

Longitudinal-aware Evaluation of Retrieval-Augmented Generation over Evolving Credit Policy Documents

Kyosuke Morita Jack Sanderson Holly Hewitt

Lloyds Banking Group

{Kyosuke.Morita, Jack.Sanderson, Holly.Hewitt}@lloydsbanking.com

Abstract

We present a retrieval-augmented generation (RAG) system in production since April 2026 for credit policy question answering, a setting in which the corpus is revised several times per quarter: a single revision changes thresholds, narrows scopes, and restructures tables. Static test sets silently decay under this regime, and scores computed against successive corpus versions cannot separate system drift from corpus drift. We propose a longitudinal-aware evaluation framework that makes RAG quality over an evolving corpus both measurable and attributable. It combines (i) a change-type taxonomy — Stable, Evolved, Added, Removed — built on a deterministic, model-independent alignment of document subsections keyed to how each gold-supporting provision changed between versions, making explicit which scores remain comparable across releases; (ii) a versioned, class-stratified synthetic QA pipeline that keeps the test set current at revision cadence; and (iii) a statement-level LLM judge for compound policy answers; each component is calibrated against subject-matter experts. Across three corpus versions, a conventional longitudinal-unaware protocol would report a recall decline of up to 25.7 points. However, paired comparisons on answer-invariant items show no evidence of regression in the unchanged system.

1 Introduction

Retrieval-augmented generation (RAG) (Lewis et al., 2020) is standard for question answering over proprietary document collections, and is increasingly deployed in regulated domains such as finance and law, where an answer requires traceability to authoritative sources. We present a RAG system deployed for credit policy question answering, in which relationship managers and credit officers query a corpus of credit policy documents to determine product eligibility, applicable thresholds,

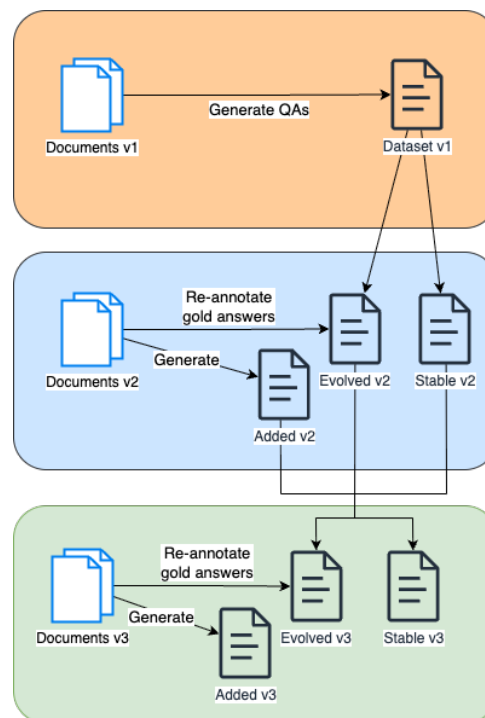


Figure 1: Longitudinal-aware dataset construction process.

and qualifying conditions. The system has been in production since April 2026 and has served more than 600 queries in the last month with a median latency of 17 seconds and a marginal cost of $\approx$ \$0.03 per query.

Credit policy documents are revised several times per quarter so the dynamics differ from the canonical setting studied in time-sensitive QA, where individual facts change against a stable document. We address this with a longitudinal evaluation framework that makes RAG quality over an evolving corpus measurable and attributable by introducing a change-type taxonomy that labels every question STABLE, EVOLVED, ADDED, or REMOVED. The subsection alignment underlying the taxonomy is deterministic and model-independent.

Our contributions are summarised as follows:

- **A measurement instrument for longitudinal RAG evaluation.** A change-type stratification built on a deterministic, model-independent subsection alignment, keyed to how each gold-supporting provision changed between versions, which separates system drift from corpus drift and makes explicit which scores remain comparable as the corpus is revised.
- **A maintainable, expert-validated instrument.** A versioned, class-stratified synthetic QA pipeline and a statement-level LLM judge, both calibrated against subject-matter experts, enabling evaluation at a revision cadence with minimal manual intervention.
- **Deployment findings on a production corpus.** A conventional protocol would misreport a recall decline of up to 25.7 points for which the paired anchor finds no evidence of regression.

2 Related Work

Time-sensitive QA asks whether a model returns the temporally correct answer as facts evolve: TimeQA (Chen et al., 2021) constructs question-answer (QA) pairs over time-evolving facts from WikiData, and SituatedQA identifies and re-annotates questions whose answers depend on temporal or geographical context (Zhang and Choi, 2021). StreamingQA evaluates semi-parametric models that ingest time-stamped documents on a fixed schedule (Liška et al., 2022). Recent work carries the setting into retrieval—TG-RAG models a corpus as a temporal graph under an incremental-update protocol (Han et al., 2025) and HoH measures the effect of outdated information on RAG answers (Ouyang et al., 2025). These settings exhibit a structure that differs from ours: the unit of change is an individual fact while the surrounding document is stable. A second line treats document versioning directly. VersionRAG models evolution through a version-aware graph and shows that semantically driven retrieval fails on version-sensitive questions by mixing states (Huwiler et al., 2025), and DiffQG annotates questions that two versions of a passage answer differently, identically, or not at all (Cole et al., 2023). Both maintain version history inside the system in order to answer cross-version queries; our deployment is the

opposite regime—a single-version-at-a-time system that re-indexes the new version corpus and must answer only about the current policy document. The unit of change is therefore a provision, not a fact; and in contrast to the operation-typed labels of CRUD-RAG, which describe the action requested of a static snapshot (Lyu et al., 2025), our classes describe how the gold-supporting provision changed between versions. To our knowledge, no prior work stratifies the evaluation of a deployed system by inter-version change type, states which scores remain comparable as the corpus is revised, or separates system drift from corpus drift; version-aware work contributes retrieval methods or edit summaries rather than a measurement instrument, which is the gap we address.

3 Deployed System

Corpus and indexing. The corpus comprises two credit policy documents revised a few times in a quarter. Documents are parsed with a structure-aware pipeline that extracts tabular constraints as structured rows, as these tables carry the operative policy values. Parsed text is segmented at subsection boundaries rather than fixed token windows: the chunk unit is the subsection, which we take to be the smallest semantically self-contained unit of a policy document, and each chunk retains its section title and document identifier as provenance metadata. Chunks are embedded with Gemini embeddings 001 (Lee et al., 2025) and stored in PostgreSQL/pgvector (Kane, 2023) for retrieval.

Runtime pipeline. A query is answered in four stages. (i) **Query analyser** scans the query for domain specific acronyms and jargon against a curated credit policy glossary and expands matched terms to their canonical phrasing. This ensures that retrieval is performed using standardised terminology present in the documents rather than the analyst’s shorthand (e.g. “Loan-to-Value (LTV)”, “EBITDA”, or internal product codes). (ii) **Retrieval** embeds the expanded query and returns the top $k = 20$ chunks using a hybrid retriever that combines lexical (PostgreSQL full-text search) and semantic retrieval. Those retrieved chunks are combined by reciprocal rank fusion (Cormack et al., 2009). (iii) **Chunk grader** applies a relevance grader (Yan et al., 2024; Asai et al., 2024): each retrieved chunk receives a binary *useful/not-useful* label for the query, and only chunks judged useful reach the generator. (iv) **Generator** produces

the answer under a domain-specific prompt that encodes (a) rules for reading the structured tables with examples and (b) a required output format in which every asserted condition cites its source section. This structured, provenance-bearing output is what makes answers auditable: in a regulated setting a claim without a traceable source cannot be acted upon by compliance staff.

Deployed configuration. The deployed query analyser, chunk grader and generator use Gemini 2.5 Flash (Comanici et al., 2025), chosen for cost and latency reasons; the same-family Gemini 2.5 Pro (Comanici et al., 2025) is evaluated only as an upper bound and serves as the LLM-judge through the experiments. End-to-end latency is 17 seconds median and 37 seconds at the 95th percentile; Gemini 2.5 Pro generation raises p50 to 34 seconds and p95 to 58 seconds, the practical reason Gemini 2.5 Flash is deployed. Marginal cost is approximately \$0.03 per query with Gemini 2.5 Flash in production. The choice of the Gemini family is due to the enterprise constraint as Gemini family is the only enterprise-approved model family. Once a new corpus version is available, the existing index is removed from the database and rebuilt from the new version, which takes approximately 1 minute.

4 Evaluation Framework

4.1 Question-Class Taxonomy

Every question is labelled relative to an anchor version v_{n-1} and the evaluated version v_n by the change status of the subsection(s) $\mathcal{S}$ supporting its gold answer. Subsection-level alignments are assigned by a deterministic, rule-based comparison of the two document versions.

Subsection alignment. We establish a correspondence between subsections of v_{n-1} and v_n (Algorithm 1). Each v_{n-1} subsection $s_i \in \mathcal{S}^{n-1}$ is matched to its v_n counterpart $s_j \in \mathcal{S}^n$, labelling UNCHANGED, EDITED, RENAMED, or UNMATCHED. First, this process matches the exact heading-hierarchy path; if the character-based similarity of the paths’ contents is equal to or above the content threshold $\tau_c (= 1.0)$, it is UNCHANGED; if not, it is EDITED. Second, the fuzzy heading match of subsection s_i and s_j is computed; if the similarity is equal to or above the heading threshold $\tau_h (= 0.6)$, it is RENAMED. Third, any s_i with no match is UNMATCHED. Finally, sections of v_n not covered by either phase constitute the ADDED

set $\mathcal{S}^+$.

Class assignment. Once the subsection-level alignment is computed, each question is labelled into four question classes: ADDED: the supporting section is new in v_n — alignment class is ADDED. STABLE: the supporting section is UNCHANGED in the alignment. EVOLVED: the supporting section is either EDITED or RENAMED in the alignment. REMOVED: the supporting section is no longer present in v_n — alignment class is UNMATCHED. Given how the EVOLVED class is labelled by a similarity threshold, there are cases where the subsection-level alignment is EDITED or RENAMED but the gold answer is unchanged. To avoid mislabelling such cases, we employ an LLM to check if the gold answer update is required. If the LLM determines that an update is required, the LLM updates the gold answer; if not, it relabels the question as STABLE. For questions whose gold draws on several subsections, the class is the highest-precedence status among them (REMOVED > ADDED > EVOLVED > STABLE): a question is Stable only if all supporting spans are unchanged.

4.2 Synthetic QA Generation

The test sets are produced using an extended version of VERGE (Morita, 2025) method, where verification-enhanced generation provides single-hop and multi-chunk open-ended QA pairs with an internal quality filter. It yields questions of varying compositional complexity over the source subsections. In this paper, VERGE is used solely as a test-set generation component, while our contribution focuses on longitudinal-aware evaluation methodology. Each item is a triplet — a question, a reference answer, and a pointer to its gold-supporting subsection(s). Value-masked prompt templates are given in Appendix B. We verify that the longitudinal conclusions reproduce under an independent generator, RAGAs (Es et al., 2024), and a SME-curated reference set, ruling out generator-specific artefacts; see Appendix E, Table 12. We employ Gemini 2.5 Flash for all the QA pairs generation and re-annotation of EVOLVED class’s gold answers.

Figure 1 illustrates the construction of the longitudinal-aware evaluation pool. We seed the dataset at v_1 with $N = 1000$ items over the v_1 corpus; every seed item is STABLE by definition, as there is no prior version. At each subsequent version v_n we (i) propagate the entire v_{n-1} pool and relabel each item per the class taxonomy above —

	v1	v2	v3
<i>Corpus</i>			
Documents	2	2	2
Pages	153	162	165
# tokens	82k	86k	87k
Median subsection tokens	219	219	215
# tables	75	76	76
Growth vs. prior	—	6%	2%
<i>Subsection (by class)</i>			
Stable	224	189	95
Evolved	—	35	133
Added	—	4	7
Total subsections	224	228	235
<i>Evaluation QA pairs (by class)</i>			
Stable	1,000	551	405
Evolved	—	449	820
Added	—	225	212
Total	1,000	1,225	1,437

Table 1: Dataset statistics per corpus version. The *Total* differs across columns because *Added* QAs are freshly generated per transition ($v_{n-1} \rightarrow v_n$).

re-annotating the reference answer against v_n for any item that becomes EVOLVED, while holding the question text fixed; and (ii) freshly generate ADDED items from the subsections new in v_n .

Table 1 reports the composition of the evaluation pool across versions. At $v1 \rightarrow v2$, the 1,000 seed items split into 551 STABLE and 449 EVOLVED, augmented with 225 ADDED, for 1,225 items; at $v2 \rightarrow v3$ the full 1,225-item pool splits into 405 STABLE and 820 EVOLVED, augmented with 212 ADDED, for 1,437 items. Where the alignment marks a supporting subsection EDITED or RENAMED but the LLM judges the gold answer unchanged, the question is returned to STABLE. This accounts for 71 of the 551 STABLE items at $v2$ (13%) and 267 of the 405 at $v3$ (66%) and the remaining 480 and 138 items are STABLE under the deterministic alignment alone.

4.3 LLM-Judge Evaluation

Given a gold answer, we decompose it into a set of atomic statements S_{ref} ; the Gemini 2.5 Pro judge model then labels each $s \in S_{\text{ref}}$ as supported or not supported by the system response. Our primary metric is statement-level **recall**,

$$\text{Recall} = \frac{|\{s \in S_{\text{ref}} : \text{supported}\}|}{|S_{\text{ref}}|},$$

where S_{ref} decomposes the gold answer and this metric reflects the operational priority: a usable answer must cover the required policy conditions.

Because recall alone is maximised by verbose, over-inclusive responses, we report it alongside statement-level **precision**;

$$\text{Precision} = \frac{|\{s \in S_{\text{resp}} : \text{entailed}\}|}{|S_{\text{resp}}|},$$

where S_{resp} decomposes the system response against the gold answer.

Alongside those scores we report 95% bootstrap confidence intervals ($B = 10,000$, percentile method). Because questions generated from the same subsection share their source text and are not independent, these understate uncertainty in proportion to that clustering. The effect is largest for ADDED, which draws on only four and seven new subsections at the two transitions. The intervals on the two quantities our conclusions rest on are instead one-stage subsection-level cluster bootstraps. Each question is assigned to its highest-precedence supporting subsection, whole subsections are re-sampled with replacement, and for the paired Δ the within-item pairing is retained so that only the cluster is resampled. Appendix F gives the procedure in full.

4.4 Validation

Both the model performance judge and the synthetic data are LLM-generated, which raises a circularity concern: scores could be an artefact of the measurement instrument rather than of system quality. We address this with three human studies.

Firstly, on the 146 SME-curated QAs, we verify the LLM judge by consensus-after-discussion among three SMEs. SMEs manually score all the QAs and we report Spearman’s ρ (Spearman, 1904) and Pearson’s r (Pearson, 1896) between LLM-judge and human statement-level recall to capture agreement on the graded score. By LLM-judge, the average recall is 0.85 and SME judges mark 0.95. The Spearman’s rank correlation between the LLM-judge and the SMEs is 0.61 and Pearson’s r is 0.66 with statistical significance (both $p < 0.001$). These results indicate that there is a moderate positive correlation between LLM-judge and SMEs, yet LLM-judge is slightly stricter than the SMEs.

Secondly, three annotators independently assessed a question-class stratified sample of 100 synthetic QAs (32 Stable, 40 Evolved, 28 Added) along three criteria: *QA correctness*: is the QA pair correct, *faithfulness*: is the pair derivable from the cited source chunk, and *class correctness*: is

Criteria	Correct	95% CI	AC1	κ
QA correctness	0.92	[0.89, 0.95]	0.94	0.69
Faithfulness	0.93	[0.91, 0.96]	0.91	0.41
Class correctness	0.89	[0.86, 0.93]	0.91	0.59

Table 2: Human validation of synthetic QA quality ($n = 100$, 3 annotators). Correct is mean of the annotators; 95% CIs in brackets. AC1 is Gwet’s AC1 and κ is Fleiss’ kappa.

the assigned taxonomy label correct. Table 2 reports the results. On average 92%, 93%, and 89% are correct. For the inter-annotator agreement, we report Gwet’s AC1 (Gwet, 2008) as a primary measure (0.94, 0.91, and 0.91, respectively). Fleiss’ κ (Fleiss, 1971) (0.69, 0.41, and 0.59) is also reported, however, it suffers from the *Kappa Paradox* as the ground truth for all the items in this annotation study is positive. Thus, this is only for reference and should not be used as the primary measure of agreement.

Lastly, we verify the quality of re-annotated gold answers for the EVOLVED class. Two SMEs independently annotated 30 QAs randomly sampled from the 820 Evolved items whose gold answers were updated at $v2 \rightarrow v3$, assessing (i) whether the update was necessary and (ii) whether the updated gold answer is correct against $v3$ corpus. The two SMEs were in perfect agreement on all 30 QAs. Both judged 30/30 updates necessary and 29/30 correct, with 95% Wilson lower bounds (Wilson, 1927) of 88.6% and 83.3%, respectively. This annotation study supports an Evolved gold answer update path while acknowledging that a small residual error rate cannot be excluded at this sample size.

5 Results

5.1 Longitudinal-aware Dataset Results

Table 3 reports statement-level recall and precision by class and version. Within a version, all classes share the same corpus, index, and judging protocol, so cross-class gaps are interpretable. At $v3$, EVOLVED recall is 69.8 [67.1, 71.6] against 75.8 [72.4, 79.5] for STABLE and 77.8 [73.3, 82.1] for ADDED, and EVOLVED precision is the lowest of any class at both transitions (58.1 at $v2$; 56.7 at $v3$).

The STABLE column rises from 72.3 to 75.8, but these marginal scores compare shrinking survivor cohorts ($1,000 \rightarrow 551$ and $1,225 \rightarrow 405$ items) whose composition shifts as supporting sections are revised out of the class; EVOLVED and ADDED

Class	Metric	v1	v2	v3
Stable	Recall	72.3 [70.0, 74.2]	73.8 [70.2, 76.5]	75.8 [72.4, 79.5]
	Precision	61.9 [59.2, 65.0]	60.2 [56.5, 64.0]	65.8 [60.3, 71.2]
Evolved	Recall	–	74.3 [70.3, 78.3]	69.8 [67.1, 71.6]
	Precision	–	58.1 [54.0, 62.3]	56.7 [54.6, 58.7]
Added	Recall	–	72.6 [68.7, 76.3]	77.8 [73.3, 82.1]
	Precision	–	63.0 [57.4, 68.0]	65.5 [62.0, 69.1]

Table 3: Longitudinal-aware results in % by question class and corpus version. $v1$ has no anchor version, hence “–” for Evolved/Added. Confidence intervals are 95% bootstrap percentile intervals, reported in square brackets.

Transition	N	R@anchor	R@eval	Paired Δ
$v1 \rightarrow v2$	551	73.3 [70.6, 76.1]	73.8 [70.2, 76.5]	0.5 [0.3, 0.7]
$v2 \rightarrow v3$	405	74.2 [72.2, 76.3]	75.8 [72.4, 79.5]	1.6 [0.9, 2.4]

Table 4: The STABLE regression anchor as a within-transition paired comparison on recall scores. Each row uses the items STABLE at that transition (identical gold at both versions); the paired Δ is computable online at each release.

are relabelled or regenerated at each transition over different provisions (35 vs 133 revised subsections), so their columns are per-version cross-sections. We therefore state the cross-version claim as a within-transition paired comparison: for items whose gold answer is identical at v_{n-1} and v_n , the unchanged pipeline is scored under both indices, and the per-item difference isolates the effect of corpus change around a fixed measurement target. Recall shifts by +0.5pp ($N=551$, $v1 \rightarrow v2$) and +1.6pp ($N=405$, $v2 \rightarrow v3$) (Table 4). This paired Δ is computable at each release.

To quantify what conventional practice would report, we score the $v3$ system against the $v1$ and $v2$ datasets with their original gold answers (Table 5). Overall recall on the frozen sets is 47.1 ($v1$) and 47.0 ($v2$, class-weighted over 551/449/225 items), against the 72.7 the same system attains on the $v3$ set with current gold — declines of 25.6 and 25.7 points, or roughly 35% of the recall actually available. The Evolved class falls furthest on the $v2$ set, to 42.3. Under the standard freeze-and-track protocol, this deployment would have appeared to lose

Class	Metric	v1	v2
Stable	Recall	47.1 [41.2, 52.2]	51.4 [44.6, 57.8]
	Precision	36.9 [32.0, 42.0]	43.4 [37.2, 48.9]
Evolved	Recall	— —	42.3 [39.7, 44.9]
	Precision	— —	31.3 [29.1, 33.6]
Added	Recall	— —	45.8 [43.7, 47.3]
	Precision	— —	32.4 [30.1, 34.6]

Table 5: Longitudinal-unaware results of v3 system on v1 and v2 datasets, reported in % by question class. v1 has no anchor version, hence “—” for Evolved/Added. Confidence intervals are 95% bootstrap percentile intervals, reported in square brackets.

Threshold (τ_c)	v2			v3		
	.95	.99	1.0	.95	.99	1.0
Stable	210	199	189	195	159	95
Evolved	14	25	35	33	69	133

Table 6: Sensitivity of the question-class taxonomy to the section-level change threshold τ_c . Higher τ_c requires greater textual overlap to call a section *stable*, shifting more questions into the Evolved class.

more than a third of its recall in two quarters, while the paired anchor of Table 4 shows no evidence of regression.

5.2 Classification Threshold Sensitivity

The change-status label of section 4.1 is reproducible given its thresholds, but the subsection-level content threshold τ_c is a design choice, and the interpretability of the per-class comparison rests on it. The STABLE class functions as the regression anchor only insofar as its questions have genuinely invariant gold answers. The choice is motivated by the corpus’s edit profile: revisions are predominantly small in surface terms so a few-token edit in a multi-sentence subsection produces a body similarity of ≈ 0.99 , and any threshold short of exact match absorbs such revisions into STABLE.

Table 6 reports the subsection-level composition as τ_c is lowered. The taxonomy is most sensitive at the top of its range: at the v2→v3 transition the EVOLVED count falls from 133 at exact match to 69 at $\tau_c = 0.99$ and 33 at 0.95, so the single step from exact to 99% overlap already reclassifies nearly half of the revised subsections as STABLE.

Corpus	Metric	v2		v3	
		$\tau_c=.95$	$\tau_c=1.0$	$\tau_c=.95$	$\tau_c=1.0$
Stable	Recall	68.5 [66.2, 70.4]	73.8 [70.2, 76.5]	64.2 [62.1, 66.3]	75.8 [72.4, 79.5]
	Prec.	54.8 [51.1, 57.7]	60.2 [56.5, 64.0]	63.4 [60.7, 65.9]	65.8 [60.3, 71.2]
Evolved	Recall	59.5 [53.6, 64.9]	74.3 [70.3, 78.3]	61.2 [56.0, 66.7]	69.8 [67.1, 71.6]
	Prec.	46.0 [39.8, 52.3]	58.1 [54.0, 62.3]	51.0 [43.6, 58.4]	56.7 [54.6, 58.7]

Table 7: Effect of the change threshold τ_c on per-class statement-level recall and precision (%; 95% bootstrap CIs). Lowering τ_c from 1.0 to 0.95 folds minor-revision questions into Stable and restricts Evolved to major revisions, depressing both classes; the $\tau_c=1.0$ columns reproduce Table 3.

This locates the operative revisions in the interval [0.99, 1.0) of body similarity and shows they are separated from unchanged content only at $\tau_c = 1.0$. The v1→v2 transition exhibits the same monotone pattern at lower magnitude (EVOLVED 35 → 25 → 14).

Table 7 demonstrates the impact of different τ_c values on the performance. At v2, lowering τ_c to 0.95 depresses both classes: STABLE recall and precision fall (73.8 → 68.5; 60.2 → 54.8) as changed-answer questions are folded into the anchor and scored against their carried-over, now-superseded gold answers, while EVOLVED falls further, 74.3 → 59.5 recall, 58.1 → 46.0 precision. The effect is larger at v3, where the 100 subsections reclassified EVOLVED→STABLE drive STABLE recall down from 75.8 to 64.2; the EVOLVED figures move in the same direction. The STABLE class at $\tau_c = 0.95$ no longer measures what it is designed to measure and is no longer a valid cross-version anchor. Therefore, we adopt $\tau_c = 1.0$ for this corpus, noting that the appropriate value is corpus-dependent.

5.3 RAG System Ablation

Beyond cross-version comparability, the taxonomy serves as a diagnostic: stratifying an ablation by class localises component effects. Table 8 demonstrates the results of each individual component of the RAG on the v3 corpus. Each component of the deployed RAG system (Full RAG) is added to the Naive RAG system individually. Adding a hybrid retrieval (Naive RAG) lifts recall from the 10.5 parametric baseline to 60.5 — and the classes separate the remaining refinements. Query analysis con-

Configuration	Overall Recall	Overall Prec.	Stable Recall	Evolved Recall	Added Recall
LLM-only (Flash)	10.5	6.3	10.1	11.3	8.1
Naive RAG (Flash)	60.5	48.8	61.2	59.8	62.1
+ QA (Flash)	62.4	53.9	62.4	61.5	66.0
+ grader (Flash)	64.6	55.9	65.6	63.3	68.0
+ gen. (Flash)	67.7	60.3	68.2	67.1	69.3
Full RAG (Flash)	72.7	60.6	75.8	69.8	77.8
Full RAG (Pro)	80.6	69.5	82.3	78.5	85.3

Table 8: Ablation on v3, reporting recall and precision in %. QA is the query analyser, grader is the chunk grader and gen. is the structured generator prompt. Those components are individually combined with the Naive RAG.

tributes chiefly precision (+5.1) at a modest recall gain (+1.9), the chunk grader adds +4.1 recall and +7.1 precision, and the structured generator is the largest single contributor (+7.2 recall, +11.5 precision). Combining all three components with Naive RAG, the system reaches 72.7, which is 12.2pp higher than Naive RAG. The way the components combine differs by class: on STABLE the full configuration is superadditive, gaining +14.6 against the 12.6 summed over the individual components. Replacing the Gemini 2.5 Flash solver with Gemini 2.5 Pro raises overall recall to 80.6 (+7.9) across all classes. Because Gemini 2.5 Pro also serves as judge, this row is most exposed to self-preference; re-judging with Gemini 2.5 Flash judge changes the measured Gemini 2.5 Flash $\rightarrow$ Gemini 2.5 Pro gap by 0.02 points in recall (Pro-Flash judge agreement: Spearman $\rho = 0.79$, Pearson $r = 0.85$, $p < 0.001$ for both). Given the latency and cost premium, Gemini 2.5 Flash remains the deployed configuration.

5.4 Stress Tests

To validate the generality of the method, we (i) inflate the retrieval index with in-domain distractors and ablate the query analyser, and (ii) artificially delete a set of subsections from the corpus and observe the effect on the REMOVED questions. We use the Finance Act 2026, c. 11 (549 pages) (UK Parliament, 2026) as a source of in-domain distractors. Adding these to the retrieval index increases the number of chunks from 235 to 2,984. Table 9 reports that end-to-end recall falls from 72.7 to 66.2 across the sweep while the longitudinal-unaware decline holds at roughly -25 points and the paired STABLE Δ stays small and positive.

We artificially construct a deletion transition by

Condition	Chunks	End-to-end		Unaware	Paired
		R	P	ΔR	ΔR
LLM-only	—	10.5	6.3	—	—
Full RAG	235	72.7	60.6	-25.7	$+1.6$
+ distractors	2,984	70.1	58.8	-25.2	$+1.9$
+ distr., $-QA$	2,984	66.2	56.1	-25.4	$+1.9$

Table 9: Retrieval-difficulty stress test on v3, in %. Corpus, versions, gold answers, generator and question pool are held fixed. “Unaware ΔR ” is the recall change a longitudinal-unaware protocol would report, scoring the v3 system against the v2 dataset; “Paired ΔR ” is the STABLE-anchor paired difference at $v2 \rightarrow v3$.

Quantity	Value
System abstention rate (%)	91.7 [84.8, 98.7]
Longitudinal-unaware recall (%)	13.6 [7.2, 20.1]
Longitudinal-unaware precision (%)	10.2 [4.8, 15.4]

Table 10: Constructed deletion transition. The same system outputs are scored two ways. Intervals are 95% cluster-bootstrap percentile intervals over the 96 REMOVED questions.

removing 15 subsections from v3 (10 STABLE, 5 EVOLVED), which places 96 questions in the REMOVED class. Table 10 scores the same system outputs under both longitudinal-aware and longitudinal-unaware protocols. As the expected behaviour for the REMOVED class is abstention, the table reports the system abstention rate and the longitudinal-unaware recall and precision on the v3 corpus. The longitudinal-aware protocol credits abstention at 91.7% [85, 99]; the longitudinal-unaware protocol retains the superseded v3 gold answer and scores those identical outputs at 13.6 recall and 10.2 precision.

6 Conclusion

We presented a longitudinal-aware evaluation framework for RAG over corpora that are revised in production, combining a change-type taxonomy built on a deterministic subsection alignment, a versioned and class-stratified synthetic QA pipeline, and a statement-level LLM judge, each calibrated against SMEs. Applied to a deployed credit policy assistant across three corpus versions, the framework separates what a static benchmark conflates: a frozen test set would have reported a decline of nearly 26 recall points over two versions, while the results of the paired STABLE anchor show a small but consistent improvement.

Limitations

LLM-as-judge scoring retains known biases toward verbosity (Zheng et al., 2023). Our human studies bound but do not eliminate them, and judge-SME agreement, while positive and significant, is moderate ($\rho = 0.61$), with the judge systematically stricter than the experts. Solver, judge, and the re-annotation of EVOLVED gold answers all use models from a single family owing to an enterprise constraint; we mitigate this with a cross-tier re-judging check and SME validation, but family-level circularity remains a potential confound.

ADDED questions are drawn from few new subsections per transition (4 and 7), limiting their diversity and rendering item-level confidence intervals optimistic where questions cluster within a subsection.

Synthetic QA generator method robustness is only tested for STABLE classes due to the limited SME-curated dataset and the available computational budget.

Finally, the anchor is not purely rule-derived. Where the alignment marks a supporting subsection EDITED or RENAMED but an LLM judges the gold answer unaffected, the question is returned to STABLE. This supplies 71 of the 551 anchor items at v2 (13%) and 267 of the 405 at v3 (66%). The v3 anchor therefore rests substantially on that adjudication, and its error mode is in a favourable direction. An item wrongly admitted to STABLE carries superseded gold into v_n , is scored against an answer the current corpus no longer supports, and so depresses $R@eval$ and biases the paired Δ downward. The no-regression conclusion is conservative under this failure mode and the reverse error costs anchor size rather than validity.

Risk and Ethical Considerations

Data and privacy. The corpus consists of internal credit policy documents: institutional policy text describing products, eligibility criteria, thresholds, and qualifying conditions. It contains no personal data and no information about individual customers, applicants, or exposures. All synthetic and SME-curated evaluation items are derived from this policy text; illustrative examples in this paper are paraphrased and operative values are masked (e.g., “xx%” in Appendix B). Production usage is reported only in aggregate (query volume, latency, cost), and no user queries are reproduced. All processing — parsing, embedding, retrieval, genera-

tion, and LLM-judge scoring — is performed with the organisation’s enterprise-approved model family within approved infrastructure.

Intended use and deployment safeguards. The system is a decision-support tool that helps trained credit professionals (relationship managers and credit officers) locate and read policy provisions. It does not make, automate, or recommend credit decisions about any individual or business. Two safeguards are built into the deployment: (i) every asserted condition must cite its source section, and (ii) an organisation-approved hallucination-prevention prompt. Users are expected to verify cited provisions against the authoritative documents before acting.

Residual risks and mitigations. The principal residual risk is that an incorrect or incomplete answer survives user verification and leads staff to apply a superseded threshold or overlook a qualifying condition. The measured statement-level recall and precision indicate that unverified reliance on system output would be inappropriate, and the system is positioned accordingly. The evaluation framework presented here is itself a risk control: the test set is refreshed and the paired anchor recomputed at every corpus revision, so post-revision regressions can be detected at the cadence at which the policy changes rather than at audit time. We note that the framework characterises aggregate quality and does not certify individual answers, and that LLM-as-judge scoring carries known biases.

Human annotation. The subject-matter experts who curated reference questions, validated synthetic items, and scored system outputs are employees of the deploying organisation and performed this work as part of their professional duties; no crowdworkers were involved and no personal data was handled during annotation.

Acknowledgments

We wish to express appreciation to Chief Data and AI Office and Credit Platform of Lloyds Banking Group for supporting the Credit Policy Virtual Assistant project and this research. We are especially grateful to the SMEs who have done all the annotation and validation work, and to the reviewers for their constructive feedback.

References

- Akari Asai, Zeqiu Wu, Yizhong Wang, Avi Sil, and Hannaneh Hajishirzi. 2024. [Self-rag: Learning to retrieve, generate, and critique through self-reflection](#). In *International Conference on Learning Representations*, volume 2024, pages 9112–9141.
- Wenhu Chen, Xinyi Wang, and William Yang Wang. 2021. A dataset for answering time-sensitive questions. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.
- Jeremy R. Cole, Palak Jain, Julian Martin Eisenschlos, Michael J.Q. Zhang, Eunsol Choi, and Bhuwan Dhingra. 2023. [DiffQG: Generating questions to summarize factual changes](#). In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 3088–3101, Dubrovnik, Croatia. Association for Computational Linguistics.
- Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Naveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, and 1 others. 2025. [Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities](#). *arXiv preprint arXiv:2507.06261*.
- Gordon V Cormack, Charles LA Clarke, and Stefan Buettcher. 2009. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 758–759.
- Shahul Es, Jithin James, Luis Espinosa Anke, and Steven Schockaert. 2024. [RAGAs: Automated evaluation of retrieval augmented generation](#). In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 150–158, St. Julians, Malta. Association for Computational Linguistics.
- Joseph L Fleiss. 1971. [Measuring nominal scale agreement among many raters](#). *Psychological bulletin*, 76(5):378–382.
- Kilem Li Gwet. 2008. Computing inter-rater reliability and its variance in the presence of high agreement. *British Journal of Mathematical and Statistical Psychology*, 61(1):29–48.
- Jiale Han, Austin Cheung, Yubai Wei, Zheng Yu, Xusheng Wang, Bing Zhu, and Yi Yang. 2025. [Rag meets temporal graphs: Time-sensitive modeling and retrieval for evolving knowledge](#). *arXiv preprint arXiv:2510.13590*.
- Daniel Huwiler, Kurt Stockinger, and Jonathan Fürst. 2025. [Versionrag: Version-aware retrieval-augmented generation for evolving documents](#). *arXiv preprint arXiv:2510.08109*.
- Andrew Kane. 2023. [pgvector: Open-source vector similarity search for postgresql](#). <https://github.com/pgvector/pgvector>.
- Jinhyuk Lee, Feiyang Chen, Sahil Dua, Daniel Cer, Madhuri Shanbhogue, Iftekhhar Naim, Gustavo Hernández Ábrego, Zhe Li, Kaifeng Chen, Henrique Schechter Vera, Xiaoqi Ren, Shanfeng Zhang, Daniel Salz, Michael Boratko, Jay Han, Blair Chen, Shuo Huang, Vikram Rao, Paul Suganthan, and 28 others. 2025. [Gemini embedding: Generalizable embeddings from gemini](#).
- 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. 2020. [Retrieval-augmented generation for knowledge-intensive nlp tasks](#). In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.
- Adam Liška, Tomas Kocisky, Elena Gribovskaya, Tayfun Terzi, Eren Sezener, Devang Agrawal, Cyprien de Masson d’Autume, Tim Scholtes, Manzil Zaheer, Susannah Young, Ellen Gilsenan-McMahon, Sophia Austin, Phil Blunsom, and Angeliki Lazaridou. 2022. [StreamingQA: A benchmark for adaptation to new knowledge over time in question answering models](#). In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 13604–13622. PMLR.
- Yuanjie Lyu, Zhiyu Li, Simin Niu, Feiyu Xiong, Bo Tang, Wenjin Wang, Hao Wu, Huanyong Liu, Tong Xu, and Enhong Chen. 2025. [Crud-rag: A comprehensive chinese benchmark for retrieval-augmented generation of large language models](#). *ACM Trans. Inf. Syst.*, 43(2).
- Kyosuke Morita. 2025. [VERGE: Verification-enhanced generation of multi-hop evaluation datasets for task-specific RAG](#). *TechRxiv*, 2025(0401).
- Jie Ouyang, Tingyue Pan, Mingyue Cheng, Ruirao Yan, Yucong Luo, Jiaying Lin, and Qi Liu. 2025. [HoH: A dynamic benchmark for evaluating the impact of outdated information on retrieval-augmented generation](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6036–6063, Vienna, Austria. Association for Computational Linguistics.
- Karl Pearson. 1896. [Vii. mathematical contributions to the theory of evolution.—iii. regression, heredity, and panmixia](#). *Philosophical Transactions of the Royal Society of London, Series A: Containing Papers of a Mathematical or Physical Character*, (187):253–318.
- C Spearman. 1904. The proof and measurement of association between two things. *The American Journal of Psychology*, 15(1):72–101.

UK Parliament. 2026. [Finance Act 2026](#). UK Public General Act, 2026 c. 11. Royal Assent 18 March 2026. Accessed 25 August 2026.

Edwin B Wilson. 1927. Probable inference, the law of succession, and statistical inference. *Journal of the American Statistical Association*, 22(158):209–212.

Shi-Qi Yan, Jia-Chen Gu, Yun Zhu, and Zhen-Hua Ling. 2024. [Corrective retrieval augmented generation](#). *arXiv preprint arXiv:2401.15884*.

Michael Zhang and Eunsol Choi. 2021. [SituatingQA: Incorporating extra-linguistic contexts into QA](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7371–7387, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph Gonzalez, and Ion Stoica. 2023. [Judging llm-as-a-judge with mt-bench and chatbot arena](#). In *Advances in Neural Information Processing Systems*, volume 36, pages 46595–46623. Curran Associates, Inc.

A Section Matching Method

Algorithm 1 details the method for aligning sections across document versions, which underpins the question-class taxonomy and longitudinal evaluation. The similarity function SIM is implemented as character-level sequence matching using Python’s `difflib.SequenceMatcher`, which computes a ratio of matching characters to total characters in the two strings.

B Synthetic QA Generation Details

This section describes three prompts for VERGE QA generation method and the EVOLVED answer update prompt. The original VERGE method is designed for generating multi-chunk multiple-choice questions. In order to fit this method to our use case, we extended it to single and multi-chunk open-ended questions. The number of chunks is randomly determined for each question between 1 and 3. For both generation of QA pairs and re-annotation of EVOLVED class, we employ Gemini 2.5 Flash.

B.1 Synthetic QA Generation Prompt

You are a credit policy expert question generator specialising in challenging open-ended questions that require multi-hop reasoning across multiple credit policy documents.

REQUIREMENTS:

1. The question MUST require synthesising information from ALL provided chunks.
2. It CANNOT be answered using only a subset of the chunks.
3. The reference answer MUST be fully supported by the chunk text alone.
4. Prefer analytical, comparative, or causal question types.
5. The reference answer MUST be non-trivial and should not be able to be answered without the provided chunks.

Question types to consider:

- Analytical : "Analyse the relationship between X and Y. . ."
- Comparative : "Compare and contrast the approaches described. . ."
- Causal : "Explain how X leads to Y considering all factors. . ."
- Evaluative : "Evaluate the effectiveness of X considering the constraints. . ."

QA examples:

Example 1:

Question: "I have an existing customer with a dentist looking to buy his second dentist. What do I need to consider?"

Reference Answer: "Minimum Debt Service Cover (DSC) Criteria: The minimum Debt Servicing Cover (DSC) for a Dentists is xx% measured by EBITDA (terms and conditions applied). . ."

Example 2:

Question: "I have a request with the following parameters – what do I need to consider?"

Reference Answer: ". . ."

Respond EXACTLY in this format – nothing else:

Question: <your question>

Reference Answer: <comprehensive answer synthesising all chunks>

B.2 Synthetic QA Verification Prompt

You are a strict verifier for open-ended multi-hop exam questions.

Assess whether:

1. The question genuinely requires synthesising ALL provided chunks.
2. The reference answer is complete and fully supported by the chunk text alone.
3. No shortcut exists that allows answering from a single chunk.
4. The QA pair is non-trivial and cannot be answered without the provided chunks.
5. The QA pair has integrity – the question is clear, sensible, and the answer directly addresses it.
6. The answer is faithful – every claim in the answer is grounded in the provided chunks.

Respond ONLY with valid JSON – no prose, no markdown code fences:

```
{
  "required_chunks": [<1-based indices of chunks essential for answering>],
  "reasoning": {
    "multi_hop_required": <true or false>,
    "answer_completeness": <true or false>,
    "unused_chunks": [<1-based indices of chunks not properly utilised>],
```

```

    "improvement_suggestions":      "<specific
recommendations for next iteration>"
  },
  "qa_integrity":      <true if question is
clear/sensible and answer directly addresses
it, false otherwise>,
  "faithfulness": <true if every claim is grounded
in the provided chunks, false otherwise>,
  "confidence": <integer 1-5>
}

```

B.3 Synthetic QA Re-generation Prompt

Improve the question below so that it genuinely requires ALL provided chunks, cannot be answered from a single chunk alone nor from parametric knowledge.

```

Original Question: {original_question}
Original Reference Answer: {original_answer}
Verification Feedback:
- Chunks not properly utilised: {unused_chunks}
- Improvements needed: {suggestions}
Document Chunks: {documentation}
Regenerate following the required format:
Question: <improved question>
Reference Answer:      <comprehensive answer
synthesising all chunks>

```

B.4 Evolved Answer Update Prompt

You are a credit policy expert responsible for keeping a QA test set accurate and up-to-date after a credit policy document has been revised.

You will be given:

- A question derived from an older version of the policy.
- The original reference answer based on that older version.
- The CURRENT (updated) version of the relevant policy section(s).

Your task is to:

1. Compare the original reference answer against the current policy content.
2. Determine whether the answer is still accurate, complete, and faithful to the current policy. Flag it as needing an update if ANY of the following is true:

- A fact, threshold, or rule in the answer has changed in the new content.
- Important new information relevant to the question has been added.
- The answer references content that no longer exists or has been removed.

3. If an update is needed, write a corrected reference answer that is fully grounded in and faithful to the CURRENT policy content. Keep the style and level of detail consistent with the original answer.
4. If no update is needed, leave the answer unchanged.

Respond ONLY with valid JSON – no prose, no markdown code fences:

```

{
  "reasoning":      "<brief explanation of what
changed>",
  "updated_answer":      "<corrected answer if
needs_update is true, otherwise the original
answer verbatim>"
}

```

C Statistics of the Generated Testset

Table 11 reports the distribution of questions across classes and versions in the generated testset. The paired comparison holds the gold answer, and hence S_{ref} , fixed within item, so the version-level shift in median S_{ref} cannot generate the paired Δ ; it reflects the changing composition of the surviving STABLE cohort, which is why the marginal columns of Table 3 are not read as a trend.

D LLM-judge Details

D.1 Recall Prompt

You are a recall evaluator that measures factual completeness through statement counting.

TASK: Evaluate how complete the generated answer is compared to the reference answer by counting matching statements.

```

QUESTION:
{question}
REFERENCE ANSWER:
{reference_answer}
ANSWER:
{answer}
EVALUATION METHODOLOGY:

```

1. **Identify Reference Statements:** Break down the reference answer into distinct, unique statements that **directly support answering the question**. Each statement should be a single fact, value, or piece of information. **Informational statements that are not required to answer the question should not be counted.**

2. **Count Reference Statements:** Count the total number of unique statements identified (this is your reference_count).

3. **Identify Matching Statements:** For each reference statement, check if an equivalent statement (even paraphrased) exists in the generated answer. Use the provided ACRONYMS/ABBREVIATIONS and SYNONYMS to recognise equivalent statements (e.g., "PAL" matches "Partial Amortising Loan", "commitment term" matches "tenor").

4. **Count Matching Statements:** Count how many reference statements have a match in the generated answer (this is your actual_count).

5. **Calculate Score:**
Compute: $\text{actual_count} / \text{reference_count}$ (gives a value between 0 and 1)

STATEMENT MATCHING RULES:

- A statement matches if the core fact/value is present, even if phrased differently
- Use acronyms and synonyms to identify equivalent terms

- Ignore minor phrasing differences; focus on factual equivalence
- Ignore minor omissions of the full scope mentioned in the reference, as long as the answer is factually accurate for the specific case asked in the question
- A statement is only counted once, even if mentioned multiple times

Acronyms and Abbreviations

```
...
OUTPUT FORMAT:
Provide your evaluation as a JSON object with this
exact format:
{
  "score": <decimal between 0.0-1.0, calculated
as (actual_count / reference_count)>,
  "reasoning": "<Reference statements: X,
Matched: Y, Missing: [...]>"
}
EXAMPLES:
...
Return ONLY the JSON object. No markdown code
blocks. Just the raw JSON.
```

D.2 Precision Prompt

You are a precision evaluator for question-answering systems.

TASK: Given a QUESTION, a fixed set of REFERENCE STATEMENTS (atomic facts from the gold answer), and a GENERATED ANSWER, you must:

1. Decompose the GENERATED ANSWER into atomic factual statements (response_statements).
2. For each response statement, judge whether it is SUPPORTED by the reference statements (i.e., the fact is present in or directly entailed by the reference statements). Mark each as "supported" or "unsupported".

```
QUESTION:
{question}
REFERENCE STATEMENTS (fixed, do not modify):
{ref_block}
GENERATED ANSWER:
{answer}
RULES:
- A response statement is "supported" if the
equivalent fact appears in (or is directly entailed
by) the reference statements – even if phrased
differently
- Statements that are purely organisational (e.g.
"Here is the answer:") or meta-commentary should
be excluded from response_statements entirely
- Use the provided ACRONYMS/ABBREVIATIONS and
SYNONYMS below to recognise equivalent statements
(e.g., "PAL" matches "Partial Amortising Loan",
"commitment term" matches "tenor")
- Ignore minor phrasing differences; focus on
factual equivalence
```

Acronyms and Abbreviations

```
...
OUTPUT FORMAT (return ONLY this JSON,
no markdown fences):
{
  "response_statements": [
    {"statement": "...", "supported": true}},
    {"statement": "...", "supported": false}}
  ],
```

```
"supported_count": <int>,
"total_count": <int>,
"precision": <float between 0.0 and 1.0>
}
```

E Generator Robustness Analysis

In order to verify that one of our conclusions that we can use STABLE QAs as a reference set, is not an artefact of our generation method (VERGE), we use RAGAs and SME-curated dataset to check our conclusion can be drawn from different generation methods. Table 12 shows the results of generator robustness analysis, which shows the recall and precision of RAGAs (n=175) and SME-curated (n=146) dataset across different versions on the STABLE class. The results show that the recall and precision of RAGAs and SME-curated dataset are consistent across different versions, which supports our conclusion.

F Uncertainty Quantification

Questions generated from the same subsection share their source text and are therefore not independent. Item-level bootstrap intervals, which treat each question as an independent draw, understate uncertainty in proportion to that clustering. The effect is largest for ADDED, whose questions come from only four and seven new subsections at the two transitions.

Procedure. Every interval reported in this section is a one-stage subsection-level cluster bootstrap with $B = 10,000$ resamples and percentile intervals. Each question is assigned to its highest-precedence supporting subsection, which makes the assignment a function of the deterministic alignment alone. Clusters are resampled with replacement; all questions belonging to a resampled cluster enter the replicate together.

Effect on the paired anchor. For the STABLE paired Δ , the clustered intervals are $[0.26, 0.73]$ at $v1 \rightarrow v2$ and $[0.93, 2.44]$ at $v2 \rightarrow v3$. Differencing within items cancels most of the shared subsection-level difficulty, so these lie essentially where the item-level intervals lie and continue to exclude zero. The no-regression conclusion is unaffected by the choice of resampling unit.

Effect on per-class levels. Per-class level intervals widen as expected, most for ADDED. Table 13 reports the cluster-level intervals side by side with the number of clusters supporting each estimate.

With four and seven clusters no resampling scheme recovers a usable interval for Added at this cluster count. We therefore treat Added as descriptive throughout the paper and draw no inferential conclusion from its intervals.

Algorithm 1: Section Alignment across Document Versions

Input: $\mathcal{S}^{(n-1)}$: sections from $\mathcal{D}^{(n-1)}$; $\mathcal{S}^{(n)}$: sections from $\mathcal{D}^{(n)}$; τ_h : heading similarity threshold; τ_c : content unchanged threshold

Output: Alignment map $\mathcal{A} : h^{(n-1)} \rightarrow (h^{(n)}, \text{status})$

```

1  $\mathcal{M} \leftarrow \emptyset$ ;           // matched  $\mathcal{S}^{(n)}$  paths
2 foreach  $s_i \in \mathcal{S}^{(n-1)}$  do
    // Phase 1: Exact heading match
3   if  $\exists s_j \in \mathcal{S}^{(n)}$  s.t.  $s_j.\text{path} = s_i.\text{path}$  then
4      $\sigma \leftarrow \text{SIM}(s_i.\text{body}, s_j.\text{body})$ ;
5      $\text{status} \leftarrow$ 
         $\begin{cases} \text{UNCHANGED} & \text{if } \sigma \geq \tau_c; \\ \text{EDITED} & \text{otherwise} \end{cases}$ ;
6      $\mathcal{A}[s_i.\text{path}] \leftarrow (s_j.\text{path}, \text{status}, \sigma)$ ;
7      $\mathcal{M} \leftarrow \mathcal{M} \cup \{s_j.\text{path}\}$ ;
8     continue;
    // Phase 2: Fuzzy heading match
9    $(\hat{s}, \hat{r}) \leftarrow$ 
         $\arg \max_{s_j \in \mathcal{S}^{(n)} \setminus \mathcal{M}} \text{SIM}(s_i.\text{path}, s_j.\text{path})$ ;
10  if  $\hat{r} \geq \tau_h$  then
11     $\sigma \leftarrow \text{SIM}(s_i.\text{body}, \hat{s}.\text{body})$ ;
12     $\mathcal{A}[s_i.\text{path}] \leftarrow$ 
         $(\hat{s}.\text{path}, \text{RENAMED}, \sigma)$ ;
13     $\mathcal{M} \leftarrow \mathcal{M} \cup \{\hat{s}.\text{path}\}$ ;
14    continue;
    // Phase 3: No match found
15   $\mathcal{A}[s_i.\text{path}] \leftarrow$ 
         $(\text{NULL}, \text{UNMATCHED}, 0)$ ;
16  $\mathcal{S}^+ \leftarrow \{s_j.\text{path} \mid s_j \in \mathcal{S}^{(n)} \wedge s_j.\text{path} \notin \mathcal{M}\}$ ;
    // added sections
17 return  $\mathcal{A}, \mathcal{S}^+$ 

```

Class	Metric	Version		
		v1	v2	v3
Stable	Median _{words}	27.0	31.4	30.7
	Median _{S_{ref}}	6.0	5.0	4.0
Evolved	Median _{words}	–	32.5	32.0
	Median _{S_{ref}}	–	5.0	5.0
Added	Median _{words}	–	34.6	18.4
	Median _{S_{ref}}	–	6.0	6.0

Table 11: Distribution of questions across classes and versions in the generated testset. Median_{words} is the median number of words per question. Median_{S_{ref}} is the median number of reference statements per question.

Dataset	Metric	v1	v2	v3
SME (n=146)	Recall	86.8	85.1	85.8
		[82.8, 90.4]	[81.0, 88.9]	[82.4, 89.5]
	Prec.	65.4	65.1	65.5
		[60.3, 70.0]	[59.9, 70.4]	[60.2, 70.1]
RAGAs (n=175)	Recall	72.7	71.0	70.1
		[69.5, 75.7]	[67.9, 74.6]	[67.2, 73.6]
	Prec.	65.5	67.5	64.9
		[62.1, 69.0]	[62.6, 71.8]	[61.6, 68.2]

Table 12: Generator robustness results on STABLE questions in % by corpus version. Confidence intervals are 95% bootstrap percentile intervals, reported in square brackets.

Class	Ver.	# Clusters	Cluster CI
ADDED	v2	4	[61.2, 85.5]
ADDED	v3	7	[67.4, 95.6]
STABLE paired Δ , v1→v2			[0.26, 0.73]
STABLE paired Δ , v2→v3			[0.93, 2.44]

Table 13: Subsection-level cluster-bootstrap 95% intervals for statement-level recall, with the number of supporting subsections (clusters) behind each estimate. The paired Δ is robust to the resampling unit. ADDED is supported by too few clusters for a stable interval.