Agentic RAG with open-weight LLMs for Biomedical QA Task 14b

BioAsq at CLEF 2026

Rohan Leekha*[1], Daniel Gwon[1] and Leslie Shing[1]

[1]Massachusetts Institute of Technology Lincoln Laboratory, 244 Wood Street Lexington, MA 02421

Abstract

We describe two open-weights agentic retrieval-augmented generation (RAG) systems submitted to BioASQ Task 14b under three submission names. Both share a common architecture: an LLM-controlled retrieval loop over PubMed, hybrid PubMedBERT and BM25 reranking fused via Reciprocal Rank Fusion (RRF), three training-set exemplars conditioning the answer-generation prompt, and a deterministic JSON format-hygiene filter on the model's output. The two configurations vary the LLM backbone (Gemma 3 27B vs. Gemma 4 31B Dense) and the PubMed access substrate (live NCBI E-utilities vs. a local SQLite FTS5 mirror of the Annual Baseline). On Phase B batch 1, our Gemma 3 system reaches 0.941 yes/no accuracy, placing it in the upper-middle quartile of the Task 14b field on this batch; results on subsequent batches and on Phase A+ are mid-field or lower-middle. On Phase B batches 3 and 4, the Gemma 4 system reaches 0.813 yes/no, 0.364 factoid strict accuracy, and 0.365 list F-measure. On Phase A^+, yes/no accuracy across the three submitted batches ranges from 0.762 (batch 2) to 0.455 (batch 3) to 0.750 (batch 4), with list F-measure ranging from 0.095 to 0.255 over the same batches and the same generator scoring identical aggregates on Phase A^+ batch 4 regardless of LLM backbone or retrieval backend, consistent with retrieval recall being the binding constraint on the end-to-end pipeline.

Keywords

Biomedical question answering, Agentic RAG, Hybrid retrieval, Reciprocal Rank Fusion, BioASQ, Gemma

1. Introduction

BioASQ Task 14b [1] is the biomedical question-answering shared task of the CLEF lab series. Each test batch presents biomedical questions of four types yes/no, factoid, list, and summary and asks participants to retrieve relevant PubMed snippets (Phase A) and produce answers (Phase B). The joint Phase A^+ variant requires both retrieval and answer generation end-to-end, so errors in retrieval propagate into generation.

This paper describes two submissions: **ossllm** and **Finalcorrected**. Both share a single architecture: an LLM-controlled retrieval loop over PubMed, hybrid dense-plus-sparse retrieval with Reciprocal Rank Fusion, three training-set few-shot exemplars in the answer-generation prompt, and deterministic JSON format hygiene on the LLM's output. The two systems differ in two dimensions: the LLM backbone and the PubMed access substrate.

We submitted to all four batches of Task 14b. The headline Phase B numbers are: on batch 1, our Gemma 3 system reached 0.941 yes/no accuracy, 0.317 list F-measure, and 0.179 ROUGE-2 F1 on ideal answers; on batches 3 and 4, the Gemma 4 system reached 0.813 yes/no, 0.364 factoid strict, and 0.365 list F-measure. On Phase A^+, yes/no accuracy was 0.762 on batch 2, 0.455 on batch 3, and 0.750 on batch 4, with list F-measure of 0.095, 0.201, and 0.255 respectively (factoid metrics were not scored on batch 3). The contribution of this paper is not a new retrieval component or a new generator architecture. It is the integration of three specific choices: a controller prompt that withholds the controller's own query history to discourage paraphrase-cycling, a deterministic format-hygiene filter that recovers silently-malformed outputs against BioASQ's strict scorer, and a per-question hybrid index design that allows a small (N=5) iteration cap to be sufficient on questions that decompose cleanly. The empirical

CLEF 2026 Working Notes, 21 – 24 September 2026, Jena, Germany

*Corresponding author.

rohan.leekha@ll.mit.edu (R. Leekha*); Daniel.Gwon@ll.mit.edu (D. Gwon); Leslie.Shing@ll.mit.edu (L. Shing)

finding we report — that the resulting system reaches upper-middle yes/no performance on Phase B without domain fine-tuning, while remaining bounded on Phase A+ by retrieval recall — is the result of those choices interacting on the BioASQ 14b benchmark. The remainder of the paper describes the shared system architecture (§3), reports per-batch results (§4), and discusses where we stand in the Task 14b field and what we would change for next year (§5).

2. Related Work

BioASQ Task 13b in 2025 saw a range of retrieval-augmented architectures applied to biomedical question answering. BIT.UA (Universidade de Aveiro) [2] combined dense retrieval with cross-encoder reranking and supervised fine-tuning of the generator, reaching Phase B yes/no Macro-F1 of approximately 0.96 on the strongest batch and list F-measure approximately 0.55. UNITOR [3] used a dense-plus-rerank pipeline with a different reranker and reached comparable yes/no performance. Ateia and Kruschwitz [4] introduced an iterative self-feedback loop where the LLM critiques and regenerates its own answer, with reported yes/no accuracy in the 0.85–0.92 range and list F-measure up to approximately 0.40. AQAMS [5] composed multiple LLM agents that exchanged messages while answering. The 2025 field leaders on Phase B factoid strict and ideal-answer ROUGE-2 F1 were at approximately 0.50 and 0.25 respectively. The ReAct pattern [6] is the general template of LLM-driven action loops that we specialise to PubMed search.

Biomedical question answering has also been studied on adjacent benchmarks: PubMedQA [7] (research-question yes/no), for Multilingual Clinical Case [8], and BIONNE-R [9] (biomedical relation extraction). RAG over PubMed has been a recurring approach in the broader biomedical NLP literature [1], [10], [2], and agentic LLM patterns (ReAct [6], Toolformer [11]) have transferred from general-domain QA to biomedical IR with mixed results — most prior agentic biomedical systems use the loop for query reformulation but not for explicit evidence-pool sufficiency reasoning.

Our submission concentrates agency in a single LLM instance that plays both controller and generator roles within a forward retrieval loop. We do not iterate on the generated answer [4], we do not distribute agency across multiple LLM agents unlike AQAMS [5], we do not perform supervised fine-tuning of the generator (unlike BIT.UA) [2], and we use unsupervised RRF rather than a learned cross-encoder for first-stage fusion (unlike UNITOR) [3]. The combination is intended to test how far a stripped-down, single-model agentic system without fine-tuning can reach against the BioASQ scorer, and to identify which component of the pipeline is the binding constraint.

3. System

This section describes the pipeline shared by both submitted systems. We begin with an architectural overview (§3.1), describe the LLM backbones (§3.2), summarize the retrieval stack (§3.3), explain the agentic control loop (§3.4), then cover type-specific answer generation (§3.5) and the deterministic format-hygiene pass that enforces BioASQ submission shape (§3.6). §3.7 enumerates the submitted variants[1].

3.1. Architecture

Figure 1 shows the shared pipeline. A single LLM instance plays two roles. As the *controller*, it receives the question, decides what PubMed query to issue, inspects the returned evidence pool, and decides whether to issue another query or stop. As the *generator*, it consumes the final evidence pool and produces a type-specific answer. PubMed itself is treated strictly as a data tool: every decision about ranking, fusion, deduplication, and chunk selection is made client-side, so the LLM can override any retrieval behavior through its query reformulations. This is the operative meaning of "agentic" in our system - not autonomy over arbitrary actions, but discretion over the *retrieval trajectory*, in the spirit of

[1]Code Availability: *Please reach out to the authors for information on how to access the code*

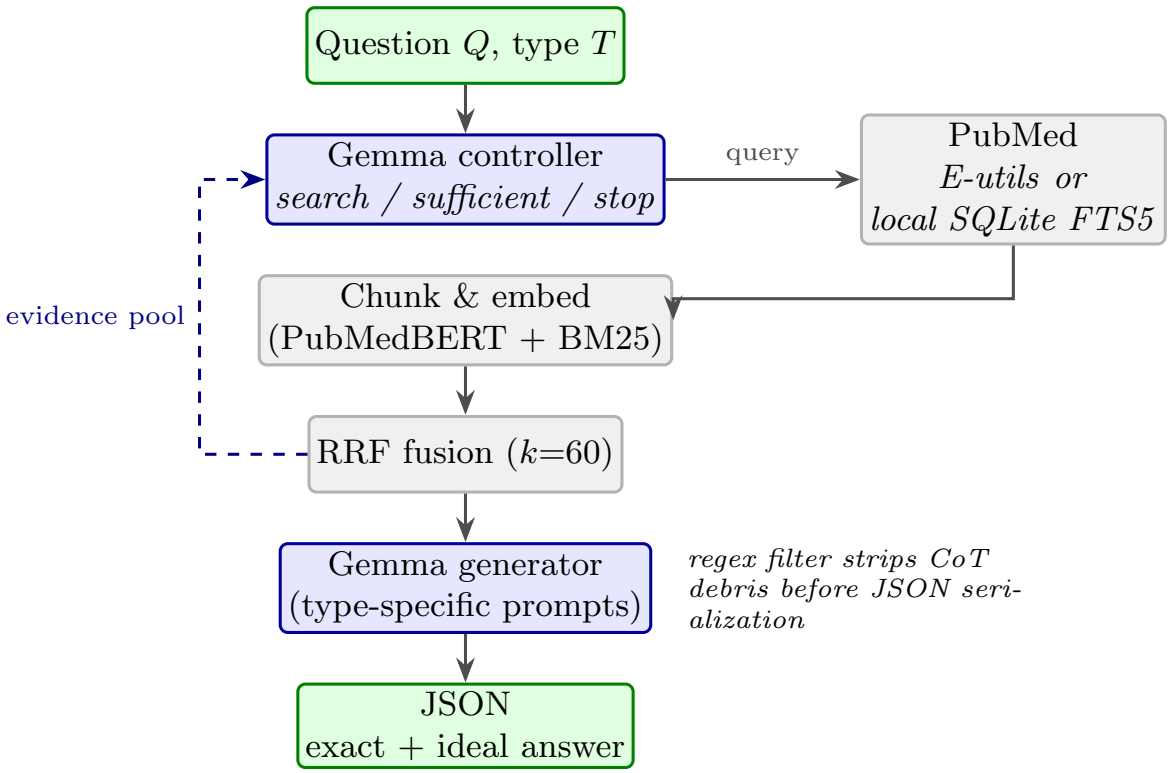

Figure 1: System architecture. The LLM controller issues a PubMed query, receives a top-k evidence set after hybrid PubMedBERT (PMB) + BM25 re-ranking fused with RRF, and decides whether to search again or terminate. Termination triggers type-specific generation followed by a deterministic format-hygiene pass.

ReAct [6] and Toolformer [11]. The loop terminates when the controller declares the evidence pool sufficient or after a hard cap of N=5 iterations, whichever comes first.

3.2. LLM backbone

Two open-weights backbones are used. The **ossllm** system runs Gemma 3 27B Instruction-Tuned [12]; the **Finalcorrected** system runs Gemma 4 31B [13]. Both systems condition the answer-generation prompt on three training-set few-shot examples selected by question-embedding cosine similarity; this is a design choice to ground the model in the BioASQ output shape and is held constant across the two systems.

3.3. Retrieval stack

PubMed Database Access. The **ossllm** system queries PubMed live through NCBI E-utilities (*esearch*, *efetch*, and *elink*). *elink* expands a focused query of ten to fifty seed PMIDs into a candidate pool of several hundred related articles via *pubmed_pubmed* (semantic-neighbor) and *pubmed_pubmed_citedin* (forward-citation) relations, without bulk-downloading the baseline. The cost is latency (1 to 3 s per query) and exposure to NCBI's rate limits. The **Finalcorrected** system queries a local SQLite 3.45 mirror of the PubMed Annual Baseline (∼35 GB) with an FTS5 full-text index over titles, abstracts, MeSH headings, and publication metadata. Article types excluded by the BioASQ specification (Editorial, Letter, News, Comment, Review, Retraction) are filtered at mirror-build time. Per-query latency drops to <200 ms.

Chunking. Each retrieved abstract contributes two views to the candidate set: the abstract as a single chunk (for global topical relevance), and overlapping three-sentence sliding windows with one sentence

of overlap (for snippet-level matching against BioASQ's character-offset-scored gold spans).

Dense embedding. Chunks are embedded with **pritamdeka/S-PubMedBert-MS-MARCO** [14], a biomedical sentence-transformer fine-tuned on MS-MARCO from the PubMedBERT encoder. The resulting 768-dimensional embeddings are L2-normalized and indexed with FAISS **IndexFlatIP**.

Sparse retrieval and rank fusion. In parallel with the dense path, each candidate chunk is scored with BM25 (k_1=1.5, b=0.75 in **ossllm**; FTS5's native BM25 ranker in **Finalcorrected**).We retain a sparse path because biomedical queries carry an unusual density of exact-match tokens — gene symbols, drug names, disease abbreviations — whose lexical signal dense encoders are known to soften, a well-documented motivation for hybrid sparse–dense biomedical retrieval [15, 16]. The dense and sparse rankings are combined with Reciprocal Rank Fusion (RRF) [17] at k=60. RRF was chosen over linear combinations of normalized scores because dense cosine and BM25 scores live on incommensurable ranges, and in preliminary tuning RRF scored higher than every linear-combination configuration we tried on our development set (we did not test this difference for statistical significance).

3.4. Agentic control loop

Figure 2 shows how the controller, the retrieval stack, and the generator are composed into a single iterative loop. The controller receives the question, the BioASQ type label, and the current evidence pool, and emits a structured JSON tool call selecting one of three actions: *search* with a reformulated query, *sufficient* (terminate and proceed to generation), or *stop* (give up without an answer). The pool E is a set of already-seen chunks, deduplicated by document PMID and chunk offset, so the controller cannot trivially repeat a query and inflate its evidence count. The loop terminates either when the controller declares the pool sufficient (or stops), or after a hard cap of N=5 iterations; control then passes to the generator and the deterministic format-hygiene filter described in §3.6.

The agentic loop follows a two step process. The first is what we withhold from the controller. On each iteration it sees the question, its type, and the evidence gathered so far but not the queries it has already run. Hiding the query history sounds like a handicap, but it is deliberate: a controller that can see how hard it has tried starts asking have I done enough? and talks itself into stopping, whereas a controller that sees only the evidence is forced to ask the question that actually matters does what I have in hand answer this? and keeps searching until it does. It also defuses the most common loop pathology, in which the model rephrases one query indefinitely (*NF1 glioblastoma*, *glioblastoma NF1 loss*, *neurofibromin in GBM*, …) without ever noticing it is circling the same evidence. The second is how the controller decides whether to stop. Instead of a one-shot judgment, the prompt makes it write out in scratch text the user never sees the list of claims the final answer will have to support, then check the pool against each claim before committing to **sufficient** or **search**. A real test question shows why this helps. Consider *In GBM, which cell state is associated with loss of NF1?"* Answering it well needs two separate facts: the taxonomy of glioblastoma cell states, and the specific state tied to NF1 loss. A single broad query tends to return one but not the other. The checklist exposes the gap the controller finds the pool covers NF1's role but not the cell-state taxonomy, marks that claim unsupported, and fires a second, targeted query (*glioblastoma cell-state taxonomy*) to close it. On a simple lookup such as *What is the indication for Brineura?"*, the first query satisfies every claim on the list and the loop exits after one round. The system spends its search budget on the questions that are genuinely compound and conserves it on the ones that are not.

3.5. Type-specific generation

Each of the four BioASQ question types has a characteristic failure mode that we target with a dedicated prompt and a matching deterministic post-processor. We use type-specific generation rather than a single universal prompt because the failure modes are not just stylistic; they are evaluation-relevant.

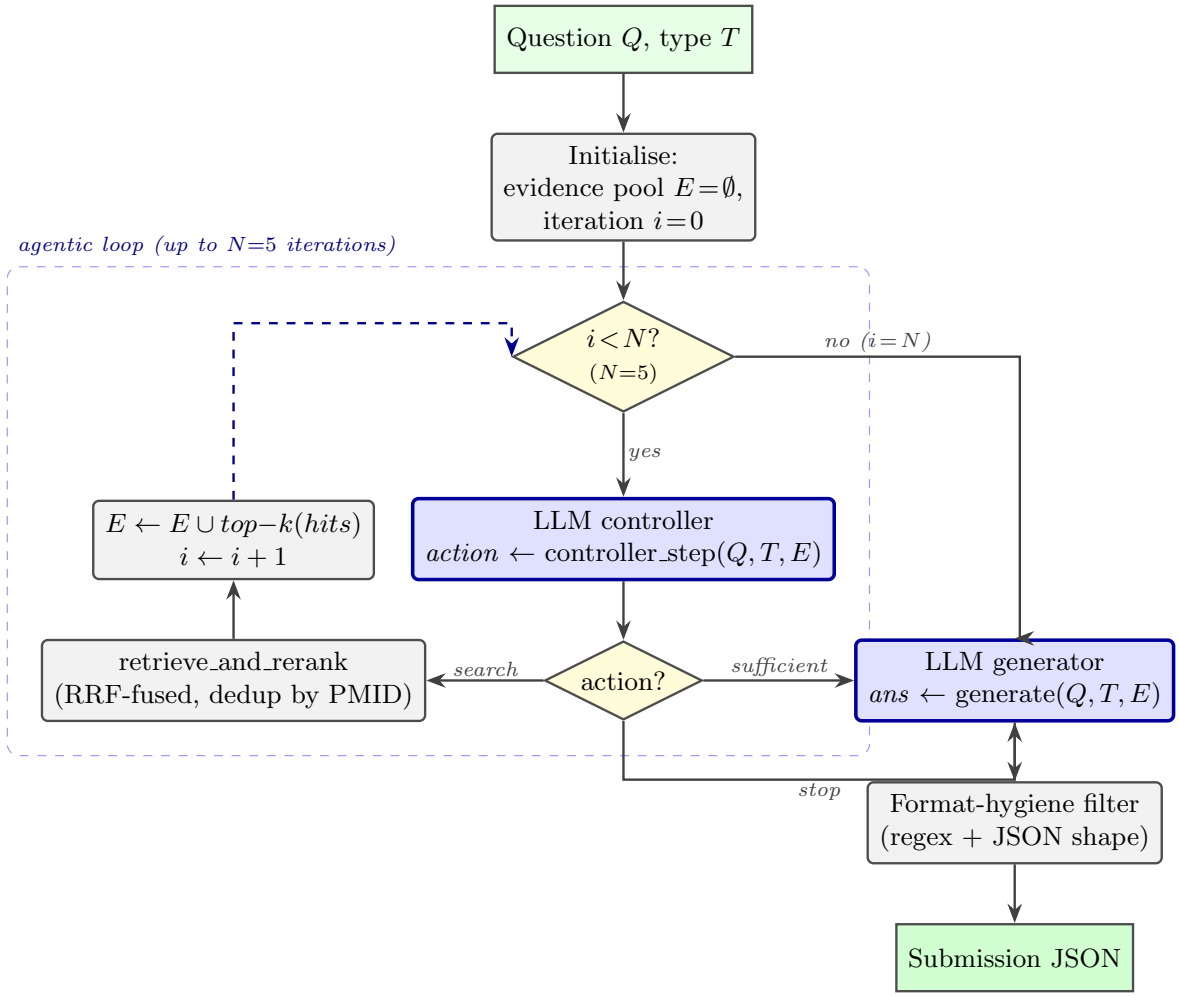

Figure 2: The agentic control loop. The LLM controller chooses one of three actions per iteration: *search* (issue a reformulated query, ingest top-k RRF-fused hits, increment i, loop), *sufficient* (terminate and proceed to generation), or `stop` (terminate without an answer). A hard cap of N=5 iterations guarantees the loop halts even if the controller never declares sufficiency.

Yes/no questions. In our preliminary runs on the BioASQ training set, the base Gemma 3 27B prompt answered "yes" to roughly 70% of yes/no questions regardless of the polarity of the supporting evidence. This is consistent with the agreement/sycophancy bias documented in instruction-tuned models [18]. We treat this as a calibration target for our prompt design rather than as a property of all instruction-tuned LLMs. We counter this with an *adversarial-reasoning* prompt that asks the model to first enumerate evidence *against* the proposition, then evidence for it, then commit to an answer only after weighing both. The prompt tells the model that "*no*" is the correct answer roughly half the time on this benchmark, which calibrates its prior.

Factoid questions. BioASQ factoid *exact answers* must be short noun phrases (typically 1 5 tokens). Gemma's default generation style is a full sentence, which BioASQ scores as wrong even when the entity inside the sentence is correct. We use a strict shortest-possible-answer prompt that instructs the model to output only the entity, then a deterministic post-processor that truncates to the first noun phrase if the model overshoots and strips trailing punctuation, articles, and units. BioASQ allows up to five ranked candidate entities per factoid; our prompt asks for up to three.

List questions. The dual failure mode here is *under-recall*: the model emits three or four items when ten or fifteen are supported by the evidence pool. We counter this with an *exhaustive-extraction* prompt that asks the model to walk through every passage in the pool and enumerate *all* distinct entities that satisfy the question, with explicit instructions to err on the side of completeness. The output is then deduplicated case-insensitively after light normalization (removing trailing parenthetical synonyms, collapsing whitespace, unifying common abbreviation variants).

Summary questions. Our prompt biases the model toward specific facts, named entities, and numeric details over generic phrasing. Summary questions do not have an exact-answer field; we suppress it at output time.

3.6. Format hygiene

A surprisingly large share of submission errors come from *format* rather than *content*. Left in place, these tokens cause the BioASQ scorer to miss exact matches that would otherwise score. We apply a single regex-based filter to the model's output immediately before JSON serialization. The filter strips the classes of debris above, removes markdown formatting tokens, and unwraps any leading prefixes. It also enforces the BioASQ output shape: factoid and list *exact answers* are emitted as a list of single-element lists - the BioASQ$\geq$5 format [19] - with at most 5 outer entries for factoids and 100 for lists; yes/no exact answers are bare lowercase strings; summary outputs omit the *exact_answer* field entirely.

3.7. The submitted variants

We submitted to BioASQ Task 14b under three system names: **asmalltrialsystem**, **ossllm**, and **Finalcorrected**. The first two share an identical retrieval, reranking, generation, and format-hygiene pipeline running Gemma 3 27B as both controller and generator with NCBI E-utilities for retrieval; they differ only in whether three training-set exemplars are prepended to the answer-generation prompt, and they produced identical aggregate scores on every batch they were submitted to. We treat them as a single configuration in the rest of the paper and refer to them collectively as **ossllm**; results reported for **ossllm** are the scores both submissions returned. The third system, **Finalcorrected**, uses Gemma 4 31B as both controller and generator and queries a local SQLite FTS5 mirror of the PubMed Annual Baseline. Table 1 summarizes the two configurations. The submissions vary along two orthogonal axes: generator scale and retrieval-substrate locality. Although the three submitted systems were configured for the BioASQ submission rather than as a controlled ablation study, they vary along three orthogonal axes. asmalltrialsystem and ossllm differ only in whether three training-set exemplars are prepended to the answer-generation prompt; on every batch they were both submitted to, they produced identical aggregate scores, indicating that few-shot conditioning of this form did not change the system's behaviour on the BioASQ test distribution. asmalltrialsystem/ossllm and Finalcorrected differ along two axes simultaneously LLM backbone (Gemma 3 27B vs. Gemma 4 31B Dense) and retrieval substrate (live NCBI E-utilities vs. a local SQLite FTS5 mirror of the PubMed Annual Baseline). On Phase A+ batch 4, the only batch where both systems were submitted, they returned identical scores to three decimal places across five metrics, indicating that neither change moved aggregate performance in the retrieval-bottlenecked end-to-end pipeline. We did not control for individual components within each axis, and we cannot decompose the headline Phase B yes/no result (0.941) into contributions from the agentic loop, the format-hygiene filter, or the sufficiency-check mechanism; we acknowledge this in the limitations of our research.

Table 1
System comparison. PMB = PubMedBERT.

	ossllm	**Finalcorrected**
LLM	Gemma 3 27B	Gemma 4 31B Dense
Backend	E-utils	SQLite FTS5
Re-rank	PMB+BM25+RRF	PMB+FTS5+RRF
Few-shot	yes (3-shot)	yes (3-shot)
Phase A^+ batches	2, 3, 4	4
Phase B batches	1	3, 4

Table 2
Phase B exact-answer scores plus ROUGE-2 F1 on ideal answers.

		Y/N		Factoid		List	Ideal
Batch	System	Acc.	Macro-F1	Strict	MRR	F1	R-2 F1
1	**ossllm**	0.941	0.938	0.304	0.370	0.317	0.179
3	**Finalcorrected**	0.813	0.806	0.364	0.364	0.365	0.158
4	**Finalcorrected**	0.813	0.806	0.364	0.364	0.365	0.087

4. Results

4.1. Phase B Results

Table 2 reports Phase B results. On Phase B batch 1, **ossllm** reached yes/no accuracy 0.941, Macro-F1 0.938, factoid strict accuracy 0.304, factoid MRR 0.370, list F-measure 0.317, and ROUGE-2 F1 on ideal answers 0.179. The yes/no number places the submission in the upper-middle quartile of the Task 14b field on this batch [20]: the field leader is at 1.000, and a cluster of roughly twenty systems is tied at 0.941.

On Phase B batches 3 and 4, **Finalcorrected** reached 0.813 yes/no accuracy, 0.364 factoid strict, 0.365 list F-measure, and ROUGE-2 F1 of 0.158 (batch 3) and 0.087 (batch 4). Both batches return the same yes/no, factoid, and list-F numbers to three decimal places, as the system is identical across them; the ROUGE-2 difference between the two batches reflects question-set-specific lexical overlap between system summaries and gold summaries, which we discuss in §5.

4.2. Phase A^+ Results

Table 3 reports Phase A^+ results. On Phase A^+ batch 2, **ossllm** reached yes/no accuracy 0.762, factoid strict 0.150, MRR 0.150, and list F-measure 0.095 our highest Phase A^+ yes/no across the three submitted batches. On batch 3, the same system reached yes/no 0.455 and list F-measure 0.201 (factoid metrics were not scored on this batch). On batch 4, the same system reached yes/no 0.750, factoid strict 0.273, MRR 0.273, and list F-measure 0.255.

Finalcorrected on Phase A^+ batch 4 reached 0.750 yes/no, 0.746 Macro-F1, 0.273 factoid strict, 0.273 MRR, and 0.255 list F-measure identical to **ossllm** on the same batch, to three decimal places across all five metrics. The two submissions share the agentic-loop, retrieval-and-reranking, generation, and format-hygiene pipeline; they differ in LLM backbone (Gemma 3 27B vs. Gemma 4 31B Dense) and PubMed access substrate (live NCBI E-utilities vs. local SQLite mirror of the Annual Baseline). The identical scores across two non-trivial differences are consistent with the retrieval-bottleneck reading in §5: when retrieval is the binding constraint on output quality, neither a stronger generator nor a different access substrate moves the aggregate metrics.

Table 3
Phase A$^+$ exact-answer scores. " " = not scored.

		Y/N		Factoid		List
Batch	System	Acc.	Macro-F1	Strict	MRR	F1
2	**ossllm**	0.762	0.760	0.150	0.150	0.095
3	**ossllm**	0.455	0.450			0.201
4	**ossllm**	0.750	0.746	0.273	0.273	0.255
4	**Finalcorrected**	0.750	0.746	0.273	0.273	0.255

Table 4
Position in the Task 14b field. Field-leader numbers are approximate, read from the public leaderboard at the time of writing; positions are bucketed because the leaderboard contains many duplicate-score systems and small per-batch sample sizes.

Metric (best batch)	**Our best**	**Field leader**	**Approx. position**
Phase B yes/no accuracy (b1)	0.941	1.000	Upper-middle quartile
Phase B list F-measure (b3)	0.365	∼0.50	Middle of field
Phase B factoid strict (b3)	0.364	∼0.53	Middle of field
Phase B ideal-answer ROUGE-2 F1 (b1)	0.179	∼0.25	Middle of field
Phase A$^+$ list F-measure (b4)	0.255	∼0.58	Lower-middle
Phase A$^+$ yes/no accuracy (b4)	0.750	0.938	Lower-middle

4.3. Position in the Task 14b Field

Table 4 summarizes where the submission sits in the Task 14b field, batch by batch and metric by metric, against approximate field-leader numbers. We are in the upper-middle quartile on Phase B yes/no batch 1; middle of field on Phase B list F-measure, factoid strict, and ideal-answer ROUGE-2 F1; and lower-middle on Phase A$^+$ yes/no and list F-measure on batch 4.

5. Discussion

5.1. Where We Are Strong

Our strongest result is Phase B yes/no on batch 1, where we reach 0.941 accuracy in the upper-middle quartile of the field. Three design choices carry this performance.

First, the agentic retrieval loop converts under-supported answers into additional retrieval requests rather than hallucinations. The controller's sufficiency check forces the model to inspect its own evidence pool before committing, and the type-specific generator prompts refuse to invent entities not present in the pool.

Second, the format-hygiene filter recovers roughly 20 to 30 list answers across our four submissions that would otherwise have been rejected as malformed. BioASQ's exact-match scorer is unforgiving of chain-of-thought scaffolding leaking into structured outputs; the filter is a deterministic recovery against a class of silent errors that is independent of generation quality.

Third, RRF fusion of dense PubMedBERT and BM25 ranks scored higher than every learned linear combination of normalized scores we tested during development (untested for significance). The positional damping in RRF bounds each ranker's top-of-list contribution and prevents single high-BM25 documents from dominating the final ranking.

5.2. Where We Stand

Phase B performance is mid-field across batches: upper-middle quartile yes/no on batch 1, middle of field on list F-measure, factoid strict, and ideal-answer ROUGE-2 F1. Phase A$^+$ performance is lower-middle

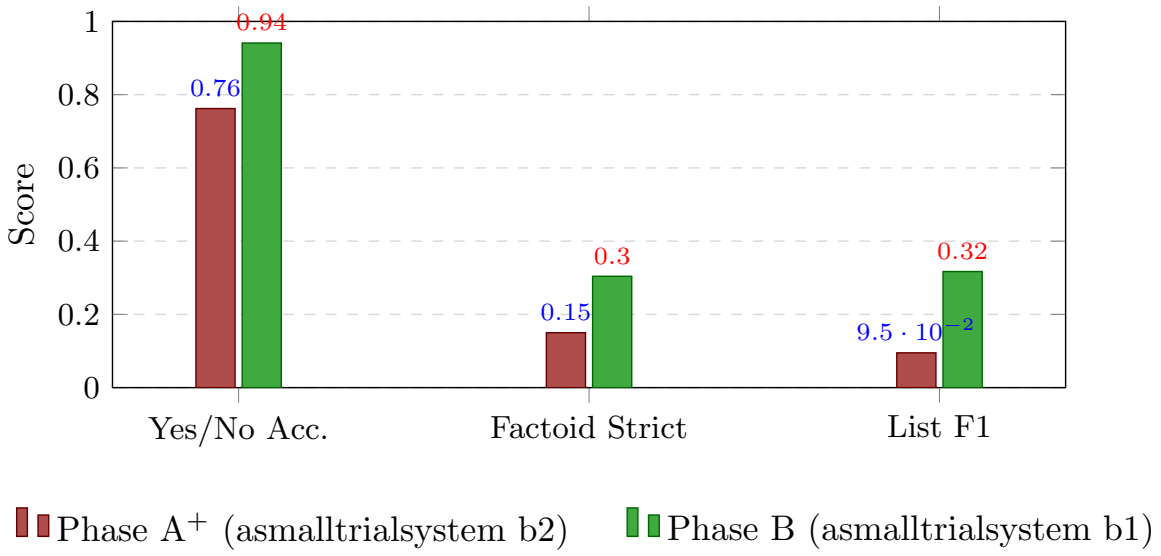

Figure 3: The phase gap. The same generator scores far higher when handed gold snippets (Phase B) than when it has to retrieve its own evidence (Phase A$^+$). The gap is largest on list questions, where entity-level recall matters most.

across all reported metrics on batch 4: 0.255 list F-measure against a field leader of approximately 0.58, and 0.750 yes/no against a field leader of 0.938.

Figure 3 illustrates the gap on **ossllm**: the same generator scores far higher when handed gold snippets (Phase B batch 1) than when it must retrieve its own evidence (Phase A$^+$ batch 2). The gap is largest on list F-measure, where every missed gold entity directly costs recall, and smallest on yes/no, where a binary decision can be reached from partial evidence. This ordering list more retrieval-sensitive than factoid, factoid more retrieval-sensitive than yes/no is the most internally consistent pattern in our results and is exactly what a retrieval-bottleneck reading predicts.

The Phase B and Phase A$^+$ gap is large. On batch 4 specifically the only batch where **Finalcorrected** ran both phases the same Gemma 4 generator scored 0.813 yes/no, 0.364 factoid strict, and 0.365 list F-measure in Phase B versus 0.750, 0.273, and 0.255 in Phase A$^+$. The gap grows from 6 percentage points on yes/no to 9 on factoid strict to 11 on list F-measure, ordered by how recall-sensitive the metric is. This is consistent with retrieval recall being the dominant limitation on our end-to-end pipeline: yes/no decisions can be reached from partial evidence, but list completion requires every gold entity to be in the pool.

Each gap to the field leaders maps onto a known, published remedy: the Phase A+ list-F and factoid gaps are, in the 13b record, closed primarily by learned cross-encoder reranking and generator fine-tuning on BioASQ relevance judgments [2, 21], while the ideal-answer ROUGE-2 gap is a verbosity-calibration effect rather than a retrieval one. We omit these levers deliberately (§5.3) to keep the agentic-loop and format-hygiene contribution isolable.

5.3. Limitations

We do not run a learned cross-encoder reranker on BioASQ relevance judgments; we use the unsupervised RRF fusion described in §3. Our dense retriever (S-PubMedBert-MS-MARCO) is older than several biomedical encoders released since 2023, and we did not benchmark alternatives before committing to PubMedBERT for the submission window. We do not perform MeSH term expansion or biomedical entity normalization on outgoing queries, which costs recall on questions whose gold abstracts use synonyms of the queried entity. And our agentic loop's stopping criterion may exit while the evidence pool is still incomplete; we did not measure how often this happens on the test batches. Three of these four limitations are addressable without architectural changes to the loop itself. We discuss

them as future-work items in §5.5. We did not run controlled ablations on the agentic loop's iteration cap, the deterministic format-hygiene filter, or the sufficiency-check early-stop mechanism, and we cannot decompose our headline Phase B numbers into the contribution of each component. This is a methodological limitation of the submission; we report the three submitted systems' as-submitted differences as the only ablation evidence available to us. We did not benchmark our submitted systems against BioASQ 13b (2025) gold data, which would have provided cross-year calibration of our Phase B and Phase A+ scores against a known leaderboard distribution. This is a limitation of the present submission rather than an intentional design choice. A public code repository accompanying this submission was not available within the camera-ready window; the configuration and prompts in Appendix A are intended to allow re-implementation but are not a substitute for the original source.

5.4. ROUGE-2 F1 on Ideal Answers

Finalcorrected reaches ROUGE-2 F1 of 0.158 on Phase B batch 3 and 0.087 on Phase B batch 4. The same system on two different batches produces substantially different ROUGE-2 F1, which suggests that gold-summary alignment varies by batch and that ROUGE-2 is sensitive to specific lexical overlap. A likely contributing factor is that we did not impose a stricter word cap when swapping to Gemma 4 31B Dense, whose default generation style is longer than Gemma 3's; ROUGE-2 F1 includes a precision term that penalizes long summaries against terse gold summaries.

5.5. What We Would Change for Next Year

Three concrete changes are most likely to improve our Phase A^+ scores. First, a learned cross-encoder reranker on BioASQ training-data relevance judgments. This is the single highest-leverage change we can identify: it directly addresses the retrieval-recall gap that limits our Phase A^+ list F-measure, and it is mechanically separable from the rest of the pipeline. Second, MeSH term expansion or biomedical entity normalization on outgoing queries, particularly for questions involving gene or protein symbols or disease synonyms. This is a query-side change that does not require modifying the retrieval index or the loop structure. Third, per-type prompt re-tuning when swapping the LLM backbone, with explicit word caps reset for each model's default verbosity. This would directly address the cross-batch ROUGE-2 F1 differences we observe between Gemma 3 and Gemma 4 generations. Two additional changes would help but require more development effort: an ensemble over multiple model proposals at generation time, and a controlled study of the agentic loop's stopping criterion to determine whether a minimum-iteration floor improves recall without prohibitive latency cost.

6. Conclusion

We submitted two open-weights agentic RAG systems to BioASQ Task 14b, sharing an architecture and varying the LLM backbone, the PubMed access substrate, and the answer-generation prompt. The shared architecture reached upper-middle-quartile Phase B yes/no performance on batch 1 (0.941 accuracy), middle-of-field results on Phase B list F-measure and factoid strict, and lower-middle Phase A^+ results bounded by retrieval recall. A learned cross-encoder reranker on BioASQ training data is the single change most likely to close the Phase A^+ gap to the leading systems in next year's submission.

Acknowledgements

DISTRIBUTION STATEMENT A. Approved for public release. Distribution is unlimited. This material is based upon work supported by the Department of the Air Force under Air Force Contract No. FA8702-15-D-0001 or FA8702-25-D-B002. Any opinions, findings, conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Department of the Air Force.

any copyright notice, U.S. Government rights in this work are defined by DFARS 252.227-7013 or DFARS 252.227-7014 as detailed above. Use of this work other than as specifically authorized by the U.S. Government may violate any copyrights that exist in this work.

Declaration on Generative AI

During the preparation of this work, the author(s) used GPT and Claude in order to: Grammar and spelling check. Further, the author(s) used GPT for figures in order to: Generate images. After using these tool(s)/service(s), the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content. Claude was also used to help generate tables and deal with overleaf issues.

References

[1] A. Nentidis, G. Katsimpras, A. Krithara, G. Paliouras, BioASQ at CLEF 2026: The fourteenth edition of the large-scale biomedical semantic indexing and question answering challenge, in: Advances in Information Retrieval (ECIR 2026), Lecture Notes in Computer Science, Springer Nature Switzerland, Cham, 2026. doi:10.1007/978-3-032-21321-1_42.

[2] R. A. A. Jonker, T. M. Almeida, J. R. Almeida, S. Matos, BIT.UA at BioASQ 13b: Revisiting evaluation, DPRF-enhanced retrieval and fine-tuned LLMs, in: CLEF 2025 Working Notes, CEUR Workshop Proceedings, CEUR-WS.org, 2025. URL: https://ceur-ws.org/Vol-4038/paper_22.pdf.

[3] F. Borazio, A. Shcherbakov, D. Croce, R. Basili, Unitor at bioasq 2025: Modular biomedical qa with synthetic snippets and multiple task answer generation, in: Working Notes of CLEF 2025 – Conference and Labs of the Evaluation Forum, volume 4038 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2025, pp. 165–197.

[4] S. Ateia, U. Kruschwitz, Can language models critique themselves? investigating self-feedback for retrieval augmented generation at bioasq 2025, arXiv preprint arXiv:2508.05366 (2025). URL: https://arxiv.org/abs/2508.05366. arXiv:2508.05366.

[5] J. Angulo, V. Yeste, Aqams and aqams2: Multi agent systems for biomedical question answering, in: Working Notes of CLEF 2025 – Conference and Labs of the Evaluation Forum, volume 4038 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2025, pp. 120–135.

[6] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, Y. Cao, ReAct: Synergizing reasoning and acting in language models, in: Proceedings of the 11th International Conference on Learning Representations (ICLR), 2023. arXiv:2210.03629.

[7] Q. Jin, B. Dhingra, Z. Liu, W. Cohen, X. Lu, PubMedQA: A dataset for biomedical research question answering, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 2019, pp. 2567–2577. doi:10.18653/v1/D19-1259.

[8] M. Rodríguez-Ortega, E. Rodríguez-López, S. Lima-López, A. Mash, M. Krallinger, Overview of MultiClinSum-2 at CLEF 2026: Data, Evaluation, and Results for Multilingual Clinical Case Report Summarization Across 15 Languages, in: E. Sánchez Salido, A. Barrón-Cedeño, A. García Seco de Herrera, S. MacAvaney, J. M. Struß (Eds.), CLEF 2025 Working Notes, 2026.

[9] I. Rozhkov, N. Loukachevitch, E. Tutubalina, Overview of the BioASQ BioNNE-R Task on Nested Biomedical Relation Extraction at CLEF 2026, in: E. Sánchez Salido, A. Barrón-Cedeño, A. García Seco de Herrera, S. MacAvaney, J. M. Struß (Eds.), CLEF 2026 Working Notes, 2026.

[10] A. Nentidis, G. Katsimpras, A. Krithara, G. Paliouras, Overview of BioASQ Tasks 14b and Synergy14 in CLEF2026, in: E. Sánchez Salido, A. Barrón-Cedeño, A. García Seco de Herrera, S. MacAvaney, J. M. Struß (Eds.), CLEF 2025 Working Notes, 2026.

[11] T. Schick, J. Dwivedi-Yu, R. Dessì, R. Raileanu, M. Lomeli, L. Zettlemoyer, N. Cancedda, T. Scialom, Toolformer: Language models can teach themselves to use tools, in: Advances in Neural Information Processing Systems (NeurIPS 36), 2023. arXiv:2302.04761.

[12] Gemma Team, A. Kamath, J. Ferret, S. Pathak, N. Vieillard, R. Merhej, S. Perrin, T. Matejovicova,

A. Ramé, M. Rivière, et al., Gemma 3 technical report, arXiv preprint arXiv:2503.19786 (2025). doi:10.48550/arXiv.2503.19786. arXiv:2503.19786.
[13] D. Chou, C. Chang, Gemma 4: Byte for byte, the most capable open models, Google blog, 2026. URL: https://blog.google/innovation-and-ai/technology/developers-tools/gemma-4/, released under Apache 2.0; family includes E2B, E4B, 26B MoE, and 31B Dense variants. Accessed 18 May 2026.
[14] Y. Gu, R. Tinn, H. Cheng, M. Lucas, N. Usuyama, X. Liu, T. Naumann, J. Gao, H. Poon, Domain-specific language model pretraining for biomedical natural language processing, ACM Transactions on Computing for Healthcare (HEALTH) 3 (2021) 1–23.
[15] Y. Gu, R. Tinn, H. Cheng, M. Lucas, N. Usuyama, X. Liu, T. Naumann, J. Gao, H. Poon, Domain-specific language model pretraining for biomedical natural language processing, ACM Transactions on Computing for Healthcare 3 (2021) 1–23. doi:10.1145/3458754.
[16] V. Boteva, D. Gholipour, A. Sokolov, S. Riezler, A full-text learning to rank dataset for medical information retrieval, in: Advances in Information Retrieval (ECIR 2016), 2016, pp. 716–722. doi:10.1007/978-3-319-30671-1_58.
[17] G. V. Cormack, C. L. A. Clarke, S. Buettcher, 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 (SIGIR), 2009, pp. 758–759.
[18] M. Sharma, M. Tong, T. Korbak, et al., Towards understanding sycophancy in language models, in: International Conference on Learning Representations (ICLR), 2024. ArXiv:2310.13548.
[19] G. Balikas, I. Partalas, A. Kosmopoulos, S. Petridis, P. Malakasiotis, I. Pavlopoulos, I. Androutsopoulos, N. Baskiotis, E. Gaussier, T. Artières, P. Gallinari, Evaluation Framework Specifications, BioASQ Project Deliverable D4.1, BioASQ Project Consortium, 2013.
[20] BioASQ, Bioasq — large-scale biomedical semantic indexing and question answering, https://www.bioasq.org/, 2025. Accessed: 2026-05-27.
[21] F. Borazio, A. Shcherbakov, D. Croce, R. Basili, UniTor at BioASQ 2025: Modular biomedical QA with synthetic snippets and multiple task answer generation, in: CLEF 2025 Working Notes, CEUR Workshop Proceedings, CEUR-WS.org, 2025.

A. Reproducibility Details

A public code repository accompanying this submission was not available within the camera-ready window. This appendix documents the prompts, hyperparameters, and configuration in full so that the system can be re-implemented without access to the original source. If still needed, please reach out to the authors for access to the source code.

A.1. Decoding settings

All LLM calls are served by a local vLLM instance. Decoding parameters vary by call type (Table 5). Each call retries up to three times with exponential backoff (3, 6, 9 seconds) under a 180-second timeout.

Call type	max_tokens	temp.	top_p
Query generation (controller)	200	0.3	0.95
Sufficiency check (controller)	20	0.1	0.95
Relevance rating (LLM rerank)	256	0.1	0.95
Answer generation (generator)	1024	0.3	0.95

Table 5. Per-call-type decoding settings.

A.2. Model checkpoints

Component	Checkpoint
ossllm / asmalltrialsystem	`google/gemma-3-27b-it`
Finalcorrected	`google/gemma-4-31b-it` (Dense)
Dense embedding	`pritamdeka/S-PubMedBert-MS-MARCO`
Sparse (ossllm)	`BM25Okapi` (k_1=1.5, b=0.75)
Sparse (Finalcorrected)	SQLite FTS5 native BM25

Table 6. Model and retrieval-component checkpoints.

A.3. Retrieval hyperparameters

Parameter	Value
Dense index	FAISS `IndexFlatIP`, 768-dim, L2-norm.
Per-question index	rebuilt fresh per question
BM25 rebuild	on every new-chunk insertion
RRF constant k	60
RRF dense weight	0.6
RRF sparse weight	0.4
Loop cap N	5 iterations
Sufficiency check	enabled when index ≥ 20 passages
Neighbour expansion	≤ 10 `pubmed_pubmed` / top hit
Citation expansion	≤ 5 `pubmed_pubmed_citedin` / top hit
Expansion scope	top 2 passages per iteration

Table 7. Retrieval and agentic-loop hyperparameters.

A.4. Chunking

Every retrieved abstract contributes three chunk types to the per-question index: (i) the title as a single chunk; (ii) the full abstract as a single chunk (length ≥ 50 characters); and (iii) overlapping three-sentence sliding windows with one-sentence stride, where sentences are split on the regular expression `(?<=[.!?])\s+`. Windows shorter than 50 characters or duplicates of previously seen chunks are dropped. Chunks are deduplicated within each per-question index; there is no cross-question deduplication.

A.5. Top-k schedule

Stage	k
Per-search FAISS retrieval (pre-fusion)	$3k$, capped at index size
Per-search BM25 ranking	$3k$
Per-search output (post-RRF)	15
Agentic-loop sufficiency inspection	top 10
Pre-rerank candidate pool	20
Post-LLM-rerank candidate pool	re-sorted top 20
Generator context	top 5–10

Table 8. Top-k at each retrieval stage.

A.6. Prompts

The following prompts are reproduced verbatim. Placeholders in braces are replaced at run time with the question, the retrieved evidence, and the few-shot exemplars.

Query generation.

```
Generate 3 short PubMed queries (3-6 words, plain
keywords) for: {question}

[If previous queries exist:]
Previous didn't find enough. Use COMPLETELY
DIFFERENT terms:
  - {prev_query_1}
  - {prev_query_2}
  - {prev_query_3}

Write ONLY 3 queries numbered. Nothing else.

1.
```

Sufficiency check.

```
Can '{question}' be answered from these?

[1] {passage_1_truncated_to_150_chars}
...
[8] {passage_8_truncated_to_150_chars}

SUFFICIENT or INSUFFICIENT?
```

The controller terminates the loop early if the response contains the substring SUFFICIENT (case-insensitive).

Yes/no generation.

```
You are an expert biomedical QA system.

INSTRUCTIONS:
1. Find evidence supporting YES.
2. Find evidence supporting NO.
3. Choose the side with STRONGER direct evidence.
4. If evidence shows PROBLEMS (toxicity, failure,
   side effects, contradictory results, lack of
   clinical evidence), answer NO.
5. If evidence is mixed, unclear, or insufficient,
   answer NO.
6. 'Promising preclinical results' or 'under
   investigation' does NOT mean YES.
7. A drug being 'studied' or 'tested' does NOT
   mean it works.

[two few-shot examples: Q / EVIDENCE /
 EXACT_ANSWER / IDEAL_ANSWER]

---
Q: {question}
EVIDENCE:
{retrieved_snippets}

EVIDENCE FOR YES:
EVIDENCE FOR NO:
EXACT_ANSWER:
```

Factoid generation.

```
You are an expert biomedical QA system.

STRICT RULES:
1. EXACT_ANSWER must be 1-5 words. A specific
   name, number, or term.
2. Copy the EXACT terminology from the evidence.
3. Do NOT paraphrase or elaborate in the answer.
4. If the evidence does not contain a clear
   specific answer, write: unknown
5. Prefer named entities: drug names, protein
   names, gene symbols, disease names, numbers.
6. Then write IDEAL_ANSWER: a 2-4 sentence
   explanation (max 200 words).

GOOD: 'transsphenoidal surgery', 'NF1', '45,X',
      'palivizumab', 'mesenchymal'
BAD:  'multiple causative factors',
      'a type of bleeding'

[two few-shot examples]

---
Q: {question}
EVIDENCE:
{retrieved_snippets}
EXACT_ANSWER:
```

List generation.

```
You are an expert biomedical QA system.

STRICT RULES:
1. List EVERY relevant item in the evidence.
   Be EXHAUSTIVE.
2. Better to include too many than too few.
3. Go through EACH passage systematically.
4. Each item is a specific name or term (1-5
   words).
5. Aim for 5-15 items. Many questions have
   10-20+ answers.
6. Prefix each item with '- ' on its own line.
7. Do NOT group or combine items. One per line.
8. After the list, write IDEAL_ANSWER: 2-4
   sentences (max 200 words).

[one few-shot example with list-formatted answer]

---
Q: {question}
EVIDENCE:
{retrieved_snippets}

Now list EVERY relevant item from ALL passages:

EXACT_ANSWER:
```

Summary generation.

```
You are an expert biomedical QA system. Write a
comprehensive 3-6 sentence answer (max 200 words)
using the evidence.

[two few-shot examples]

---
Q: {question}
EVIDENCE:
{retrieved_snippets}
IDEAL_ANSWER:
```

LLM relevance reranking.

```
Rate each passage's relevance (0-2) to
answering: {question}

[1] {passage_1_truncated_to_200_chars}
...
[20] {passage_20_truncated_to_200_chars}

Return [N] SCORE per line:
```

The relevance score (0, 1, or 2) is fused with the RRF score using weights 0.6 (RRF) and 0.4 (LLM relevance /2).

A.7. Hardware and serving

We run the LLM hosted via vLLM with tensor_parallel_size=2, gpu_memory_utilization=0.85, and bfloat16 precision; the context window is capped at 8192 tokens during generation. PubMedBERT embeddings are computed on CPU (batch size 32, FP32) due to GPU memory contention with the served 27B/31B model. The Finalcorrected system queries a ∼35 GB SQLite 3.45 mirror of the 2025 PubMed Annual Baseline (release 18 December 2024) with an FTS5 full-text index over titles, abstracts, MeSH headings, and publication metadata; `Editorial`, `Letter`, `News`, `Comment`, `Review`, and `Retraction` article types are excluded at mirror-build time. While Live E-utilities queries for the ossllm system were issued during the Task 14b Phase A+ submission windows on 18th March 2026 to April 29th 2026

A.8. Randomness

Controller calls run at `temperature` 0.1–0.3; the sufficiency check at 0.1 is effectively deterministic on the prompts we observed. Generator calls run at `temperature` 0.3; we did not fix a serving-level random seed at submission time, which we estimate contributes ∼1–2 points of aggregate-metric variance from informal repeated runs. Few-shot exemplar selection is deterministic given a fixed development set (cosine similarity to the test question, ties broken by training-set order).