-8B. This indicates that documents returned by BM25 are less useful in the iterative deep research process, whereas Qwen3-Embedding-8B provides more relevant and informative documents.

J Effect of Reranking

To evaluate the impact of reranking, we apply listwise reranking [**?** **?**] over the top–20 and top–100 retrieved candidates using RankLLM [**?**] with Qwen3-8B/32B and ReasonRank-7B/32B [**?**] models. The reranker operates with a sliding window of 20 candidates and a stride of 10, using a 16k-token context and a 16k-token thinking budget (output token count) to balance coverage and compute. Longer candidates are truncated to fit within the context window as needed.

Table 6 reports the effect of reranking after first-stage retrieval with Qwen3-Embed-8B, in the retrieval-only setting. For like-sized models, Qwen3 and ReasonRank perform similarly, with differences typically within 1 point. Overall, reranking yields sizable gains, improving Recall@5 by 8.4–24.0 points. With top-20 reranking, model size matters little (only ~2–3 points difference). Expanding the reranking candidate set to 100 improves all models, with larger gains for the 32B models, thereby widening the effectiveness gap between 8B and 32B models at higher rerank depths.

Table 7 reports the effect of integrating reranking into end-to-end performance of two search agents, GPT-4.1 and gpt-oss-20B (high reasoning effort), using Qwen3-Embed-8B as the first-stage retriever and Qwen3-8B to rerank the top 20 candidates. For both models, Accuracy (judged by GPT-4.1) and Recall improve substantially. This further indicates that reranking improves the precision and recall of retrieved evidence at higher ranks, helping the agent surface more relevant information.

K API Cost

Table 8 Shows the API costs of the experiments in Table 1.

Table 8: Overall API costs of proprietary agents for the experiments in Table 1.

LLM	Retriever	Accuracy	Price (USD)
GPT-4.1	BM25	14.58%	$106.96
	Qwen3-Embed-8B	35.42%	$89.81
o3	BM25	49.28%	$836.35
	Qwen3-Embed-8B	63.49%	$740.79
GPT-5	BM25	55.90%	$400.36
	Qwen3-Embed-8B	70.12%	$360.71
Sonnet 4	BM25	14.34%	$352.04
	Qwen3-Embed-8B	36.75%	$325.75
Opus 4	BM25	15.54%	$2,043.95
	Qwen3-Embed-8B	36.14%	$1,842.48
Gemini 2.5 Flash	BM25	15.54%	$47.32
	Qwen3-Embed-8B	33.01%	$41.29
Gemini 2.5 Pro	BM25	19.04%	$138.64
	Qwen3-Embed-8B	28.67%	$99.92

L Future Work and Discussion

We believe that our BrowseComp-Plus opens new avenues for advancing research in the Deep-Research area. BrowseComp-Plus retains the challenging nature of the original BrowseComp while providing a more controlled and transparent experimental setup similar to early pivotal evaluation benchmarks like Natural Question (NQ) [27] and HotpotQA [30]. Like how NQ and HotpotQA have facilitated the design, comparison, and of modern neural QA systems, we hope that BrowseComp-Plus will serve similar roles for Deep-Research agent studies. Here, we list some immediate research directions.

While our current work focuses on how different retrievers influence inference performance, a promising future direction is to examine the role of the retriever during agent optimization. For example, optimizing a search agent may be more challenging when paired with BM25 than with a modern embedding-based retriever, simply because BM25 surfaces fewer relevant documents. Understanding how retriever quality affects the learning dynamics of an agent remains an open question.

Another important extension is to study the agent's "out-of-distribution" tool-use capabilities. For instance, if an agent is optimized using a BM25 search tool, how well does its performance generalize when switched to an embedding-based search tool?

A more creative research could be an attempt on a breakdown of the commercial search engine. As much as a folktale, a commercial search solution employs tiered, composed, and multi-facet search solution. Is the LLM able to orchestrate a set of search tools to perform federated search [41], or even a sub-agent, to get quality results similar to those from Google?

A further direction is to design retrieval models that are tolerant of, or even adaptive to, a specific agent. In the Deep Research setting, the primary consumer of retrieved documents is no longer a human, but a tool-augmented LLM agent. This raises the possibility that retrieval models could be co-optimized with the agent for achieving overall answer accuracy, rather than developed and evaluated in isolation.

Finally, as shown in this work, an oracle retriever capable of surfacing gold or highly relevant documents can greatly improve accuracy. Such retrievers may also reduce the number of search iterations required, improving the overall efficiency of the research process. Developing high-precision retrieval systems for reasoning-intensive, complex queries could yield substantial benefits for real-world applications.

Overall, BrowseComp-Plus serves as an ideal testbed for pursuing these directions, enabling systematic and fine-grained analyses of agent–retriever interactions within the Deep-Research paradigm.

Table 9: Evidence document retrieval effectiveness on the Fineweb 10BT corpus.

Retriever	Corpus	Recall@5	Recall@100	Recall@1000	nDCG@10
BM25	Original	1.2	4.7	13.6	1.6
BM25	Original + Fineweb	2.2	8.0	19.4	3.1
Qwen3-Embed-8B	Original	14.5	47.7	76.7	20.3
Qwen3-Embed-8B	Original + Fineweb	11.6	37.6	64.2	16.4
ReasonIR-8B	Original	12.2	43.6	73.9	16.8
ReasonIR-8B	Original + Fineweb	8.6	30.7	56.3	11.8

Table 10: Accuracy of end-to-end search agents on our BROWSECOMP-PLUS original 100k corpus vs. FineWeb 10BT corpus.

LLM	Retriever	Corpus	Accuracy
SearchR1-32B	BM25	Original	3.86%
	BM25	Original + Fineweb	4.72%
	Qwen3-Embed-8B	Original	10.36%
	Qwen3-Embed-8B	Original + Fineweb	8.33%
Qwen3-32B	BM25	Original	3.49%
	BM25	Original + Fineweb	5.42%
	Qwen3-Embed-8B	Original	10.36%
	Qwen3-Embed-8B	Original + Fineweb	7.11%

M Effect of Corpus Size

The corpus in BROWSECOMP-PLUS contains approximately 100K documents. While real-world agents often operate over much larger, web-scale corpora, we aim to assess whether our designed corpus size is sufficient to support valid experimental observations. To this end, we augment our benchmark corpus with the Fineweb-edu [42] document collection (10 billion tokens),[4] deduplicated by URL. This expansion results in a significantly larger corpus of 9,771,311 documents—roughly 10 times larger than the original.

Table 9 shows retrieval performance before and after adding Fineweb documents. For BM25, retrieval effectiveness improves across all metrics, likely due to better inverse document frequency (IDF) estimation in the larger corpus, which strengthens BM25's lexical scoring.

In contrast, neural retrievers (Qwen3-Embedding-8B and ReasonIR-8B) show degraded performance on the Fineweb-augmented corpus. This drop is theoretically expected: the relative ranking of documents from the original small corpus remains unchanged, but the newly added Fineweb documents can now appear in the top ranks. Since these additional documents are unjudged, they are treated as non-relevant under standard TREC-style evaluation, inevitably lowering measured retrieval effectiveness.

It is important to note that lower retrieval scores for embedding models on Fineweb do not necessarily indicate worse final answers, some unjudged, top-ranked Fineweb documents may be "false negatives" that still provide useful evidence. However, as shown in Table 10, adding Fineweb does not improve answer accuracy for embedding-based retrievers. For example, Qwen3-32B with Qwen3-Embedding-8B drops from 10.36% to 7.11% accuracy.

Overall, expanding the corpus size by a factor of 10 does not lead to different conclusions about the ranking or effectiveness level among the retrievers and LLM search agents, supporting our claim that the original 100K corpus offers both strong positive coverage and sufficient challenge for robust evaluation.

[4] https://huggingface.co/datasets/HuggingFaceFW/fineweb-edu/viewer/sample-10BT

N Usage of LLM

ChatGPT is used during the writing to polish text (e.g., correct grammar) and format tables.