Retrieval-Bound Generation: Lean RAG pipelines for Biomedical QA

Notebook for the BioASQ Lab at CLEF 2026.

Diogo M. C. Antunes[1,*], Paulo R. C. Lopes[1] and Francisco M. Couto[1]

[1]LASIGE, Departamento de Informática, Faculdade de Ciências, Universidade de Lisboa, 1749-016 Lisboa, Portugal

Abstract

Biomedical question answering requires retrieving relevant evidence from large literature collections and generating accurate answers grounded in the retrieved evidence, a task made challenging by the scale of PubMed and the characteristics of biomedical terminology. In this work we describe our participation in BioASQ Task 14b, covering all three phases of the challenge (A, A+, and B), with a focus on lean, locally deployable Retrieval-Augmented Generation. We developed four retrieval pipeline variants that share a common query expansion, snippet extraction, and cross-encoder re-ranking stage, differing in their retrieval strategy: sparse-only, sparse-dense hybrid, dense pseudo-relevance feedback, and a cross-encoder ensemble. For answer generation, we used lean and instruction-tuned large language models under a few-shot prompting strategy with a few-pass inference scheme, in which an exact answer is generated first and then used to condition a more comprehensive ideal natural language answer. These systems were designed to run on a single NVIDIA A30 GPU, prioritizing resource efficiency and local deployability throughout the retrieval and generation stages. The code and data used is publicly available in GitHub and in Hugging Face, respectively.

Keywords

Question Answering, Retrieval-Augmented Generation, Lean, Locally deployable, BioASQ Task 14b

1. Introduction

The continuous growth of the biomedical literature, with PubMed indexing over a million new citations every year, makes manual evidence retrieval increasingly impractical for researchers and clinicians. Large Language Models (LLMs) have shown strong performance on open-domain Question Answering (QA), but their direct application in the biomedical field is limited by static knowledge, hallucination risks, and the impracticality of fine-tuning a dedicated model for every specialized area. Retrieval-Augmented Generation (RAG) mitigates these issues by grounding a generative model on documents fetched from an external knowledge base at inference time, producing answers that are both up-to-date and traceable to their sources [1, 2]. The BioASQ Challenge [3] has established itself as the reference benchmark for evaluating such systems on real biomedical information needs, providing expert-curated questions paired with relevant PubMed documents, snippets and gold-standard answers.

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

*Corresponding author.

diogomantunes.da@gmail.com (D. M. C. Antunes); lopescrpaulo@gmail.com (P. R. C. Lopes); fjcouto@ciencias.ulisboa.pt (F. M. Couto)

0000-0003-0627-1496 (F. M. Couto)

In these notes, we describe our participation in BioASQ Task 14b [4, 5], covering Phase A (information retrieval), Phase B (answer generation) and Phase A+ (retrieval and generation). Our submissions are built around a deliberately lean, modular RAG pipeline designed to remain viable on a single workstation rather than on distributed GPU clusters. The retrieval stage combines sparse lexical search with dense retrieval, fused through Rank Fusion algorithms. A cross-encoder is then used to re-rank the top candidates and text snippets, that together with the question, are passed to an LLM for an answer. The remainder of this paper details each component, the methodology, and the results obtained on BioASQ test batches.

2. Related Work

2.1. RAG in Biomedicine

Retrieval-Augmented Generation, introduced by Lewis et al.[1], combines a parametric language model with a non-parametric retrieval component, conditioning generation on documents fetched at inference time. This grounding mitigates the hallucinations characteristic of standalone LLMs and has rapidly become the dominant architecture for knowledge-intensive QA, where factual Precision and terminology specificity are critical. Recent surveys highlight that domain-adapted retrieval components, biomedical embedding models, specialized re-rankers, and highly curated corpora, are essential for reliable performance on benchmarks such as BioASQ [6, 7]. A complementary finding from Cuconasu et al.[8] shows that the type of irrelevant context matters as much as its presence, semantically related but non-answer-bearing documents are substantially more harmful to generation quality than fully unrelated documents, motivating retrieval pipelines that prioritize Precision over volume in the candidates passed to the final LLM.

2.2. BioASQ challenge

The BioASQ Task b challenge [5] is structured around three evaluation phases, each isolating a different component of the QA pipeline. Phase A focuses on retrieval, given a biomedical question, participants return the most relevant PubMed documents and extract answer-bearing snippets, evaluated with standard information retrieval metrics [3]. Phase A+ requires generating a complete answer end-to-end, using only the documents and snippets produced by the participant's own Phase A pipeline. This evaluates the full system under realistic conditions, where retrieval errors propagate to generation. In contrast, Phase B provides gold-standard snippets curated by the organizers, isolating answer generation quality from retrieval quality.

A QA training file is provided by the BioASQ team, containing 5729 questions, each complemented by the gold-standard documents containing the answers and text snippets from those documents that were selected by experts [9]. Through an exploratory data analysis of the provided training file, it's noticeable that all gold-standard text snippets are contained in the abstract or the title of the gold-standard documents (with some exceptions pointing to a *"section.0"* that is not specified in the documentation) and that exist 4 types of questions that require different answers from the systems:

- **Yes/No questions**: 1541 questions that require either a "yes" or "no" as an answer;
- **List questions**: 1130 questions that require a list of entities (e.g. a list of genes) as an answer;
- **Factoid questions**: 1695 questions that require a particular entity (e.g. a disease, drug, or gene) as an answer;
- **Summary questions**: 1363 questions that can only be answered by producing a short text summarizing relevant information.

Although Yes/No, List and Factoid questions require an *exact answer* specific to each type, they all share a required *ideal answer* which consists in a short paragraph written in natural language that also answers the question.

2.3. Sparse-Dense Hybrid Retrieval and Rank Fusion

The first stage of any RAG pipeline must navigate a fundamental tension between two complementary methods of information retrieval. Lexical matching methods (specifically BM25) excel at handling the precise terminology of biomedical literature, where gene names, protein identifiers, and drug compounds are typically expressed verbatim across both questions and answer-bearing passages [10]. On the other side, dense retrievers capture semantic relationships beyond surface-form overlap, recovering documents that discuss the same concept with different phrasing. To achieve the highest Recall on retrieval neither signal is sufficient alone, for example, a question about *"long non-coding RNA splicing"* may have as a match abstracts that never use that exact phrase yet describe the underlying process, while a question targeting a specific identifier such as *"HIF-1α"* is most reliably answered by documents that mention the term rather than paraphrasing its function.

Hybrid retrieval addresses this tension by combining both signals before the intermediate step of re-ranking. Across recent BioASQ editions, sparse–dense hybrid pipelines have consistently outperformed single-engine baselines, with BM25 providing Precision on exact entity matches and dense biomedical encoders, contributing semantic generalization [11, 12]. The combination is particularly suited to the heterogeneous question types in BioASQ Task b mentioned above (Section 2.2).

Combining heterogeneous ranks into a single fused list requires a fusion strategy, and two families of methods dominate the literature. Rank-based methods such as Reciprocal Rank Fusion (RRF), which discard the raw scores entirely and operate only on document positions in each input list [13]. RRF is commonly used because it requires no score normalization and is robust to the very different score magnitudes produced by sparse and dense retrievers. Score-based methods, in contrast, fuse the raw similarity scores after normalization (typically *min-max norm*). The most direct instance, Weighted Sum (WSum), assigns a tunable weight to each retriever and combines normalized scores linearly. WSum is more expressive than RRF when one retriever is known to dominate, as the optimal weights are learned directly from a training set against a chosen retrieval metric.

In this work both fusion strategies are employed in different pipelines. WSum is used in the Hybrid pipeline to maximize Recall at the candidate pool level prior to re-ranking, while RRF is

used as the late-stage combination strategy in the DPRF and Ensemble pipelines, where the inputs to be fused already come from Recall-oriented stages.

2.4. Cross-Encoder Re-ranking and Ensemble

While bi-encoder retrievers are computationally efficient, their architecture imposes a fundamental limitation making queries and documents be encoded independently into fixed vector representations. Cross-encoders address this limitation by jointly encoding the (query, document) pair through a single pass, allowing every token in the question to attend to every token in the candidate and vice versa [14]. The cost is substantially higher inference latency, which is why cross-encoders are typically deployed as a second-stage re-ranker over a small candidate pool produced by a faster first-stage retriever rather than applied to the full corpus.

This two-stage retrieval methodology, one stage Recall-oriented and the other Precision-oriented cross-encoder re-ranking, has become the standard Precision-oriented design across recent BioASQ submissions and more broadly in the BEIR benchmark [12, 15, 16]. A practical consequence is that the choice of cross-encoder model has a disproportionate impact on final retrieval quality, motivating careful selection based on training domain, parameter count, and context window. Models trained on MS MARCO provide strong general-purpose baselines, while domain-adapted variants based on ModernBERT can offer better alignment with biomedical terminology at comparable parameter scales [17, 15]. A natural extension of the single re-ranker design is to run multiple cross-encoders in parallel over the same candidate pool and combine their rankings. The intuition is that re-rankers trained on different domains (or even just different datasets) and with different architectures make complementary reasoning, this meaning, documents ranked highly by all models are more reliably relevant than documents favored by any single model in isolation.

2.5. Dense Pseudo Relevance Feedback (DPRF)

Pseudo-Relevance Feedback is a classical query expansion technique with origins predating modern neural retrieval [18, 19]. The core assumption is that the top-ranked documents from an initial retrieval pass, while not yet verified as relevant, contain useful signal about what relevant documents look like for the given query.

Dense Pseudo-Relevance Feedback (DPRF) adapts this idea to the embedding space. Rather than extracting terms from the seed documents, the seed documents themselves are encoded with a dense retriever, and their nearest neighbors in the vector space are added to the candidate pool. This sidesteps the vocabulary mismatch that limits term-based expansion, for example, a seed document containing the phrase *"hepatic glucose production"* can recover neighbors discussing *"liver glycogenolysis"* even when neither phrase appears in the original query. The technique is particularly valuable in biomedical retrieval, where the gap between question vocabulary and abstract vocabulary can be structurally wide.

2.6. LLM-Based Answer Generation in Biomedical QA

The answer generation component of biomedical RAG pipelines has converged on a main design: a general-purpose or biomedically-adapted instruction-tuned LLM is conditioned on retrieved

snippets through a prompt, and generates both exact and ideal answers in a inference call, with no parameter updates [20, 15, 12].

A consistent finding across past submissions is that prompt design matters more than model scale for BioASQ-style outputs. Few-shot prompting with carefully selected in-context examples can significantly improve format adherence, particularly for the structured exact answers of factoid and list questions, where the model must produce parsable output [21, 15]. Multi-pass strategies, in which the exact answer is generated first and then provided as additional context for ideal-answer generation, have also been adopted to enhance coherence between the two outputs.

Compute constraints have shaped model selection in parallel with prompting strategy. Lean, locally deployable LLMs have proven viable for BioASQ Phase B, as demonstrated by Lopes et al.[20] with a 4-bit quantized 7B parameter model. This direction is particularly relevant for biomedical institutions where data privacy and financial constraints can rule out commercial cloud APIs, motivating system designs that prioritize local inference over peak benchmark performance achievable only on expensive distributed GPU clusters.

3. Methodology

All components of this work, including corpus indexing, cross-encoder fine-tuning, retrieval, and answer generation, were carried out on a single NVIDIA A30 GPU. This constrained hardware setting directly motivated the lean design philosophy adopted throughout the system.

3.1. Corpus and Indexes

The knowledge base is the 2026 PubMed Annual Baseline, comprising approximately 35 million documents released on January 30, 2026. For the dense retrieval process we use a FAISS index with embeddings produced by NeuML/pubmedbert-base-embeddings-matryoshka[1]. The index uses an IVF-SQ8 configuration with 2^{16} clusters trained on a random sample of vectors, with inner product as the similarity metric. The *nprobe* parameter, which controls how many clusters are visited per query, was tuned on the BioASQ training and set to $nprobe = 512$ offering the best trade-off between Recall and per-query latency.

Sparse retrieval is performed using the PISA engine via PyTerrier[2] with BM25 scoring. The index was constructed over the PubMed Annual Baseline mentioned above. The preprocessing tools tested were Porter2 stemming and Terrier stop-word removal, and also the BM25 hyperparameters were tuned via grid search on the BioASQ training set, targeting Recall as the relevant metric. The grid search covered values of $k1$ and b commonly seen in the literature, which were also tested across four preprocessing configurations: with stemming and stop-word removal, no treatment and stemming or stop-word removal configurations alone. The optimal values and configurations used in this work were $k_1 = 0.6$ and $b = 0.4$, utilizing an index configured with Porter2 stemming and Terrier stop-word removal.

[1] https://huggingface.co/NeuML/pubmedbert-base-embeddings-matryoshka
[2] https://pyterrier.readthedocs.io/en/latest/

3.2. Phase A: Retrieval Pipelines

3.2.1. Query Expansion

All BioASQ questions are natural language interrogatives, which contain auxiliary and interrogative words (”are”, ”what”, ”does”, ”which”) that carry no discriminative retrieval signal and dilute the term-frequency weight of the relevant biomedical entities under the BM25 scoring function. To address this, prior to BM25 retrieval a simple type-specific query expansion strategy is applied. A regular expression strips auxiliary and interrogative words from the question body, isolating the key entity terms, which are then appended to the original query with a repetition frequency that varies by question type: factoid questions receive the key terms appended twice, yes/no and list questions receive the key terms appended once and summary questions receive no expansion. The intuition behind this strategy is to try compensate for the structural mismatch between interrogative natural language and the declarative style of biomedical abstracts.

3.2.2. Pipeline Variants

We developed four Phase A pipeline variants, sharing the corpus management, query expansion, snippet extraction, and cross-encoder re-ranking components. The variants differ only in how candidate documents are retrieved and fused prior to the final re-ranking stage.

The sparse-only pipeline (system name **lean_rag_ft_sparse**) uses BM25 as the sole retrieval component, forwarding the top-1000 BM25 candidates directly to the cross-encoder without any dense retrieval or fusion step. This variant serves as a controlled baseline that isolates the contribution of dense retrieval and fusion to overall pipeline performance.

The hybrid pipeline seen in Figure 1, combines BM25 and dense retrieval through WSum fusion implemented via the *ranx*[3] library. Each retriever independently produces a ranked list of top-1000 candidate documents, which are subsequently fused using a weighted combination with weights of 0.83 for the sparse run and 0.17 for the dense run. The weights were obtained by grid search over all combinations with step size 0.01 under min-max score normalization, targeting Recall on the BioASQ training set. The dominance of the sparse weight is consistent with the broader observation that BM25 remains a strong baseline on biomedical retrieval benchmarks, where dense models often struggle with domain shift and highly specialized terminology [16]. The top-500 fused candidates are then passed to the cross-encoder for re-ranking. We submitted two variants of this pipeline: one using the base (non-fine-tuned) cross-encoder (system name **lean_rag**), and one using the cross-encoder fine-tuned as described in Section 3.2.5 (system name **lean_rag_ft**). This pairing allows a controlled comparison of the effect of domain-specific fine-tuning on the re-ranking stage while keeping all other pipeline components identical.

[3] https://github.com/AmenRa/ranx

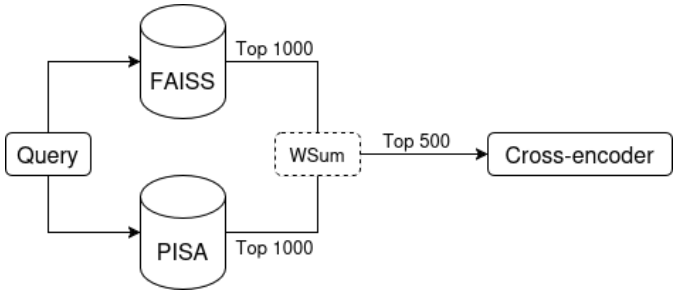

Figure 1: Architecture of the hybrid retrieval pipeline combining dense (FAISS) and sparse (PISA) retrievers, followed by WSum fusion and cross-encoder re-ranking. The sparse-only pipeline is similar with only the sparse component (PISA) providing candidates to the cross-encoder.

The DPRF pipeline (system name **lean_rag_dprf**) extends the hybrid approach with a feedback expansion step, Figure 2. After an initial sparse retrieval, a preliminary cross-encoder pass identifies the top pseudo-relevant seed documents. Each seed is then encoded with the same embedding model used by the dense index, and its nearest neighbors in the vector space are recovered via FAISS, capturing semantically related documents that keyword-based retrieval misses. The expansion results are integrated with the BM25 and FAISS retrievals using a 3-way RRF.

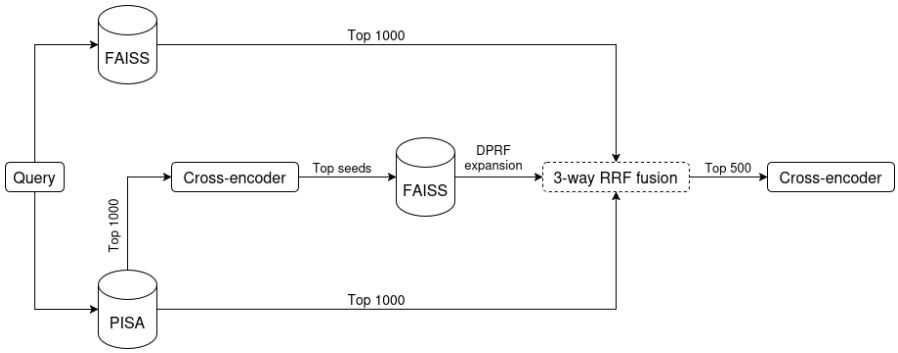

Figure 2: Architecture of the DPRF pipeline utilizing sparse-seeded cross-encoder expansion and a final 3-way RRF fusion before passing the candidate pool through the cross-encoder a last time.

Inspired by the consistent observation across recent BioASQ overviews and top-performing submissions, we developed a cross-encoder ensemble pipeline (system name **lean_rag_ensemble**) [3, 12, 15]. Rather than selecting a single cross-encoder re-ranker, this variant runs three cross-encoders in parallel over the same candidate pool: BAAI/bge-reranker-v2-m3, Alibaba-NLP/gte-reranker-modernbert-base finetuned on BioASQ and BEIR data (Section 3.2.5), and cross-encoder/ms-marco-MiniLM-L12-v2. Each model independently scores and ranks the candidates, and the three ranked lists are combined via RRF. A per-model confidence threshold is also applied before fusion, requiring a document to exceed the score threshold of each enabled model to be included in the final submission, reducing false positives by requiring consensus across models of different architectures, parameter scales, and training domains.

3.2.3. Cross-encoders utilized

All four pipeline variants employ at least one cross-encoder re-ranker as the final document scoring stage. Three cross-encoder models were evaluated in this work: BAAI/bge-reranker-v2-m3[4] (568M parameters, multilingual XLM-RoBERTa base), Alibaba-NLP/gte-reranker-modernbert-base[5] (149M parameters, ModernBERT base), and cross-encoder/ms-marco-MiniLM-L12-v2[6] (33M parameters, MiniLM base). The three models span a broad range of parameter scales and training domains, deliberately covering the accuracy-latency trade-off spectrum.

Following re-ranking, a confidence threshold cutoff is applied to determine the final set of documents submitted for each question, retaining only documents whose cross-encoder score exceeds a model-specific threshold tuned on the BioASQ training set. This mechanism is motivated by the BioASQ evaluation protocol, which penalizes submissions containing irrelevant documents, for example returning ten documents of which only three are relevant is scored substantially worse than returning the three relevant documents alone. It is therefore preferable to submit fewer high-confidence documents than to consistently return the maximum of ten. In the Ensemble pipeline, the same logic is applied at a per-model level, where a document must exceed the confidence threshold of each of the three cross-encoders before being included in the final RRF-fused submission. We note that, due to an implementation issue, this threshold cutoff was active only on batch 4 submissions, earlier batches returned a fixed set of up to ten documents per question. The effect of this correction is visible in the Phase A results (Section 4).

3.2.4. Snippet Extraction

Snippet extraction is performed independently of document ranking and is applied to the top documents returned by each pipeline. For each candidate document, the title and abstract are segmented into sentences using the ScispaCy[7] biomedical sentence segmenter (the en_ner_bionlp13cg_md model), which handles the abbreviations, numerical expressions, and notations that cause general-purpose segmenters to produce incorrect sentence boundaries on biomedical text [22]. Candidate snippets are then constructed using a sliding-window approach that generates all 1-, 2-, and 3-sentence windows from each document. The multi-sentence windowing mirrors the typical length of BioASQ gold-standard snippets in the training set, where they routinely span two or three contiguous sentences.

To control the size of the candidate pool fed to the cross-encoder, the windows are first pre-filtered using cosine similarity between the embeddings of the query and each candidate window, retaining the top candidates per document. The full set of pre-filtered candidates across all documents is then scored by the cross-encoder, which evaluates each (query, snippet) pair jointly and returns the highest-scoring snippets for submission.

[4] https://huggingface.co/BAAI/bge-reranker-v2-m3
[5] https://huggingface.co/Alibaba-NLP/gte-reranker-modernbert-base
[6] https://huggingface.co/cross-encoder/ms-marco-MiniLM-L12-v2
[7] https://allenai.github.io/scispacy/

3.2.5. Cross-encoder finetuning

To better adapt the cross-encoder re-ranker to the biomedical retrieval domain, we fine-tuned the Alibaba-NLP/gte-reranker-modernbert-base model on a combination of the BioASQ training set and two biomedical splits of the BEIR benchmark, TREC-COVID and SciFact [16, 23, 24]. The two BEIR splits were selected for their (query, relevant abstract) structure and also their use of PubMed abstracts, which mirrors BioASQ Task b, while introducing complementary phrasing styles and domain vocabulary that complement the BioASQ training set.

Reaching a stable fine-tuning configuration was not immediate, and several design choices were the result of iteration. Our initial setup used a pointwise objective, where each (query, document) pair was independently classified as relevant or non-relevant with Binary Cross-Entropy (BCE) loss. While conceptually simple, this formulation converged to weaker rankings on the validation set and was less stable than the alternatives explored. Eventually, this converged on a pairwise objective using Margin Ranking Loss with margin $m = 0.3$, which directly optimizes the relative ordering that the re-ranker must produce at inference time, this meaning, for each query, a positive document and a negative are pushed apart in score space by at least the m margin.

The number of hard negatives per positive was also tuned iteratively. We initially sampled 15 negatives per positive, on the intuition that more negatives would provide a richer contrastive signal. In practice, this produced severe training instability, with gradient norms fluctuating between consecutive steps and the model failing to converge to a useful ranking. Reducing the ratio to 5 negatives per positive stabilized training. Hard negatives themselves were drawn from candidates by a FAISS engine pass, which produced documents that are topically related but not answer-bearing, the precise type of error that the cross-encoder needs to learn to suppress. Even with a stable training objective, we observed consistent overfitting in our first runs, with validation NDCG (Normalized Discounted Cumulative Gain) peaking early and degrading thereafter. This motivated the adoption of Layer-wise Learning Rate Decay (LLRD), which soften the learning rate for lower layers while maintaining the classification head and the final encoder layers with a more aggressive value. This protects the pre-trained representations of the lower transformer layers, which already encode useful general language structure. The final training configuration uses AdamW optimizer for 3 epochs, weight decay of 0.01, a linear learning rate schedule with 10% warmup and gradient clipping at $\ell 2$ norm 1.0. NDCG was selected as the validation metric for the evaluation split, over alternatives such as MAP or Precision due to its sensitivity to the position of relevant documents within the ranking rather than only their presence.

3.3. Phase B: Answer Generation

3.3.1. Model and Inference Setup

Two instruction-tuned LLMs were used across submissions: Intelligent-Internet/II-Medical-8B[8], an 8B-parameter biomedical model obtained through supervised fine-tuning of Qwen3-8B on a curated corpus of medical literature and clinical question-answering datasets, and

[8]https://huggingface.co/Intelligent-Internet/II-Medical-8B

google/gemma-4-E4B-it[9], a recent instruction-tuned model from the Gemma 4 family with a Mixture-of-Experts architecture that activates approximately 4 billion parameters per Forward Pass. II-Medical-8B was used in the earlier submission batches, while Gemma-4-E4B-it was incorporated in the final batch following its public release in late April 2026. Both models were loaded in *bfloat16* Precision. The 8B-parameter ceiling on model size was a deliberate design choice consistent with the lean architecture philosophy of this work and the infrastructure available.

3.3.2. Prompting and Inference

To condition the model on the biomedical domain and on the specific output formats required by the BioASQ challenge, a few-shot prompting strategy is employed with $n = 5$, $n = 0$ and $n = 10$ examples per question type, drawn from the BioASQ training set. For each inference call, the prompt is assembled as a system message followed by n (user, assistant) example pairs and a final user turn containing the actual question and its retrieved context. The system prompt seen in Figure 3 was created to be simple and clear, instructing the model to follow rules passed in the user prompt. The four query type specific formatting instructions, one for each BioASQ question type, are reproduced in full in Appendix 6.

[System]You are an expert biomedical question-answering assistant. Answer questions using the provided passages and your medical knowledge. Always return the answer STRICTLY in the requested format.
[Few-shot examples, repeated *n* times]
[User] Passage: {snippets}
Question: {question body}
[Query type specific formatting instruction]

Figure 3: Overall prompt box for both LLMs used in this work. The last instruction is specific to each query type.

For questions requiring both an exact and an ideal answer (yes/no, factoid, and list types), the pipeline performs two separate inference passes per question. The first pass generates the exact answer using a type-specific formatting instruction. The second pass generates the ideal answer, a free-text summary of up to 200 words, with the exact answer from the first pass embedded as a hint in the user prompt. This conditioning mechanism allows the ideal answer to be generated with prior knowledge of the short answer, improving coherence between the two outputs. Summary questions, which have no exact-answer component, undergo only a single pass targeting the ideal answer directly.

[9]https://huggingface.co/google/gemma-4-E4B-it

3.4. Evaluation metrics

Phase A submissions are evaluated separately for documents and snippets. Documents are scored using Precision, Recall, F-measure, Mean Average Precision (MAP), and Geometric Mean Average Precision (GMAP). Snippets are scored with the same set of measures, with Precision and Recall redefined at the character level via article–offset pairs to account for partial overlap between returned and gold snippets that are not identical. The main ranking metric for documents is MAP and for snippets is F-measure [25].
Phase A+ and Phase B share the same evaluation protocol, differing only in the source of the documents and snippets provided to the generation module. Both phases evaluate exact and ideal answers separately, and the metrics used are type-specific. Yes/no exact answers are scored with macro-averaged F-measure across the two classes (”yes” and ”no”), factoid answers are scored with Mean Reciprocal Rank (MRR) and list answers are scored with F-measure. The ideal answers, are evaluated automatically with ROUGE-2 and ROUGE-SU4 against the gold-standard answers, complemented by manual scoring from the biomedical experts. However, according to the BioASQ team, the official and final scores for Phase B and Phase A+ are based solely on manual evaluation, while the automatic metrics based on ROUGE are presented as preliminary results that may not perfectly align with human judgment [25].

4. Results

In this section we report official BioASQ preliminary scores across all three phases. Due to a temporary infrastructure outage at our institution, we were unable to submit results for batch 1 of Phases A+ and B, results for those phases therefore begin at batch 2. In Phase A, the five systems were introduced incrementally across batches as the development of this work progressed.

4.1. Phase A: Document and Snippet Retrieval

Document and snippet retrieval results are reported in Tables 1 and 2, respectively. In batch 1, the hybrid pipelines (**lean_rag** and **lean_rag_ft**) perform around the submission median for document MAP (0.169 vs. a median of 0.170), whereas in batch 2 they clearly exceed it (0.179 vs. 0.147), with **lean_rag_ft** ranking 20th of all submissions. One initially surprising finding is that fine-tuning the cross-encoder re-ranker produces negligible gains, in batch 2, the fine-tuned variant (**lean_rag_ft**, MAP 0.1791) is within 0.001 of the non-fine-tuned variant (**lean_rag**, MAP 0.1781). This suggests that the main retrieval bottleneck lies in the first-stage candidate pool rather than in the Precision of the re-ranking stage.
The sparse-only pipeline (**lean_rag_ft_sparse**) performs comparably to the hybrid pipelines across most batches, with MAP differences rarely exceeding 0.01. This indicates that, in our configuration, the dense FAISS channel contributes limited additional Recall over BM25 alone, a finding consistent with the relatively low standalone Recall of our dense index. The DPRF variant (**lean_rag_dprf**) is consistently the weakest of our systems despite its more complex retrieval path, with MAP of 0.1649 in batch 2 and 0.1113 in batch 3, in both cases the lowest among our submissions for those batches. This suggests that the additional expansion step

propagates noise from the seed selection stage into the candidate pool, and that a more precise seed selection strategy would be needed for the expansion to contribute positively.

As described in Section 3.2.3, the confidence threshold cutoff was active only from batch 4 onward in our official submissions. In that batch, the ensemble variant (**lean_rag_ensemble**) achieved a mean precision of 0.1224, the highest value among all participants, while maintaining a competitive MAP of 0.1852. Since the test set size and the number of gold-standard documents were constant between batches 3 and 4, this gain is directly attributable to the cutoff, which discards low-confidence documents and submits a compact, high-precision set instead of the maximum number of candidates allowed. The contrast with the rest of the field makes the effect concrete, the top-ranked system by MAP reached only 0.0967 precision, well below ours, and owed its higher MAP (0.2310 vs. 0.1852) to substantially greater recall (0.3698 vs. 0.2741). MAP is largely recall-driven, so it rewards a different objective than the one our cutoff pursues. The two metrics measure different aspects of retrieval quality, and our design deliberately optimizes precision.

This behavior is not confined to batch 4. Across all batches our systems sit in the upper range of submissions on mean precision while trailing on recall, a direct consequence of favoring high-confidence documents over recall maximization at the re-ranking stage. Batch 4 simply makes the trade-off explicit, because it is the only batch in which the cutoff was active.

Snippet retrieval follows a similar pattern to document retrieval, with MAP values ranging from 0.02 to 0.06 across batches. The gap between snippet and document performance is structural, snippet extraction operates over the candidate pool produced by document retrieval, so any weakness in document Recall propagates directly into snippet quality. Among the pipeline variants, differences in snippet MAP within the same batch are negligible, reinforcing the conclusion that the first-stage retrieval, not the re-ranking or snippet extraction stage, is the primary bottleneck for snippet retrieval too.

Table 1
Document retrieval results for Phase A. Ranks are assigned by BioASQ according to MAP, with rank 1 being the highest-MAP system. *Best Competitor* cells display the metrics of the top-ranked system in each batch (rank 1 by MAP), submitted by another participating team and *Median* cells are the median of the corresponding metric across all systems submitted to that batch by all participating teams, and therefore do not correspond to a single system or team. Bold marks the best MAP among our systems in each batch. The confidence threshold cutoff was active only in batch 4.

System	Mean Prec.	Recall	MAP	Rank
Batch 1				
lean_rag	0.0800	0.2483	0.1691	27
lean_rag_ft	0.0825	0.2620	0.1650	28
lean_rag_ft_sparse	0.0805	0.2547	0.1630	29
Best Competitor	0.1188	0.4090	0.2835	1
Median	0.0892	0.2769	0.1696	–
Batch 2				
lean_rag	0.0650	0.2790	**0.1781**	21
lean_rag_ft	0.0650	0.2790	**0.1791**	20
lean_rag_ft_sparse	0.0613	0.2657	0.1624	27
lean_rag_dprf	0.0600	0.2600	0.1649	25
Best Competitor	0.0800	0.3214	0.2395	1
Median	0.0971	0.2048	0.1474	–
Batch 3				
lean_rag	0.0367	0.1742	0.1201	37
lean_rag_ft	0.0367	0.1742	0.1192	38
lean_rag_ft_sparse	0.0383	0.1825	**0.1215**	35
lean_rag_ensemble	0.0400	0.1960	0.1209	36
lean_rag_dprf	0.0350	0.1687	0.1113	43
Best Competitor	0.0783	0.3206	0.2114	1
Median	0.0633	0.2845	0.1289	–
Batch 4				
lean_rag	0.1030	0.2772	0.1759	28
lean_rag_ft	0.0977	0.2745	0.1726	31
lean_rag_ft_sparse	0.0967	0.2731	0.1700	33
lean_rag_ensemble	**0.1224**	0.2741	**0.1852**	20
lean_rag_dprf	0.0964	0.2804	0.1754	29
Best Competitor	0.0967	0.3698	0.2310	1
Median	0.0717	0.2492	0.1663	–

Table 2
Snippet retrieval results for Phase A. Ranks are assigned by BioASQ according to F-measure, with rank 1 being the highest F-measure system. *Best Competitor* cells display the metrics of the top-ranked system in each batch (rank 1 by F-measure), submitted by another participating team and *Median* cells are the median of the corresponding metric across all systems submitted to that batch by all participating teams, and therefore do not correspond to a single system or team. Bold marks the best F-measure among our systems in each batch.

System	Mean Prec.	Recall	F-Measure	Rank
Batch 1				
lean_rag	0.0518	0.1369	**0.0606**	30
lean_rag_ft	0.0505	0.1537	0.0601	29
lean_rag_ft_sparse	0.0504	0.1537	0.0601	28
Best Competitor	0.0824	0.2188	0.0914	1
Median	0.0481	0.1211	0.0554	–
Batch 2				
lean_rag	0.0755	0.1469	**0.0809**	28
lean_rag_ft	0.0743	0.1466	0.0802	29
lean_rag_ft_sparse	0.0667	0.1361	0.0761	32
lean_rag_dprf	0.0688	0.1348	0.0762	30
Best Competitor	0.0674	0.1918	0.0846	1
Median	0.0463	0.1388	0.0578	–
Batch 3				
lean_rag	0.0209	0.0754	0.0301	39
lean_rag_ft	0.0217	0.0757	0.0305	40
lean_rag_ft_sparse	0.0220	0.0757	0.0309	38
lean_rag_ensemble	0.0203	0.0636	0.0275	42
lean_rag_dprf	0.0237	0.0772	**0.0327**	37
Best Competitor	0.0690	0.1524	0.0780	1
Median	0.0196	0.1409	0.0319	–
Batch 4				
lean_rag	0.0581	0.1424	0.0693	38
lean_rag_ft	0.0627	0.1442	**0.0726**	39
lean_rag_ft_sparse	0.0612	0.1444	0.0717	43
lean_rag_ensemble	0.0525	0.1571	0.0678	42
lean_rag_dprf	0.0594	0.1414	0.0699	40
Best Competitor	0.0774	0.2055	0.0992	1
Median	0.0364	0.0966	0.0442	–

4.2. Phase A+: End-to-End generation

Phase A+ results are shown in Table 3. In this phase, all systems use a 5-shot prompting strategy with II-Medical-8B in batches 2 and 3, and Gemma-4-E4B-it in batch 4. The system names in this phase reflect the Phase A retrieval pipeline used to produce the documents and snippets, not differences in the generation component. Yes/no exact answers are the strongest category, with macro-F1 reaching 0.83 in batch 2 and 0.87 for the ensemble variant with the Gemma4 model in batch 4. Factoid and list scores are more modest, which is expected given the retrieval limitations documented in Phase A.

The drop in batch 3 relative to batch 2 is consistent with Phase A retrieval quality, the generation module receives lower-quality context, and answer quality suffers accordingly. Batch 4 shows partial recovery and introduces an interesting insight, the ensemble pipeline (**lean_rag_ensemble**) achieves the best yes/no macro-F1 (0.87) despite not being the top variant by document MAP, suggesting that the multi-model consensus produces a qualitatively different candidate set that, while not strictly better by MAP, provides more useful context for the generation model.

ROUGE scores for ideal answers are modest across all batches, with ROUGE-2 F1 between 0.10 and 0.16. Differences between system variants within a batch are small and unlikely to reflect meaningful distinctions given the test set sizes (between 80 and 60 queries), also these automatic metrics don't capture the full alignment of ideal answers needing a manual curation to see the

Table 3
Phase A+ results per batch. Left block shows exact answer official metrics (macro-F1 for yes/no, MRR for factoid, F-measure for list) and right block shows ideal answer ROUGE scores. Bold values indicate the best results per batch and metric. The LLM was II-Medical-8B with 5-shot up to batch 3, and Gemma-4-E4B-it with 5-shot in batch 4.

	Exact answers			Ideal answers (ROUGE)			
System	**Y/N (maF1)**	**Fact. (MRR)**	**List (F)**	**R-2 (Rec)**	**R-2 (F1)**	**R-SU4 (Rec)**	**R-SU4 (F1)**
Batch 2							
lean_rag	**0.8329**	0.2667	0.2721	0.2057	0.1537	0.2187	0.1577
lean_rag_ft	**0.8329**	**0.3000**	**0.2772**	0.2122	**0.1564**	0.2210	0.1565
lean_rag_ft_sparse	**0.8329**	0.2500	0.2766	**0.2147**	0.1560	**0.2253**	**0.1582**
lean_rag_dprf	0.7667	**0.3000**	0.2544	0.2136	0.1534	0.2235	0.1565
Batch 3							
lean_rag	0.5417	**0.2941**	**0.2358**	0.1219	0.1035	0.1293	0.1088
lean_rag_ft	0.6857	0.2353	**0.2358**	**0.1374**	**0.1148**	0.1468	0.1213
lean_rag_ft_sparse	0.6857	0.2353	**0.2358**	0.1361	0.1141	**0.1483**	**0.1229**
lean_rag_dprf	**0.7708**	0.1765	0.1822	0.1327	0.1071	0.1452	0.1168
lean_rag_ensemble	**0.7708**	0.2353	0.2338	0.1373	0.1100	0.1463	0.1149
Batch 4							
lean_rag	0.7091	0.0909	0.2703	**0.2030**	0.1561	**0.2144**	0.1628
lean_rag_ft	0.7091	0.0909	0.3059	0.1891	0.1489	0.2014	0.1554
lean_rag_ft_sparse	0.7091	0.0909	**0.3226**	0.1836	0.1425	0.1983	0.1486
lean_rag_dprf	0.7091	**0.1364**	0.3068	0.1865	0.1452	0.1966	0.1514
lean_rag_ensemble	**0.8667**	0.0909	0.3065	0.1880	**0.1641**	0.1959	**0.1671**

true capacity of the LLM in generating a good natural language answer.

4.3. Phase B: Generation with Gold Snippets

Phase B results for exact and ideal answers are shown in Table 4. In this phase, the system names encode the few-shot count used for generation rather than the retrieval architecture: **lean_rag** uses 5-shot, **lean_rag_ft** uses 10-shot, and **lean_rag_ft_sparse** uses 0-shot. These settings were introduced incrementally: batch 2 covers the 5-shot setting only, batch 3 covers all three, and batch 4 covers the 5-shot and 0-shot settings. II-Medical-8B was used in batches 2 and 3, and Gemma-4-E4B-it in batch 4. In batch 4, only the 5-shot and 0-shot settings could be submitted: under Gemma-4-E4B-it the longer 10-shot prompt exceeded the available GPU memory (producing out-of-memory errors), whereas it had fit under II-Medical-8B in batch 3. Phase B represents the strongest results in this work, and the contrast with Phase A+ is a clear effect of the noise propagation. In batch 3, all three submitted systems achieve a perfect yes/no macro-F1 of 1.00, factoid MRR reaches 0.47 for the 10-shot variant and list F-measure reaches 0.42. These scores are substantially higher than the corresponding Phase A+ results from the same batch, where the same generation module with the same LLM and the same few-shot strategy produced lower scores across all categories. Since the only difference between Phase A+ and Phase B is the source of the snippets, this gap directly quantifies the cost of our retrieval limitations on downstream generation quality. In batch 4, only the 5-shot and 0-shot settings could be evaluated. Among these, the 5-shot variant obtained the best list F-measure across all of our submissions (0.4916), while the 0-shot variant produced the highest ROUGE scores. Because the 10-shot setting could not be run under Gemma-4-E4B-it, its effect in this batch remains unmeasured, given that it had been the strongest setting in batch 3, we do not interpret its absence as evidence that fewer in-context examples are preferable.

Regarding the effect of few-shot count, batch 3 is the only batch in which all three settings were submitted, and there the 10-shot variant was the strongest on the exact-answer metrics, obtaining the best factoid MRR (0.4706) and list F-measure (0.4192), differences on yes/no and on the ROUGE scores were marginal. This indicates that additional in-context examples helped mainly on the structured factoid and list outputs rather than uniformly across question types. In batch 4, the 0-shot variant produced the highest ROUGE scores under the newly introduced Gemma-4 model, possibly because the absence of in-context examples let the model generate more fluently, producing text that overlaps more with the reference ideal answers. As mentioned for Phase A+, these observations rest on automatic metrics and may not align with the manual scores assigned by the BioASQ experts.

Table 4
Phase B results per batch. Left block shows exact answer official metrics and right block shows ideal answer ROUGE scores. Bold values indicate the best result(s) per batch and metric. The LLM was II-Medical-8B up to batch 3 and Gemma-4-E4B-it in batch 4, the same as phase A+.

	Exact answers			Ideal answers (ROUGE)			
System	**Y/N (maF1)**	**Fact. (MRR)**	**List (F)**	**R-2 (Rec)**	**R-2 (F1)**	**R-SU4 (Rec)**	**R-SU4 (F1)**
Batch 2							
lean_rag	0.8929	0.3000	0.4188	0.2607	0.2306	0.2630	0.2310
Batch 3							
lean_rag	**1.0000**	0.4118	0.3563	0.2379	**0.2276**	0.2269	0.2170
lean_rag_ft	**1.0000**	**0.4706**	**0.4192**	**0.2419**	0.2272	**0.2347**	**0.2202**
lean_rag_ft_sparse	**1.0000**	0.4118	0.3339	0.2350	0.1998	0.2328	0.1946
Batch 4							
lean_rag	**0.9352**	**0.1818**	**0.4916**	0.2647	0.2462	0.2557	0.2380
lean_rag_ft_sparse	**0.9352**	**0.1818**	0.4243	**0.3332**	**0.2721**	**0.3295**	**0.2656**

5. Discussion and Conclusion

Our participation in BioASQ Task 14b set out to evaluate whether a lean, locally deployable RAG system could remain competitive on biomedical QA. The modular separation of the pipeline into a retrieval phase and a generation phase allowed us to isolate where the system succeeds and where it falls short, and the clearest finding of our participation is that retrieval was the primary bottleneck.

This conclusion is most directly supported by the contrast between Phase A+ and Phase B. Across all batches, Phase B scores are substantially higher than the corresponding Phase A+ scores, which quantifies the cost that our retrieval limitations impose on downstream answer quality. When supplied with high-quality context, our lean generation module produces strong results, including perfect macro-F1 on yes/no questions and competitive factoid and list scores in Phase B.

Within the retrieval phase, several design choices yielded informative results. Fine-tuning the cross-encoder re-ranker produced negligible gains over the base model, and the sparse-only pipeline performed comparably to the hybrid pipelines, both pointing to the first-stage candidate pool, rather than the re-ranking stage, as the limiting factor. The DPRF variant, despite its greater complexity, was consistently the weakest of our systems, suggesting that the benefit of dense expansion is contingent on a more precise seed selection mechanism than the one we employed. In contrast, the confidence threshold cutoff proved very effective. In batch 4, the only batch in which it was active, the ensemble pipeline achieved the highest mean Precision among all submissions, confirming that filtering low-confidence documents is a sound strategy. A related and somewhat counter-intuitive observation is that this ensemble pipeline, while not the strongest by document MAP, produced the best yes/no results in Phase A+, suggesting that standard retrieval metrics such as MAP do not fully capture which retrieved context is most

useful for downstream generation.
These results should be interpreted with an important caveat. All scores reported here are based on automatic metrics, and the manual evaluation conducted by the BioASQ biomedical experts was not yet available at the time of writing. Prior participants have documented a persistent misalignment between automatic metrics and human judgment in this task [12], where systems ranked highly by automatic measures were often rated poorly by human evaluators and vice versa.
Taken together, our findings indicate that a lean architecture constrained to a single GPU can be competitive in biomedical question answering, particularly in the generation phase and in Precision-oriented retrieval, while remaining limited in raw retrieval Recall. This pattern is consistent with the trade-offs expected from a deliberately lightweight design, which prioritizes local deployability and efficiency over the raw Recall achievable with larger retrieval infrastructures. These trade-offs are the central lesson of our participation, targeted and careful design choices can compensate for limited resources in some components of the pipeline, but not uniformly across all of them. A concrete illustration emerged in Phase B, where, under the Gemma-4-E4B-it model, the 10-shot prompt exceeded the memory of our single GPU and could not be run, even though the same configuration had fit under II-Medical-8B and had been the strongest few-shot setting in batch 3. Here the resource constraint did not merely lower a score it removed a competitive configuration from consideration altogether, underscoring that the cost of a lean design is not always a graceful degradation.
Future work will focus on the retrieval phase, which our results identify as the main bottleneck, and specifically on the first-stage candidate pool rather than the re-ranking stage. Since fine-tuning the cross-encoder yielded negligible gains, a next step is to adapt the first-stage dense retriever itself, by the fine-tuning of the bi-encoder used on BioASQ (query, abstract) pairs, using hard negatives as in our re-ranker fine-tuning, should raise the standalone recall that currently limits both the hybrid and DPRF pipelines. We also plan to implement hypothetical-document expansion (HyDE), in which a draft answer generated by the LLM already present in our pipeline is embedded and used for a better dense retrieval. In parallel, we will quantify how much recall is lost to the IVF-SQ8 quantization of our dense index by comparing it against an exact or HNSW index, in order to determine whether the dense channel is limited by the embedding model or by the index configuration. For the DPRF pipeline, we intend to gate seed selection with the same confidence threshold that proved effective in Phase A, so that dense expansion proceeds only from high-confidence seeds rather than diluting the pool with noisy ones. Also, we will apply the confidence threshold cutoff consistently across all batches, with per-model score calibration before fusion in the ensemble. Finally, after our finding that the ensemble system improved downstream yes/no answers without being the strongest by MAP, seems plausible to optimize retrieval directly for downstream answer quality, rather than for MAP alone. All of these directions will be pursued within the single-GPU budget that defines our lean setting.

Acknowledgments

This work is funded by national funds through FCT – Fundação para a Ciência e a Tecnologia, I.P. under the LASIGE Research Unit, ref. UID/00408/2025 , DOI https://-

doi.org/10.54499/UID/00408/2025. This work was supported by the Digital European Programme under the Grant Agreement 101083770 and from the Recovery and Resilience Plan (PRR) within the scope of the Recovery and Resilience Mechanism (MRR) of the European Union (EU), framed in the Next Generation EU, for the period 2021-2026, within project ATTRACT, with reference 774; and by Unicage (https://unicage.eu/).

Declaration on Generative AI

During the preparation of this work, the author(s) used Claude (Opus 4.7) in order to: Grammar and Spelling check, improve academic writing style and formatting assistance (adjusting tables and figure placement in LaTeX). After using this tool, the author(s) reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W. tau Yih, T. Rocktäschel, S. Riedel, D. Kiela, Retrieval-augmented generation for knowledge-intensive nlp tasks, 2021. URL: https://arxiv.org/abs/2005.11401. arXiv:2005.11401.

[2] M. Cheng, Y. Luo, J. Ouyang, Q. Liu, H. Liu, L. Li, S. Yu, B. Zhang, J. Cao, J. Ma, D. Wang, E. Chen, A survey on knowledge-oriented retrieval-augmented generation, arXiv preprint arXiv:2503.10677 (2025). URL: https://arxiv.org/abs/2503.10677.

[3] A. Nentidis, G. Katsimpras, A. Krithara, M. Krallinger, M. Rodríguez-Ortega, E. Rodriguez-López, N. Loukachevitch, A. Sakhovskiy, E. Tutubalina, D. Dimitriadis, G. Tsoumakas, G. Giannakoulas, A. Bekiaridou, A. Samaras, G. M. D. Nunzio, N. Ferro, S. Marchesin, M. Martinelli, G. Silvello, G. Paliouras, Overview of bioasq 2025: The thirteenth bioasq challenge on large-scale biomedical semantic indexing and question answering, 2025. URL: https://arxiv.org/abs/2508.20554. arXiv:2508.20554.

[4] A. Nentidis, G. Katsimpras, A. Krithara, M. Krallinger, M. Rodríguez-Ortega, E. Rodriguez-López, N. Loukachevitch, I. Rozhkov, E. Tutubalina, D. Dimitriadis, V. Patsiou, G. Tsoumakas, G. Giannakoulas, A. Bekiaridou, A. Samaras, G. M. Di Nunzio, N. Ferro, S. Marchesin, M. Martinelli, G. Silvello, G. Paliouras, Overview of BioASQ 2026: The fourteenth BioASQ Challenge on Large-Scale Biomedical Semantic Indexing and Question Answering, in: M. Hagen, M. Potthast, B. Stein, P. Schaer, E. Zangerle, S. MacAvaney, J. M. Struß, E. Sánchez Salido, A. Barrón-Cedeño, A. García Seco de Herrera (Eds.), Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Seventeenth International Conference of the CLEF Association (CLEF 2026), Lecture Notes in Computer Science, Springer, 2026.

[5] 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 2026 Working Notes, 2026.

[6] M. Abo El-Enen, S. Saad, T. Nazmy, A survey on retrieval-augmentation generation (rag) models for healthcare applications, Neural Computing and Applications (2025). doi:10.1007/s00521-025-11666-9.

[7] J. He, B. Zhang, H. Rouhizadeh, Y. Chen, R. Yang, J. Lu, X. Chen, N. Liu, D. Teodoro, Retrieval-augmented generation in biomedicine: A survey of technologies, datasets, and clinical applications, 2025. URL: https://arxiv.org/abs/2505.01146. arXiv:2505.01146.

[8] F. Cuconasu, G. Trappolini, F. Siciliano, S. Filice, C. Campagnano, Y. Maarek, N. Tonellotto, F. Silvestri, The power of noise: Redefining retrieval for rag systems, 2024. URL: https://arxiv.org/abs/2401.14887. doi:https://doi.org/10.1145/3626772.3657834.

[9] A. Krithara, A. Nentidis, K. Bougiatiotis, G. Paliouras, Bioasq-qa: A manually curated corpus for biomedical question answering, Scientific Data 10 (2023) 170. URL: https://doi.org/10.1038/s41597-023-02068-4. doi:10.1038/s41597-023-02068-4.

[10] S. E. Robertson, S. Walker, S. Jones, M. Hancock-Beaulieu, M. Gatford, Okapi at trec-3, in: Proceedings of the Third Text REtrieval Conference (TREC-3), 1994. URL: https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/okapi_trec3.pdf.

[11] 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). URL: http://dx.doi.org/10.1145/3458754. doi:10.1145/3458754.

[12] R. A. A. Jonker, T. Almeida, J. R. Almeida, S. Matos, BIT.UA at BioASQ 13b: Revisiting evaluation, DPRF-enhanced retrieval and fine-tuned LLMs, in: G. Faggioli, N. Ferro, P. Rosso, D. Spina (Eds.), Working Notes of the Conference and Labs of the Evaluation Forum (CLEF 2025), volume 4038 of *CEUR Workshop Proceedings*, CEUR-WS.org, Madrid, Spain, 2025. URL: https://ceur-ws.org/Vol-4038/paper_22.pdf.

[13] 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, Association for Computing Machinery, 2009. URL: https://doi.org/10.1145/1571941.1572114. doi:10.1145/1571941.1572114.

[14] Z. Xu, F. Mo, Z. Huang, C. Zhang, P. Yu, B. Wang, J. Lin, V. Srikumar, A survey of model architectures in information retrieval, 2025. URL: https://arxiv.org/abs/2502.14822. arXiv:2502.14822.

[15] S. Verma, F. Jiang, X. Xue, Beyond retrieval: Ensembling cross-encoders and GPT rerankers with LLMs for biomedical QA, in: G. Faggioli, N. Ferro, P. Rosso, D. Spina (Eds.), Working Notes of the Conference and Labs of the Evaluation Forum (CLEF 2025), volume 4038 of *CEUR Workshop Proceedings*, CEUR-WS.org, Madrid, Spain, 2025. URL: https://ceur-ws.org/Vol-4038/paper_50.pdf.

[16] N. Thakur, N. Reimers, A. Rücklé, A. Srivastava, I. Gurevych, Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models, 2021. URL: https://arxiv.org/abs/2104.08663.

[17] X. Zhang, Y. Zhang, D. Long, W. Xie, Z. Dai, J. Tang, H. Lin, B. Yang, P. Xie, F. Huang, M. Zhang, W. Li, M. Zhang, mGTE: Generalized long-context text representation and reranking models for multilingual text retrieval, in: Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track, Association for Computational Linguistics, 2024. doi:10.18653/v1/2024.emnlp-industry.103.

[18] J. Xu, W. B. Croft, Query expansion using local and global document analysis, in: Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Devel-

opment in Information Retrieval, Association for Computing Machinery, New York, NY, USA, 1996. URL: https://doi.org/10.1145/243199.243202. doi:10.1145/243199.243202.

[19] J. J. Rocchio, Relevance feedback in information retrieval, in: G. Salton (Ed.), The SMART Retrieval System: Experiments in Automatic Document Processing, Prentice-Hall, Englewood Cliffs, NJ, 1971, pp. 313–323.

[20] P. R. C. Lopes, S. I. R. Conceição, M. Fernandes, F. M. Couto, lasigeBioTM: A lean biomedical QA system empowered by structured knowledge, in: G. Faggioli, N. Ferro, P. Rosso, D. Spina (Eds.), Working Notes of the Conference and Labs of the Evaluation Forum (CLEF 2025), volume 4038 of *CEUR Workshop Proceedings*, CEUR-WS.org, Madrid, Spain, 2025. URL: https://ceur-ws.org/Vol-4038/paper_30.pdf.

[21] S. Ateia, U. Kruschwitz, Can open-source llms compete with commercial models? exploring the few-shot performance of current gpt models in biomedical tasks, 2024. URL: https://arxiv.org/abs/2407.13511. arXiv:2407.13511.

[22] M. Neumann, D. King, I. Beltagy, W. Ammar, Scispacy: Fast and robust models for biomedical natural language processing, in: Proceedings of the 18th BioNLP Workshop and Shared Task, Association for Computational Linguistics, 2019. URL: http://dx.doi.org/10.18653/v1/W19-5034. doi:10.18653/v1/w19-5034.

[23] D. Wadden, S. Lin, K. Lo, L. L. Wang, M. van Zuylen, A. Cohan, H. Hajishirzi, Fact or fiction: Verifying scientific claims, in: EMNLP, 2020.

[24] K. Roberts, T. Alam, S. Bedrick, D. Demner-Fushman, K. Lo, I. Soboroff, E. Voorhees, L. L. Wang, W. R. Hersh, TREC-COVID: rationale and structure of an information retrieval shared task for COVID-19, Journal of the American Medical Informatics Association (2020). URL: https://doi.org/10.1093/jamia/ocaa091. doi:10.1093/jamia/ocaa091.

[25] P. Malakasiotis, I. Pavlopoulos, I. Androutsopoulos, A. Nentidis, Evaluation Measures for Task B, Technical Report, BioASQ, 2020. URL: http://participants-area.bioasq.org/Tasks/b/eval_meas/.

6. Appendix: Query Type Specific Formatting Instructions

This appendix reproduces the four type-specific formatting instructions referenced in Section 3.3.2. One of these instructions is appended to the final user turn of every inference call.

Yes/No questions

You must answer ONLY with lowercase “yes” or “no”, even if you are unsure.

Factoid questions

Your response MUST be a single JSON object and nothing else.
Use EXACTLY this format: {“entities”: [“answer1”, “answer2”]}

Rules:

- Include ALL plausible answers, up to 5, ordered by decreasing confidence.
- Do NOT stop at one answer if multiple valid answers exist.
- Each entry must be a short, direct answer (a name, number, or expression).
- Do not include synonyms or duplicate meanings.
- Avoid acronyms unless the full name is unavailable.
- When multiple numerical answers exist, merge into a single range.
- If truly unknown, return an empty array: {“entities”: []}

Example: {“entities”: [“Metformin”, “Insulin”, “Glipizide”]}

List questions

Your response MUST be a single JSON object and nothing else.
Use EXACTLY this format: {“entities”: [“answer1”, “answer2”]}

Rules:

- Up to 100 entries, max 100 characters each.
- Answers may be supplemented with your own knowledge if needed.
- Prefer the shortest, cleanest, and most direct entry from the passage that accurately answers the question.
- All phrases, even if incomplete, may be relevant.
- Do not include synonyms or duplicate meanings.
- If unknown, return an empty array.

Example: {“entities”: [“entity1”, “entity2”]}

Summary questions

Please give a concise and correct answer to the question. STRICT LIMIT: your answer must be at most 200 words.

Rules:

- Prefer answers that combine direct phrases or sentences from the passage.
- You may add minimal connecting words (such as “and”, “which”, “that”, “because”, “thus”) to make the answer grammatically correct.
- Do NOT start your answer with phrases like “Based on the passage” or “The answer is”. Go straight to the answer.