Aion-BibleQA: Evaluating Retrieval and Citation Faithfulness in Verse-Grounded Bible RAG Systems

Mohammad Raouf Abedini
Department of Computing
Macquarie University
Sydney, Australia
mohammadraouf.abedini@students.mq.edu.au
https://raoufabedini.dev

Abstract

We introduce **Aion-BibleQA**, a 40-question pilot benchmark for evaluating citation faithfulness and false-premise robustness in verse-grounded Bible retrieval-augmented generation (RAG). Existing RAG evaluations measure whether the right document was retrieved, but not whether the system uses retrieved content as citation support in its answer (a gap that matters most when user trust depends on accurate scripture attribution). We build a layered retrieval pipeline that combines exact verse lookup, direct chapter-aware database queries, per-chapter coverage guarantees, and hybrid semantic re-ranking, and evaluate it with two metrics: Recall@5 for retrieval coverage and citation_support (an LLM-as-judge score, 0–1) for answer faithfulness. On gold_40 v0.3, our v3 system achieves R@5 = 0.941, mean citation_support = 0.978, zero unsupported citations, and a false-premise refusal rate of 1.000 (6/6). To guard against same-family judge bias, a cross-family judge (OpenAI `gpt-4.1`) re-scores all 40 rows under the same rubric: the two judges agree within one rubric level on every row and report identical unsupported-claim (0.000) and refusal (6/6) rates, so the headline findings are not an artifact of single-judge scoring. Failure analysis shows that the remaining errors trace to retrieval scope (semantic drift in thematic queries and within-chapter verse selection), not to citation misuse, suggesting that retrieval and faithfulness are separable failure modes in Bible RAG systems. Because the benchmark is small (40 questions) and developer-constructed, these results should be read as pilot evidence rather than a general estimate of Bible QA system performance.

1 Introduction

Bible study applications face a sharper faithfulness challenge than most RAG domains. A user who asks "What does John 3:16 say about love?" expects the system to cite John 3:16, not a topically adjacent verse. If the system retrieves the right chapter but cites the wrong verse within it, the answer sounds plausible while misattributing scripture, a failure invisible to standard retrieval metrics.

Retrieval-augmented generation (Lewis et al., 2020) is the dominant architecture for knowledge-intensive QA. Evaluation typically measures whether retrieval returned relevant documents (Recall@k, MRR) and whether the final answer matches a reference (BLEU, BERTScore, exact match). Neither metric captures whether the system used retrieved content correctly. A system can score R@5 = 1 while citing a retrieved passage for a claim it does not support.

We call this gap the **retrieval-faithfulness split**: retrieval correctness and citation faithfulness are distinct failure modes that require separate measurement.

This paper makes four contributions:

- **Aion-BibleQA**, a 40-question benchmark with gold verse annotations across six categories (direct, interpretive, thematic, multi-hop, false-premise, adversarial) designed to stress-test both retrieval and faithfulness.
- A **layered retrieval architecture** (exact verse lookup → chapter-aware DB query → per-chapter coverage guarantee → hybrid semantic re-ranking) achieving R@5 = 0.941 on the benchmark.
- **Empirical results** on gold_40 v0.3 using a Gemini LLM-as-judge citation protocol (Table 3): R@5 = 0.941, mean citation_support = 0.978, unsupported claim rate = 0.000, false-premise refusal rate = 1.000 (6/6), with a cross-family (GPT) judge corroborating every score within one rubric level (Section 5.4).
- A **failure taxonomy** identifying retrieval scope as the remaining bottleneck: semantic

drift and within-chapter verse selection, not citation misuse.

In this pilot, citation failures disappear when the retrieved verse set contains the required evidence. The split clarifies the engineering agenda: improve what gets retrieved, not how it is cited.

2 Related Work

2.1 Retrieval-Augmented Generation

Lewis et al. (2020) introduced RAG as a parametric–nonparametric hybrid: a dense retrieval component fetches supporting documents, and a seq2seq generator conditions on both the query and retrieved content. Subsequent work improved dense retrieval representations (Karpukhin et al., 2020) and showed that generative models benefit from conditioning on multiple retrieved passages simultaneously (Izacard and Grave, 2021). RAG has been extended to multi-hop settings where answers require aggregating evidence across documents (Tang and Yang, 2024). Retrieval quality caps answer quality: systems cannot generate faithful answers from content they did not retrieve.

In Bible RAG, retrieval quality and answer faithfulness are separable: a system can retrieve the right chapter while citing the wrong verse within it.

2.2 Faithfulness and Grounding Evaluation

Faithfulness (whether an answer is entailed by its supporting context) is distinct from factual correctness (Maynez et al., 2020). RAGAS (Es et al., 2023) decomposes RAG evaluation into faithfulness, answer relevance, and context precision, using an LLM to judge each dimension without ground-truth annotations. G-Eval (Liu et al., 2023) demonstrates strong correlation between GPT-4 judgments and human ratings for faithfulness criteria across summarization and dialogue tasks. Our Gemini judge scores citation_support; false_premise_refusal adds an orthogonal robustness dimension.

LLM-as-judge suffers from same-family bias: a judge rates outputs from the same model family higher (Zheng et al., 2023). Our judge is from the same model family as the answer model.

2.3 Bible NLP

Resnik et al. (1999) established the Bible as a parallel corpus for cross-lingual alignment, a line extended by the eBible Corpus (Akerman et al., 2023) covering 833 languages. Lima et al. (2025) survey AI applications to biblical text analysis and identify semantic search and question answering as emerging tasks with limited benchmark coverage. Citation-faithfulness evaluation has not previously been applied to Bible RAG systems.

2.4 False-Premise Robustness

False-premise robustness has been studied in general QA. Hu et al. (2023) construct a dataset of false-premise questions and show that current systems answer false-premise questions as if the premise were true. Bible QA has a specific form of this problem: users frequently misquote or misattribute verses. We include false-premise and adversarial categories in Aion-BibleQA to measure whether the system refuses fabrication requests and corrects misattributions.

3 System Architecture

3.1 Overview

Aion is a verse-grounded Bible QA system deployed as a Supabase Edge Function. Given a user query, the system parses explicit Bible references, retrieves candidate verses, re-ranks them semantically, and generates a grounded answer citing specific verses. Figure 1 shows the pipeline.

3.2 Reference Parsing

A two-pass parser identifies explicit scripture references in the query.

Pass 1 (verse-level): Regex matches `BOOK CHAPTER:VERSE` patterns (e.g., "John 3:16", "Ps. 23:1"). An alias map normalizes common abbreviations and alternate spellings to canonical book IDs.

Pass 2 (chapter-only): A separate regex captures patterns like "Psalm 23" or "1 Corinthians 15" where no verse is specified. A backtrack fix (`lastIndex = match.index + 1` on alias miss) prevents tokens such as "Does 1" from consuming the leading digit of "1 Corinthians".

Numeric keyword suppression: Parsed chapter references suppress numeric token extraction in the downstream semantic query. Without this, a query for "Psalm 23" generates keyword "23", retrieving unrelated numerical content (census records, inventory lists).

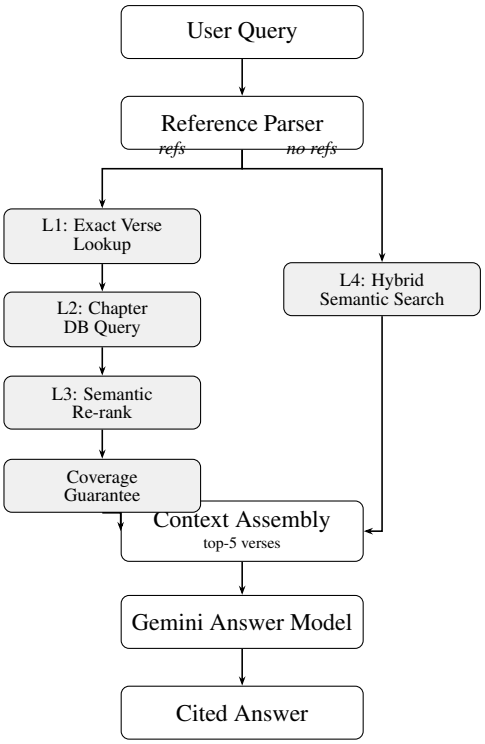

Figure 1: Aion v3 retrieval pipeline. Queries with explicit scripture references fork left through exact lookup (L1), chapter-aware DB query (L2), in-chapter semantic re-ranking (L3), and a per-chapter coverage guarantee. Queries without references fork right through hybrid semantic search (L4). Both paths merge at context assembly, which selects the top-5 verses for the answer model.

3.3 Layered Retrieval (v3 Direct-Chapter)

Retrieval proceeds in four layers, ordered by specificity.

Layer 1 — Exact verse lookup. For verse-level references, the system queries the `bible_verses` table directly by (*book_id*, *chapter*, *verse*). These rows are added to the candidate set without a semantic ranking step.

Layer 2 — Chapter-aware DB query (`lookupChapterVerses`). For chapter-only references, all verses in the referenced chapter are fetched directly from `bible_verses` by (*book_id*, *chapter*). This avoids the v2 design flaw of running an unconstrained semantic search then filtering, which could return semantically close verses from other chapters.

Layer 3 — Semantic re-ranking within chapter (`selectWithinChapters`). Fetched chapter verses are re-ranked by cosine similarity to the query embedding. An IVFFlat index on the `embedding` column provides approximate nearest-neighbor search. The top-k results are selected.

Per-chapter coverage guarantee. After semantic re-ranking, the system verifies that at least one verse from each referenced chapter appears in the final candidate set. If not, it appends the chapter's first verse. This ensures multi-hop queries referencing two chapters return at least one verse from each.

Layer 4 — Hybrid semantic fallback. For queries with no parsed references, the system runs an unconstrained semantic search over all verses using the query embedding. Keywords extracted from the query bias results toward verses with matching lexical content.

Context assembly. The top-5 verses across all layers are assembled into a context block formatted as `BOOK CHAPTER:VERSE — {text}` on separate lines.

3.4 Answer Generation

We pass the assembled verse context and original query to `gemini-3.1-flash-lite` (Google DeepMind, 2026) with a system prompt that instructs it to: (1) answer using only the provided verses and not recall verses from memory; (2) cite each referenced verse inline using the format `Book Chapter:Verse`; and (3) acknowledge when the provided verses do not address the question. The prompt contains no explicit instruction to detect false premises or refuse fabrication requests; the 1.000 refusal rate in Table 4 is an emergent consequence of these three constraints (see Appendix A).

3.5 Coarse Retrieval Ablation

The v1→v3 progression doubles as a coarse ablation: each version swaps in one retrieval strategy, so the version-over-version deltas indicate the direction of each design change rather than the isolated contribution of a single layer. v3 matches v2 on the same dataset (R@5 = 0.882; Table 1) despite replacing the retrieval architecture. It fixes aion_035 (Psalm 23 + John 10 multi-hop) but correctly rebreaks aion_036, an accidental success in v2 from an unrestricted semantic fallback.

4 Experiments

4.1 Benchmark: Aion-BibleQA gold_40

Dataset: `aion_bibleqa_gold_40_v0.3.jsonl`, 40 questions across 6 categories (Table 2). Each question has a `gold_verse` (primary expected verse coordinate, e.g., `JHN.3.16`) and an `acceptable_verse_clusters` list. R@5 = 1 if

System	Dataset	R@5	MRR
v1 hybrid-ref	v0.1	0.676	0.552
v2 chapter-ref	v0.2	0.882	0.700
v3 direct-chapter	v0.2	0.882	0.714
v3 direct-chapter	v0.3	**0.941**	**0.773**

Table 1: Coarse retrieval ablation across system versions. Each version swaps in one retrieval strategy, isolating the direction of each change rather than individual-layer contribution. The v0.3 gain (+0.059 R@5) is attributable to annotation expansion for two questions (aion_023, aion_033), not architecture change.

Category	n	Description
direct	10	Names a specific verse (e.g., "What does John 3:16 say?")
interpretive	7	Asks for meaning of a named passage
thematic	12	Theme query without naming a verse
multi_hop	5	Requires verses from two different chapters
false_premise	4	Contains a factual error about scripture
adversarial	2	Asks the system to fabricate content

Table 2: Aion-BibleQA category distribution.

any retrieved verse matches the gold or any cluster member. We verified all 40 questions against the live `bible_verses` table (BSB translation).

Annotation note: Aion-BibleQA is a pilot benchmark created for this study. Questions cover six failure modes identified during v1 system development, not a random sample of user queries. Results should be interpreted in this context.

4.2 Retrieval Metrics

Recall@5 (R@5): Binary. Score 1 if any of the top-5 retrieved verse coordinates matches `gold_verse` or any cluster member; else 0.

MRR (Mean Reciprocal Rank): $1/r$ where r is the rank of the first matching verse in the ranked list; 0 if no match in top-5.

We compute both metrics per row and report averages per category and overall.

4.3 Citation-Faithfulness Evaluation

Judge model: `gemini-3.1-flash-lite` (Google DeepMind, 2026).

For each row, the judge receives: (1) the question, (2) the category, (3) expected behavior from the dataset annotation (category-level guidance and refusal expectation, but *not* gold verse coordinates), (4) the retrieved verse block with full verse texts, and (5) the system answer. The judge scored only whether the answer's claims were supported by the cited retrieved verse text; it did not score retrieval relevance against gold annotations.

Score	Meaning
1.00	Every claim directly grounded in the cited verse text
0.75	Mostly grounded; minor over-reach on one citation
0.50	Some claims go beyond what the verses say
0.25	Most claims loosely connected to the text
0.00	Verses decorative; claims unsupported or fabricated

Table 3: citation_support scoring rubric used by the LLM judge.

The judge produces a JSON response with two fields:

- `citation_support` (0.0–1.0) for direct/interpretive/thematic/multi_hop rows
- `false_premise_refusal` (0 or 1) for false_premise/adversarial rows

Implementation: We judged all 40 rows in a single pass with three-attempt retry per row and exponential backoff; 0 judge call failures.

4.4 Aggregate Metrics

- **mean_citation_support:** Mean cs across non-refusal rows (n = 34)
- **unsupported_claim_rate:** Fraction of rows with cs < 0.5
- **decorative_citation_rate:** Fraction of rows with cs $\leq$ 0.25
- **fp_refusal_rate:** Fraction of false_premise/adversarial rows with false_premise_refusal = 1

4.5 Experimental Setup

Bible translation: BSB (Berean Standard Bible) stored in a Supabase `bible_verses` table. Embedding model: `text-embedding-3-small` (OpenAI; 1,536 dimensions), stored as `halfvec(1536)` in pgvector with an IVFFlat index (`halfvec_cosine_ops`). Answer model: Gemini (same model family as the primary judge; see the Limitations section), deployed as a Supabase Edge Function on Deno Deploy infrastructure. Judges: `gemini-3.1-flash-lite` (primary) and OpenAI `gpt-4.1` (secondary, cross-family robustness check; Section 5.4). Both receive an identical rubric and prompt at

Metric	Score
R@5	0.941
MRR	0.773
Mean citation_support	0.978
Unsupported claim rate (cs < 0.5)	0.000
Decorative citation rate (cs ≤ 0.25)	0.000
False-premise / adversarial refusal	1.000 (6/6)

Table 4: v3 system on gold_40 v0.3: overall metrics.

Category	n	R@5	MRR	cs	fp
direct	10	1.000	1.000	1.000	—
interpretive	7	1.000	0.929	1.000	—
thematic	12	0.917	0.524	0.979	—
multi_hop	5	0.800	0.700	0.900	—
false_premise	4	—	—	—	1.000
adversarial	2	—	—	—	1.000

Table 5: Per-category results. cs = citation_support; fp = fp_refusal. 95% Wilson CIs for R@5: overall [0.80, 0.97], multi_hop [0.38, 0.96].

temperature 0.1. Benchmark runner: calls the live Edge Function over HTTPS with SSE streaming from a consumer MacBook. All benchmark runs are frozen as JSONL artifacts; results are reproducible from those frozen files.

5 Results

5.1 Main Results

v3 on gold_40 v0.3 achieves R@5 = 0.941, citation_support = 0.978, and fp_refusal = 1.000 (Table 4), suggesting that once retrieval scope is correct, the system cites retrieved verses faithfully.

5.2 Per-Category Breakdown

Direct and interpretive categories. R@5 = 1.00 and citation_support = 1.00 for all 17 questions (Table 5). The correct verse appears in the top-5 for every question in these categories. MRR = 0.929 for interpretive (vs. 1.000 for direct) reflects one question (aion_006) where the correct verse was retrieved at rank 2; no citation failure resulted.

Thematic and multi-hop contain the remaining retrieval failures. Two questions fail R@5: aion_027 (thematic, grace semantic drift) and aion_036 (multi-hop, IVFFlat boundary). Citation faithfulness remains high for both categories (cs = 0.979 and 0.900), showing that the system cites retrieved content correctly even when retrieval returned suboptimal verses.

Multi-hop coverage note. Standard R@5 passes a multi-hop question if *any* required verse is retrieved. Under a stricter per-chapter criterion (success only if every required chapter is represented in the top-5), multi-hop all-chapters recall is also 0.800, but with a different failure: aion_034 passes standard R@5 (JAS.2.17 retrieved) yet misses the ROM.3 chapter entirely, while aion_036 has both required chapters present but neither specific gold verse. The two metrics agree on score but expose complementary failure modes.

Metric	Gemini	GPT
Mean citation_support (n = 34)	0.978	0.941
Unsupported rate (cs < 0.5)	0.000	0.000
Decorative rate (cs ≤ 0.25)	0.000	0.000
False-premise refusal	1.000	1.000

Table 6: Two-judge panel on gold_40 v0.3. The cross-family judge reproduces the unsupported-claim and refusal rates exactly; mean citation_support differs by 0.037.

Refusal categories. All six refusal-category questions scored 1 (4 false-premise, 2 adversarial). No fabricated content appeared in any answer. The adversarial sample (n = 2) is too small for confident generalization; these results are directionally encouraging.

5.3 Sub-1.0 Rows

Two rows scored below citation_support = 1.0.

aion_021 (thematic, cs = 0.75): EPH.6.19 cited for a gratitude context. The verse records Paul requesting prayer for his ministry, not a general statement about thankfulness. A minor over-reach.

aion_035 (multi-hop, cs = 0.50): JHN.10.1 cited instead of JHN.10.11. The per-chapter coverage guarantee appends the first verse of JHN.10 (the parable of the gate) rather than the theologically central verse ("I am the good shepherd"). Notably, R@5 = 1 for this question (PSA.23.1 retrieved correctly); the failure is within-chapter verse selection, not retrieval coverage. aion_035 achieves R@5 = 1 yet cs = 0.50, a direct illustration of the retrieval-faithfulness split.

5.4 Multi-Judge Robustness

The primary judge (Gemini) shares a model family with the answer model, which risks same-family leniency (Zheng et al., 2023). To test whether the headline scores survive a change of judge, we rescored all 40 rows with a cross-family judge (OpenAI `gpt-4.1` (OpenAI, 2025)) under the identical rubric and prompt (Table 6).

The two judges agree exactly on 31/40 rows

Class	Example	Fix
Semantic drift	aion_027: "grace" → salutation formulas	Query expansion + salutation penalty (v3.1)
IVFFlat boundary	aion_036: PHP.4.6 inconsistently absent	Per-chapter vector RPC (v4)
Per-chapter wrong verse	aion_035: JHN.10.1 vs JHN.10.11	Per-chapter vector RPC (v4)

Table 7: Remaining failure taxonomy. None is an unsupported-citation or decorative-citation failure; the system does not fabricate verse support. aion_035 scores cs = 0.50 because the per-chapter guarantee returned the wrong verse within the correct chapter, not because the model cited a verse it did not retrieve.

and *within one rubric level on all 40* (every disagreement is a single 0.25 step). Critically, the conclusions that do not depend on a fractional score are judge-invariant: both judges place zero rows below the unsupported threshold and confirm all 6 refusals. The nine disagreements are minor over-reach calls (e.g. `gpt-4.1` rates aion_021 at 1.00 where Gemini gives 0.75, and the reverse on aion_027); none crosses a decision boundary. We report this as an automated multi-model robustness check, not as expert human adjudication.

6 Discussion

6.1 The Retrieval-Faithfulness Split

Retrieval correctness and citation faithfulness are separable. The two metrics diverge in multi_hop: R@5 = 0.800 while citation_support = 0.900. A system can retrieve the wrong verse (R@5 = 0) yet cite whatever it retrieved correctly (cs = 1.00): aion_027 illustrates this. Conversely, a system can achieve R@5 = 1 while citing the wrong verse within a chapter (cs = 0.50): aion_035 illustrates this.

Both metrics are necessary: high R@5 does not prevent unsupported-citation failures, and high citation_support does not confirm that retrieved content was relevant. We use "citation misuse" to mean citing verses that are decorative or do not support the stated claim; aion_035 is not this kind of failure (JHN.10.1 was retrieved and cited faithfully; the per-chapter guarantee returned the wrong verse within the correct chapter).

6.2 Failure Taxonomy

All remaining failures fall into one of three classes (Table 7).

In every case where retrieval returned relevant content, the answer model cited it correctly. Fixing retrieval scope addresses both remaining failure classes.

6.3 The aion_027 Grace Drift

"Grace" in Pauline letters has two semantic neighborhoods: salvific grace (EPH.2.8–9: "For by grace you have been saved through faith") and epistolary grace (greeting formulas of the form "Grace and peace to you…"). The epistolary cluster dominates the embedding space: these short, formulaic verses have embeddings closer to the unqualified query "grace" than the longer salvific passage.

Two candidate fixes: (1) query expansion, prepending "saved by" or "salvation" to thematic grace queries at retrieval time; (2) salutation suppression, a post-retrieval filter penalizing verses matching the pattern "Grace and peace to you from X". The second is more targeted; the first generalizes more broadly to other semantic drift cases.

6.4 v4 Architecture Roadmap

aion_035 and aion_036 share a root cause: the per-chapter guarantee and IVFFlat search operate at chapter granularity, not verse-within-chapter granularity.

The v4 fix is a `search_verses_in_chapter` Supabase RPC running constrained vector search within a single chapter, parameterized by query embedding, book ID, chapter number, and return count k. For JHN.10, the intended effect is that a query embedding for "God as shepherd" would rank JHN.10.11 above JHN.10.1; this has not been experimentally confirmed. The per-chapter guarantee would then return the semantically best verse within the chapter rather than the chapter's first verse. This is a well-defined engineering task with a clear specification derived from the failure analysis.

7 Conclusion

We introduced Aion-BibleQA, a 40-question benchmark for citation-faithfulness and false-premise robustness in verse-grounded Bible RAG. Our v3 retrieval system, combining exact verse lookup, direct chapter-aware database queries, per-chapter coverage guarantees, and hybrid semantic re-ranking, achieves R@5 = 0.941 on gold_40 v0.3. A Gemini LLM-as-judge evaluation shows mean citation_support = 0.978, zero unsupported citations, and a 1.000 false-premise refusal rate.

Retrieval and citation faithfulness are distinct failure modes. In this pilot, citation failures disappear when the retrieved verse set contains the required evidence. The remaining errors (semantic drift in thematic queries, IVFFlat boundary effects in multi-hop retrieval, and within-chapter verse selection) are retrieval scope problems.

Future priorities:

- Expert (theological) human annotation, beyond the automated cross-family judge agreement already reported.
- Expansion to 200+ questions sampled from real users.
- Thematic query expansion for grace-like semantic drift (v3.1).
- Per-chapter vector search RPC for within-chapter verse selection (v4).

We release Aion-BibleQA, the retrieval system, the judge harness, and all benchmark artifacts alongside this paper at `https://github.com/Raoof128/Aion` (code) and (Abedini, 2026) (frozen dataset, Zenodo DOI: 10.5281/zenodo.20522874).

Limitations

Automated judging, no expert human validation. Citation-faithfulness scores come from LLM judges, and the primary judge (Gemini) shares a model family with the answer model, which risks same-family leniency (Zheng et al., 2023). We mitigate this with a cross-family judge (`gpt-4.1`) on all 40 rows: the two judges agree within one rubric level on every row and report identical unsupported-claim and refusal rates (Section 5.4), so the headline conclusions are robust to the choice of judge. We do not, however, claim expert theological validation: we evaluate only whether generated claims are supported by the cited verse text, under a reproducible multi-model protocol. Expert human annotation, and a larger automated panel (a third judge harness is included but unrun for lack of an API key), remain future work.

Pilot benchmark size. gold_40 v0.3 contains 40 questions. Statistical uncertainty is high: R@5 is averaged over the 34 non-refusal rows, so one question shifts it by roughly 3% (1 of 34). Results in the multi_hop category (n = 5) and adversarial category (n = 2) are especially noisy. A 95% CI on multi_hop R@5 = 0.800 with n = 5 spans roughly ± 0.30.

Constructed benchmark, not user-sampled. Questions target known failure modes from v1 development, not the distribution of queries real users ask. The benchmark was created by the same team that built the system under evaluation; scores may overstate performance relative to an independently assembled evaluation set. Real-world R@5 may differ.

Single translation (BSB). All verses come from the Berean Standard Bible. Results may not generalize to systems using KJV, NIV, ESV, or other translations, where verse wording differs and embeddings shift.

IVFFlat non-determinism. Across three repeated runs on gold_40 v0.2, three questions showed R@5 variance (aion_006, aion_007, aion_036). Run-to-run R@5 ranged from 0.853 to 0.882 on the same question set. Benchmark runs in this paper were each performed once and frozen as canonical JSONL artifacts; results are reproducible from those frozen files.

Coarse, not controlled, ablation. The v1→v2→v3 progression (Table 1) acts as a coarse ablation showing the direction of each design change, but each version alters more than one component at once, so the isolated contribution of an individual retrieval layer is not quantified.

Acknowledgments

We thank the Berean Bible translators for dedicating the BSB to the public domain with all uses freely permitted, which made this corpus-based study possible.

Ethics Statement

This study uses the Berean Standard Bible (BSB), whose texts are dedicated to the public domain with all uses freely permitted. No personal data were collected; Aion-BibleQA contains no human subject information. The judge and answer models are commercial APIs; no model weights were modified or redistributed. The benchmark targets a low-risk religious reference task; nonetheless, retrieval failures that misattribute scripture could mislead users about biblical content, and results should not be interpreted as endorsing any theological position.

References

Mohammad Raouf Abedini. 2026. Aion-BibleQA: Citation-faithfulness and false-premise robustness benchmark for bible RAG.

Vesa Akerman, David Baines, Damian Daspit, and Ulf Hermjakob. 2023. The eBible corpus: Data and model benchmarks for Bible translation for low-resource languages. ArXiv:2304.09919.

Shahul Es, Jithin James, Luis Espinosa-Anke, and Steven Schockaert. 2023. RAGAS: Automated evaluation of retrieval augmented generation. ArXiv:2309.15217.

Google DeepMind. 2026. Gemini 3.1 Flash-Lite. Model card, Google AI for Developers. Generally available May 2026. https://ai.google.dev/gemini-api/docs/changelog.

Shengding Hu, Yulong Luo, Huadong Wang, Xingyi Cheng, and Zhiyuan Liu. 2023. Won't get fooled again: Answering questions with false premises. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*. ACL 2023.

Gautier Izacard and Edouard Grave. 2021. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics*. EACL 2021. arXiv:2007.01282.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*. EMNLP 2020. arXiv:2004.04906.

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*. ArXiv:2005.11401.

B. C. Lima, N. Omar, I. Avansi, and L. N. de Castro. 2025. Artificial intelligence applied to the analysis of biblical scriptures: A systematic review. *Analytics*, 4(2):13.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-Eval: NLG evaluation using GPT-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. EMNLP 2023. arXiv:2303.16634.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. ACL 2020. arXiv:2005.00661.

OpenAI. 2025. GPT-4.1. OpenAI API release notes. https://platform.openai.com/docs/models.

Philip Resnik, Mari Broman Olsen, and Mona Diab. 1999. The Bible as a parallel corpus: Annotating the 'Book of 2000 tongues'. *Computers and the Humanities*, 33(1–2).

Yixuan Tang and Yi Yang. 2024. MultiHop-RAG: Benchmarking retrieval-augmented generation for multi-hop queries. ArXiv:2401.15391.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. In *Advances in Neural Information Processing Systems*. NeurIPS 2023 Datasets and Benchmarks Track. arXiv:2306.05685.

A Answer Generation System Prompt

The following prompt is used verbatim by `buildPrompt()` in `supabase/functions/chat/index.ts`. `{versesBlock}` is populated at runtime with the top-5 retrieved verses formatted as `BOOK CH:VS — "text"` on separate lines; `{userMessage}` is the verbatim user query.

```
You are Aion, a wise and warm Bible
companion.  You help people explore
Scripture with clarity and warmth.

RULES:
- Answer using ONLY the provided verses
below. Do not invent or recall
verses from memory.
- Cite each verse you reference using the
format: Book Chapter:Verse.
- If the provided verses don't answer the
question, say so honestly and
suggest what the user might search for
instead.
- Keep your response concise and
conversational (2-4 short paragraphs
max).

[Retrieved Verses]
{versesBlock}

[User Question]
{userMessage}
```

Design note. The prompt contains no explicit false-premise detection or fabrication-refusal instruction. The refusal behavior follows from the grounding-only constraints: Rule 1 prohibits recalled knowledge; when the retrieved verse block contains no relevant content (as occurs for non-existent passages or adversarial queries), Rule 3 produces a natural refusal. Robustness to false premises is therefore a consequence of the retrieval constraints rather than an explicitly programmed detection rule.