COVERSYNTH: COVERAGE-GUIDED GENERATION OF RETRIEVAL EVALUATION SETS

Anonymous authors
Paper under double-blind review

ABSTRACT

Enterprise agents need retrieval evaluation sets that expose domain-specific failures and guide developers towards efficient deployment tradeoffs. Neither expert-human-authored sets nor current synthetic generation methods provide a methodology for achieving the necessary coverage, reliability, efficiency, and discriminability required for such decision-making. Simultaneously, the field does not have an assessment framework for determining when an evaluation set has satisfied these properties. We introduce an evaluation set assessment framework for these criteria and CoverSynth – a generator specifically designed to satisfy these criteria without incurring the high cost of human expert labeling. Starting from an unannotated corpus, CoverSynth targets coverage across retrieval demands and system capabilities. Deterministic checks locate evidence quotes in their sources, while model-based checks assess semantic validity to guarantee reliable ground truth. Across five public enterprise-style corpora, our method provides informative variation in difficulty, revealing unseen failure modes such as an inability of agents to handle underspecified or ambiguous requests. On the RepLiQA corpus, 91% of human-generated queries can be answered by all members on a panel of 12 retrieval systems, compared to only 12% of CoverSynth-generated queries. Finally, in a repeated TechQA/RepLiQA study, 12 of 12 within-harness model contrasts have positive nominal 95% intervals on generated sets, compared with 4 of 12 on human sets. CoverSynth makes coverage and evidence quality explicit construction objectives, producing corpus-grounded evaluations that reveal retrieval headroom and differences between agent configurations.

1 INTRODUCTION

Enterprise agents operate over large, heterogeneous private corpora with organization-specific terminology, document relationships, and operational rules (Sun et al., 2026). Beyond filling gaps in pre-trained knowledge (Kandpal et al., 2023), agents must interpret information in its local context (Agarwal et al., 2026), and when retrieved evidence drives decisions, missing dependencies or unsupported assumptions propagate through later workflow steps. Evaluation must therefore test both whether an agent can assemble supporting evidence and whether it recognizes requests the corpus cannot resolve.

An industry survey finds that only 5% of organizations have a custom generative AI tool in production, attributing most failures to brittle workflows and a lack of contextual learning (Challapally et al., 2025). Retrieval alone does not guarantee grounding: retrieval-based legal research tools hallucinate on 17–33% of queries (Magesh et al., 2025), and a search-augmented model leaves about 30% of its medical statements unsupported by its cited sources (Wu et al., 2025). Regulators also increasingly require pre-deployment testing of high-risk AI systems (e.g., European Parliament and Council, 2024).

Enterprises often need these evaluations without existing questions, gold annotations or query logs. Authoring by subject-matter experts (SMEs) is costly: ViDoRe V3 required 12,000 hours of annotation (Loison et al., 2026); confidentiality also limits crowd annotation of regulated corpora (Cook et al., 2025). Recent benchmarks combine model drafting with human or LLM-judge verification (Sun et al., 2026; Cohen et al., 2025; Loison et al., 2026), or automate generation as in MiRAGE (Sahu et al., 2026). Verification of individual items, however, does not establish coverage of an in-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

tended workload. Our five-corpus profiles show that native benchmark and MiRAGE reference sets leave 31–59% of concept clusters untested (Table 1). The two MiRAGE reference sets contain only answerable questions, leaving abstention and clarification behavior untested.

Such omissions can conceal failures on particular retrieval demands; incomplete gold and questions on which systems saturate can also obscure useful performance differences. We translate these risks into an assessment framework for evaluation sets (Section 4). Coverage and diversity metrics quantify concentration across question types and evidence structures, representation of user roles, and untested corpus regions (Wang et al., 2026b; Klearman et al., 2026). Source-localization checks and oracle evidence probes address annotation weaknesses that can distort retrieval scores. Measurements of headroom, repeat-based system separation and resource use assess whether the set can inform choices among systems and budgets. These complementary diagnostics apply across construction methods and are interpreted relative to the declared workload and evidence representation.

Guided by these criteria, we present CoverSynth, a framework that synthesizes retrieval evaluation sets from an unannotated corpus (Figure 1). It makes coverage across corpus regions, personas, intents, evidence structures and failure modes an explicit construction objective. Deterministic checks locate retained evidence quotes exactly in their sources, while model-based checks assess semantic validity. The pipeline infers lenses and personas and requires no expert-authored questions or evidence annotations; expert effort can instead focus on corpus preparation, reviewing the intended workload and auditing outputs (Section 5.2).

We use this assessment framework to evaluate CoverSynth against native and synthetic references on five enterprise-style corpora (Section 6). The reported profiles show higher joint attribute diversity (0.77–0.91 vs. 0.35–0.70) and no uncovered concept clusters on three corpora. On RepLiQA, all 12 tested systems cite every gold document in every available repeat on 91% of human questions versus 12% of generated questions. In the repeated TechQA/RepLiQA study, 12 of 12 within-harness model contrasts have positive nominal 95% intervals on generated sets, compared with 4 of 12 on human sets. Answerability verdicts further expose difficulty recognizing underspecified and ambiguous requests.

Our contributions are:

- **An assessment framework for retrieval evaluation sets**, mapping practical requirements for coverage, evidence validity, and system comparison to complementary, automatically computed diagnostics (Section 4).
- **A method for constructing retrieval evaluation sets directly from unannotated corpora**, with verifiable evidence annotations and explicit coverage objectives, evaluated against native and synthetic references across five corpora (Sections 3 and 6).
- **A taxonomy and controlled generation procedure for unanswerable queries**, covering missing evidence, false premises, under-specification, and ambiguity. By applying controlled changes to answerable questions, it creates cases that test whether agents recognize when a request cannot be answered as posed (Section 3.5).

2 Related Work

Generation of a retrieval evaluation set. Enterprise retrieval benchmarks such as EnterpriseRAG-Bench (Sun et al., 2026), WixQA (Cohen et al., 2025), and ViDoRe V3 (Loison et al., 2026) rely on human authoring or curation, and report that different question types need different construction procedures and that negative coverage requires deliberate corpus exploration. CoverSynth reproduces these properties synthetically.

InPars (Bonifacio et al., 2022) and Promptagator (Dai et al., 2023) introduced synthetic query generation via domain-agnostic and few-shot prompting. Later work added control and verification: attribute guidance with programmatic checks (DataGen; Huang et al., 2025), multi-agent verifiers (MiRAGE; Sahu et al., 2026), and chunk-level knowledge graphs for single- and multi-hop synthesis (RARE, Zeng et al., 2025; RAGAS, Es et al., 2024). Other frameworks target visual page retrieval (Loison et al., 2026), preference arenas (DocArena; Wang et al., 2026a), or multi-step reasoning trajectories (DRAGON, Shen et al., 2026; S^3-R1, Goel et al., 2026). None guarantees byte-exact ev-

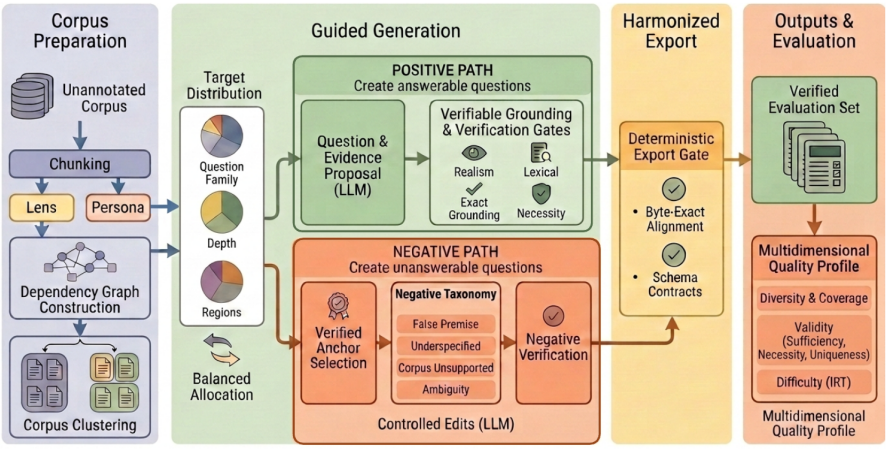

Figure 1: Overview of CoverSynth. An inferred lens guides chunk annotation and concept-graph construction. Graph communities, personas, and coverage controls guide positive and negative query generation. Model checks assess semantic validity; the export gate enforces exact span localization and record contracts.

idence by construction or isolates the sequential dependency depth that iterative multi-hop retrieval requires.

Coverage of the corpus and of question types. Klearman et al. (2026) show that aggregate metrics hide localized failures and propose semantic stratification to cover the whole corpus. CCQGen (Kang et al., 2025) conditions generation on uncovered concepts, and OBLIQ-Bench (Tchuindjo et al., 2026) clusters documents by a latent attribute before writing queries. CoverSynth also clusters the corpus, but targets joint coverage of question types, answerability classes, and evidence structure rather than topics alone.

Unanswerability and abstention. RefusalBench (Muhamed et al., 2025) categorizes informational uncertainty as ambiguity, contradiction, missing information, or false premise; HopRefusalBench (Xie et al., 2026) shows that where unanswerability occurs in a reasoning chain (root, intermediate, or terminal) affects refusal behavior; and GENERATE-PAIRWISE (Chaudhary et al., 2024) jointly synthesizes relevant and irrelevant queries as hard negatives. CoverSynth turns these findings into a generation procedure: it applies one controlled edit to a verified answerable anchor and keeps the item only if the corpus supports the intended failure, fixing the locus and cause of unanswerability by construction.

Assessing synthetic data quality. SynQuE (Chen et al., 2025) scores synthetic data against unlabeled samples using embedding distribution and diversity metrics plus LLM-based structural rubrics (LENS). Our framework instead measures intrinsic properties of the evaluation set and needs no reference data.

Table 3 (Appendix A.1) compares CoverSynth with representative methods.

3 METHOD

CoverSynth constructs an evaluation set from an unannotated corpus $\mathcal{C}$. Models propose lenses, personas, questions, and quotes while keeping track of question attributes that support non-redundant coverage as well as precise downstream error analysis. Deterministic components localize quotes, validate schemas, and analyze the graph. Positive and negative synthesis paths meet at an independent export gate (Figure 1).

3.1 CORPUS REPRESENTATION AND DEPENDENCY GRAPH

Layout-aware parsing splits documents into chunks, from which models infer one *lens* and a small set of *user personas*. Inspired by OBLIQ-Bench's human-defined lens (Tchuindjo et al., 2026), the lens specifies how to interpret passages, such as the engineering requirement each establishes. Personas describe expected users, their domain familiarity, question style, and example questions.

The lens guides annotation of concepts, references, and acronyms from each chunk. Canonicalized concept labels form the vertices $V = \{\ell_1, \ldots, \ell_M\}$. Semantic edges connect each concept to its $k = 15$ nearest neighbors with cosine similarity at least $\tau = 0.55$, using 256-dimensional `text-embedding-3-large` embeddings. Citation edges connect concepts whose passages cross-reference one another, with fixed weight 0.55. Resulting in a graph $G = (V, E_{\text{sem}} \cup E_{\text{cite}})$

By partitioning the concept graph into localized communities, Leiden clustering maps each chunk to its concepts' communities, grouping related passages across the corpus. Within each community, a planner classifies passage subsets as *co-dependent* (2–4 jointly required passages), *competing* (opposing or conditionally bounded claims), or *isolated* (single-passage facts) to identify feasible question types.

3.2 QUERY TOPOLOGY: DEPTH VERSUS BREADTH

For evidence set $S = \{s_1, \ldots, s_b\}$, breadth $b = |S|$ counts evidence units and depth d counts sequential dependencies:

- **Single lookup ($b = 1, d = 0$):** One evidence unit supplies the answer.
- **Parallel aggregation ($b > 1, d = 0$):** Independent units jointly supply the answer.
- **Bridge chain ($b > 1, d = b - 1$):** Resolving s_t yields the key needed to locate s_{t+1}.

Only bridges contribute to hop depth; parallel aggregation increases breadth.

3.3 VERIFIABLE GROUND-TRUTH SPANS

To construct answerable evaluation items, the model drafts a question from a feasible passage subset (Section 3.1) and proposes verbatim supporting quotes. Each quote $s_k \in S$ is drawn from a source chunk c_k. A deterministic locator checks each quote against its source and records the matching character offsets.

A quote is admitted only if it occurs in its source:

$$g_{\text{ground}}(s_k, c_k) \;=\; \mathbf{1}[\,\exists\,[i, j] \text{ s.t. } c_k[i : j] = s_k\,]\,. \tag{1}$$

The matching offsets $[i_k, j_k]$ are stored in the item record. Every cited span must pass this check during generation and independently at export, after negative generation (Section 3.5) as well.

3.4 POSITIVE GENERATION AND ADVERSARIAL PRUNING

A draft uses $\mathcal{T} = (\text{structure}, \text{persona}, \text{intent}, \text{steer})$: the query topology, a persona, one of eight intents (Table 4), and a bias toward under-produced difficulty and phrasing, respectively. Ordered checks impose two hard gates: persona realism (g_{real}; Appendix A.4) and exact quote localization (g_{ground}). Two soft checks prompt repairs for question–passage bag-of-words overlap above 0.5 and distinctive wording or a question that lists its own answer options (the semantic "own-words" critic; Appendix A.5). A final adversarial verifier (g_{verify}) checks answerability, gold-span sufficiency, that the question neither reveals the answer nor points to its location, independent support per evidence unit, and sequential dependency for bridges (Appendices A.6–A.7). The hard admission criteria are

$$A(q, \{s_k\}) \;=\; g_{\text{real}} \;\wedge\; \left(\bigwedge_{k=1}^{|S|} g_{\text{ground}}(s_k, c_k) \right) \;\wedge\; g_{\text{verify}}. \tag{2}$$

A draft still failing a hard gate after at most two repairs is rejected. An accepted positive becomes a *verified anchor*.

3.5 NEGATIVE GENERATION AND DIFFICULTY

Each negative applies one generation-control edit (Table 5) to a verified anchor. Two separate axes target refusal difficulty: *topology* places the defect at the root, intermediate, or terminal hop, leaving preceding hops answerable (Xie et al., 2026); *intensity* (low, medium, high) sets the variation's severity and intended detectability (Muhamed et al., 2025). Two independently framed adjudicators assess validity. A *direct verifier* checks the intended failure, such as corpus silence, a false premise, or reference ambiguity; an *adversarial solver* tries to resolve it against the corpus. Admission requires the verifier to confirm the failure and the solver to find no resolution.

3.6 COVERAGE CONTROL

Positive candidates are enumerated over the graph under fixed per-source caps. A round-robin schedule prioritizes under-sampled documents; an equal-share hop quota controls depth. Soft steering discourages over-produced phrasing and complexity tiers. Stable hashing of candidate tuples assigns valid persona–intent pairs, with realized coverage assessed post-hoc (Section 4).

For negatives, each attribute axis—failure family, generation control, topology, intensity, and complexity—is apportioned separately. Given normalized target weights w and budget n, largest-remainder (Hamilton) apportionment assigns

$$n_j = \lfloor n\, w_j \rfloor + r_j, \qquad \sum_j n_j = n, \tag{3}$$

where $r_j \in \{0, 1\}$ assigns remaining slots to the largest fractional parts. Greedy emission minimizes the maximum relative deficit, $\max_j (n\, w_j - \hat{n}_j)/(n\, w_j)$, among values feasible for the active anchor; $\hat{n}_j$ counts emitted or reserved items. Marginal apportionment avoids enumerating the full Cartesian grid, but anchor feasibility can still constrain realized coverage.

4 ASSESSMENT FRAMEWORK

Retrieval evaluation should help developers diagnose failures, compare candidate systems, and make informed deployment decisions for an intended workload. An evaluation set that omits relevant information needs or corpus regions can leave important failures untested. Unreliable reference evidence can confound system errors with annotation artifacts, while questions on which candidate systems uniformly succeed or fail offer little signal for choosing between them. These risks motivate three complementary requirements: coverage of retrieval demands within the intended workload, validity of gold evidence, and informative variation in difficulty. Our assessment framework operationalizes these requirements through automated diagnostics, interpreted relative to a declared corpus and workload.

Intended users and scope. Personas specify the user roles and information needs an enterprise evaluation targets. Coverage is assessed within this declared scope. For a declared persona set $\mathcal{P}$ with question counts q_h and shares $\pi_h = q_h / \sum_{h' \in \mathcal{P}} q_{h'}$, persona coverage is the normalized entropy

$$H_{\text{norm}} \;=\; -\frac{\sum_{h \in \mathcal{P}} \pi_h \log \pi_h}{\log |\mathcal{P}|} \;\in\; [0, 1], \tag{4}$$

which equals 1 when every persona receives the same number of questions and 0 when all questions target a single persona. Questions that match no declared persona are excluded from π_h.

Coverage and diversity. Retrieval performance can vary with both question demands and corpus regions: success on direct lookup need not transfer to cross-file aggregation or sequential reference resolution, and omitted topics leave local failures untested. We assess eight question intents (Appendix A.3) jointly with multi-file and multi-hop requirements. For the declared feasible cells Ω of this capability grid, let p be the empirical item distribution. Attribute diversity (Wang et al., 2026b) is the normalized joint entropy

$$\text{AD} \;=\; -\frac{\sum_{\omega \in \Omega} p(\omega)\, \log p(\omega)}{\log |\Omega|} \;\in\; [0, 1], \tag{5}$$

which captures interaction imbalances missed by marginal entropies. Following Klearman et al. (2026), we measure corpus coverage with the sufficient-corpus-coverage ratio (SCC) and zero-query cluster proportion (ZQC). For corpus document set $\mathcal{C}$, assessment clusters $\mathcal{K}$, and queries Q_i hitting cluster i,

$$\text{SCC} = \frac{|\mathcal{D}_{\text{covered}}|}{|\mathcal{C}|}, \qquad \text{ZQC} = \frac{\left|\{i \in \mathcal{K} : |Q_i| = 0\}\right|}{|\mathcal{K}|} \in [0, 1], \tag{6}$$

where $\mathcal{D}_{\text{covered}}$ contains documents in at least one cluster with $|Q_i| \geq n_{\min} = 5$. SCC measures document coverage through repeatedly queried clusters; ZQC identifies clusters receiving no questions. Taken together, AD, SCC, and ZQC measure a benchmark's ability to detect performance variances across dimensions which matter for retrieval.

Gold-evidence validity. Incomplete or redundant gold can confound system errors with annotation artifacts. Exact localization (Equation 1) checks that evidence exists; oracle-reader probes assess its semantic usefulness. *Sufficiency* is the fraction of probed questions answered correctly from gold alone. Conditional on gold sufficiency, *uniqueness* is the fraction answered incorrectly from selected distractors, and *necessity* is the fraction of gold units whose removal makes the reader incorrect. These probes depend on the reader and evidence granularity; results appear in Table 2, with formal definitions in Appendix A.12.

Difficulty. An evaluation set should exercise routine and demanding cases, revealing failures and where additional capability may help. We examine document-retrieval saturation and the stability of system differences across repeats as evidence of diagnostic value within the tested panel (Section 6; Appendix A.11 and A.14).

5 Experimental Setup

5.1 Corpus selection and preparation

We evaluate CoverSynth on five public collections with varied retrieval demands. TechQA contains short, sectioned IBM support technotes; RepLiQA contains fictional prose. EnterpriseRAG-Bench (ERAG-Bench) spans nine source systems—Confluence, Jira, Slack, GitHub, Google Drive, Gmail, Linear, HubSpot, and Fireflies—with a Confluence subset used for generation. eCFR contains nested, cross-referenced regulations. Harvey Firm Knowledge (Harvey FK) contains legal documents organized by client, where near-duplicate content across clients makes it easy to ground an answer in the wrong client's records.

We convert corpora to Markdown and chunk each document along its native structure (headings, capitalized section labels, or numbered clauses), falling back to paragraph blocks where none exist. Subsets approximate deployment scope: a Confluence space, an organization's policies, whole legal matters, or a set of regulators. Gold-seeded TechQA and ERAG subsets contain a 0.40 document fraction of distractors. We inspect chunks-per-document distributions and resample degenerate subsets, including those with large single-chunk tails. Each corpus is capped at 5,000 chunks to limit generation cost. Sampling choices are fixed before generation. Detailed corpus sub-sampling methodology can be found in Appendix A.2.

5.2 Evaluation protocol and baselines

Generation details. GPT-5.4 annotates concepts, entities, and references; GPT-5.6-sol drafts questions and adjudicates semantics. Lens and personas are inferred, with target hop depths of 0, 1, or 2.

Retrieval systems and studies. Our nine-system base panel combines BM25, dense retrieval (`all-MiniLM-L6-v2`), and their reciprocal-rank fusion with ReAct and Claude Code, each using Haiku-4.5, Sonnet-5, and Opus-5. Agents iteratively search, inspect documents, and select evidence; returned spans are scored for precision, recall, and F_2 (Appendix A.8). Single-pass Claude Code measurements across the five corpora appear in Appendix A.13.

Table 1: Benchmark assessment metrics as defined in Section 4 on the five corpora using both CoverSynth as well as reference benchmarks. Note that persona entropy values are calculated using CoverSynth's persona set, excluding questions without a matching persona. Computation of reference datasets' SCC and ZQC are detailed in Appendix A.10. Arrows indicate the preferred direction for the first four assessment metrics. For difficulty skew, values closer to zero are preferred (with more negative values representing a skew towards hard and more positive values representing a skew towards easy). Statistically significant differences marked with *.

Corpus	Source	Queries	AD ↑	Persona H_{norm} ↑	SCC ↑	ZQC ↓	Difficulty skew ⊙
ERAG-Bench	CoverSynth	150	**0.88***	**0.98***	**0.96**	**0.00**	-0.69
	Reference	53	0.70	0.79	0.86	0.54	-0.90
Harvey FK	CoverSynth	150	**0.89***	**0.99***	0.76	**0.00**	-0.67^*
	Reference	150	0.59	0.87	**0.99**	0.59	-1.60
TechQA	CoverSynth	128	**0.91***	**0.99***	0.87	**0.18**	$+0.27$
	Reference	374	0.41	0.54	0.54	0.31	-0.09
eCFR	CoverSynth	148	**0.84***	**0.99***	**1.00**	**0.00**	-0.29
	Reference	150	0.66	0.76	0.28	0.44	-0.43
RepLiQA	CoverSynth	147	**0.77***	**0.99***	**0.98**	**0.02**	-0.36^*
	Reference	685	0.35	0.79	0.67	0.57	$+2.31$

Repeated two-corpus study. On TechQA and RepLiQA, whose human reference sets provide passage-level gold, we run the nine-system panel on both the generated and the human sets, repeating each agent three times; indexed retrieval is deterministic. This study examines headroom and model separation (Section 6.3). Settings and scoring populations are in Appendix A.14, and additional single-run agent configurations in Appendix A.16.

Baselines. We use native ERAG, TechQA and RepLiQA references, and MiRAGE (Sahu et al., 2026) for Harvey and eCFR. MiRAGE uses chunk-level gold, scored separately from spans (Appendix A.13).

6 Results and Discussion

We first assess the composition, evidence validity and retrieval behavior of the generated sets across all five corpora. A focused repeated study on TechQA and RepLiQA then examines measurement headroom and system separation against human sets with passage-level gold.

6.1 Coverage and evidence validity

Broader coverage, with corpus-dependent tradeoffs. CoverSynth achieves higher joint attribute diversity than references across all corpora (Table 1). ZQC is lower in every comparison and is reported as zero on ERAG-Bench, Harvey and eCFR. SCC improves on ERAG-Bench, eCFR and RepLiQA, but is lower on Harvey (0.76 versus 0.99) and TechQA (0.87 versus 1.00). Reaching a concept cluster at least once therefore differs from meeting the repeated-coverage threshold in clusters associated with a document. Retroactively labeling reference queries with CoverSynth's persona set reveals that CoverSynth yields near-uniform persona coverage, while reference benchmarks achieves significantly lower spread. These profiles support broader measured coverage while exposing how repeated coverage is distributed under the specified inputs.

Exact localization and semantic validity. Source-offset checks inherently ensure exact localization of evidence span offsets. In addition, Table 2's oracle-reader probes show that CoverSynth achieves sufficiency of 0.70–0.97, uniqueness of 0.68–0.99, and leave-one-out necessity of 0.49–0.67 scores. These indicate higher validity scores on five of six quality dimensions relative to MiRAGE, i.e., the other synthetically generated baseline. However, this only remains true for half the dimensions when comparing to human reference baselines. This suggests that human reference baselines may yield more precise evidence spans, offering room for improvement by CoverSynth.

Table 2: Gold-span validity, as defined in Section 4, from oracle-reader probes. The oracle reader (Claude-Opus-5) is a different model from the generator (GPT-5.6 Sol). Uniqueness contrasts each gold span against the three highest lexical-overlap non-gold passages. †The human TechQA and RepLiQA sets carry a single gold span per item, so leave-one-out necessity is trivially 1.00 and not comparable to the multi-span CoverSynth rate.

	ERAG-Bench		Harvey FK		TechQA		eCFR		RepLiQA	
Rate	**CoverSynth**	**Human**	**CoverSynth**	**MiRAGE**	**CoverSynth**	**Human**	**CoverSynth**	**MiRAGE**	**CoverSynth**	**Human**
Sufficiency	**0.70**	0.64	0.80	**0.95**	**0.72**	0.61	**0.97**	0.95	**0.91**	0.90
Uniqueness	0.99	**1.00**	**0.68**	0.65	0.96	0.96	**0.99**	0.94	0.90	**0.97**
Necessity (leave-one-out)	0.58	**0.64**	**0.61**	0.24	0.67	—†	**0.49**	0.41	0.49	—†

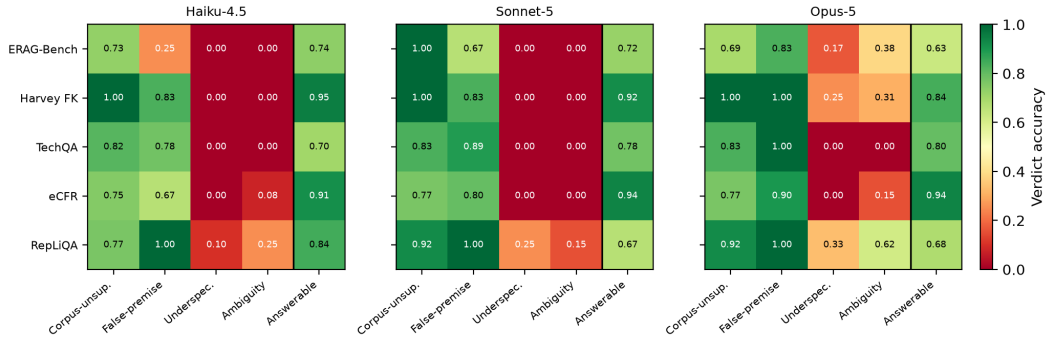

Figure 2: Single-pass answerability-verdict accuracy by corpus and model under the Claude Code harness, for all five CoverSynth outputs (the same runs as Table 9). The first four columns are the negative families; the final column is answerable-item verdict accuracy. Failed runs are excluded.

6.2 Retrieval behavior and negative-query diagnostics

Retrieval agent discriminability In Figure 3 we see that both human and CoverSynth discriminate reliably between agentic vs. indexed retrievers in span F_2 (with the indexed retrievers reliably performing worse than agentic retrievers). However, only CoverSynth can consistently discern the expected Haiku-4.5 $<$ Sonnet-5 $<$ Opus-5 base model hierarchy. Further, in appendix A.13, figure 5, CoverSynth is able to discern each harness type by the expected context token consumption rate (unlike the reference benchmarks). Complete per-model measurements and protocol details appear in Appendix A.13.

Negative families expose different failure modes. In Figure 2, we see that CoverSynth not only discerns the expected model backbone hierarchy, but reveals the specific failure modes (underspecified and ambiguous questions) that the stronger reasoning model (Opus-5) is most effective at improving upon. The taxonomy therefore reveals distinctions obscured by aggregate answerability accuracy. See Appendix A.17 for the detailed breakout.

6.3 Performance headroom and retrieval system separation

We next examine whether the generated sets support informative system comparisons on TechQA and RepLiQA, using the repeated nine-system panel described in Section 5.2.

Performance varies by corpus. Figure 3 shows all configurations separately. Agents score lower on generated RepLiQA questions, while TechQA differences vary by model. Within the base ReAct and Claude Code harnesses, 12 of 12 ordered model contrasts have positive nominal 95% intervals on generated sets, compared with 4 of 12 on human sets. Contrasts are oriented according to the expected Haiku–Sonnet–Opus capability order. Full scores and model contrasts are in Appendix A.14.

Headroom and reliability. In Table 1's difficulty skewness metric (also visualized in Appendix A.11's Figure 4), we see that CoverSynth produces a less skewed distribution of difficulty levels

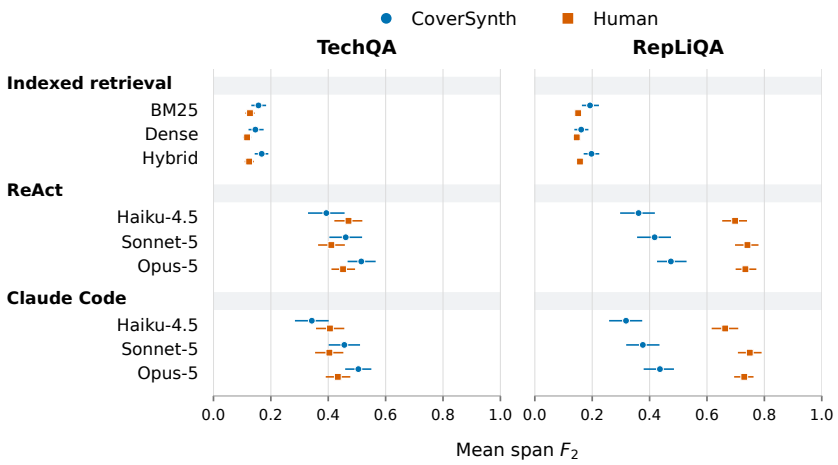

Figure 3: Retrieval performance on CoverSynth and human evaluation sets. Points show mean span F_2 on answerable questions; whiskers show 95% bootstrap intervals. Protocol details are in Appendix A.14.

on four out of the five corpora as compared to their respective reference baselines. In particular, it avoids the extreme skew towards hard questions that MiRAGE creates on Harvey FK (a skew of only -0.67 by CoverSynth vs. MiRAGE's -1.60) and the extreme skew towards easy questions that RepLiQA ships with (a skew of of only -0.36 by CoverSynth vs. the reference benchmark's +2.31). This difficulty distribution directly translates into more reliable discernment of different retrieval systems. For example, 91% of the answerable subset of benchmark queries from the native RepLiQA corpus are correctly addressed by every member of the nine of panel of retrieval systems. Making these retrieval systems effectively indistinguishable on the native benchmark. This is only true for 12% of CoverSynth's generated queries, allowing for more informative deployment choices.

6.4 Discussion

Measured against the assessment framework outlined in Section 4, we show that CoverSynth consistently provides more coverage than what native and MiRAGE references provide on all five corpora. This includes more extensive retrieval demands (including underspecified and ambiguous demands), better distribution over the domain's natural personas as measured by entropy, as well as less skewed difficulty distributions. Persona coverage is computed against CoverSynth's corpus-derived persona sets, and evaluating those sets against a more comprehensive topic structure is left to future work. The focused study adds further evidence of head room and model separation not only on span F_2, but also along other performance dimensions such as token consumption. The rich taxonomy supported by CoverSynth also allows drill downs into specific failure modes that explain the separation between systems not only at an aggregate score level, but also at fine grained retrieval demands (such as underspecified intents). Finally, this greater coverage and discernment is accompanied by high gold label validity scores relative to the other synthetic baseline benchmark, making the resulting automated corpus driven benchmark an effective tool for high quality deployment decision making in enterprise scenarios.

7 Conclusion

We introduced an assessment framework for retrieval evaluation sets and CoverSynth, a generator that builds such sets from an unannotated corpus with coverage and exact evidence localization as construction objectives. On five enterprise-style corpora, the resulting sets cover the declared workload more broadly than native and MiRAGE references and separate retrieval systems where human sets saturate.

AI USE STATEMENT

We used a generative AI assistant to draft and edit the prose of this manuscript, to format its equations and tables, and to organize the bibliography. The method, the mathematical formulation, the experimental design, and all reported results are the authors' own work, but the implementation was completed with the assistance of AI coding tools. The assistant did not design experiments or produce results. The authors reviewed and verified all text, claims, and citations, and take responsibility for the final content. See the ICLR 2027 AI Policy for Authors for details.

ETHICS STATEMENT

Our method generates evaluation data from public corpora only, all under permissive or public-domain licenses. Grounding every item in verifiable evidence reduces the risk of unsupported claims in downstream measurement. We use no proprietary or personal data in this work.

REFERENCES

Shubham Agarwal, Asim Biswal, Sepanta Zeighami, Alvin Cheung, Joseph Gonzalez, and Aditya G. Parameswaran. Arming data agents with tribal knowledge, 2026. URL https://arxiv.org/abs/2602.13521.

Luiz Bonifacio, Hugo Abonizio, Marzieh Fadaee, and Rodrigo Nogueira. InPars: Data Augmentation for Information Retrieval using Large Language Models. 2022.

Aditya Challapally, Chris Pease, Ramesh Raskar, and Pradyumna Chari. The GenAI divide: State of AI in business 2025. Technical report, MIT NANDA, July 2025. URL https://mlq.ai/media/quarterly_decks/v0.1_State_of_AI_in_Business_2025_Report.pdf.

Aditi Chaudhary, Karthik Raman, and Michael Bendersky. It's All Relative! A Synthetic Query Generation Approach for Improving Zero-Shot Relevance Prediction. *Findings of the Association for Computational Linguistics: NAACL*, 2024. doi: 10.18653/v1/2024.findings-naacl.107.

Arthur Chen, David R. Cheriton, and Victor Zhong. SynQuE: Estimating Synthetic Dataset Quality Without Annotations. 2025.

Dvir Cohen, Lin Burg, Sviatoslav Pykhnivskyi, Hagit Gur, Stanislav Kovynov, Olga Atzmon, and Gilad Barkan. WixQA: A Multi-Dataset Benchmark for Enterprise Retrieval-Augmented Generation. 2025.

Thomas Cook, Richard Osuagwu, Liman Tsatiashvili, Vrynsia Vrynsia, Koustav Ghosal, Maraim Masoud, and Riccardo Mattivi. Retrieval augmented generation (rag) for fintech: Agentic design and evaluation. In *2025 3rd International Conference on Foundation and Large Language Models (FLLM)*, pp. 93–100, 2025. doi: 10.1109/FLLM67465.2025.11391155.

Zhuyun Dai, Vincent Y. Zhao, Ji Ma, Yi Luan, Jianmo Ni, Jing Lu, Anton Bakalov, Kelvin Guu, Keith B. Hall, and Ming-Wei Chang. Promptagator: Few-shot Dense Retrieval From 8 Examples. *International Conference on Learning Representations (ICLR)*, 2023.

Shahul Es, Jithin James, Luis Espinosa-Anke, and Steven Schockaert. RAGAS: Automated Evaluation of Retrieval Augmented Generation. *Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations (EACL)*, 2024.

European Parliament and Council. Regulation (EU) 2024/1689 of the european parliament and of the council laying down harmonised rules on artificial intelligence and amending regulations (AI Act), June 2024. URL http://data.europa.eu/eli/reg/2024/1689/oj.

Harsh Goel, Akhil Udathu, Susmija Jabbireddy, Pradnesh Kalkar, and Atharva Parulekar. s^3-r1: Learning to retrieve and answer step-by-step with synthetic data. 5 2026. URL https://arxiv.org/pdf/2605.01248.

Yue Huang, Siyuan Wu, Chujie Gao, Dongping Chen, Qihui Zhang, Yao Wan, Tianyi Zhou, Jianfeng Gao, Chaowei Xiao, Lichao Sun, and Xiangliang Zhang. DataGen: Unified Synthetic Dataset Generation via Large Language Models. *International Conference on Learning Representations (ICLR)*, 2025.

Nikhil Kandpal, Haikang Deng, Adam Roberts, Eric Wallace, and Colin Raffel. Large language models struggle to learn long-tail knowledge. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

Seong Ku Kang, Bowen Jin, Wonbin Kweon, Yu Zhang, Dongha Lee, Jiawei Han, and Hwanjo Yu. Improving Scientific Document Retrieval with Concept Coverage-based Query Set Generation. *Proceedings of the 18th ACM International Conference on Web Search and Data Mining (WSDM)*, 2025. doi: 10.1145/3701551.3703544.

Andrew Klearman, Radu Revutchi, Rohin Garg, Rishav Chakravarti, Samuel Marc Denton, and Yuan Xue. Coverage, Not Averages: Semantic Stratification for Trustworthy Retrieval Evaluation. 2026.

António Loison, Quentin Macé, Antoine Edy, Victor Xing, Tom Balough, Gabriel de Souza P. Moreira, Bo Liu, Manuel Faysse, Celine Hudelot, and Gautier Viaud. ViDoRe V3: A comprehensive evaluation of retrieval augmented generation in complex real-world scenarios. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 16570–16600, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.755. URL https://aclanthology.org/2026.acl-long.755/.

Varun Magesh, Faiz Surani, Matthew Dahl, Mirac Suzgun, Christopher D. Manning, and Daniel E. Ho. Hallucination-free? Assessing the reliability of leading AI legal research tools. *Journal of Empirical Legal Studies*, 22(2):216–242, 2025. doi: 10.1111/jels.12413. URL https://doi.org/10.1111/jels.12413.

Aashiq Muhamed, Leonardo F. R. Ribeiro, Markus Dreyer, Virginia Smith, and Mona T. Diab. RefusalBench: Generative Evaluation of Selective Refusal in Grounded Language Models. 2025.

Chandan Kumar Sahu, Premith Kumar Chilukuri, and Matthew Hetrich. MiRAGE: A Multiagent Framework for Generating Multimodal Multihop Question-Answer Datasets for RAG Evaluation. 2026.

Haiyang Shen, Hang Yan, Zhongshi Xing, Mugeng Liu, Yue Li, Zhiyang Chen, Yuxiang Wang, Jiuzheng Wang, and Yun Ma. DRAGON: Domain-specific Robust Automatic Data Generation for RAG Optimization. *Findings of the Association for Computational Linguistics: EACL*, 2026.

Yuhong Sun, Joachim Rahmfeld, Chris Weaver, Weijia Chen, Roshan Desai, Wenxi Huang, and Mark H. Butler. EnterpriseRAG-Bench: A RAG Benchmark for Company Internal Knowledge. 2026.

Diane Tchuindjo, Devavrat Shah, and Omar Khattab. OBLIQ-Bench: Exposing Overlooked Bottlenecks in Modern Retrievers with Latent and Implicit Queries. 2026.

Jiamian Wang, Ruiyi Zhang, Tong Yu, Jing Shi, Samyadeep Basu, Rajiv Jain, Zhiqiang Tao, and Tong Sun. DocArena: Turning Raw Documents into Controllable Training Environments for Document Search Agents. 2026a.

Shuaiqi Wang, Aadyaa Maddi, Zinan Lin, and Giulia Fanti. Synae: A framework for measuring the quality of synthetic data for tool-calling agent evaluations. 5 2026b. URL https://arxiv.org/pdf/2605.22564.

Kevin Wu, Eric Wu, Ally Cassasola, Angela Zhang, Kevin Wei, Teresa Nguyen, Sith Riantawan, Patricia Shi Riantawan, Daniel E. Ho, and James Zou. An automated framework for assessing how well LLMs cite relevant medical references. *Nature Communications*, 16:3615, 2025. doi: 10.1038/s41467-025-58551-6. URL https://www.nature.com/articles/s41467-025-58551-6.

Jianan Xie, Xin Sun, Zhongqi Chen, Xing Zheng, Qiang Liu, and Bowen Song. HopRefusalBench: Diagnosing Refusal Failures in Search-Augmented Agents for Multi-Hop Reasoning. 2026.

Yixiao Zeng, Tianyu Cao, Danqing Wang, Xinran Zhao, Zimeng Qiu, Morteza Ziyadi, Tongshuang Wu, and Lei Li. RARE: Retrieval-Aware Robustness Evaluation for Retrieval-Augmented Generation Systems. 2025.

A Appendix

A.1 Comparison to existing methodologies for evaluation set generation

Table 3: Positioning of CoverSynth against representative synthetic generation methods (references in Section 2) on five dimensions: reliance on human annotation or guidance (SME-free), validation of byte-exact text spans (grounding), inclusion of multi-hop and negative questions (multi-hop and negative, respectively), and coverage-aware generation (coverage). ✓: supported; ∼: partial or heuristic; –: not addressed. Each prior method covers a subset of these properties; CoverSynth is distinguished by combining byte-exact grounding with a formal structural approach to generating multi-hop positives and hard negatives under enforced coverage.

Method	SME-free	Grounding	Multi-hop	Negatives	Coverage
InPars	✓	–	–	–	–
Promptagator	∼	–	–	–	–
DataGen	✓	∼	–	–	∼
RARE	✓	∼	✓	–	–
CCQGen	✓	–	–	–	✓
OBLIQ-Bench	∼	∼	–	–	✓
EnterpriseRAG-Bench	–	✓	∼	✓	∼
MiRAGE	✓	∼	–	–	∼
DocArena	✓	∼	∼	–	–
RefusalBench	∼	✓	–	✓	–
HopRefusalBench	∼	✓	✓	✓	–
CoverSynth (ours)	✓	✓	✓	✓	✓

A.2 Corpora Sampling

Baseline domain filtering. Each collection is first restricted to the sub-domain that matches the material enterprise retrieval agents are actually deployed over, rather than to the incidental bulk of the collection. EnterpriseRAG-Bench is filtered on (`source_type`) to Confluence: wiki pages hold the documented, canonical knowledge — runbooks, specifications, internal policy — that an enterprise knowledge agent is typically pointed at, whereas the systems that dominate by volume, Slack and Gmail, are transient conversational traffic. RepLiQA is filtered on (`document_topic`) to Company Policies, the one topic of seventeen that is workplace material rather than local-news reportage, with the filter applied before sampling so the document budget draws within the topic. Harvey contributes the legal knowledge-work setting and is sampled at the level of whole matters rather than files, because a matter is the unit its gold addresses and a partial matter cannot answer a question about it. eCFR contributes the compliance setting, cut to five titles — banking, securities, food and drug, labor, and public welfare — fixed in advance as the regulators a compliance-facing retrieval agent would plausibly be pointed at. TechQA needs no filter: every document is an IBM product support technote, already the enterprise support scenario.

Gold-first seeding and distractors. Where a collection ships human questions, the corpus is seeded with the documents its gold cites rather than taken as an arbitrary slice — an arbitrary slice would leave most human questions unanswerable against the prepared corpus, confounding any generated-versus-human comparison with missing evidence. TechQA, EnterpriseRAG-Bench and RepLiQA are each then filled out with a sampled distractor set held at a matched fraction of 0.40 of the corpus, so that retrieval difficulty is comparable across the three. Harvey takes no separate draw,

being sampled by whole matter, and eCFR ships no questions at all, so in both the remainder of the corpus serves as the distractor pool for any given question. Selection is seeded throughout.

Conversion and segmentation. No collection ships in a format the generator reads natively, so each is converted to Markdown and segmented into chunks — the unit over which gold spans and multi-chunk questions are built. Segmentation uses whatever structure the source provides: markdown headings for Confluence pages, capitalised section labels for TechQA technotes, and numbered clauses for eCFR, whose sections are individually chunk-sized and are therefore aggregated upward into parts so that a document is a complete regulation rather than a fragment. Documents with no recoverable structure fall through to a paragraph-block segmenter. This step is verified rather than assumed: a document that yields no segmentation collapses to a single whole-file chunk, which removes within-document questions silently rather than failing, so we check the realized chunk distribution per corpus. All corpora but RepLiQA segment more than 98% of their documents, at medians of 6 to 22 chunks. RepLiQA averages 1.9, its text being PDF-derived prose with no internal structure, so we treat it as a single-chunk corpus and report within-document breadth only for the others.

A.3 Question intents and negative taxonomy

Table 4: The eight question intents (`scenario_type`) a positive item can realize, each naming the analytical work the answer requires. The generator steers a candidate toward one intent and an adversarial gate confirms the drafted question matches it. *Factual lookup* is the default, used when no other intent fits.

Question intent	**What the answer requires**
Factual lookup	Retrieve a stated fact, definition, value, threshold, name, or reference precisely, as written (the default when no other intent fits).
Compliance	Whether a given situation satisfies a rule, or what is required to satisfy it: eligibility, permissibility, exemption, or which requirement governs the case.
Procedural	How something is done or constructed; the answer is a method, a sequence of steps, or a process.
Diagnostic	Explains an existing state: the reason or mechanism behind a rule or behavior, or where a defect or failure lies.
Impact	What follows from a state or change: its effect, consequence, penalty, or knock-on obligation.
Temporal	Turns on time or order: what must happen before or after what, a stage, duration, waiting period, frequency, or deadline; or how guidance on a topic changed across dated sources.
Comparative	How two concepts relate, differ, or compare, on an axis other than time or source agreement.
Discrepancy resolution	Whether the sources agree: reconciling an inconsistency, deciding which of two positions governs, or confirming they align; requires passages that genuinely diverge.

Table 5: The negative-question taxonomy: four failure families realized by eight generation controls. Answerability and expected behavior are determined by the family. Only the corpus-unsupported family ships an empty gold set; the others cite the passage that makes the item fail.

Failure family	Generation control(s)	Answerability	Desired agent behavior
Corpus-unsupported	Entity, property, condition, time, scope	Unsupported	Abstain with rationale
False premise	Contradicted assertion	Unsupported	Correct the premise
Underspecified	Remove disambiguator	Ambiguous	Request clarification
Ambiguity	Ambiguous reference	Ambiguous	Request clarification

A.4 VERIFICATION PROMPT: REALISM GATE (g_{REAL})

```
You are {persona}.

The question below was written for a file-search assistant that has NOT
↪  seen any source text and must
SEARCH a corpus to answer it. Make TWO independent judgments.

1. self_contained - is the question about the SUBJECT itself rather than
↪  the PAPERWORK? Naming a
   SPECIFIC, real-world entity the question is ABOUT is FINE and often
   ↪  realistic - a court case, statute
   or section citation, ruling, party, product, or a named
   ↪  standard/spec/identifier a practitioner would
   cite ("In [Case v. Party], what did the court hold about [issue]?" is
   ↪  self_contained=TRUE:
   the case is the subject). Mark self_contained=FALSE only if, in ANY
   ↪  phrasing, it:
   - refers to the SOURCE ARTIFACT or how it is organized - "the document
   ↪  / guide / manual / section /
     table / annex", a GENERIC "the standard" / "the guidance" / "the
     ↪  local practice", "which section
     covers X", "where is X documented", "point me to...";
   - asks what a document SAYS instead of the substantive answer (really
   ↪  "where is this written?");
   - refers deictically to an unseen passage ("the scenario described",
   ↪  "as shown", "used here",
     "listed here", "the provided material");
   - only makes sense if the reader is already looking at the document.
   The line: naming the thing the answer is ABOUT (a named case, statute,
   ↪  spec, product, or procedure) =
   subject = TRUE; pointing at the file/section it lives in, or a
   ↪  *generic/unnamed* "the document /
   the standard" = paperwork = FALSE.

2. realistic - would YOU genuinely ask this in day-to-day work? Mark
↪  realistic=FALSE if it is
   over-specified, telegraphs its own answer, pre-lists or enumerates the
   ↪  answer's parts (e.g.
   "(a)...(b)...", "list each...", "...including X, Y and Z", "such as
   ↪  ..."), or is longer / more precise than a
   busy person would type. (Two closely-related clauses like "what does X
   ↪  mean and how is it used?" are
   fine.)

QUESTION: {question}

Respond ONLY JSON: {"self_contained": true|false, "realistic":
↪  true|false,
  "reason": "<short - which check failed and why>",
```

```
  "suggestion": "<a self-contained, natural rephrasing that asks about
  ↪  the subject>"}
```

A.5 Verification Prompt: Own-words critic (lexical gate)

```
You are auditing whether a retrieval question is too EASY - answerable
↪  without really
searching or reasoning over the material. You are shown the question and
↪  the EXACT source passage(s) it
is meant to be answered from.

Mark searchable=true if EITHER of these holds:
A. KEYWORD LIFT - the question reuses the passage's distinctive wording
↪  (its rare multi-word technical
   phrases or sentence structure), so a reader could find the passage by
   ↪  keyword search without
   understanding the concept.
B. ANSWER TELEGRAPHED - the question STATES or LISTS the answer itself:
↪  it names the specific value(s),
   option(s), or alternatives that ARE the answer, so the agent only has
   ↪  to CONFIRM a choice rather than
   find and reason it out. Examples:
   - "...do I use the household-income test or the individual-income
   ↪  test?" when those two ARE the answer
     -> ask "...which income test applies here?"
   - "...do I just carry over the unadjusted baseline rate?" when that IS
   ↪  the answer
     -> ask "...what rate basis do I use for this case?"
   - "...is it 14 days or 30 days?" when the value is the answer -> ask
   ↪  "...what filing deadline applies?"

Mark searchable=false if answering requires understanding: the question
↪  asks about the same underlying
point in DIFFERENT words / framing than the passage AND does not hand
↪  over the answer's values/options.

IMPORTANT: do NOT penalize ordinary domain vocabulary a real user would
↪  naturally use (a term of art,
the standard name for a thing). Penalize only DISTINCTIVE lifted phrasing
↪  or a telegraphed answer. A
question can be perfectly natural and domain-specific while still being
↪  open and in the asker's own words.

QUESTION: {question}

SOURCE PASSAGE(S):
{quotes}

Respond ONLY JSON: {"searchable": true|false, "reason": "<short: 'lift'
↪  or 'telegraphs answer' + why>",
  "suggestion": "<the same question re-asked openly, in the asker's own
  ↪  words, WITHOUT the answer in it>"}
```

A.6 Adversarial Verifier Prompt: Single and Aggregation Queries

```
You are an adversarial grader checking a retrieval-eval item.

QUESTION: {question}

EVIDENCE QUOTES (verbatim from the corpus; each is a separate piece of
↪  evidence, id in brackets):
{quotes}

Decide, using ONLY these quotes:
```

```
1. answerable: can the question be answered correctly from these quotes
↪  alone?
2. sufficient: is the evidence complete (nothing essential missing)?
3. per_doc: for EACH evidence id, does that quote INDEPENDENTLY support
↪  the part of the answer it
   contributes? (catches a "multi-hop" question that is really answerable
   ↪  from a single piece of
   evidence - including two passages from the SAME document.)
4. necessary: for EACH quote in order, is it actually needed (true) or
↪  redundant/spurious (false)?
5. leak: does the QUESTION give away the answer, or point at WHERE in the
↪  source it is
   written (a line or page number, "see the table below", "in section 4
   ↪  of the guide")? Naming a
   real-world entity the question is ABOUT - a case, a statute or section
   ↪  citation, a named
   standard - is NOT a leak; that is how a practitioner
   ↪  asks.{intent_criterion}

Respond with ONLY JSON:
{"answerable": bool, "sufficient": bool, "per_doc": {"<evidence_id>":
↪  bool},
  "necessary": [bool, ...], "leak": bool,{intent_field} "notes":
  ↪  "<short>"}
```

A.7 Adversarial Verifier Prompt: Bridge (multi-hop) Queries

```
You are an adversarial grader checking a MULTI-HOP (scavenger-hunt)
↪  retrieval item.
The evidence quotes are given IN INTENDED CHAIN ORDER: each earlier quote
↪  should name a "bridge" entity
you must resolve before you can even locate the next quote.

QUESTION: {question}

EVIDENCE QUOTES (verbatim, in chain order, id in brackets):
{quotes}

Bridge entities the question must NOT name (naming one lets the agent
↪  skip a hop): {bridges}

Decide, using ONLY these quotes:
1. answerable: can the question be answered correctly from these quotes
↪  together?
2. sufficient: is the evidence complete (nothing essential missing)?
3. dependency: is at least one OTHER quote genuinely REQUIRED to reach
↪  the answer - either because it
   supplies a distinct necessary fact, OR because it DECODES a
   ↪  term/acronym or IDENTIFIES an entity the
   question uses that you must resolve to interpret the question or
   ↪  LOCATE the answer? Answer TRUE if
   removing that quote would leave a searcher unable to correctly
   ↪  identify, interpret, or find the
   answer - EVEN IF the quote that states the answer reads as
   ↪  self-contained once you are looking at it.
   Answer FALSE only if every other quote is genuinely redundant (e.g.
   ↪  the answer quote already uses the
   resolved/expanded form, so nothing needed decoding, and no other quote
   ↪  is needed to locate it).
   Example of a REAL dependency: the question uses an acronym; quote A
   ↪  spells it out; quote B states the
   rule using the full form - B alone never tells you what the acronym
   ↪  means, so A is required.
4. bridge_leak: does the QUESTION hand over the RESOLVED form of a
↪  linking entity - its expansion /
```

```
   definition, or the specific intermediate identity - so the reader can
   ↪  skip a hop and jump straight to
   the answer? (Merely USING an acronym or an unresolved term is FINE -
   ↪  that IS the hop; giving its full
   form or definition is the leak.)
5. necessary: for EACH quote in order, is it actually needed to traverse
↪  the chain (true) or
   redundant/spurious (false)? An intermediate quote whose removal still
   ↪  leaves the chain resolvable is
   NOT necessary.
6. leak: does the QUESTION give away the answer, or point at WHERE in the
↪  source it is
   written (a line or page number, "see the table below", "in section 4
   ↪  of the guide")? Naming a
   real-world entity the question is ABOUT - a case, a statute or section
   ↪  citation, a named
   standard - is NOT a leak; that is how a practitioner
   ↪  asks.{intent_criterion}

Respond with ONLY JSON:
{"answerable": bool, "sufficient": bool, "dependency": bool,
↪  "bridge_leak": bool,
  "necessary": [bool, ...], "leak": bool,{intent_field} "notes":
  ↪  "<short>"}
```

A.8 Retrieval System Configurations

Indexed baselines. BM25, dense retrieval, and hybrid retrieval share a common passage segmentation: documents are split into passages of at most 250 tokens using the `cl100k_base` tokenizer, with up to 50 tokens of overlap. Each baseline returns up to 10 ranked passages as evidence. BM25 uses the BM25S implementation with Lucene scoring ($k_1 = 0.9$, $b = 0.4$), lowercasing, English stopword removal, and Snowball English stemming. Dense retrieval uses `all-MiniLM-L6-v2` embeddings.

Hybrid retrieval. Hybrid retrieval retrieves up to 20 candidates from each of the same BM25 and dense retrievers, then combines their rankings using equal-weight reciprocal-rank fusion with a rank constant of 60. A passage at rank r contributes $1/(60 + r)$ to its fused score from each candidate list in which it appears, with ranks starting at one. The top 10 passages under the fused ranking are returned without an additional reranking stage.

Agent harnesses and budgets. The local ReAct and Claude Code harnesses each run with Haiku-4.5, Sonnet-5, and Opus-5, yielding six agent configurations. Agents iteratively search and inspect the corpus and select supporting passages as evidence. Agent runs are capped at 50 turns and a 300 s wall-clock timeout, with one retry permitted for rate-limit errors.

A.9 Construction profiles and exact localization

Table 6 retains the corpus-specific document and community counts. Community counts depend on corpus size, the concept representation, and clustering settings; they are not a controlled measure of domain complexity. Table 7 reports realized properties of the current generated TechQA/RepLiQA files. Multi-document counts use distinct gold source files; multi-hop counts use the recorded dependency-depth labels and do not reestablish the semantic necessity of every bridge.

Table 6: Reported Leiden community counts (communities with at least two members) on the corpus subsets used for the construction profiles.

Corpus	Documents	Communities
ERAG-Bench	95	95
Harvey FK	137	515
TechQA	500	176
eCFR	120	398
RepLiQA	275	65

Table 7: CoverSynth-generated set composition. Structural counts and AD use the 100 answerable items per corpus; intent presence uses all records. AD is normalized over the declared 32-cell grid. Negative-family counts refer to distinct items, not repeated runs.

Property	TechQA	RepLiQA
All items	128	147
Answerable items	100	100
Multi-document answerable items	50	72
Labeled multi-hop answerable items	34	11
Occupied capability cells	28/32	19/32
AD	0.906	0.768
Intents present	7/8	7/8
Corpus-unsupported negatives	12	13
False-premise negatives	9	9
Underspecified negatives	5	12
Ambiguous negatives	2	13

The offset check covers every gold quote in the four current TechQA/RepLiQA files, including evidence records attached to negatives: 244 generated TechQA quotes, 405 generated RepLiQA quotes, and 280 human-reference quotes, for 929 matches with no missing sources or mismatched substrings. This establishes occurrence at the stored character offsets. It does not test whether an answer is correct, whether a passage is necessary, or whether a query is representative of user traffic.

The human ERAG retrieval subset contains 73 items, including 53 answerable items; 43 of those answerable items cite multiple documents. Consequently, the single-document structure of the human TechQA/RepLiQA subsets cannot be generalized to human retrieval benchmarks as a whole.

A.10 Coverage profiling

The MiRAGE SCC/ZQC profiles on Harvey and eCFR use the implementation accompanying Klearman et al. (2026). These measure question coverage, independently of the chunk retrieval scores below. The assessment partition $\mathcal{K}$ in Section 4 need not be the graph partition used for generation. Interpreting the profiles requires a common document inventory, assessment partition, and query-assignment rule within each corpus comparison.

A.11 Difficulty analysis.

For the historical backbone panel, we fit a two-parameter logistic item response model. System a succeeds on item i with probability

$$P(Y_{ai} = 1 \mid \theta_a, \alpha_i, \beta_i) \;=\; \sigma(\alpha_i(\theta_a - \beta_i)) \;=\; \frac{1}{1 + \exp(-\alpha_i(\theta_a - \beta_i))}, \tag{7}$$

where θ_a is latent system ability, β_i item difficulty, and α_i item discrimination. Maximum a posteriori estimation uses a standard-normal ability prior and a zero-mean ridge prior on (α_i, β_i). Span F_2 is binarized at the panel's mean-matching threshold before fitting.

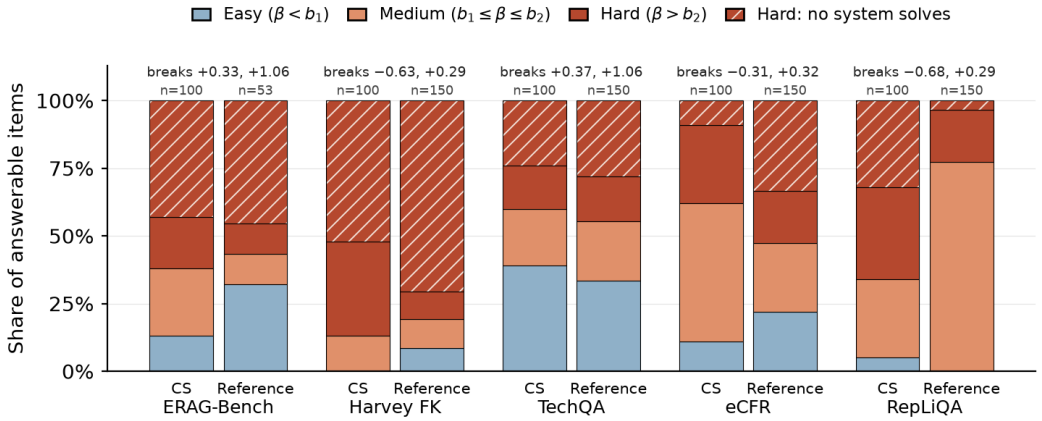

Figure 4: Difficulty distribution of answerable questions for CoverSynth and each reference set. Difficulty is IRT item difficulty β from a joint fit per corpus over six systems (three indexed retrievers and three Claude Code agents). Cut-offs for each corpus's difficulty category were determined using Jenks natural breaks optimization. Hatched: hard questions that no system solves.

Table 8: Shape of the item difficulty distribution on answerable questions. An item's failure rate is the share of the six panel systems whose binarized span-F_2 falls below the corpus's pooled mean-matching threshold. Skew is the adjusted Fisher–Pearson sample skewness of item failure rates; positive values indicate mass at the easy end with a tail of harder items. Δ skew gives CoverSynth minus *reference* with a 95% bootstrap interval (2,000 resamples, drawing items independently within each set).

Corpus	Set	Skew	Δ skew [95% CI]
ERAG-Bench	CoverSynth Reference	-0.69 -0.90	$+0.21$ $[-0.54, +0.92]$
Harvey FK	CoverSynth Reference	-0.67 -1.60	$+0.93$ $[+0.37, +1.53]$
TechQA	CoverSynth Reference	$+0.27$ -0.09	$+0.35$ $[-0.21, +0.90]$
eCFR	CoverSynth Reference	-0.29 -0.43	$+0.14$ $[-0.27, +0.77]$
RepLiQA	CoverSynth Reference	-0.36 $+2.31$	-2.67 $[-3.38, -1.98]$

A.12 Oracle-reader probes and evidence granularity

Table 2 reports reader probes, which are distinct from retrieval precision, recall, and F_1. The gold evidence units for CoverSynth are localized spans. Sufficiency depends on the amount of supplied evidence, uniqueness on the chosen distractors, and necessity on the unit removed. A whole-chunk deletion and a localized-quote deletion therefore test different interventions. We do not infer relative evidence quality across unmatched representations. MiRAGE is excluded from these oracle comparisons because its native annotations identify chunks and a matched span-level reader protocol has not been established.

Let the oracle reader's correctness on evidence S be $r(S) \in \{0, 1\}$, and let S_i^* be item i's gold evidence units and D_i its distractor passages. Over N probed items, *sufficiency* is the share answered correctly from gold alone: $\text{Sufficiency} = \frac{1}{N}\sum_{i=1}^{N} r(S_i^*)$. Conditional on gold sufficiency, *uniqueness* is the share answered incorrectly from distractors alone: $\text{Uniqueness} = \sum_{i:r(S_i^*)=1} \mathbf{1}[r(D_i) = 0] \,/\, \sum_{i=1}^{N} r(S_i^*)$. *Necessity* is the fraction of gold units in sufficient items whose removal makes the

oracle incorrect:

$$\text{Necessity} = \frac{\sum_{i:\, r(S_i^*)=1} \sum_{s \in S_i^*} \mathbf{1}[\, r(S_i^* \setminus \{s\}) = 0\,]}{\sum_{i:\, r(S_i^*)=1} |S_i^*|} \tag{8}$$

Human TechQA sufficiency (0.61) illustrates that oracle answerability limits also affect human-authored sets. A bridge passage may help an agent locate an answer while being unnecessary to a reader already given the final answer passage; redundant parallel evidence can also lower leave-one-out necessity. These probes are not estimates of human annotation error or proofs of global answer uniqueness.

A.13 Five-corpus retrieval measurements

Tables 9, 10 and 11 report retrieval performance on answerable questions from CoverSynth, native benchmark references and MiRAGE, respectively, using Claude Code with Opus-5, Sonnet-5 and Haiku-4.5. The generated and reference arms were run separately, with turn budgets of 50 (CoverSynth) and 40 (native references and MiRAGE). All three tables exclude failed runs. In the CoverSynth arm, Haiku excludes 12 failed answerable runs on TechQA, 10 on RepLiQA, 9 on ERAG-Bench, 6 on eCFR and one on Harvey, and Sonnet one on RepLiQA. In the native-reference arm, TechQA Haiku excludes six, and RepLiQA includes 20 answerable items without span gold. MiRAGE Harvey Haiku excludes one failure, leaving 149 items; the other MiRAGE rows contain 150. Excluding failures raises the scores of models that fail, mainly Haiku: generated TechQA Haiku span F_1 is 0.40 with exclusion and 0.35 with failures scored zero. These tables are descriptive, not a matched comparison. The repeated comparison in Section 6.3 matches systems across evaluation sets, requires span gold for retrieval scoring, and applies consistent failure handling.

MiRAGE chunk scoring. MiRAGE identifies gold by native chunk IDs. Each final predicted span is mapped to the native chunk containing its midpoint, and repeated chunk IDs count once. Table 11 reports the resulting chunk precision, recall and F_1 separately from character-overlap span metrics. With whole-chunk gold, a short quote inside a relevant chunk can have high character precision but recover few gold characters; predictions in non-gold chunks still incur false positives. Chunk scores measure recovery of evidence regions and are not directly comparable to CoverSynth's span scores. Different gold granularity and execution budgets preclude attributing cross-table differences to benchmark difficulty alone.

Figure 5 separately summarizes the original single-pass comparison of indexed retrievers, ReAct agents and Claude Code agents. Its scoring populations are described in the caption.

Table 9: Retrieval performance on the answerable questions from the five-corpus CoverSynth outputs against a Claude code harness for three differently sized Claude models. File and span retrieval metrics include average precision (P), recall (R), and F1. Resource use is the mean number of tool calls and of evidence-context tokens per question; the turn budget is 50. Failed runs are excluded (Appendix A.13).

		File			Span			Resource use	
Corpus	**Model**	**P**	**R**	**F1**	**P**	**R**	**F1**	**Tool Calls**	**Tokens (k)**
ERAG-Bench	Opus-5	0.52	0.62	0.53	0.17	0.47	0.23	13.7	11.0
	Sonnet-5	0.59	0.55	0.54	0.26	0.41	0.29	18.4	13.9
	Haiku-4.5	0.43	0.40	0.39	0.20	0.29	0.22	30.0	30.1
Harvey FK	Opus-5	0.46	0.66	0.51	0.19	0.38	0.23	7.6	7.9
	Sonnet-5	0.50	0.54	0.49	0.20	0.29	0.22	6.7	8.7
	Haiku-4.5	0.42	0.40	0.38	0.23	0.19	0.19	12.3	24.0
TechQA	Opus-5	0.88	0.86	0.86	0.34	0.67	0.41	7.0	2.3
	Sonnet-5	0.82	0.77	0.79	0.35	0.53	0.39	10.6	3.5
	Haiku-4.5	0.79	0.73	0.75	0.43	0.45	0.40	23.8	16.1
eCFR	Opus-5	0.99	0.98	0.98	0.58	0.79	0.64	6.1	2.7
	Sonnet-5	0.96	0.93	0.94	0.71	0.63	0.63	6.7	4.1
	Haiku-4.5	0.91	0.89	0.90	0.72	0.57	0.60	16.5	13.2
RepLiQA	Opus-5	0.67	0.74	0.68	0.38	0.51	0.41	8.4	3.7
	Sonnet-5	0.63	0.62	0.61	0.40	0.41	0.38	12.2	3.4
	Haiku-4.5	0.64	0.65	0.63	0.43	0.40	0.38	18.8	7.5

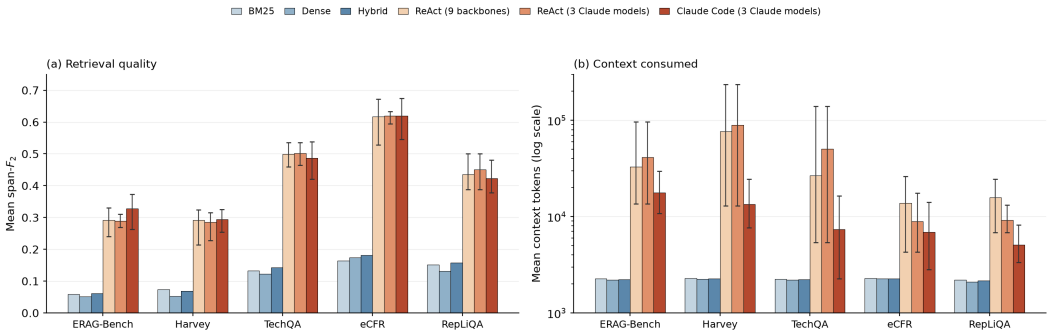

Figure 5: Five-corpus system comparison: (a) mean span F_2 and (b) mean context tokens, on a logarithmic scale. Agent bars summarize the broader ReAct backbone panel, the three-Claude ReAct panel and the three-Claude Claude Code panel; whiskers show the within-group range. The three-Claude agent groups on TechQA and RepLiQA include negative questions, with credit for correct abstention; the tables and repeated study report retrieval scores on answerable questions.

Table 10: Retrieval on native ERAG-Bench, TechQA and RepLiQA reference sets using Claude Code. File and span metrics report mean precision (P), recall (R), and F_1. Resource use is the mean number of tool calls and evidence-context tokens per question; the turn budget is 40. Failed runs are excluded (Appendix A.13). ERAG reference gold is document-level, so span metrics are omitted. MiRAGE chunk metrics appear separately in Table 11.

		File			**Span**			**Resource use**	
Corpus	**Model**	**P**	**R**	**F1**	**P**	**R**	**F1**	**Tool Calls**	**Tokens (k)**
ERAG-Bench	Opus-5	0.95	0.88	0.90	—	—	—	10.8	8.8
	Sonnet-5	0.91	0.67	0.73	—	—	—	9.9	7.3
	Haiku-4.5	0.94	0.56	0.66	—	—	—	10.5	10.9
TechQA	Opus-5	0.83	0.91	0.86	0.23	0.72	0.32	8.1	2.5
	Sonnet-5	0.77	0.80	0.78	0.24	0.61	0.31	11.4	4.5
	Haiku-4.5	0.79	0.81	0.80	0.28	0.51	0.32	17.2	13.4
RepLiQA	Opus-5	0.98	0.99	0.98	0.61	0.88	0.67	3.9	1.2
	Sonnet-5	0.98	0.99	0.98	0.74	0.85	0.74	3.8	0.81
	Haiku-4.5	0.97	0.98	0.97	0.77	0.73	0.70	4.0	1.5

Table 11: MiRAGE retrieval using native chunk gold on Harvey and eCFR. File and chunk precision (P), recall (R) and F_1 use the evidence units named in their column groups. Chunk scores are not directly comparable to character-overlap span scores. Claude Code uses a 40-turn budget. N counts successful answerable runs; Harvey Haiku excludes one failure. Resource columns report mean tool calls and evidence-context tokens.

			File			Chunk			Resource use	
Corpus	Model	N	P	R	F_1	P	R	F_1	Calls	Tokens (k)
Harvey FK	Opus-5	150	0.673	0.616	0.596	0.317	0.491	0.335	7.4	7.740
	Sonnet-5	150	0.621	0.485	0.498	0.350	0.365	0.310	6.3	8.489
	Haiku-4.5	149	0.636	0.455	0.486	0.411	0.316	0.309	6.4	17.632
eCFR	Opus-5	150	0.997	0.977	0.982	0.625	0.828	0.651	5.4	2.435
	Sonnet-5	150	0.907	0.880	0.889	0.712	0.721	0.665	4.9	3.306
	Haiku-4.5	150	0.920	0.890	0.900	0.753	0.700	0.671	8.4	12.202

A.14 Focused repeated-study protocol

Populations and scoring. The two evaluation sets for each corpus share the prepared document collection. Generated TechQA and RepLiQA evaluation sets each contribute 100 answerable items with span gold. Human TechQA contributes 150; human RepLiQA contributes 130 after excluding 20 answerable items with no span gold. Negatives are scored separately. Span precision and recall measure character overlap between located predicted and gold spans, pooled across files within an item; span F_2 weights recall more than precision. Scores are averaged over repeats, then over items. File recall measures the share of gold documents from which a located snippet appears in the final citations. Failed runs remain in the denominator with score zero. Complete document recall in Table 12 requires every system to cite all gold documents in every repeat, not merely to have read a gold document during its trajectory.

Panels. The nine-system panel combines BM25, dense and hybrid retrieval with six base agents: ReAct and Claude Code with Haiku-4.5, Sonnet-5 and Opus-5, each run three times with 50 turns and 300 s per question (Appendix A.8). The deterministic indexed runs are reused across repeats. Three additional single-run agent configurations are reported separately in Appendix A.16 and are excluded from the main analysis.

Table 12: Focused comparison on answerable items with span gold. The first block uses all nine systems; complete document recall requires every system to cite all gold documents in every repeat. The second block uses the six base agents. A pair meets the detection criterion when nominal paired tests detect a difference in at least 80% of 400 resamples of 100 items with replacement.

	TechQA		RepLiQA	
Metric	CoverSynth	Human	CoverSynth	Human
All nine systems, three repeats				
Complete document recall by all systems	23/100	75/150	12/100	118/130
Best-system mean span F_2	0.52	0.47	0.47	0.75
Six base agents, three repeats				
Span-F_2 reliability $G(100)$	0.94	0.74	0.93	0.86
Pairs meeting detection criterion	8/15	3/15	8/15	4/15

Headroom and score interpretation. Universal complete document citation occurs on 50% of human TechQA items versus 23% of generated items, and on 91% versus 12% for RepLiQA (Table 12). The best observed mean span F_2 is 0.47 versus 0.75 on RepLiQA but 0.52 versus 0.47 on TechQA (CoverSynth versus human). Added document-citation headroom therefore does not imply uniformly greater difficulty under every metric. Multiple gold documents and lower lexical overlap are plausible contributors, not isolated causal effects. Document recall concerns final citations, not every document an agent may have read.

The per-system score intervals in Figure 3 use 1,000 bootstrap resamples of the eligible items after averaging each system's repeats. They describe item-sampling uncertainty conditional on those runs; they do not estimate variability across new runs, regenerated questions or new corpora. The two evaluation sets contain different questions, so the displayed means are not a matched-question comparison. These intervals are distinct from the reliability and paired-contrast intervals below.

Reliability and uncertainty. Relative generalizability is estimated from a systems $\times$ items $\times$ repeats variance decomposition. Projected to $N = 100$ items and R repeats,

$$G(N, R) = \frac{\sigma_s^2}{\sigma_s^2 + \sigma_{si}^2/N + \sigma_{sr}^2/R + \sigma_e^2/(NR)}. \tag{9}$$

Here s, i and r index systems, items and repeats; estimated variance components are truncated at zero. The six-agent analysis uses $R = 3$. Confidence intervals resample items and, when present, repeats with replacement. Pooled ΔG is the arithmetic mean of the TechQA and RepLiQA differences; it does not represent sampling across a larger population of corpora. G is a conditional variance-based reliability coefficient, not a probability that a ranking is correct. Adding the three indexed retrievers to the six base agents drives $G(100)$ to 0.976–0.997 on all four sets, because the retriever–agent gap dominates system variance; at this ceiling ΔG is slightly negative (Table 13). Separating retrievers from agents is therefore not a distinctive property of either set, and the reliability comparison is informative only within the agent panel.

Table 13: Span-F_2 reliability and sensitivity to panel choice. CS = CoverSynth; Hu = human. All estimates project to 100 items. Intervals are 95% bootstrap intervals for $\Delta G = G_{\mathrm{CS}} - G_{\mathrm{Hu}}$; pooled rows average the two corpus differences. The last panel adds BM25, dense, and hybrid retrieval to the six base agents (three repeats).

Panel	Corpus	G_{CS}	G_{Hu}	ΔG	95% interval
Six agents, three repeats	TechQA	0.942	0.739	+0.203	$[+0.050, +0.659]$
	RepLiQA	0.927	0.857	+0.070	$[-0.056, +0.337]$
	Pooled	—	—	+0.136	$[+0.041, +0.414]$
Six agents + retrievers	TechQA	0.986	0.988	-0.002	$[-0.011, +0.005]$
	RepLiQA	0.976	0.997	-0.022	$[-0.039, -0.016]$
	Pooled	—	—	-0.012	$[-0.022, -0.008]$

Contrasts and pair detection. Capability contrasts compare all three pairs of Haiku, Sonnet, and Opus within each of two harnesses on each corpus. Their nominal 95% intervals bootstrap items and repeats; the 12 intervals per evaluation-set type are correlated and not multiplicity-adjusted. All generated-set contrasts are positive in each repeat. Pair detection is a separate calculation: each system's item score is averaged over repeats, and 400 samples of 100 items are drawn with replacement from the full eligible set. A two-sided paired t-test uses nominal $\alpha = 0.05$; a pair meets the criterion if at least 80% of draws detect a difference. These are estimated detection frequencies. Several pairs lie close to the cutoff: generated RepLiQA ReAct Sonnet–Opus (0.805) and Claude Code Sonnet–Opus (0.84) meet it, whereas generated TechQA ReAct Haiku–Sonnet (0.775), generated TechQA ReAct-Haiku versus Claude Code-Haiku (0.76) and generated RepLiQA Claude Code Haiku–Sonnet (0.755) do not. Pair counts are therefore sensitive to the threshold. The procedure is distinct from the separately computed Holm-adjusted all-pairs significance tests, and nondetection does not establish equivalence.

All 12 within-harness model contrasts have positive nominal 95% intervals on the generated sets, compared with four positive intervals, one negative interval, and seven crossing zero on the human sets. More six-agent pairs meet the separate detection criterion on the generated sets (Table 12), but no same-model ReAct–Claude Code pair meets it on either generated set (the closest is generated TechQA Haiku at 0.76). These results locate separation in the tested model contrasts; expected model order does not establish benchmark validity, and nondetection of harness differences does not establish harness equivalence.

Detailed scores. Tables 14 and 15 report each system's precision, recall, F_2, and cited-document recall. Retriever means average the three indexed methods; base-agent means average the six ReAct/Claude Code configurations. In all four corpus/set combinations, retrievers have higher mean span recall but lower F_2 than the base agents: recall is 0.69 versus 0.54 on generated TechQA and 0.96 versus 0.80 on human RepLiQA, for example. Broad returned chunks incur low precision; agents can delimit shorter spans. Citation style also interacts with gold length. A controlled output-budget comparison would be needed to isolate the causal effect of citation length, so F_2 should be read alongside precision, recall, and cited-document recall.

Table 14: TechQA: mean scores on answerable items with span gold (CS = CoverSynth, Hu = human; n = 100 CS, 150 human). Reps = independent runs over the full evaluation set (not agent turns); agent scores average their repeats, and failed runs score 0. CC-Opus-5.5 and CC-Max are single-run extensions outside the nine-system panel (Appendix A.16).

System	Reps	Span F2		Span P		Span R		File R	
		CS	Hu	CS	Hu	CS	Hu	CS	Hu
BM25	1	0.16	0.13	0.04	0.03	0.72	0.83	0.83	0.95
Dense	1	0.15	0.12	0.04	0.03	0.61	0.77	0.72	0.95
Hybrid	1	0.17	0.12	0.05	0.03	0.75	0.82	0.84	0.97
ReAct-Haiku	3	0.39	0.47	0.39	0.31	0.43	0.64	0.65	0.86
ReAct-Sonnet	3	0.46	0.41	0.36	0.23	0.57	0.63	0.78	0.78
ReAct-Opus	3	0.52	0.45	0.32	0.24	0.69	0.73	0.89	0.90
CC-Haiku	3	0.34	0.41	0.37	0.30	0.37	0.53	0.63	0.83
CC-Sonnet	3	0.46	0.40	0.38	0.24	0.53	0.59	0.76	0.78
CC-Opus	3	0.50	0.43	0.32	0.23	0.67	0.70	0.88	0.88
CC-Opus-5.5	1	0.48	0.47	0.37	0.29	0.57	0.67	0.86	0.89
CC-Max-Haiku	1	0.42	0.45	0.42	0.31	0.46	0.60	0.79	0.90
CC-Max-Opus-5.5	1	0.50	0.51	0.40	0.32	0.58	0.70	0.89	0.93

Table 15: RepLiQA: mean scores on answerable items with span gold (CS = CoverSynth, Hu = human; n = 100 CS, 130 human). Reps = independent runs over the full evaluation set (not agent turns); agent scores average their repeats, and failed runs score 0. CC-Opus-5.5 and CC-Max are single-run extensions outside the nine-system panel (Appendix A.16).

System	Reps	Span F2		Span P		Span R		File R	
		CS	Hu	CS	Hu	CS	Hu	CS	Hu
BM25	1	0.19	0.15	0.06	0.03	0.56	0.98	0.71	0.99
Dense	1	0.16	0.15	0.05	0.03	0.49	0.91	0.69	0.99
Hybrid	1	0.20	0.16	0.06	0.04	0.58	0.99	0.78	1.00
ReAct-Haiku	3	0.36	0.70	0.35	0.72	0.39	0.75	0.59	0.98
ReAct-Sonnet	3	0.42	0.74	0.40	0.70	0.45	0.82	0.69	0.99
ReAct-Opus	3	0.47	0.73	0.35	0.54	0.57	0.87	0.82	0.98
CC-Haiku	3	0.32	0.66	0.37	0.73	0.33	0.69	0.57	0.98
CC-Sonnet	3	0.38	0.75	0.42	0.72	0.39	0.81	0.63	0.99
CC-Opus	3	0.44	0.73	0.36	0.56	0.49	0.85	0.77	0.99
CC-Opus-5.5	1	0.42	0.73	0.45	0.74	0.44	0.77	0.75	1.00
CC-Max-Haiku	1	0.35	0.62	0.37	0.67	0.37	0.67	0.70	0.99
CC-Max-Opus-5.5	1	0.46	0.70	0.47	0.67	0.48	0.75	0.79	1.00

A.15 Repeat variability and failure robustness

The standard deviation of each base system's mean across repeats ranges from 0.002 to 0.022. Removing every item on which any base agent failed retains 66 generated and 128 human TechQA items, and 74 generated and 130 human RepLiQA items. The resulting $G(100)$ point estimates remain higher on the generated sets: 0.91 versus 0.74 on TechQA and 0.92 versus 0.86 on RepLiQA. These are projections to 100 items from the retained populations, not additional significance results. Repeating retrieval evaluation does not measure variability from regenerating the questions or choosing new corpora.

Corpus preparation and prospective uses. Corpus preparation uses existing gold documents where available, although generation does not require expert-authored questions as input. The experiments do not isolate the contribution of closed-book answerability. Continuous regeneration and training-data applications remain prospective uses; neither a corpus-update benefit nor a training gain is measured here.

A.16 Single-run agent extensions

Three further agent configurations were run once on the two-corpus study. They are excluded from the main panel because they have one repeat instead of three and, for CC-Max, a different prompt, architecture, and budget. *CC-Opus-5.5* is the base Claude Code agent with Opus-5.5. *CC-Max* is a Claude Code orchestrator that can only delegate: a hook refuses its own file-reading and search tools, so retrieval is performed by scout, reader, and verifier subagents of the same model (Haiku-4.5 or Opus-5.5), guided by a retrieval playbook, with 100 turns and 900 s per question. Answer and abstention instructions match the base agents. Scores appear in Tables 14 and 15. CC-Opus-5.5 does not exceed CC-Opus-5 on either generated set (0.48 versus 0.50 span F_2 on TechQA, 0.42 versus 0.44 on RepLiQA).

Adding the three extensions to the six base agents requires averaging repeats per system and item and using a systems $\times$ items decomposition. The reliability advantage then shrinks: $G(100)$ is 0.916 versus 0.792 on TechQA ($\Delta G = +0.124$, 95% interval $[+0.009, +0.239]$) and 0.911 versus 0.903 on RepLiQA ($+0.008$, $[-0.063, +0.085]$), and the pooled difference, $+0.066$ $[-0.001, +0.135]$, includes zero.

CC-Max illustrates a reducible failure mode under a changed prompt, retrieval procedure and budget. On answerable generated items, base CC-Haiku ends without an answerable verdict and without citing a gold document in 82/300 TechQA runs and 66/300 RepLiQA runs, compared with 9/100 and 8/100 for CC-Max-Haiku. Base turn-cap failures account for 38/300 and 32/300 runs; CC-Max

has three timeout failures per 100 runs on each corpus. Because the base has three repeats and CC-Max one, with differing instructions and budgets, this does not isolate an architecture effect.

A.17 NEGATIVE-QUERY RESULTS AND ORIGINAL-FIGURE CONVENTIONS

Table 16 reports the answerability-verdict accuracy for a focused analysis of TechQA and RepLiQA, counting failures as incorrect. The generated families contain only 2–13 items per corpus (Table 7), so rates can move substantially when a small number of judgments change. These results concern verdict agreement, not the quality of a follow-up clarification or premise correction.

The heatmap (Figure 2) uses single-pass Claude Code summaries for all five corpora and excludes failed runs in every cell. Its displayed values must therefore not be combined with the repeated, failure-inclusive table as though they shared one denominator. Correct abstention on an empty-gold negative can receive a span score of one for an agent, whereas the retrievers do not return an answerability verdict. This is an additional reason that the agent–retriever F_2 gap cannot be interpreted solely as better evidence retrieval.

Table 16: Answerability-verdict accuracy on negatives, by failure family (CoverSynth) and on the human 'not in corpus' negatives. Failed runs count as wrong; agents averaged over repeats. CC-Opus-5.5 and CC-Max are single-run extensions (Appendix A.16).

	TechQA					RepLiQA				
System n items	Unsupp. 12	False prem. 9	Underspec. 5	Ambig. 2	Hu unsupp. 120	Unsupp. 13	False prem. 9	Underspec. 12	Ambig. 13	Hu unsupp. 38
ReAct-Haiku	0.67	1.00	0.00	0.00	0.54	0.79	0.96	0.06	0.08	0.35
ReAct-Sonnet	0.75	0.89	0.00	0.00	0.79	0.95	0.85	0.14	0.21	0.39
ReAct-Opus	0.81	1.00	0.00	0.00	0.71	0.97	1.00	0.25	0.64	0.43
CC-Haiku	0.81	0.85	0.13	0.00	0.66	0.72	0.85	0.06	0.05	0.26
CC-Sonnet	0.83	0.96	0.00	0.00	0.83	0.92	1.00	0.31	0.36	0.45
CC-Opus	0.89	1.00	0.13	0.00	0.83	0.95	1.00	0.42	0.62	0.47
CC-Opus-5.5	0.92	1.00	0.00	0.50	0.78	0.85	1.00	0.33	0.69	0.34
CC-Max-Haiku	0.75	0.78	0.00	0.00	0.60	0.69	1.00	0.08	0.23	0.21
CC-Max-Opus-5.5	0.92	1.00	0.00	0.00	0.78	0.92	1.00	0.17	0.46	0.45