BioASQ Phase B: Exploring Diverse Optimization Techniques and Testing Frameworks for Enhanced Bio-Answering Performance

CSA-IISR at CLEF 2026

Yen-Sung Chu[1,†], Cheng-Yu Hou[1,†], Chen-Hau Hsu[1,†] and Richard Tzong-Han Tsai[2,3,*]

[1]Chingshin Academy, Taipei, Taiwan
[2]Department of Computer Science and Information Engineering, National Central University, Taiwan
[3]Department of Medical Research, Cathay General Hospital, Taipei, Taiwan

Abstract

This paper describes our system for the BioASQ Task 14b Phase B challenge at CLEF 2026 [27, 28]. Without any model fine-tuning, our training-free framework systematically explores a broad range of inference-time optimization strategies across multiple frontier LLMs, implemented within a feedback-driven automated prompt optimization loop. Our empirical results indicate that strategic prompt engineering under a multi-model comparative setup yielded superior scalability over more complex pipelines. The resulting system achieved first place on all four competition batches for list questions, outperforming all competing teams. Furthermore, our approach demonstrated robust generalizability across other question types, securing first place in two batches and second place in the other two batches for factoid questions, as well as first place in one batch for yes/no questions. Our results demonstrate that, within the BioASQ Phase B setting where gold-standard snippets are provided, a well-designed instruction optimization pipeline can rival or surpass more complex architectures involving retrieval augmentation or model fine-tuning.All prompts and code are publicly available at https://github.com/yensungchiu/BioASQ [2].

Keywords

BioASQ, biomedical question answering, inference-time optimization, prompt engineering, automated prompt optimization, retrieval-augmented generation, large language models, training-free framework

1. Introduction

The BioASQ challenge [1], running since 2013, is one of the most established benchmarks for biomedical question answering [3], attracting research groups from leading institutions worldwide. Task B requires systems to answer questions from provided PubMed snippets across four types: yes/no, factoid, list, and summary [4]. In this work, we focus on the three exact-answer types: yes/no, factoid, and list. Recent BioASQ editions have seen increasing adoption of LLMs, with top systems employing RAG pipelines [5, 6], ensemble strategies [7, 8], and fine-tuned BERT-based architectures [9, 10]. However, the space of inference-time prompt engineering remains underexplored: most systems optimize for surrogate objectives rather than official evaluation metrics, and self-feedback mechanisms have shown mixed results in practice [11]. We present our system for BioASQ Task 14b Phase B at CLEF 2026. Without any model fine-tuning, our training-free framework combines structured snippet preprocessing, multi-model comparative analysis, and a feedback-driven automated prompt optimization loop that uses official BioASQ evaluation scores directly as the optimization signal—distinct from DSPy [12] and OPRO [13], which rely on proxy objectives. Our system achieved first place on all four competition

CLEF 2026 Working Notes, 21 – 24 September 2026, Jena, Germany
*Corresponding author.
†These authors contributed equally.
11335007@st.chjhs.tp.edu.tw (Y. Chu); 11335036@st.chjhs.tp.edu.tw (C. Hou); aaads10111217@gmail.com (C. Hsu); thtsai@g.ncu.edu.tw (R. T. Tsai)
0009-0008-6190-8211 (Y. Chu); 0009-0000-9335-0912 (C. Hou); 0009-0003-1380-7700 (C. Hsu); 0000-0003-0513-107X (R. T. Tsai)

batches for list questions, first or second place across all batches for factoid questions, and first place in one batch for yes/no questions. Our main contributions are:

1. A unified, training-free biomedical QA framework achieving state-of-the-art results across yes/no, factoid, and list questions through systematic prompt engineering and automated optimization.
2. Type-specific ablation studies revealing: the failure of self-feedback and snippet re-ranking for yes/no questions; model-specific strengths and ranked output formatting for factoid extraction; and the failure of BM25 re-ranking for list questions [14, 15].
3. A feedback-driven prompt optimization loop that improves list F1 by over 7%, with the optimized prompt generalizing to all four task 14b phase b competition batches.

The remainder of this paper is organized as follows. Section 2 reviews related work. Section 3 describes the task and evaluation metrics. Section 4 presents our system and per-type methodology. Section 5 reports development and official competition results. Section 6 provides analysis and discussion, and Section 7 concludes.

2. Related Work

2.1. BioASQ Challenge

The BioASQ challenge [3, 1] has run since 2013, evolving from BM25-based retrieval to neural and LLM-based approaches [16, 6]. BioASQ 13b [4] attracted systems spanning fine-tuned encoders, RAG pipelines, and prompting frameworks across a curated corpus of 5,389 questions [17]. Notable participants include: DMIS-KU [7], who used log-probability ensemble decoding across snippet conditions to top three of four batches; Evidence Prime [8], showing that ensembles of small open-weight models can rival GPT-4o and Claude 3.5; NCU-IISR [15, 14], with a multi-model RAG framework using GPT-4o, o3-mini, and Gemini 2.0; the University of Regensburg [11], finding that self-feedback with reasoning models yields inconsistent gains and few-shot prompting often outperforms it; UniTor [18], a modular RAG pipeline with synthetic snippet generation and open-source LLMs; and PJs-team [19], using embedding-based reranking with a LoRA-finetuned Qwen2.5-3B, confirming that proprietary LLMs outperform finetuned smaller models on exact-answer tasks.

2.2. LLMs for Biomedical QA

The emergence of frontier Large Language Models (LLMs) such as GPT-5 (OpenAI), Claude (Anthropic), and Gemini (Google DeepMind) has opened new avenues for biomedical question answering [20]. Several studies have demonstrated that these models achieve strong performance on medical benchmarks without domain-specific training, particularly when prompted with retrieved evidence. Domain-adapted pre-trained models such as BioBERT [9] and PubMedBERT [10] established strong baselines for biomedical NLP tasks and remain competitive for entity-level extraction; however, frontier LLMs with appropriate prompting have progressively narrowed or closed this gap in BioASQ evaluations.

2.3. Prompt Engineering and Optimization

Prompt engineering has emerged as a practical alternative to fine-tuning for adapting LLMs to specific tasks. Common strategies include zero-shot prompting, few-shot in-context learning [21], chain-of-thought reasoning [22], and structured output formatting. Automated prompt optimization methods such as DSPy [12], OPRO [13], and APE [23] have proposed systematic approaches to prompt refinement using surrogate objectives or model-internal signals. Our method differs in that it uses the official BioASQ evaluation service as the optimization target, ensuring that improvements are measured directly against ground-truth metrics rather than proxy objectives.

2.4. Retrieval-Augmented Generation

RAG [5] combines parametric LLM knowledge with non-parametric retrieval from external document collections, reducing hallucination and improving factual grounding. In the biomedical domain, RAG has been applied to clinical QA [24] and literature-based question answering, where grounding answers in PubMed abstracts is critical. In BioASQ Phase B, gold-standard snippets are provided alongside each question, simplifying the retrieval component to snippet selection and reranking. Several BioASQ systems have exploited this by reranking or filtering the provided snippets before generation [6, 18, 19]. Our system extends this line of work with a lightweight IDF-based snippet reranker for factoid questions, while our ablation studies for list questions reveal that BM25-based filtering—a common technique in prior systems [14]—severely degrades performance due to its inability to capture biomedical synonym relationships.

3. Task Description and Evaluation Metrics

In BioASQ Task B Phase B, each question is provided together with a set of gold-standard relevant articles and text snippets retrieved from PubMed. Participating systems must return an exact answer whose format depends on the question type. The participants are told the type of each question. We describe the three exact-answer types and their official metrics below.

3.1. Yes/No Questions

Each yes/no question requires a single binary label — *yes* or *no* — as the exact answer, compared against the gold label associated by the BioASQ team of biomedical experts. Since BioASQ 6b, the official evaluation metric is the **Macro-averaged F1 score (Macro-F1)**, defined as the unweighted mean of the per-class F1 scores for the *yes* and *no* classes:

$$\text{Macro-F1} = \frac{F_1^{\text{yes}} + F_1^{\text{no}}}{2}, \tag{1}$$

where F_1^{yes} and F_1^{no} are the standard F1 scores[1] computed by treating "yes" and "no" as the positive class, respectively. Accuracy is also reported for completeness but is not the official ranking criterion.

3.2. Factoid Questions

Factoid questions require the system to identify one or a small number of specific entities (e.g., a gene name, a drug, or a disease name) as the exact answer. Unlike yes/no or list questions, factoid questions demand precise entity extraction—the returned answer must match the gold standard at the surface-form level rather than merely capturing the correct semantic meaning. Systems return a ranked list of up to five candidate answer strings per question. Performance is evaluated using three complementary metrics. Strict accuracy requires the top-ranked candidate to exactly match the gold answer. Lenient accuracy relaxes this requirement, counting a question as correct if any of the five returned candidates matches the gold answer. Mean Reciprocal Rank (MRR) provides a ranking-aware measure that rewards systems for placing the correct answer higher in the list, and serves as the primary metric for the factoid subtask:

$$\text{MRR} = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{\text{rank}_i}, \tag{2}$$

where $|Q|$ is the total number of factoid questions and rank_i is the position of the first correct answer in the ranked list returned for question i. A reciprocal rank of zero is assigned when no correct answer appears among the five candidates, penalising systems that fail to retrieve the correct answer entirely. MRR therefore incentivises systems to both retrieve the correct answer and rank it as highly as possible.

[1] For a binary class c, the per-class F1 score is defined as $F_1^c = 2 \cdot \frac{P_c \cdot R_c}{P_c + R_c}$, where P_c and R_c denote precision and recall with class c treated as positive.

3.3. List Questions

List questions require the system to extract all qualifying entities that answer the question from the provided snippets. The evaluation penalizes both omission (missed correct answers, reducing recall) and over-generation (spurious answers, reducing precision) equally through Mean Precision, Mean Recall, and the standard F1 score (see Section 3, footnote 1). The official measure for list questions is the mean F1.

4. Methodology

4.1. System Overview

Our system consists of a unified pipeline applied independently to each question type (yes/no, factoid, list), with type-specific prompt designs and optimization strategies. The shared pipeline stages are: (1) snippet processing, (2) prompt construction, (3) LLM generation, and (4) answer extraction. Each stage was independently evaluated through controlled ablation experiments. In addition to the forward pipeline, we employ an automated prompt optimization loop for each question type: after each submission to the BioASQ evaluation service, the official scores are fed back to the LLM along with the current prompt and the full optimization history. The LLM then generates an improved prompt targeting the identified weaknesses.

4.2. Yes/No Question Answering

We adopt a prompt-based, Retrieval-Augmented Generation (RAG) [5] architecture in which a large language model receives the question text and the associated gold snippets as context. The overall pipeline and the family of experimental variations are described below.

Table 1 summarizes the dataset, sample size, model(s), and experimental status (development vs. held-out vs. official) for every yes/no ablation reported in Sections 4.2 and 5.1.

Baseline system. Given a question q and gold snippets $S = \{s_1, \ldots, s_n\}$, we prompt the model with a system message defining its role and a user message containing the question and concatenated snippets. Two output variants are evaluated: **baseline_prompt** elicits step-by-step reasoning followed by a final "yes" or "no"; **baseline_yesno_only** restricts the output to a single binary token with no surrounding text. Full prompt texts are available at https://github.com/yensungchiu/BioASQ/tree/master/yesno. The backbone is `Claude Opus 4.6` via OpenRouter[2] at temperature 0.

Prompt engineering. Using `Claude Opus 4.6` (temperature 0, BioASQ 11b–13b, $N \approx 268$), we evaluate combinations of three system prompts and three user templates. *System prompts:* (i) *Default* — step-by-step expert assistant; (ii) *Conservative* — biases toward "no" under weak or mixed evidence; (iii) *Critical reviewer* — requires unambiguous evidence, defaults to "no" for conflicting snippets. *User templates:* (i) *Default* — standard step-by-step reasoning; (ii) *Conflict-aware* — treats conflicting evidence as "no"; (iii) *Snippet-analysis* — notes each snippet's evidence before reasoning.

Model comparison. We compare five frontier LLMs under two prompt conditions, both at temperature 0, evaluated on BioASQ 11b–13b ($N \approx 269$–270). Under the *default prompt*, `Claude Opus 4.6`, `Claude Sonnet 4.6`, `Gemini 3.1 Pro`, `GPT-5.5 Pro`, and `Perplexity Sonar-Reasoning-Pro` are evaluated under the identical default prompt established for the baseline. Under the *best prompt of Opus*, the same model set (excluding `GPT-5.5 Pro`, which incurred prohibitive cost and produced inferior results) is re-evaluated under the best prompt identified by prompt engineering, to assess whether prompt optimisation generalises across model families. Models were accessed via OpenRouter using the identifiers `anthropic/claude-opus-4.6`, `anthropic/claude-sonnet-4.6`,

[2]OpenRouter provides a unified API gateway to multiple LLM providers: https://openrouter.ai [25].

Table 1
Summary of experimental setup for all yes/no ablations. "Status" indicates whether the result is a development-set finding, a held-out validation result, or an official competition submission.

Experiment	Dataset	N	Model(s)	Configuration	Status
CoT vs. direct output	11b–13b	269	Opus 4.6	`baseline_prompt` vs. `baseline_yesno_only`	Dev
Prompt engineering (A2-1–A2-5)	11b–13b	≈268	Opus 4.6	3×3 system/user prompt grid	Dev
Model comparison (default prompt)	11b–13b	≈269–270	5 frontier LLMs	Default prompt	Dev
Model comparison (best prompt)	11b–13b	≈268–270	4 frontier LLMs	Best prompt (A2-5)	Dev
Temperature sensitivity	11b–13b	269	Opus 4.6	$T \in \{0, 0.3, 0.7, 1.0\}$	Dev
Iterative prompt refinement	11b+12b / 9b+13b	160 (dev)	Opus 4.6	Dev-1/2/3 rule additions	Dev + Held-out
Snippet re-ranking	12b+13b	≈182–184	Opus 4.6	Embedding re-ranking, Top-$K \in \{5, \text{all}\}$	Dev
In-context example retrieval	11b–13b	≈220–222	Opus 4.6	$K \in \{6, 10, 16\}$, yes:no ratio	Dev
Majority voting	11b–13b	≈220–222	Opus 4.6	Shuffle/Temperature voting, $N \in \{3, 5\}$	Dev
Single-snippet voting	13b	82	4 frontier LLMs	Per-snippet majority vote	Dev
Self-feedback (SF-1–5)	12b	≈102	Opus 4.6 / Sonnet 4.6	3-turn self-refinement	Dev
Document enrichment	12b+13b	182 / 158	Opus 4.6	Snippets vs. +Abstracts vs. Abstracts-only	Dev
DSPy MIPROv2 optimization	Synthetic split	100 / 50 / 100	Sonnet 4.6	H-1 (Zero-shot CoT) / H-2 (Predict+demos)	Held-out
Official competition	14b, Batches 1–4	see Table 19	Opus 4.6, Gemini 3.1 Pro, GPT-5.4/5.5 Pro, Perplexity Sonar-R-Pro	5 submission variants/batch	Official

`google/gemini-3.1-pro`, `openai/gpt-5.5-pro`, and `perplexity/sonar-reasoning-pro`. All yes/no experiments were conducted between February and March 2026, with temperature $= 0$ and all other decoding parameters (top-p, max tokens, stop sequences) left at OpenRouter defaults.

Temperature sensitivity. We evaluate temperatures $T \in \{0,\ 0.3,\ 0.7,\ 1.0\}$ on `Claude Opus 4.6` with the default prompt to assess output stability.

Iterative prompt refinement. Starting from `baseline_prompt`, we conduct an error-driven, iterative prompt refinement procedure using BioASQ 11b and 12b as the development set and BioASQ 9b and 13b as held-out test sets. Across three rounds, we analyse prediction errors, categorise failure modes into recurring patterns, and encode corrective decision rules directly into the system prompt.

Snippet relevance re-ranking. We re-rank snippets by cosine similarity to the question using `all-MiniLM-L6-v2` (384-d) and `S-PubMedBERT-MS-MARCO`, ablating the LLM backbone, Top-K truncation ($K \in \{5, \text{all}\}$), and whether similarity scores are shown in the prompt.

In-context example retrieval. We retrieve the top-K semantically similar training examples via a Chroma vector index, balanced between "yes" and "no" labels, and prepend them as demonstrations. We ablate $K \in \{6, 10, 16\}$, the yes/no ratio (0:10, 5:5, 10:0), and whether the retrieval query includes snippets.

Majority voting. To reduce stochastic sensitivity, we employ majority voting over N independent inferences per question via two strategies: **shuffle voting** (snippet order is randomly permuted per pass; temperature fixed at 0) and **temperature voting** (snippet order fixed; temperature > 0 introduces stochastic variation). We evaluate $N \in \{3, 5\}$ votes and temperatures $T \in \{0.5, 1.0\}$.

Single-snippet voting. Each snippet is presented to the model individually, and per-snippet binary predictions are aggregated via majority vote. This tests whether decomposed, per-snippet reasoning followed by ensemble aggregation can match joint multi-snippet reasoning.

Self-feedback. We implement a three-turn self-refinement pipeline: (1) **Generate** — the model produces an initial answer with chain-of-thought reasoning; (2) **Critique** — the model reviews its own reasoning using a structured checklist or a devil's-advocate framing; (3) **Finalise** — the model issues a final answer incorporating the self-critique. We ablate the critique prompt (default checklist vs. devil's advocate) and the model assignment for each turn (homogeneous `Opus` vs. heterogeneous `Opus` + `Sonnet`).

Document enrichment. Beyond the gold snippets, we experiment with augmenting the context with full PubMed abstracts corresponding to the gold document IDs, evaluated on BioASQ 12b–13b using `Claude Opus 4.6`. We compare three configurations: **snippets only** (baseline), **snippets + abstracts** (full abstracts appended after the snippet block, with section headers retained), and **abstracts only** (the snippet block is removed entirely and replaced with full abstracts, testing whether abstracts alone can substitute for curated snippets).

Automated prompt optimisation with DSPy. We apply the DSPy framework with its MIPROv2 optimiser to automatically discover improved instructions and few-shot demonstrations. Two module architectures are compared: **Zero-shot CoT (H-1)** uses `dspy.ChainOfThought` with no demonstrations, so MIPROv2 can only rewrite the instruction text; **Predict + demos (H-2)** uses `dspy.Predict` (no explicit CoT field) with up to 4 bootstrapped and 4 labelled demonstrations, so MIPROv2 optimises both instruction and demonstrations. Both use `Claude Sonnet 4.6` as the backbone ($N_{\text{train}} = 100$, $N_{\text{dev}} = 50$, $N_{\text{test}} = 100$, seed $= 42$).

4.3. Factoid Question Answering

Our factoid approach is built on systematic, error-driven prompt engineering across 13 iterative experiments on BioASQ 10b–13b development data, combined with snippet reranking and careful model selection.

System Infrastructure. All experiments share a common asynchronous pipeline built on `asyncio` and `AsyncOpenAI`, allowing concurrent processing of up to 20 questions via `asyncio.Semaphore`. Snippets are passed through a reranking module before prompt insertion, and outputs are logged to Weights & Biases (W&B) per question. Evaluation uses Mean Reciprocal Rank (MRR), computed by

finding the first predicted candidate that matches any gold answer after normalization (lowercasing, bracket removal, punctuation stripping, and whitespace compression).

LLM-assisted prompt development via structured error analysis. To systematically identify failure modes and guide prompt refinement, all prediction runs were logged to W&B in a structured per-question format capturing input question, gold answer, model output, MRR contribution, error flag, prompt template identifier, and model identifier. Run-level aggregate metrics—including total questions, correct count, accuracy, MRR, error count, and error rate—were recorded alongside, enabling systematic cross-run comparison across prompt iterations. When MRR improvement plateaued, the full prediction log was exported as a CSV file and submitted to an LLM assistant (Claude) for pattern-level error analysis. The LLM identified recurring failure categories—such as over-specification, paraphrasing mismatch, and surface-form boundary errors—and proposed targeted prompt revisions, which were reviewed and selectively incorporated by the first author into the subsequent prompt iteration. This LLM-assisted analysis involves no gradient updates or model weight modification, remaining consistent with our training-free framework, and was conducted exclusively on BioASQ 10b–13b development data.

Early Explorations (Exp 1–3). Starting from a simple GPT-4o-mini single-answer prompt (MRR $\sim$0.30), the system prompt instructed the model to act as a biomedical expert and return a single concise factoid answer, while the user prompt provided five guidelines emphasizing brevity and discouraging full sentences or explanations. Despite the seemingly reasonable setup, performance was far below acceptable levels. Error analysis identified four persistent failure modes: *over-specification* (e.g., "amenable to exon 53 skipping" instead of "exon 53"), *paraphrasing mismatch* (e.g., "below 0.5%" vs. "less than 0.5%"), *under-specification* (over-trimming required tokens), and *hallucination on null cases*. Experiment 2 added a *Trim-Aggressively* verbatim extraction rule requiring the model to output a JSON list of exactly one minimal entity copied from the snippets, with explicit bad/good examples such as `BAD: "the expression of p53"` vs. `GOOD: "p53"`, improving MRR to $\sim$0.38. Experiment 3 further added few-shot boundary annotations showing correct answers (e.g., `["BRCA1"]`) alongside marked-incorrect alternatives (e.g., `["BRCA1 gene"]`, `["the BRCA1 gene"]`), pushing MRR to $\sim$0.40. Despite these gains, the original failure modes persisted throughout, and an additional issue of *empty output* emerged when questions required specific numerical values.

Ranked Output Strategy (Exp 4–8). Since exact-match on a single answer is inherently brittle — the gold answer might be "Ventriculoperitoneal shunt" while the model outputs "VP-shunt," or the gold might be "high cholesterol" while the model extracts "LDL cholesterol" — Experiment 4 reframed the task as producing a ranked list of up to five candidates explicitly targeting MRR, with the most confident candidate at Rank 1, valid variations such as abbreviation/full-name pairs at Ranks 2–4, and a competing distinct entity at Rank 5 as an insurance policy. This structural change alone improved MRR from 0.40 to 0.45, confirming that the ranked-list formulation is fundamentally better suited to MRR optimization than single-answer extraction. Experiments 5–7 explored increasingly structured multi-layer prompts (Minimalist Entity, Standard Noun Phrase, and Contextual/Verbose layers), but counterintuitively degraded MRR to 0.33–0.44, showing that overly prescriptive instructions confuse extraction. Experiment 8 reverted to a simpler two-section structure, recovering MRR to $\sim$0.44.

Snippet Reranking (Exp 11). Hypothesizing that input quality was the remaining bottleneck, we built a lightweight dependency-free reranker. It scores each snippet by IDF-weighted token overlap with the question, with additive bonuses for bigram ($+3.0$) and phrase matches ($+2.0$). The top-6 snippets are concatenated up to a 5,000-character limit before LLM inference. This module confirmed that filtering irrelevant context is as important as prompt design, contributing to MRR $\sim$0.45.

Final Prompt Design (Exp 12–13). Upgrading to Claude Opus/Sonnet 4.6 with an eight-step structured prompt raised MRR to ∼0.50 (Exp 12). The prompt covers: (1) question format and semantic type identification across 16 categories; (2) verbatim candidate extraction with context-trimming rules; (3) surface-form variant generation including abbreviation/full-name pairs and singular/plural forms; and (4) eight type-specific extraction rules for definition, mutation/exon, numeric, route, mechanism, disease, gene/protein, and anatomical questions. Subsequent error analysis revealed overfitting to older 8b–12b patterns, particularly on fragment questions, binary alternative questions, and British English spellings. Experiment 13 addressed this by adding: fragment-question rewriting (e.g., "Adverse effects of metformin." → "What are the adverse effects of metformin?"); mandatory British/American spelling pairs with British spelling at Rank 1 (e.g., "haemophilia" before "hemophilia"); modifier-inclusive variants (entity alone, gerund form, "use of X" form, and modifier-plus-entity form); and five additional type rules (binary alternative, historical name, specificity level, and entity-qualifier variants). Tie-breaking follows: verbatim snippet match > British spelling > shorter form > answer-type alignment. These additions raised MRR to 0.60–0.70 on BioASQ 13b validation.

Model Selection and Submission. For BioASQ 14b we distributed five submission slots across Claude Opus/Sonnet 4.6, Claude Opus 4.7, Gemini 3.1 Pro Preview, and GPT-5.0 Pro, holding the prompt constant. Raw outputs were parsed through a JSON safety pipeline and deduplicated case-insensitively. The final submission pipeline was fully automatic, with no manual intervention on answer content prior to submission.

4.4. List Question Answering

For list questions we follow a four-stage ablation pipeline (model selection, snippet processing, prompt strategy, and automated optimization), visualised in Figure 1.

Snippet processing. BioASQ Phase B provides gold standard snippets retrieved from PubMed for each question. We investigated three approaches. **Raw input (baseline)**: official snippets are concatenated with whitespace, preserving the original formatting, with no preprocessing applied. **BM25 re-ranking**: we hypothesized that the official snippets may contain passages not directly relevant to the question, and that filtering could reduce noise; we applied BM25 (Okapi) scoring via the `rank-bm25` library (v0.2.2) with default parameters to re-rank snippets by relevance to the question text, retaining only the top-K most relevant snippets, testing $K \in \{5, 10, 20\}$ with lowercase alphanumeric tokenization. **Processed (final configuration)**: all official snippets are retained (no filtering) with three lightweight enhancements — (1) sequential numbering (`[S01]`, `[S02]`, ...) for unambiguous reference, (2) whitespace normalization (newlines, tabs, and multiple spaces collapsed to single spaces), and (3) truncation of the concatenated context at 5,000 characters.

Model selection. Three frontier LLMs were evaluated under identical controlled conditions: **GPT-4** (`openai/gpt-4`), **Gemini 3 Pro** (`google/gemini-3.1-pro-preview`), and **Claude Opus 4.6** (`anthropic/claude-opus-4.6`), all accessed via the OpenRouter API with temperature $= 0.0$ for deterministic outputs and all other decoding parameters (top-p, max tokens, stop sequences) left at OpenRouter defaults. All list-question experiments were conducted between February and March 2026.

Prompt engineering. We systematically evaluated five prompting strategies. **Zero-shot baseline**: a minimal instruction with no extraction rules or examples. **Few-shot (3 examples)**: three biomedical Q&A examples spanning different entity types (genes, symptoms, drugs) as in-context demonstrations. **Structured output**: a detailed rule-based prompt with explicit constraints covering JSON formatting, case-insensitive deduplication, minimal entity extraction, full-name preference over abbreviations, frequency-based ordering, and a 100-character maximum per item. **Role-based expert**: a persona-driven prompt assigning the model the role of "a senior biomedical researcher with 20+ years of

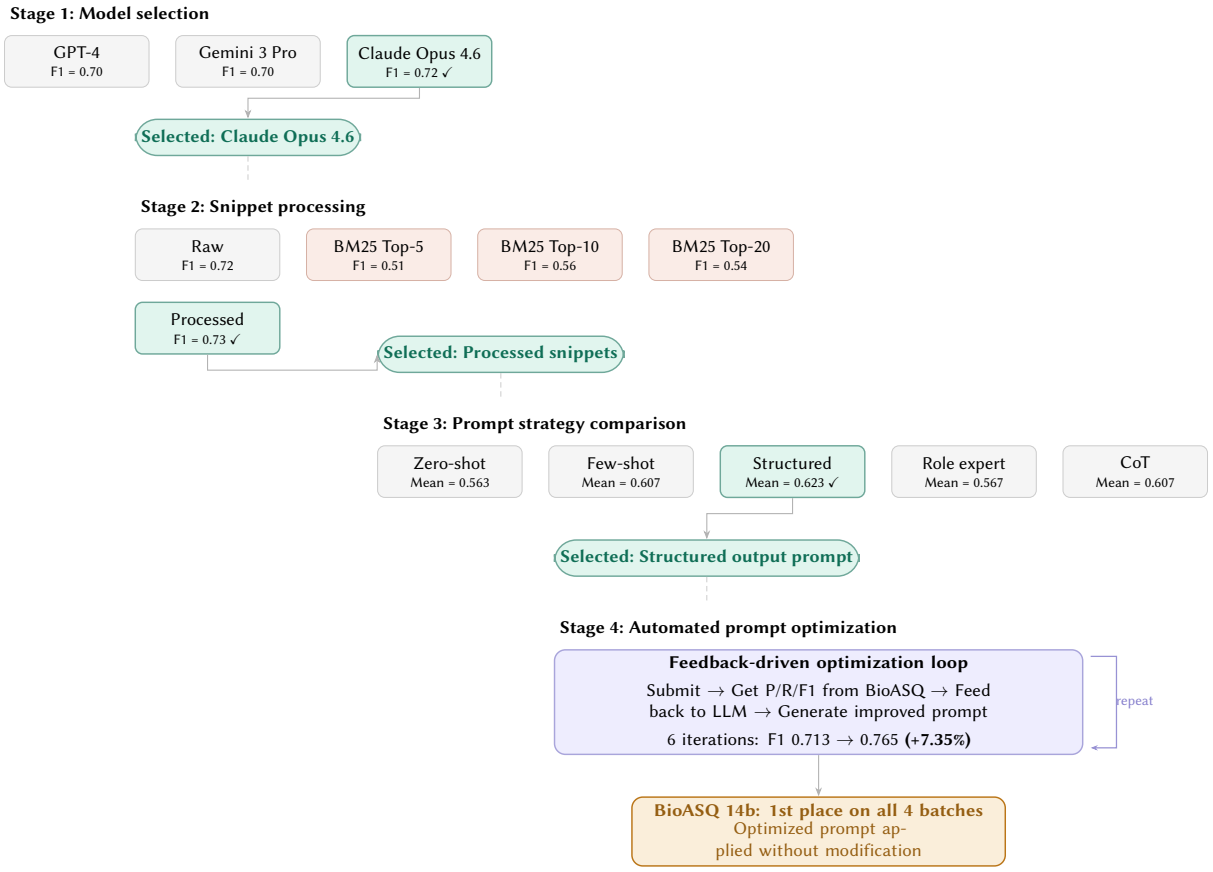

Figure 1: List question answering pipeline. Four sequential ablation stages: (1) model selection identifies Claude Opus 4.6 as the best-performing model; (2) snippet processing reveals BM25 re-ranking degrades performance while lightweight preprocessing provides marginal improvement; (3) prompt strategy comparison selects structured output for its cross-batch stability; (4) automated prompt optimization improves F1 by 7.35%. The final optimized prompt generalizes to achieve first place on all four BioASQ 14b competition batches.

experience." **Chain-of-Thought (CoT)**: a five-step reasoning process — (1) identify candidate entities, (2) verify against the question, (3) deduplicate, (4) order by frequency, (5) output the final JSON list.

Automated prompt optimization. Our core contribution for list questions is a feedback-driven prompt optimization loop using official BioASQ evaluation scores—not surrogate metrics—to iteratively refine the prompt. The procedure initializes with the structured output prompt, submits predictions to the BioASQ evaluation service, feeds back the official Precision/Recall/F1 scores along with the full optimization history, and generates an improved prompt targeting the identified weakness. This cycle repeats until F1 converges or declines. This approach is distinct from DSPy [12] and OPRO [13] in that it uses real official evaluation scores rather than surrogate objectives.

During competition, the optimization was performed on BioASQ training batch 11b (batch 4), and the resulting prompt was applied without modification to the 14b competition batches. However, the original per-iteration logs from the competition-time optimization were not preserved. To validate the method's reproducibility and provide the ablation data reported in this paper, we conducted a post-competition replication of the optimization procedure using the same method, training data (11b batch 4), model (Claude Opus 4.6), and evaluation service. The replication produced a different but comparably effective prompt, confirming that the optimization procedure reliably yields high-quality prompts across independent runs. Both the competition prompt and the replication prompt are provided in Appendix C.

The meta-prompt used to generate improved prompts at each iteration provides the LLM with: (1) its role as a prompt engineer for BioASQ, (2) the full performance history across all previous iterations

(Precision, Recall, F1), (3) the current prompt text, (4) the latest official evaluation scores, (5) a diagnostic analysis of whether precision or recall is the current weakness, and (6) an instruction to generate a self-contained improved prompt preserving the `{question}` and `{context}` placeholders. The full meta-prompt template is provided in Appendix D.

Output post-processing. All list-question submissions were fully automatic with no human review or post-editing. Model outputs were parsed by first stripping common prefixes (e.g., "answer:", "output:"), then extracting the first JSON array via regex matching. If standard `json.loads` parsing failed, single quotes were replaced with double quotes and parsing was retried; if both attempts failed, the raw string was split on commas as a fallback. Items containing parentheses were removed (to avoid partial entity references), and leading/trailing whitespace was stripped from each item. Case-insensitive deduplication was handled by the prompt instruction rather than programmatic post-processing.

5. Experiments and Results

5.1. Yes/No Questions

Chain-of-thought vs. direct output. Both variants use `Claude Opus 4.6` at temperature 0, evaluated on BioASQ 11b–13b ($N = 269$). There is no difference, indicating that the output format — whether the model externalizes its reasoning or internalizes it — has no measurable effect on classification accuracy. It suggests that frontier LLMs perform the underlying evidence assessment *regardless* of whether intermediate steps are explicitly written out. In practice, `baseline_yesno_only` reduces token usage and latency, making it preferable for large-scale inference, while `baseline_prompt` retains its value for error analysis and iterative prompt development.

Table 2
Chain-of-thought vs. direct output.

Variant	Acc.	Macro-F1	F1-Yes	F1-No
`baseline_prompt`	0.9442	0.9399	0.9560	0.9239
`baseline_yesno_only`	0.9442	0.9399	0.9560	0.9239

Prompt engineering. Evaluated with `Claude Opus 4.6` ($N \approx 268$). The best-performing combination (A2-5, Macro-F1 $= 0.9523$) pairs a "critical biomedical reviewer" system instruction with a structured snippet-analysis user template, surpassing the default prompt by $+0.0124$. The conflict-aware user prompt (A2-1) performs worst among all variants, suggesting that explicitly instructing the model to treat conflicting evidence as "no" introduces a systematic *no*-bias that reduces recall on the *yes* class (F1-Yes $= 0.9415$ vs. 0.9560 baseline). The conservative system prompt (A2-3) similarly depresses *yes*-class recall, whereas the critical-reviewer framing (A2-4, A2-5) achieves higher precision on *yes* without sacrificing recall, yielding the best balance across both classes.

Table 3
Prompt variant comparison (Claude Opus 4.6, $N \approx 268$).

ID	System / User Prompt	Acc.	Macro-F1	F1-Yes	F1-No
A2-1	Default / Conflict-aware	0.9291	0.9257	0.9415	0.9100
A2-2	Default / Snippet-analysis	0.9478	0.9439	0.9586	0.9293
A2-3	Conservative / Default	0.9403	0.9359	0.9527	0.9192
A2-4	Critical reviewer / Default	0.9515	0.9483	0.9612	0.9353
A2-5	**Critical reviewer / Snippet-analysis**	**0.9552**	**0.9523**	**0.9641**	**0.9406**

Model comparison. Under the default prompt, evaluated on BioASQ 11b–13b ($N \approx 269$–270): All

Table 4
LLM comparison under the default prompt.

Model	Acc.	Macro-F1	F1-Yes	F1-No
Gemini 3.1 Pro	0.9519	**0.9481**	0.9621	0.9340
Perplexity Sonar-R-Pro	0.9481	0.9449	0.9583	0.9314
Claude Sonnet 4.6	0.9481	0.9446	0.9586	0.9307
Claude Opus 4.6	0.9442	0.9399	0.9560	0.9239
GPT-5.5 Pro	0.9222	0.9153	0.9395	0.8912

frontier models achieve Macro-F1 above 0.91. Gemini 3.1 Pro achieves the highest score (0.9481), while GPT-5.5 Pro trails substantially, particularly on the *no* class (F1-No $= 0.8912$). Under the best prompt of Opus, the same model set (excluding `GPT-5.5 Pro`) is re-evaluated on BioASQ 11b–13b ($N \approx 268$–270): Unlike the default-prompt condition — where Gemini 3.1 Pro led the ranking — the best

Table 5
LLM comparison under the best prompt (A2-5).

Model	Acc.	Macro-F1	F1-Yes	F1-No
Claude Opus 4.6	0.9552	**0.9523**	0.9641	0.9406
Claude Sonnet 4.6	0.9219	0.9189	0.9346	0.9032
Gemini 3.1 Pro	0.8889	0.8856	0.9051	0.8661
Perplexity Sonar-R-Pro	0.8741	0.8706	0.8917	0.8496

prompt generalises poorly across model families: `Claude Opus 4.6` retains its gains ($+0.0124$ over its own default), but all other models decline substantially. This suggests that the critical-reviewer framing and structured snippet-analysis template are calibrated to the reasoning style of `Claude Opus 4.6` and do not transfer to other architectures.

Temperature sensitivity. Temperatures $T \in \{0, 0.3, 0.7, 1.0\}$ all produce **identical** outputs (Macro-F1 $= 0.9399$), indicating that the model's yes/no predictions are fully deterministic given the same prompt and snippets.

Iterative prompt refinement. Development set: 11b + 12b ($N = 160$); held-out test sets: 13b and 9b. While each refinement round improves dev-set Macro-F1, the gains do not transfer: Dev-3 degrades

Table 6
Iterative prompt refinement results.

Version	Dev Macro-F1	FP	FN	13b Macro-F1	9b Macro-F1
Dev-1 (`baseline_prompt`)	0.9548	4	3	0.9137	**0.9779**
Dev-2 (+Rules 1–6)	0.9615	2	4	0.9155	**0.9779**
Dev-3 (+Rules 7–8)	**0.9743**	1	3	0.9132	0.9667

on the held-out 9b set ($0.9779 \rightarrow 0.9667$), indicating prompt overfitting. Consequently, we adopt `baseline_prompt` for all subsequent experiments and competition submissions.

Snippet re-ranking. Evaluated on BioASQ 12b + 13b ($N \approx 182$–184), `Claude Opus 4.6`. All re-ranking configurations underperform the unranked baseline. Truncation to Top-5 produces the largest degradation (-0.076). Displaying similarity scores in the prompt further harms performance.

Table 7
Snippet re-ranking ablation.

Embedding Model	Top-K	Macro-F1
No re-ranking (baseline)	all	**0.9340**
`all-MiniLM-L6-v2`	all	0.9281
`S-PubMedBERT-MS-MARCO`	all	0.9161
`all-MiniLM-L6-v2`	5	0.8584
`all-MiniLM-L6-v2` (show scores)	all	0.9161

In-context example retrieval. Evaluated on BioASQ 11b–13b ($N \approx 220$–222), `Claude Opus 4.6`. A balanced set of 10 examples (5 yes, 5 no) with question + snippet retrieval achieves the best result

Table 8
In-context example retrieval ablation.

Total K	Yes:No	Query	Macro-F1
0 (baseline)	—	—	0.9370
10	5:5	Q + Snippets	**0.9472**
10	5:5	Q only	0.9373
6	3:3	Q + Snippets	0.8705
16	8:8	Q + Snippets	0.7609
10	0:10	Q + Snippets	0.9153
10	10:0	Q + Snippets	0.7773

($+0.0102$ over baseline). Larger example sets and imbalanced label ratios both degrade performance substantially.

Majority voting. Evaluated on BioASQ 11b–13b ($N \approx 220$–222), `Claude Opus 4.6`. All voting

Table 9
Majority voting ablation.

Strategy	N	Temp.	Macro-F1	Agreement
Baseline	1	0	**0.9370**	—
Shuffle	3	0	0.9277	0.978
Shuffle	5	0	0.9230	0.973
Temperature	3	0.5	0.9327	0.991
Temperature	3	1.0	0.9327	0.991

configurations underperform the single-inference baseline. The agreement rate exceeds 97% in all settings, confirming that the model's responses are nearly deterministic.

Single-snippet voting. Evaluated on BioASQ 13b ($N = 82$); full-context baseline Macro-F1 $= 0.9513$. All single-snippet configurations fall below the full-context baseline, with Gemini dropping to 0.695, confirming that joint multi-snippet reasoning is essential.

Self-feedback. Evaluated on BioASQ 12b ($N \approx 102$); baseline Macro-F1 $= 0.9258$. All self-feedback variants degrade performance relative to the single-turn control. Although SF-2 corrects 3 of 4 changed predictions, the overall Macro-F1 declines because the critique phase introduces a systematic shift toward 'no' in the model's output distribution—reducing F1-Yes even on questions whose final label is unchanged—which disproportionately harms the yes-class F1 under Macro-averaging.. The devil's advocate (SF-3) actively harms performance by overriding correct initial answers.

Table 10
Single-snippet voting across models.

Model	Macro-F1	Agreement Rate
Full-context baseline	**0.9513**	—
Claude Opus 4.6	0.9059	0.842
Claude Sonnet 4.6	0.8343	0.622
Perplexity Sonar-R-Pro	0.8343	0.683
Gemini 3.1 Pro	0.6947	0.598

Table 11
Self-feedback ablation.

ID	Turn 1	Turn 2	Turn 3	Acc.	Macro-F1	Δ	→correct	→wrong
SF-1	Opus 4.6	— (none)	—	0.9307	**0.9258**	—	—	—
SF-2	Opus 4.6	Default checklist	Opus 4.6	0.9216	0.9177	-0.0081	$+3$	0
SF-3	Opus 4.6	Devil's advocate	Opus 4.6	0.8824	0.8786	-0.0472	$+2$	-3
SF-4	Sonnet 4.6	Model swap	Sonnet 4.6	0.8627	0.8573	-0.0685	$+4$	-6
SF-5	Opus 4.6	Cross-model (Sonnet)	Opus 4.6	0.8922	0.8874	-0.0384	$+2$	-1

Document enrichment. Evaluated on BioASQ 12b + 13b, `Claude Opus 4.6`. Adding full abstracts

Table 12
Document enrichment results.

Configuration	N	Macro-F1
Snippets only (baseline)	182	**0.9340**
Snippets + Abstracts	158	0.9230
Abstracts only	158	0.9097

dilutes rather than enriches the evidence context, and replacing snippets with abstracts entirely degrades performance further.

DSPy automated optimisation. Using `Claude Sonnet 4.6`, evaluated on a held-out test set ($N = 100$). H-1 optimisation produces no change (the instruction is already locally optimal). H-2

Table 13
DSPy MIPROv2 optimisation results.

Exp	Stage	Acc.	Macro-F1	F1-Yes	F1-No	P-Yes
H-1	baseline	0.950	0.9447	0.9618	0.9275	0.984
H-1	optimised	0.950	0.9447	0.9618	0.9275	0.984
H-2	baseline	0.940	0.9332	0.9545	0.9118	0.969
H-2	**optimised**	**0.960**	**0.9560**	**0.9692**	**0.9429**	**1.000**

optimisation yields the highest Macro-F1 (0.9560, $+0.0228$ over its baseline) with *perfect* P-Yes $= 1.0$.

5.2. Factoid Questions

Prompt strategy comparison. Table 14 traces MRR across our 13 prompt iterations. Three interventions produced the largest gains: (1) ranked output strategy (Exp 4, +0.05 MRR over single-answer baseline), (2) snippet reranking (Exp 11, +0.04 MRR), and (3) combined model upgrade and prompt refinement (Exp 12, +0.10 MRR). Conversely, attempts to impose multi-step extraction pipelines (Exp 5–7) consistently degraded performance, demonstrating a non-monotonic relationship between prompt

complexity and MRR. Frontier LLMs perform better with moderate guidance than with rigid procedural instructions.

Table 14
Summary of key prompt engineering iterations on BioASQ 10b–13b development data.

Exp	Key Change	MRR
1	Simple guidelines, single answer	$\sim$0.30
2	Trim-Aggressively, verbatim extraction	$\sim$0.38
3	Few-shot extraction boundary examples	$\sim$0.40
4	Ranked list output (1–5 candidates)	$\sim$0.45
5–7	Multi-layer / multi-step pipelines	0.33$-$0.44
8	Relaxed structure, competitor at Rank 5	$\sim$0.40
11	+ Snippet reranking (IDF-based)	$\sim$0.45
12	8-step prompt + model upgrade	$\sim$0.50
13	Overfitting fix, edition-specific rules	0.60$-$0.70

Model comparison. Table 15 reports factoid MRR and lenient accuracy for our three development models evaluated on BioASQ 13b Phase B across four batches. Claude Opus 4.6 achieved the highest

Table 15
Model comparison on BioASQ 13b Phase B development data (factoid).

Model	B1	B2	B3	B4	Avg MRR	Avg Len. Acc.
Claude Opus 4.6	0.667	0.795	0.603	0.775	**0.710**	**0.777**
Claude Sonnet 4.6	0.635	0.765	0.517	0.739	0.664	0.752
Gemini 2.5 Pro	0.556	0.735	0.583	0.748	0.655	0.758
Gemini 3.1 Pro Preview	0.583	0.756	0.558	0.777	0.669	0.823

average MRR (0.710) and lenient accuracy (0.777), outperforming the other models on three of four batches. Gemini 3.1 Pro Preview achieved the highest average lenient accuracy (0.823) across all four batches, suggesting strong answer coverage despite a lower MRR, and outperformed Gemini 2.5 Pro on average MRR (0.669 vs. 0.655). Notably, Gemini 2.5 Pro slightly outperformed Claude Sonnet 4.6 on Batch 3 (0.583 vs. 0.517) despite trailing overall, and Gemini 3.1 Pro Preview achieved the highest single-batch lenient accuracy on Batch 4 (0.777, tied with Claude Opus 4.6). The transition from GPT-4o-mini (Exp 11) to Claude Opus 4.6 (Exp 12) produced the single largest MRR gain ($\sim$0.10), confirming that model capability is at least as important as prompt design for this task. Both Claude models benefited more consistently from the structured prompt than Gemini models, suggesting that instruction-following capability is a key differentiator for the ranked-extraction formulation. These cross-model differences motivated our multi-model submission strategy of distributing different models across submission slots rather than committing to a single model.

Error analysis. Five primary error modes were identified and tracked throughout development. *Over-specification* was the most frequent, where the model appended unnecessary modifiers beyond the gold boundary (e.g., "exon 53 skipping" instead of "exon 53"); this was partially addressed by the verbatim trimming rules introduced in Exp 2 and reinforced in the Rank 1 extraction rules of Exp 12–13. *Paraphrasing mismatch* occurred when the model produced semantically correct but lexically different answers (e.g., "below 0.5%" vs. "less than 0.5%"); the ranked-list strategy mitigated this by covering multiple surface forms across five slots. *Wrong abstraction level* caused the model to return a specific sub-type when the gold expected a broader category, or vice versa; Rule L of the final prompt explicitly instructs the model to include both levels as candidates. *Hallucination on null cases* persisted across early experiments and was only reliably suppressed after Exp 12, where the internal-knowledge fallback

step was paired with an explicit instruction to output `["None"]` only when no snippet evidence exists. *Under-specification* arose when aggressive trimming removed tokens that were part of the gold answer (e.g., returning "BRCA1" when the gold was "BRCA1 gene"); this was balanced by the surface-form variant strategy in Exp 13, which generates both the minimal entity and slightly longer noun-phrase variants.

Competition results. Combining all optimization steps—ranked output, snippet reranking, structured prompt, and model upgrade—yielded a cumulative gain of approximately $+0.30$ MRR over the Experiment 1 baseline (from $\sim$0.30 to 0.60–0.70 on the 13b validation set). On the official BioASQ 14b leaderboard, our system placed **1st** in both Batch 3 (MRR 0.5882) and Batch 4 (MRR 0.5530) among approximately 70–80 participating systems, and achieved top-5 performance in Batch 1 (MRR 0.4891, tied 5th). Batch 2 was comparatively weaker (MRR 0.3917, 9th), likely due to a higher proportion of numeric and null-answer questions that remained challenging despite our targeted rules.

5.3. List Questions

Table 16
List questions: snippet processing results on BioASQ 11b (batch 4, $N = 24$). ΔF1 is relative to the Raw baseline.

Snippet Mode	Precision	Recall	F1	ΔF1
Raw (baseline)	0.73	0.74	0.72	—
BM25 Top-5	0.53	0.52	0.51	-0.21
BM25 Top-10	0.56	0.59	0.56	-0.16
BM25 Top-20	0.55	0.57	0.54	-0.18
Processed	**0.74**	**0.75**	**0.73**	$+0.01$

Snippet processing results. BM25 re-ranking severely degrades performance (F1 drop of 16–21%). BM25 relies on lexical overlap, but biomedical text is rich in synonymous terminology: a question about "heart attack" may be answered by a snippet discussing "myocardial infarction." BM25 filters out these semantically relevant but lexically dissimilar snippets. The improvement from Raw to Processed (ΔF1 $= +0.01$) is marginal and likely within sampling variability at this batch size ($N = 24$); however, the BM25 degradation (F1 drops of 0.16–0.21) is substantively large even for small N.

Table 17
List questions: model comparison on BioASQ 11b (batch 4, $N = 24$).

Model	Precision	Recall	F1
GPT-4	0.74	0.68	0.70
Gemini 3 Pro	0.69	0.75	0.70
Claude Opus 4.6	**0.73**	**0.74**	**0.72**

Model comparison. Claude Opus 4.6 achieved the highest F1 (0.72) with the most balanced Precision/Recall profile, which is critical for list questions where both omission and over-generation are penalized equally. The F1 differences among models are small (0.02) and should be interpreted cautiously given the limited batch size ($N = 24$).

Prompt strategy comparison. Table 17 compares five prompt strategies across three independent BioASQ training batches for list questions. Structured output achieved the highest mean F1 (0.623) and ranked first overall. Although Chain-of-Thought attained the highest individual batch score (0.77 on 11b Batch 4), its performance varied considerably across batches (0.51 on 12b Batch 2, 0.54 on 13b Batch 3), resulting in a lower mean F1 of 0.607. In contrast, Structured output maintained more

Table 18
List questions: prompt strategy comparison across three BioASQ training batches (N: 11b batch 4 $= 24$, 12b batch 2 $= 18$, 13b batch 3 $= 22$).

Prompt	11b batch 4 F1	12b batch 2 F1	13b batch 3 F1	Mean	Rank
Zero-shot	0.68	0.46	0.55	0.563	4
Few-shot	0.75	0.49	0.58	0.607	2
Structured	0.72	**0.53**	**0.62**	**0.623**	**1**
Role Expert	0.70	0.47	0.53	0.567	5
Chain-of-Thought	**0.77**	0.51	0.54	0.607	2

consistent performance across all three batches (0.72, 0.53, 0.62), yielding the lowest variance (±0.078). In a competition setting where a single prompt must generalize to unseen questions from different distributions, such stability is arguably more valuable than peak performance on any individual batch.

Table 19
List questions: automated prompt optimization iterations on BioASQ 11b (batch 4, $N = 24$). Results are from a post-competition replication of the optimization procedure (see Section 4.4).

Iter	Key Modification	Prec.	Recall	F1
1	Baseline structured prompt	0.7238	0.7245	0.7129
2	Added verification step	0.7239	0.7384	0.7168
3	Explicit exclusion rules	0.7472	0.7523	0.7384
4	6-step workflow, 85% threshold	0.7546	0.7939	0.7598
5	**90% confidence, pruning**	**0.7696**	**0.7915**	**0.7653**
6	Over-constrained (declined)	0.7552	0.7776	0.7551

Automated optimization results. Table 19 illustrates a systematic progression from a structured output baseline to an optimized state based on official BioASQ metrics, obtained through a post-competition replication of the optimization procedure described in Section 4.4. While the replication produced a different final prompt than the one used during competition, the trajectory confirms the method's effectiveness: the same optimization procedure, applied to the same training data with the same model and evaluation service, yields substantial and consistent F1 gains. In **Iteration 2**, the introduction of a verification step primarily improved Recall (+0.0139) with negligible impact on Precision, demonstrating that answer validation effectively recovers missed entities. **Iteration 3** incorporated explicit exclusion rules, yielding simultaneous gains in both Precision (+0.0233) and Recall (+0.0139), which underlines that defining what to exclude is as vital to prompt engineering as specifying what to include. Restructuring the prompt into a 6-step workflow with an 85% confidence threshold in **Iteration 4** forced an exhaustive snippet examination, resulting in the largest single-iteration Recall boost (+0.0416). **Iteration 5** then focused on refining Precision (+0.0150) by raising the confidence threshold to 90% and introducing a post-generation pruning mechanism; this strategy incurred only a marginal Recall loss (0.0024) and successfully achieved the peak F1 score of 0.7653. Finally, **Iteration 6** over-constrained the prompt, causing both metrics to decline and thereby marking a natural convergence boundary beyond which further restrictions become counterproductive. Overall, this automated loop shifted dynamically from prioritizing Recall via broader scanning to sharpening Precision via stricter filtering, ultimately driving a 7.35% relative F1 gain (from 0.7129 to 0.7653) and producing a prompt configuration with a peak F1 of 0.7653.

5.4. Official Competition Results (BioASQ 14b)

Across all four batches our team **CSA-IISR** achieved first place on all four batches for list questions, first or second place across all batches for factoid questions, and first place in one batch for yes/no

questions.

Yes/No. For the yes/no subtask across all four batches of BioASQ 14b Phase B, we submitted five system variants per batch, exploring combinations of backbone model and prompting strategy. The three main models used were `Claude Opus 4.6`, `Gemini 3.1 Pro`, and `GPT-5.4/5.5 Pro`, all prompted with the default chain-of-thought template unless stated otherwise. Two additional variants per batch incorporated in-context retrieved examples: one with ten all-*no* demonstrations (10no/10) and one with a balanced five-*no* out of ten (5no/10), both retrieved via a Chroma vector index. In Batches 1 and 2, the Gemini-based default-prompt submission achieved the highest yes/no Macro-F1 among our five variants (0.9377 and 0.8929, respectively). Batch 3 was the strongest result overall: three of our five submissions — Gemini default, Claude default, and Claude with a conflict-aware rule — all attained a perfect Macro-F1 of **1.0000**. In Batch 4, the in-context 10no/10 variant (`Claude Opus 4.6`) recorded the best yes/no Macro-F1 of our submissions at 0.9262, outperforming the single-model default runs. All yes/no submissions were generated fully automatically; no human review or manual correction was applied to model outputs prior to submission.

Table 19
CSA-IISR submission results on BioASQ 14b Phase B (yes/no).

Batch	Sub.	Model	Configuration	Acc.	F1-Yes	F1-No	Macro-F1
1	1st	Gemini 3.1 Pro	Default CoT	0.9412	0.9524	0.9231	**0.9377**
1	2nd	Claude Opus 4.6	Default CoT	0.8824	0.9000	0.8571	0.8786
1	3rd	GPT-5.4 Pro	Default CoT	0.8824	0.9000	0.8571	0.8786
1	4th	Claude Opus 4.6	In-context 9 examples	0.8824	0.9000	0.8571	0.8786
1	5th	Gemini 3.1 Pro	In-context 9 examples	0.8824	0.8889	0.8750	0.8819
2	1st	Gemini 3.1 Pro	Default CoT	0.9048	0.9286	0.8571	**0.8929**
2	2nd	Claude Opus 4.6	Default CoT	0.9048	0.9286	0.8571	**0.8929**
2	3rd	GPT-5.4 Pro	Default CoT	0.9048	0.9286	0.8571	**0.8929**
2	4th	Claude Opus 4.6	In-context 10no/10	0.9048	0.9286	0.8571	**0.8929**
2	5th	Claude Opus 4.6	In-context 5no/10	0.9048	0.9286	0.8571	**0.8929**
3	1st	Gemini 3.1 Pro	Default CoT	1.0000	1.0000	1.0000	**1.0000**
3	2nd	Claude Opus 4.6	Default CoT	1.0000	1.0000	1.0000	**1.0000**
3	3rd	GPT-5.4 Pro	Default CoT	0.9091	0.9333	0.8571	0.8952
3	4th	Claude Opus 4.6	In-context 10no/10	0.9091	0.9333	0.8571	0.8952
3	5th	Claude Opus 4.6	Default + conflict rule	1.0000	1.0000	1.0000	**1.0000**
4	1st	Gemini 3.1 Pro	Default CoT	0.8667	0.9048	0.8000	0.8524
4	2nd	Claude Opus 4.6	Default CoT	0.8667	0.9048	0.8000	0.8524
4	3rd	GPT-5.5 Pro	Default CoT	0.8000	0.8750	0.6667	0.7708
4	4th	Claude Opus 4.6	In-context 10no/10	**0.9333**	**0.9524**	**0.9000**	**0.9262**
4	5th	Perplexity Sonar-R-Pro	Default CoT	0.8000	0.8571	0.7273	0.7922

Factoid. Table 20 presents all CSA-IISR factoid results in BioASQ 14b Phase B. Rankings use *dense ranking* (ties share the same rank; the next distinct score receives the next integer). Our system ranked 1st in Batches 1 (tied) and 3, and 2nd in Batches 2 and 4, among 71–82 systems per batch. In Batch 3, CSA-IISR 5th achieved MRR 0.5882—the highest single-batch score across all systems. In Batch 4, CSA-IISR 1st reached MRR 0.5530, ranking second behind ku_dmis_3/5 (0.5758). In Batch 2, our best run ranked second behind MedQA-1 (0.4000); the lower absolute scores system-wide suggest higher question difficulty for this batch. The best-performing run varied by batch (2nd in Batch 1; 5th in Batches 2 and 3; 1st in Batch 4), confirming that model diversity is valuable when question distributions are unpredictable. Table 20 summarises the best run per batch.

Table 20: CSA-IISR factoid results, all submissions, BioASQ 14b Phase B (dense ranking by MRR).

Batch (#sys)	Submission	Rank	MRR	Len. Acc.	Str. Acc.
Batch 1 (71)	CSA-IISR 2nd	**1** (tie)	0.5217	0.5217	0.5217
	CSA-IISR 1st	3 (tie)	0.4891	0.5652	0.4348
	CSA-IISR 3rd	3 (tie)	0.4891	0.5652	0.4348
	CSA-IISR 4th	10	0.4275	0.4783	0.3913
	CSA-IISR 5th	11	0.4203	0.4783	0.3913
Batch 2 (82)	CSA-IISR 5th	2 (tie)	0.3917	0.4500	0.3500
	CSA-IISR 1st	3 (tie)	0.3750	0.4000	0.3500
	CSA-IISR 2nd	3 (tie)	0.3750	0.4000	0.3500
	CSA-IISR 3rd	3 (tie)	0.3750	0.4000	0.3500
	CSA-IISR 4th	3 (tie)	0.3750	0.4000	0.3500
Batch 3 (75)	CSA-IISR 5th	**1**	0.5882	0.6471	0.5294
	CSA-IISR 1st	4 (tie)	0.5118	0.5882	0.4706
	CSA-IISR 2nd	4 (tie)	0.5118	0.5882	0.4706
	CSA-IISR 3rd	6	0.4902	0.5882	0.4118
	CSA-IISR 4th	7	0.4706	0.5294	0.4118
Batch 4 (81)	CSA-IISR 1st	2	0.5530	0.7273	0.4545
	CSA-IISR 2nd	4 (tie)	0.5227	0.6364	0.4545
	CSA-IISR 3rd	4 (tie)	0.5227	0.6364	0.4545
	CSA-IISR 4th	5	0.5076	0.7273	0.3636
	CSA-IISR 5th	—	—	—	—

Table 21
List questions: official **BioASQ 14b** competition results. N: number of list questions per batch.

Batch	system	Prec.	Recall	F1	Rank	N	2nd-place F1
14b B1	CSA-IISR 3rd	0.3675	0.3678	0.3613	**1st**	21	0.3204
14b B2	CSA-IISR 3rd	0.4564	0.4998	0.4720	**1st**	13	0.4628
14b B3	CSA-IISR 3rd	0.4930	0.5617	0.5175	**1st**	17	0.5031
14b B4	CSA-IISR 4th	0.6158	0.7274	0.6543	**1st**	20	0.6427

List. Table 21 presents the official BioASQ 14b competition results for list questions. Our system (CSA-IISR) ranked first across all four batches. The margins over the second-place system ranged from 0.0092 (Batch 2, $N = 13$) to 0.0409 (Batch 1, $N = 21$). Given the small batch sizes ($N = 13$–21), these margins should be interpreted cautiously; however, the consistency of first-place rankings across all four independent batches—each comprising a distinct set of questions—provides stronger evidence than any single-batch result alone.

Notably, the prompt used during competition was optimized on BioASQ 11b training data using the same feedback-driven procedure described in Section 4.4 and applied to the 14b competition without modification. The fact that a separately conducted post-competition replication of the same optimization procedure yielded a different but comparably effective prompt (Table 19) further supports the robustness of the method.

6. Analysis and Discussion

6.1. Yes/No Questions

Output format does not affect accuracy. `baseline_prompt` and `baseline_yesno_only` yield identical Macro-F1 (0.9399), showing that suppressing chain-of-thought output has no accuracy cost. We interpret this as frontier LLMs performing implicit evidence assessment regardless of output format, not as evidence that chain-of-thought reasoning is unhelpful. `baseline_yesno_only` is used for competition submissions for efficiency, while `baseline_prompt` is retained for error analysis.

Frontier models converge but differ in error profile. Four of five models achieve Macro-F1 within 0.93–0.95, suggesting performance convergence. Error profiles differ: Claude Opus 4.6 over-affirms ($R^{yes} = 0.976$, $R^{no} = 0.892$), while Perplexity Sonar is balanced ($P = R = 0.958$ for yes; $P = R = 0.931$ for no). GPT-5.5 Pro trails all others with FP $= 16$.

Systematic yes-bias. LLMs consistently over-predict "yes": Claude Opus 4.6 produces FP $= 11$ vs. FN $= 4$; GPT-5.5 Pro produces FP $= 16$ vs. FN $= 5$. The critical-reviewer prompt (A2-5) reduces FP to 7–8 without inflating FN. Balanced in-context examples (5 no demonstrations) reduce FP further. A conservative system prompt reduces both FP and FN non-selectively and is less effective. Epistemic framing ("require direct evidence") outperforms blanket conservatism for threshold calibration.

Prompt complexity and overfitting. Iterative rule addition improves dev-set Macro-F1 steadily (Dev-3: 0.9743) but degrades on held-out 9b (0.9779 $\rightarrow$ 0.9667). With only 7 errors in the Dev-2 set, each added rule targets 1–2 specific cases rather than encoding general reasoning principles. We term this *prompt-level overfitting*. The default baseline prompt, with no explicit rules, generalises more robustly and is used for all competition submissions.

Temperature insensitivity. Temperatures $T \in \{0, 0.3, 0.7, 1.0\}$ yield byte-identical outputs across all 269 questions. Majority-voting agreement rates exceed 97%, confirming near-deterministic behaviour. The model's logit distribution is sharply peaked for binary classification over curated evidence, leaving no variance for ensemble methods to exploit.

Evidence completeness outweighs relevance ranking. Snippet re-ranking and truncation uniformly degrade performance. Top-5 truncation drops Macro-F1 by -0.076; keeping all snippets but reordering loses -0.006 to -0.018. Gold snippets are likely to have been already relevance-curated; further filtering by embedding similarity creates a relevance–accuracy mismatch. Displaying similarity scores in the prompt adds an additional -0.012 drop, as the model anchors on scores over holistic evidence.

In-context examples: effective but fragile. Balanced retrieval (5 yes, 5 no; K $= 10$) improves Macro-F1 by $+0.0102$, the largest prompt-level gain. The method is fragile: all-yes examples (10:0) collapse Macro-F1 to 0.7773 and all-no (0:10) to 0.9153. Increasing to K $= 16$ collapses performance to 0.7609 via context dilution. Label balance and example count must both be carefully controlled.

Self-critique overcorrects. The standard critique (SF-2) corrects 3 of 4 changed predictions, yet overall Macro-F1 drops (0.9258 $\rightarrow$ 0.9177) because the critique shifts the model's distribution toward conservatism, elevating false negatives even without changing the final label. The devil's advocate critique (SF-3) worsens this to 0.8786. When the base model already exceeds 93% accuracy, critique-induced perturbations are *a priori* more likely to corrupt correct answers than to fix incorrect ones.

Joint reasoning outperforms decomposition. Per-snippet voting scores 0.9059 (best) vs. 0.9513 for full-context inference. Low per-snippet agreement rates (Opus: 0.842; Gemini: 0.598) confirm that individual snippets are frequently ambiguous; cross-snippet synthesis is essential. Document enrichment shows the same pattern: adding full abstracts dilutes rather than enriches the evidence context.

Automated optimisation: demonstrations over instructions. H-1 (instruction-only MIPROv2) produces zero gain; the initial instruction is already locally optimal and no rephrasing improves it. H-2 (Predict + 4 demonstrations) gains $+0.0228$ Macro-F1 and achieves $P^{yes} = 1.000$ after optimisation. Demonstrations calibrate the model's decision threshold through exemplars rather than verbal rules, providing qualitatively different guidance than instruction text alone.

6.2. Factoid Questions

Why over-structured prompts failed. Experiments 5–7 introduced multi-layer extraction logic and seven-step pipelines in an attempt to improve answer precision. However, MRR dropped by up to 0.12 relative to Exp 4, the ranked output baseline. Frontier LLMs interpret overly prescriptive instructions as conflicting constraints, leading to inconsistent extraction behaviour. This finding implies that for exact-answer tasks, moderate structure outperforms rigid procedural guidance.

Ranked output vs. single answer. Transitioning from single-answer to a ranked list of 1–5 candidates (Exp 4) produced an immediate gain of $+0.05$ MRR with no change in model or retrieval strategy. Placing surface-form variants and boundary alternatives in lower ranks effectively functions as a soft ensemble, increasing the probability that at least one candidate matches the gold answer. For competition settings evaluated by MRR, this output format design decision alone accounts for a substantial portion of the performance gain.

Lightweight reranking vs. full RAG. IDF-based snippet reranking (Exp 11) improved MRR by $+0.05$ over the no-reranking baseline. Rather than augmenting the provided gold-standard snippets with additional PubMed passages—an approach that prior work has shown to introduce noise rather than useful signal when evidence is already curated [26]—we adopted a lightweight reranking strategy that operates exclusively within the provided snippet set. By scoring each snippet via IDF-weighted token overlap with additive bigram (+3.0) and phrase-match (+2.0) bonuses, the reranker prioritizes the most question-relevant snippets without discarding any evidence, avoiding the contradictory context that full retrieval augmentation risks introducing. This suggests that when gold-standard evidence is already curated, lightweight in-set reranking is preferable to full retrieval pipelines.

Model capability as a performance ceiling. Upgrading from GPT-4o-mini to Claude Opus 4.6 (Exp 12) produced the single largest gain across all interventions ($+0.10$ MRR), larger than any prompt or retrieval change. Among development models, Claude Opus 4.6 led with average MRR 0.710, followed by Gemini 3.1 Pro Preview (0.669), Claude Sonnet 4.6 (0.664), and Gemini 2.5 Pro (0.655). Notably, Gemini 3.1 Pro Preview achieved the highest average lenient accuracy (0.823), indicating that despite ranking third in MRR it retrieved the correct answer within five candidates more consistently than any other model—a distinction that is useful in practice when ranking can be separately improved. The performance gap was most pronounced on harder batches, suggesting that model capacity acts as a ceiling on what prompt engineering alone can achieve.

Prompt optimization insights. The prompt evolution across 13 iterations reveals a consistent pattern: early iterations focus on extraction boundary precision, while later iterations address surface-form normalisation and question-type specialisation. Three effective refinements emerged: (1) converting flat rule lists into type-specific numbered procedures improved handling of edge cases such as fragment

questions and binary alternatives, (2) specifying what *not* to extract (e.g., surrounding context words, determiners) was as important as specifying what to extract, and (3) adding British spelling priority at Rank 1 reduced paraphrasing mismatch against BioASQ gold answers. The performance plateau at Exp 13 and the degradation observed when further rules were added suggest that automated prompt optimization has a natural stopping point beyond which additional constraints introduce ambiguity.

6.3. List Questions

Why BM25 re-ranking failed. BM25 keyword matching cannot capture semantic equivalence in biomedical terminology. When a question uses one term (e.g., "heart attack") but the relevant snippet uses a synonym ("myocardial infarction"), BM25 assigns a low score and the snippet is filtered out. This finding implies that when snippets are already curated by domain experts, additional surface-level filtering removes useful information rather than reducing noise.

Structured output vs. chain-of-thought. CoT achieved the highest single-batch F1 (0.77 on 11b) for list questions but suffered from high cross-batch variance ($\pm$0.117). Structured output provided a more reliable performance floor. For competition settings where a single prompt must handle unknown question distributions, stability is more valuable than peak performance.

Prompt optimization insights. The prompt evolution across iterations reveals a consistent pattern: early iterations focus on improving recall, while later iterations shift to precision. Three effective patterns emerged: (1) converting flat rule lists into numbered step-by-step procedures (+2.14% F1), (2) specifying what *not* to extract was as important as specifying what to extract (+2.16% F1), and (3) replacing vague "high confidence" with specific thresholds (85%$\rightarrow$90%). The convergence at Iteration 5 and decline at Iteration 6 demonstrate that automated optimization has a natural stopping point.

Statistical considerations. All list-question development experiments were conducted on small batches ($N = 13$–24). At these sample sizes, individual F1 differences of 0.01–0.02 (e.g., Raw vs. Processed snippets, or Claude vs. GPT-4/Gemini) are within the range of sampling variability and should not be over-interpreted. We do not report confidence intervals because per-question prediction logs were not retained; the aggregate scores reported are those returned by the official BioASQ evaluation service. The BM25 degradation (F1 drops of 0.16–0.21) and the automated optimization gains (+7.35% F1 across six iterations) are large enough to be substantively meaningful even at small N. For the official competition results, the consistency of first-place rankings across four independent batches provides collective evidence that is more informative than any single-batch margin.

6.4. Cross-Type Findings

Three findings recur across all three question types. First, interventions that calibrate the decision threshold (critical-reviewer framing, balanced demonstrations, DSPy demonstrations, confidence thresholds) consistently outperform those that alter the reasoning process (self-critique, re-ranking, majority voting). Second, over-engineered pipelines—BM25 re-ranking, self-feedback, and multi-step extraction pipelines—degraded performance relative to their respective baselines, because when gold-standard evidence is already curated, additional filtering or procedural rigidity removes useful signal rather than reducing noise. Third, model capability sets a hard ceiling that prompt engineering alone cannot overcome; the largest single gain in all three subtasks came from upgrading the backbone model.

6.5. Limitations

1. **Small batch size.** Competition batches contain only $N \approx 13$–24 questions, limiting statistical power. Small F1 differences in ablation tables should be interpreted cautiously; for the yes/no ablations in particular (Tables 3–9), per-question significance testing was not performed, so reported Macro-F1 differences should be read as indicative trends rather than confirmed effects.

2. **Gold snippets only.** Performance under noisy or automatically retrieved snippets may differ.
3. **Development data overlap.** Prompt refinement and in-context retrieval experiments share training data with the dev set; potential leakage is partially mitigated by held-out evaluation.
4. **TextGrad incomplete.** The TextGrad pipeline was built but not evaluated due to time and cost constraints.
5. **Post-competition replication.** The list-question prompt optimization ablation (Table 19) was replicated after the competition because per-iteration logs from the original optimization run were not preserved. While the replication used the identical method, data, model, and evaluation service, the stochastic nature of LLM-based prompt generation means the resulting prompt differs from the competition prompt. Both prompts are provided in Appendix C for transparency.

7. Conclusion

We presented a training-free biomedical QA system for BioASQ Task 14b Phase B that relies exclusively on prompt engineering and automated optimization across frontier LLMs. Our system achieved first place on all four batches for list questions, first or second place across all batches for factoid questions, and first place in one batch for yes/no questions. Three findings generalize across question types. First, a feedback-driven prompt optimization loop guided by official evaluation scores—rather than proxy metrics—yields consistent, transferable gains: the list prompt improved F1 by 7.35% on held-out training data and generalized without modification to all four competition batches. Second, over-engineered pipelines consistently underperform simpler, well-calibrated prompts: BM25 re-ranking, self-feedback, and multi-step extraction pipelines all degraded performance relative to their respective baselines. Third, model capability sets a hard ceiling that prompt engineering alone cannot overcome; the largest single gain in all three subtasks came from upgrading the backbone model. These results suggest that systematic, metric-driven prompt optimization is a viable and scalable alternative to fine-tuning and complex RAG architectures for biomedical QA, when gold-standard evidence is already provided, as in BioASQ Phase B. Whether these gains extend to end-to-end settings where retrieval quality is part of the challenge remains an open question.

Acknowledgments

We thank the BioASQ organizers for providing the evaluation infrastructure that made our automated prompt optimization approach possible.

Declaration on Generative AI

This work investigates large language models (LLMs) as the primary research subject; all LLM usage for question answering, automated prompt optimization, and error analysis is described in Sections 4–5. During manuscript preparation, an LLM-based assistant (Claude, Anthropic) was used for editing and proofreading. All authors reviewed and take full responsibility for the content of this paper.

References

[1] A. Nentidis, G. Paliouras, et al., "Overview of BioASQ Tasks," in *CLEF Working Notes*, 2024.

[2] Y.-S. Chu, C.-Y. Hou, and C.-H. Hsu, "CSA-IISR BioASQ System — Prompts and Code," GitHub repository, 2026. Available: https://github.com/yensungchiu/BioASQ

[3] G. Tsatsaronis, G. Balikas, P. Malakasiotis, I. Partalas, M. Zschunke, M. R. Alvers, D. Weissenborn, A. Krithara, S. Petridis, D. Polychronopoulos, et al., "An overview of the BioASQ large-scale biomedical semantic indexing and question answering competition," *BMC Bioinformatics*, vol. 16, pp. 1–28, 2015.

[4] A. Nentidis, G. Katsimpras, A. Krithara, G. Paliouras, "Overview of BioASQ Tasks 13b and Synergy13 in CLEF2025," in *CLEF 2025 Working Notes*, G. Faggioli, N. Ferro, P. Rosso, D. Spina (Eds.), 2025.
[5] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W. Yih, T. Rocktäschel, S. Riedel, and D. Kiela, "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, pp. 9459–9474, 2020.
[6] A. Nentidis et al., "Overview of BioASQ 2024 — Task 12b," in *CLEF Working Notes*, 2024.
[7] H. Kim, H. Lee, Y. Cho, J. Park, J. Park, S. Park, Y. T. Chok, S. Baek, D. Lee, and J. Kang, "Prompting matters: Snippet-aware strategies for biomedical QA with LLMs in BioASQ 13b," in *CLEF 2025 Working Notes*, 2025.
[8] D. Stachura, J. Konieczna, and A. Nowak, "Are smaller open-weight LLMs closing the gap to proprietary models for biomedical question answering?" in *CLEF 2025 Working Notes*, 2025.
[9] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, and J. Kang, "BioBERT: A pre-trained biomedical language representation model for biomedical text mining," *Bioinformatics*, vol. 36, pp. 1234–1240, 2020.
[10] Y. Gu, R. Tinn, H. Cheng, M. Lucas, N. Usuyama, X. Liu, T. Naumann, J. Gao, and H. Poon, "Domain-specific language model pretraining for biomedical natural language processing," *arXiv preprint arXiv:2007.15779*, 2020.
[11] S. Ateia and U. Kruschwitz, "Can language models critique themselves? Investigating self-feedback for retrieval augmented generation at BioASQ 2025," in *CLEF 2025 Working Notes*, 2025.
[12] O. Khattab et al., "DSPy: Compiling declarative language model calls into self-improving pipelines," *arXiv preprint arXiv:2310.03714*, 2023.
[13] C. Yang et al., "Large language models as optimizers," in *ICLR*, 2024.
[14] B.-C. Chih, J.-C. Han, and R. T.-H. Tsai, "NCU-IISR: Enhancing biomedical question answering with GPT-4 and retrieval augmented generation in BioASQ 12b Phase B," in *CLEF Working Notes*, vol. 3740, pp. 99–105, 2024.
[15] B.-C. Chih, J.-C. Han, H.-C. Hung, and R. T.-H. Tsai, "NCU-IISR: Biomedical question answering via Gemini and GPT APIs in the BioASQ 13b Phase B challenge," in *CLEF 2025 Working Notes*, 2025.
[16] A. Nentidis et al., "Overview of BioASQ 2023 — Task 11b," in *CLEF Working Notes*, 2023.
[17] A. Krithara, A. Nentidis, K. Bougiatiotis, and G. Paliouras, "BioASQ-QA: A manually curated corpus for biomedical question answering," *Scientific Data*, vol. 10, p. 170, 2023.
[18] F. Borazio, A. Shcherbakov, D. Croce, and R. Basili, "UniTor at BioASQ 2025: Modular biomedical QA with synthetic snippets and multiple task answer generation," in *CLEF 2025 Working Notes*, 2025.
[19] P. Vachharajani, "Exploring retrieval-reranking and LLM-based answer generation for biomedical QA," in *CLEF 2025 Working Notes*, 2025.
[20] H. Nori, N. King, S. M. McKinney, D. Carignan, and E. Horvitz, "Capabilities of GPT-4 on medical challenge problems," *arXiv preprint arXiv:2303.13375*, 2023.
[21] T. Brown et al., "Language models are few-shot learners," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 1877–1901.
[22] J. Wei et al., "Chain-of-thought prompting elicits reasoning in large language models," in *Advances in Neural Information Processing Systems*, vol. 35, 2022.
[23] Y. Zhou et al., "Large language models are human-level prompt engineers," in *ICLR*, 2023.
[24] K. Guu, K. Lee, Z. Tung, P. Pasupat, and M.-W. Chang, "REALM: Retrieval-Augmented Language Model Pre-Training," in *Proceedings of the 37th International Conference on Machine Learning (ICML)*, 2020.
[25] OpenRouter, "OpenRouter: A unified API gateway for large language models," 2024. Available: https://openrouter.ai
[26] S. Rosenthal, A. Sil, R. Florian, and S. Roukos, "CLAPnq: Cohesive Long-form Answers from Passages in Natural Questions for RAG Systems," *Transactions of the Association for Computational Linguistics*, vol. 13, pp. 53–72, 2025.

[27] 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, and G. Paliouras, "Overview of BioASQ 2026: The fourteenth BioASQ Challenge on Large-Scale Biomedical Semantic Indexing and Question Answering," in *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.

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

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

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

[31] D. Dimitriadis, V. Patsiou, E. Stoikopoulou, A. Toumpas, A. Bekiaridou, A. Samaras, G. Giannakoulas, and G. Tsoumakas, "Overview of ElCardioCC Task on Clinical Coding in Cardiology at BioASQ 2026," in *CLEF 2026 Working Notes*, E. Sánchez Salido, A. Barrón-Cedeño, A. García Seco de Herrera, S. MacAvaney, and J. M. Struß (Eds.), 2026.

[32] M. Martinelli, V. Bonato, G. M. Di Nunzio, G. Faggioli, N. Ferro, O. Irrera, B. Kantz, S. Marchesin, L. Menotti, S. Merlo, R. Michelotto, G. Tussardi, F. Vezzani, P. Waldert, and G. Silvello, "Overview of GutBrainIE@CLEF 2026: Gut-Brain Interplay Information Extraction," in *CLEF 2026 Working Notes*, E. Sánchez Salido, A. Barrón-Cedeño, A. García Seco de Herrera, S. MacAvaney, and J. M. Struß (Eds.), 2026.

[33] A. Krithara, A. Nentidis, K. Bougiatiotis, and G. Paliouras, "BioASQ-QA: A manually curated corpus for Biomedical Question Answering," *Scientific Data*, vol. 10, no. 1, p. 170, 2023, publisher: Nature Publishing Group UK London.

A. Yes/No Question Prompts

This appendix provides the full text of the three most representative yes/no prompts. `O-baseline` and `O-yesno_only` are the two baseline output formats compared in Table 2 and used throughout Section 4.2; `O-baseline` is also the prompt adopted for all competition submissions after the iterative-refinement ablation (Table 6) showed that rule-augmented variants overfit the development set. A2-5 is the best-performing system/user combination identified by the prompt engineering grid in Table 3 (Macro-F1 = 0.9523). The remaining prompt variants (A2-1–A2-4, the Dev-2/Dev-3 rule sets, and the self-feedback critique prompts SF-2/SF-3/SF-4/SF-5) are provided in full in the code repository; we summarize their designs briefly below to keep this appendix concise.

O-baseline — System Message. Used as the chain-of-thought baseline in Table 2 and adopted for all competition submissions.

```
You are an expert biomedical AI assistant. Your task is to answer clinical and biological Yes/No
    questions by reasoning step-by-step based strictly on the provided literature snippets. End your
    response with a new line containing exactly and only the word: yes or no.
```

O-baseline — User Message. Used as the chain-of-thought baseline in Table 2 and adopted for all competition submissions.

```
### INSTRUCTIONS:
1. Read the question and snippets carefully.
2. Write down your step-by-step reasoning on how the snippets answer the question.
3. Do not use outside knowledge.
4. End your response with a new line containing exactly and only the word: yes or no

### QUESTION:
{q_body}

### SNIPPETS:
{snippets_block}

### REASONING AND EXACT ANSWER:
```

O-yesno_only — System Message. The direct-output variant compared against O-baseline in Table 2; used for large-scale inference due to lower token cost and latency.

```
You are an expert biomedical AI assistant. Your task is to answer clinical and biological Yes/No
    questions based strictly on the provided literature snippets. Your response must consist of exactly
     and only the single word "yes" or "no". Do not include any reasoning, explanation, punctuation, or
     extra characters.
```

O-yesno_only — User Message. The direct-output variant compared against O-baseline in Table 2; used for large-scale inference due to lower token cost and latency.

```
### INSTRUCTIONS:
1. Read the question and snippets carefully.
2. Determine the answer based strictly on the provided literature snippets.
3. Do not use outside knowledge.
4. Your response must contain EXPLICITLY and ONLY the word "yes" or "no". Do not include any other text,
     punctuation, reasoning, or explanation.

### QUESTION:
{q_body}

### SNIPPETS:
{snippets_block}

### ANSWER:
```

A2-5 (Best Prompt) — System Message. A2-5 pairs a critical-reviewer system instruction with a structured snippet-analysis user template; it achieved the highest Macro-F1 (0.9523) among the five system/user combinations tested in Table 3, and is the prompt referred to as "best prompt of Opus" in Table 5.

```
You are a critical biomedical reviewer. Answer Yes/No questions only when the provided snippets contain
    direct and unambiguous evidence. If snippets conflict with each other, are only tangentially
    related to the question, or do not clearly support a yes answer, answer no. Your response must
    consist of exactly and only the single word "yes" or "no". Do not include any reasoning,
    explanation, punctuation, or extra characters.
```

A2-5 (Best Prompt) — User Message. Pairs with the critical-reviewer system message above to form the full A2-5 configuration.

```
### INSTRUCTIONS:
1. Read the question and each numbered snippet carefully.
2. For each relevant snippet, note its number [N] and what evidence it provides.
3. Based only on the snippets, reason toward your final answer.
4. Do not use outside knowledge.
5. Your response must consist of exactly and only the single word "yes" or "no". Do not include any
    reasoning, explanation, punctuation, or extra characters.

### QUESTION:
```

```
{q_body}

### SNIPPETS:
{snippets_block}

### SNIPPET ANALYSIS AND ANSWER:
```

Other Prompt Variants (Summary). For brevity, the remaining ablated prompts are summarized here rather than reproduced in full; complete text is available in the repository.

- **A2-1–A2-4** (Table 3): three system-prompt variants (Default, Conservative, Critical reviewer) crossed with three user-template variants (Default, Conflict-aware, Snippet-analysis), holding all other settings identical to O-baseline / A2-5 above.
- **Dev-2 / Dev-3** (Table 6): O-baseline augmented with 6 (Dev-2) or 8 (Dev-3) decision rules addressing specific error patterns (e.g., distinguishing niche-country clinical use from broad regulatory approval, requiring majority evidence across tested drugs). Dev-3 improved development-set Macro-F1 to 0.9743 but degraded on the held-out 9b set to 0.9667, motivating our choice of O-baseline for competition submissions.
- **SF-2 / SF-3** (Table 11): Turn-2 critique prompts inserted into the three-turn self-feedback pipeline. SF-2 (*default checklist*) asks the model to check for misread snippets, contradicting evidence, and question-relevance drift. SF-3 (*devil's advocate*) instructs the model to argue the opposite conclusion before deciding whether to revise its answer. SF-4 and SF-5 reuse the SF-2 checklist with different model assignments across turns (Sonnet-only and Opus/Sonnet cross-model, respectively).

B. Factoid Question Final Prompt

It was produced through the systematic, error-driven prompt engineering procedure described in Section 4.3, optimized on BioASQ 10b–13b development data across 13 iterative experiments.

```
Your objective is to maximize Mean Reciprocal Rank (MRR) by placing the answer string most likely to
     exactly match the gold answer in Rank1.
Return a JSON list containing 15 unique answer strings ordered by confidence.
The first answer is the most important.

------------------------------------------------------------------------
CoT MODE: OFF
Do not output any reasoning or explanation.
Output ONLY the final JSON array.
------------------------------------------------------------------------

STEP 1  IDENTIFY THE QUESTION FORMAT AND EXPECTED ANSWER FORM

A. Identify the question format first:
  - Proper question: ends with "?" and contains an interrogative word (what, which, where, how, who,
       when, is, does)
  - Fragment: a noun phrase or incomplete sentence that does NOT end with "?" or lacks an interrogative
        word
    Examples: "Adverse effects of metformin.", "Estimated incidence of X.", "Most common cause of Y."
  - Binary alternative: "Is X a Y or Z?" pattern where the answer is one of the two named options
    Examples: "Is aspirin an anticoagulant or antiplatelet agent?"

  If the question is a FRAGMENT, mentally rewrite it as an explicit question before proceeding:
  - "Adverse effects of X."  "What are the adverse effects of X?"
  - "Prevalence of X."  "What is the prevalence of X?"
  - "Most common Y of X."  "What is the most common Y of X?"
  - "Mean/estimated/typical Z."  "What is the mean/estimated/typical Z?"
  Apply the same answer-type rules as for explicit factoid questions after rewriting.

B. Determine the semantic type and expected answer surface form.
Common semantic types include:
- disease or medical condition
- gene / protein / enzyme / molecular target
```

```
- mutation / exon / genotype
- drug
- treatment indication
- biological process or mechanism
- diagnostic sign or biomarker
- route of administration
- numeric value (percentage, year, concentration, etc.)
- anatomical structure
- microorganism / virus / bacterium
- cell type / tissue / organ
- chemical compound / small molecule
- pathway name
- transcription factor
- species / organism name

Important:
For BioASQ factoid questions, the gold answer is usually a short answer string, such as:
- a short entity name
- a concise snippet phrase
- a mutation/exon identifier
- a numeric value with unit
- a short process phrase
Do NOT assume the gold answer is always the most canonical biomedical name.
Choose the answer form most likely to exactly match the gold answer.
----------------------------------------------------------------------
STEP 2  EXTRACT CANDIDATE ANSWERS FROM THE SNIPPETS
Extract candidate answers supported by the snippets.
Rules:
- Answers must be supported by the snippets
- Prefer wording grounded directly in the snippets
- You may trim unnecessary surrounding words
- Do not invent unsupported entities
- Do not paraphrase aggressively
- Prefer short standalone answer strings
- Do not return full sentences unless the question explicitly asks for a process or mechanism and the
    answer cannot be expressed as a short phrase

When a snippet is longer than 200 characters:
- Extract the minimal noun phrase or identifier from the most directly relevant clause
- Do NOT include surrounding context words that are not part of the answer
- Example: snippet says "...we identify FOXA1, a forkhead transcription factor, as the key driver..."
  extract "FOXA1", not "FOXA1, a forkhead transcription factor"

Negative rules (what NOT to extract):
- Do not extract full sentences as candidates unless no short form exists
- Do not extract surrounding context words that are not part of the answer
- Do not extract method names when the question asks for a result
- Do not extract author names or citation markers

Synthetic examples (illustration only):
"patients diagnosed with Huntington's disease"  "Huntington's disease"
"deletion in the PTEN tumor suppressor gene"  "PTEN"
"upregulation of the mTOR signaling pathway"  "mTOR signaling pathway"
"amenable to exon 45 skipping therapy"  "exon 45"
----------------------------------------------------------------------
STEP 3  GENERATE PLAUSIBLE SURFACE-FORM VARIANTS
BioASQ gold answers often vary in wording. When the snippets support multiple valid forms of the same
    answer, include multiple variants only if they meaningfully improve exact-match chances.
Useful variants may include:
- canonical scientific name
- shorter snippet form
- abbreviation / acronym
- singular / plural form
- adjective / adverb form when both are plausible
- British AND American spelling variants (always generate both when applicable)
- noun form AND gerund/verbal form when the question is about a process or use
- concise phrase vs slightly longer phrase

Spelling variants are especially important for BioASQ:
BioASQ uses British English in gold answers. Always include the British spelling as a candidate
alongside the American spelling when they differ.
Examples:
"haemophilia" (British) / "hemophilia" (American)  include both, British spelling in Rank 1
"oestrogen" / "estrogen"  include both, British spelling in Rank 1
```

```
"anaerobe" / "anaerobe" (same here, but "anaesthetic" / "anesthetic" differ)  British in Rank 1
Note: BioASQ gold answers sometimes use non-standard or informal spellings
(e.g. "mosquitos" instead of "mosquitoes"). When a question involves a common
organism or term with spelling variants, include both forms.

Slot allocation strategy:
- Slots 12: Most likely exact-match form and its strongest variant
- Slots 34: Alternative surface forms (abbreviation, plural, spelling variant, etc.)
- Slot 5: Only if a genuinely different competing answer exists
Do NOT use all 5 slots on near-duplicate variants of a weak answer.
----------------------------------------------------------------------
STEP 4  SPECIAL QUESTION-TYPE RULES

Rule priority when multiple rules below apply:
E (process/mechanism) > B (mutation/exon) > J (binary alternative) >
G (gene/protein abbreviation) > A (definition-style) > all others

A. Definition-style factoid questions ("What is X?")
For questions such as "What is X?" or "What is Y?":
- If X is a database or software tool  prefer a SHORT descriptive label, not the full
  description sentence. Gold answers for tools are often just 24 words.
  Example: Q "What is GeneCards?"  gold is likely "gene database", not the full description
- If X is a surgical procedure  describe what it treats or its purpose in a short phrase,
  using the specific condition named in the snippets, not a related broader disease
  Example: Q "What is Heller myotomy?"  gold is likely "surgical procedure for achalasia"
- If X is a biological concept  prefer a short descriptive noun phrase
- If X is an abbreviation being defined  prefer the full expanded form
- Include both short and long forms across slots when uncertain
Other example:
Question: What is CRISPR?
 ["clustered regularly interspaced short palindromic repeats", "RNA-guided genome editing system"]

B. Mutation / exon / genotype questions
Prefer the minimal identifier that directly answers the question.
When the gold answer may include multiple notation formats (e.g. HGVS + protein notation),
include the most complete standard form first, then shorter variants.
Example:
"shown to undergo exon 23 skipping"  "exon 23"
Question asking for a specific mutation:
 ["BRCA1 c.5266dupC", "c.5266dupC", "p.Gln1756Profs"]

C. Numeric questions
Prefer the numeric value that directly answers the question.
Keep the number and unit if present.
Examples:
"approximately 5%"  "5%"
"greater than 2 mg/dL"  "2 mg/dL"
If both a short form and a fuller statistic are plausible, include both.
Example:
["Odds Ratio 1.32", "1.32"]

When multiple numeric values appear in snippets:
- Identify the grammatical subject of the question, then find the number
  that is the predicate/complement of that exact subject
- Prefer a range over a point estimate when both appear in the snippets:
  e.g. if snippets contain both "2-8%" and "3.04%", rank the range "2-8%" first
- Preserve exact formatting from the snippet including commas in large numbers:
  e.g. "1,000" not "1000", "per 1,000 births" not "per 1000 births"
- If the question asks for "average" or "approximate", prefer a single representative number
- Include competing values in lower ranks if genuinely uncertain

D. Route / administration questions
Include route adjective/adverb forms when supported.
Example:
["intrathecally", "intrathecal"]

E. Process / mechanism questions
Return the process phrase, not the entity name.
When the question asks what a gene/protein/molecule "mediates", "is associated with",
or "is involved in", the answer is the process name  include BOTH the gerund form
and the noun form as variants.
Examples:
"X inactivation" and "X-Inactivation" and "X chromosome inactivation"  include all three
"ubiquitination" and "ubiquitin-mediated proteolysis"  include both
```

```
Prefer a short process phrase over a full sentence whenever possible.

IMPORTANT  noun form vs verb form:
When the question asks for a drug's mechanism of action or a molecule's function,
the gold answer is almost always a NOUN PHRASE (e.g. "X inhibitor", "X antagonist",
"inhibition of X"), NOT a verb phrase (e.g. "inhibits X", "blocks X").
Always generate the noun-phrase form as Rank 1 or Rank 2:
- "inhibits ATP-citrate lyase"  also generate "ATP-citrate lyase inhibitor"
- "blocks the receptor"  also generate "receptor antagonist" / "receptor blocker"
- "activates pathway Y"  also generate "activator of pathway Y"

F. Disease / indication questions
Prefer the disease or condition being treated, not the molecular mechanism.
Example:
Question: What disease does drug Y treat?
Prefer: "multiple myeloma"
Not: "inhibition of proteasome activity"

IMPORTANT  "What complication / use / purpose" questions:
When the question asks "What complication was treated with X?" or "What was X used for?"
or "What is the indication for X?", the gold answer is the CLINICAL INDICATION or
DRUG FUNCTION CATEGORY, not the disease mechanism or symptom context.
Example:
Q: "What complication of pregnancy was X used for?"
 gold is the clinical purpose/category (e.g. "antiemetic", "preventing miscarriage")
  NOT the disease name alone (e.g. "morning sickness", "nausea")
Include both the drug-class/function term AND the condition name as candidates,
with the drug-class/function term in Rank 1 when snippets describe usage purpose.

G. Gene / protein name vs abbreviation questions
- When the question uses the full name  prefer the abbreviation as Rank 1
- When the question uses the abbreviation  prefer the full name as Rank 1, abbreviation as Rank 2
- When the question asks "what is the target / name / ligand / receptor of X" and neither
 the full name nor an abbreviation appears in the question itself  prefer the FULL NAME
 as Rank 1, abbreviation as Rank 2. Never place a bare abbreviation in Rank 1 in this case.
Examples:
Question: "What gene encodes the retinoblastoma protein?"
 ["RB1", "retinoblastoma 1"]
Question: "What is the full name of VEGF?"
 ["vascular endothelial growth factor", "VEGF"]
Question: "What is the target of drug X?" (neither full name nor abbreviation in question)
 ["full protein name", "ABBREVIATION"]  full name always Rank 1 here

H. Anatomical location questions
Prefer the most GENERAL anatomical region that still correctly answers the question,
then place more specific sub-regions in lower ranks.
The gold answer in BioASQ tends to be the broader region, not the most precise sub-structure.
Example:
Question: "Where is the substantia nigra located?"
 ["midbrain", "brainstem", "brain"]  broader first
Question: "Where are sulci and gyri found?"
 ["brain", "cerebral cortex", "cerebrum"]  "brain" first, not "cerebral cortex"
If snippets do not state the location, use internal biomedical knowledge.

I. Fragment / incomplete question format
When the question body is a noun phrase or sentence fragment (identified in Step 1A):
- After mentally rewriting to an explicit question, apply the appropriate rule above
- The answer should be as short as if the question were asked explicitly
- Examples after rewriting:
 "Adverse effects of metformin."  "What are the adverse effects?"  ["lactic acidosis", "
     gastrointestinal side effects", "nausea"]
 "Estimated incidence of postpartum depression."  numeric  ["10-15%", "1 in 7 women"]
 "Most common cause of bacterial meningitis in adults."  entity  ["Streptococcus pneumoniae", "
     Neisseria meningitidis"]
 "Mean age of onset of Parkinson's disease."  numeric  ["60 years", "60"]

J. Binary alternative questions ("Is X a Y or Z?")
When the question asks the subject to be classified between two explicit options:
- Extract both alternatives
- Rank the option directly supported by snippets as Rank 1
- Return ONLY the chosen short alternative as Rank 1, not the full sentence
- Also include the modifier form if snippets support it (e.g. "strict anaerobe" not just "anaerobe")
- Include the unchosen alternative as last resort
- Example:
```

```
  Q: "Is aspirin an anticoagulant or antiplatelet agent?"
  Snippet: "Aspirin irreversibly inhibits COX-1, functioning as an antiplatelet agent"
   ["antiplatelet", "antiplatelet agent", "anticoagulant"]
  Q: "Is X an obligate aerobe or anaerobe?"
  Snippet: "strictly anaerobic gram-positive bacterium"
   ["strict anaerobe", "anaerobe", "strictly anaerobic", "aerobe"]

K. Previous / historical name or acronym questions
When the question asks for a "previous", "former", "old", or "historical"
name, acronym, or abbreviation of a current term:
- The answer is the deprecated/former term, not the current one
- If the snippet provides only the full form without the abbreviation,
  use internal biomedical knowledge to supply the standard abbreviation (Step 6 fallback)
- Include both the abbreviation and the full former name as variants
- Example:
  Q: "What is the previous acronym for MAFLD?"
   ["NAFLD", "nonalcoholic fatty liver disease"]

L. Specificity level  prefer the answer at the right granularity
BioASQ gold answers vary widely in specificity. Apply this rule:
- If a question asks "What type/kind/class of X", prefer a category-level term
  (e.g. "anesthetics" not "volatile anesthetics")
- If a question asks for a specific agent/drug/gene, prefer the specific term
- When uncertain, include BOTH the broad term AND the specific term,
  with the broader/shorter term in Rank 2
- Also apply this to adjectives/prepositions: gold answers often include
  a qualifier that model candidates miss
  Example: gold = "strict anaerobe"  always try "strict [answer]" as a variant
  Example: gold = "use of contrast agents"  include "use of [entity]" as a variant alongside just "[
      entity]"
  Example: gold = "preventing miscarriage"  include both "preventing miscarriage" and "miscarriage
      prevention" and "miscarriage"

M. Entity + qualifier variants
When extracting an entity answer, always generate these surface-form variants:
1. Entity alone: "miscarriage"
2. Gerund action + entity: "preventing miscarriage", "prevention of miscarriage"
3. "use of" + entity: "use of iodinated contrast agents"
4. Entity + abbreviated parenthetical: "human umbilical cord (hUC)"
5. Modifier + entity: "strict anaerobe", "chronic condition"
Include whichever variants are plausible given the question wording.
The question's verb or preposition often signals which variant gold expects:
- "treated with" / "used for"  "use of X" or just "X"
- "complication of" / "caused by"  noun or gerund form
- "associated with"  noun form
- "mediates" / "involved in"  process/gerund form
----------------------------------------------------------------------
STEP 5  CANONICALIZE CAREFULLY
Prefer canonical biomedical forms only when they are likely to exactly match the gold answer.
Do NOT blindly replace a snippet phrase with a more formal biomedical synonym.
Examples:
If snippets say "patients with elevated blood pressure":
Candidates: "elevated blood pressure", "hypertension"

If the question asks for an exon:
Prefer: "exon 23"
NOT: "DMD amenable to exon 23 skipping"

If the question asks for a mechanism:
Prefer: "increase T-cell activation"
NOT: "immune checkpoint blockade"

Canonicalization is helpful only when it increases the chance of exact gold-answer match.
----------------------------------------------------------------------
STEP 6  HANDLE INSUFFICIENT SNIPPETS
If the snippets do not contain sufficient information to directly answer the question:
- Use your internal biomedical knowledge to generate plausible candidates
- Clearly prioritize any snippet-grounded answers if ANY exist
- Place knowledge-based answers in lower ranks (35)
- Do not return empty output
----------------------------------------------------------------------
STEP 7  HANDLE CONFLICTING SNIPPETS
When snippets contain conflicting answers:
- Count how many snippets support each candidate
```

```
- Prefer the candidate supported by more snippets
- If tied, prefer the candidate from the snippet with more specific or direct wording
- Place the minority candidate in Rank 2 as insurance
--------------------------------------------------------------------
STEP 8  RANK CANDIDATES
Rank candidates by likelihood of exactly matching the gold answer.
Use the following heuristics:
1. Match the expected answer form for the question type:
   entity question  entity
   exon/mutation question  identifier (full notation preferred)
   numeric question  number/statistic (preserve original formatting)
   mechanism/process question  process phrase or gerund
   anatomical question  anatomical region
   binary alternative question  the chosen alternative, with modifier if snippets support it
2. Prefer candidates directly supported by snippet wording.
3. Prefer concise answers that fully answer the question.
4. When multiple surface forms of the same answer are plausible, include several strong variants.
5. Avoid ranking related but wrong entity types.
6. Prefer the candidate most likely to be used as the final standalone BioASQ gold answer string.

Slot allocation:
- Rank 1: Single most likely exact-match answer
- Rank 2: Strongest surface-form variant OR broader/shorter form of Rank 1
- Ranks 35: Additional variants OR competing answers only if genuinely plausible

Tie-breaking rules (when two candidates are equally plausible):
1. Prefer the form that appears verbatim in the snippets
2. Prefer British spelling over American spelling
3. Prefer the shorter form over the longer form
4. Prefer the form that directly matches the question's answer type word
--------------------------------------------------------------------
IMPORTANT RULES
- Do not hallucinate unsupported entities
- Do not blindly prefer the shortest span
- Do not blindly prefer the most canonical biomedical name
- Prefer short standalone answer strings
- Use full sentences only when a short phrase cannot answer the question
- For mutation/exon questions, prefer the complete standard notation first
- For binary alternative questions, include the modifier from the snippet (e.g. "strict anaerobe" not
     just "anaerobe")
- Always include British spelling variants when American/British differences exist
- Always generate "use of X", gerund, and parenthetical-abbreviation variants when relevant
- When uncertain about specificity level, include both the broad and specific term
- Do not return empty output
- Return between 1 and 5 strings
- Each string must be unique (no duplicates)
- No explanation, no markdown, no extra text outside the JSON array
--------------------------------------------------------------------
INPUT
Question:
{question}
Snippets:
{context}
--------------------------------------------------------------------
OUTPUT FORMAT
Your entire response must be a single valid JSON array of strings.
Return ONLY the JSON array.
Example:
["Answer1", "Answer2", "Answer3"]
""".strip()


def prompt_list(question: str, context: str) -> str:
   return f"""
You are an answering system for the BioASQ Phase B biomedical question answering task.

Task:
Answer a LIST question using ONLY the provided snippets.

Rules:
- Output MUST be a JSON list of short entity strings.
- Each item MUST be copied VERBATIM from the snippets.
- Do NOT include explanations or sentences.
- Do NOT include duplicates.
```

```
- If no valid items exist, output [].
- Output MUST be valid JSON.
- No markdown.

Question: {question}

Snippets:
{context}

Return ONLY the JSON exact answer.
]
```

C. List Question Competition Prompt

The prompt below was used for all four BioASQ 14b list-question competition submissions without modification. It was produced by the feedback-driven optimization procedure described in Section 4.4, optimized on BioASQ 11b (batch 4).

```
You are an answering system for the BioASQ Phase B biomedical question answering task.
Task: Answer a LIST question using ONLY the provided snippets.
Rules:
- Output MUST be a valid JSON list of strings.
- Each string MUST be an exact substring found in the snippets.
- No explanations, no keys, no extra text, no markdown.
- Do NOT include parentheses '(' or ')' in any answer string.
- Remove duplicate answers (case-insensitive matching).
- If no answer is found in snippets, return an empty list: []
- Extract minimal entities (e.g., "BRCA1" not "BRCA1 gene mutation").
- For questions regarding "Syndromes" or "Triads", extract ONLY the defining features or core symptoms,
    avoiding general associated conditions found in the text.
- Order answers by frequency of mention in snippets, then alphabetically.
- When multiple forms exist (abbreviations vs full names), PREFER the Full Name over the abbreviation (
    even if the abbreviation is more frequent).
- Maximum string length per item: 100 characters.
- Prioritize precision over recall - only include high-confidence answers clearly supported by snippets.
- SCAN ALL SNIPPETS: Don't stop after finding a few answers - check every snippet thoroughly.
- CONTEXT MATTERS: Only include entities that are explicitly stated as answers to the question type (e.g
    ., for "What genes...", only include actual gene names mentioned as associated with the condition).
- AVOID SPECULATION: Do not infer or extrapolate answers not directly stated in snippets.
- For numerical or measurement answers: include the unit if present in the snippet.
- For drug/treatment questions: include both generic and brand names if both appear.
- For protein/gene questions: normalize capitalization to match standard nomenclature (e.g., human genes
     in capitals).
Example:
Question: "What genes are associated with Alzheimer's disease?"
Snippets: "Studies show APOE, PSEN1, and PSEN2 are linked to Alzheimer's. The APOE gene is the most
    significant..."
Expected output: ["APOE", "PSEN1", "PSEN2"]
Input:
Question: {question}
Snippets: {context}
Return ONLY the JSON list of strings.
```

The prompt below was produced by the post-competition replication of the same optimization procedure (Table 19, Iteration 5). While structurally different from the competition prompt—adopting a six-step workflow with an explicit 90% confidence threshold—it achieved a comparable F1 of 0.7653 on the same training batch, confirming the method's robustness across independent runs.

```
You are a biomedical expert answering LIST questions for the BioASQ Phase B challenge.

Task: Given a biomedical question and supporting snippets, return a JSON list of answer strings.

APPROACH -- FOLLOW EACH STEP PRECISELY:

Step 1: PARSE THE QUESTION
- What exact entity type is requested? (genes, proteins, drugs, diseases, symptoms, methods, cell types,
     etc.)
```

```
- What exact relationship is asked about? (causes, treats, inhibits, components of, associated with, etc
    .)
- What are the scope constraints? (organism, tissue, disease context, time, etc.)
- If the question implies a specific number (e.g., "triad," "two main types," "three classes"), your
    answer list should match that number unless snippets clearly support more.

Step 2: EXTRACT CANDIDATES FROM SNIPPETS -- CONSERVATIVELY
- Read every snippet. For each, identify entities that are EXPLICITLY and DIRECTLY stated as fulfilling
    the asked relationship.
- The snippet must use AFFIRMATIVE language connecting the entity to the relationship. Mere co-
    occurrence or tangential mention is NOT sufficient.
- EXCLUDE entities mentioned in: negation contexts, hypothetical/speculative contexts, background/
    historical contexts that were later contradicted, comparison baselines that are not themselves
    answers.
- Example: "Unlike A, B and C are effective for X" -> only B and C are candidates, not A.

Step 3: STRICT VALIDATION -- APPLY EVERY CHECK TO EVERY CANDIDATE
For each candidate entity, it must pass ALL of the following:
(a) CORRECT TYPE: Is it exactly the entity type the question asks for?
(b) DIRECT EVIDENCE: Can you quote or closely paraphrase a specific snippet passage that explicitly
    states this entity answers the question? If you cannot, REMOVE it.
(c) AFFIRMATIVE CONTEXT: Is it stated positively as an answer, not negated, not merely hypothesized? If
    not affirmative, REMOVE it.
(d) SPECIFICITY: If both a general category and specific members appear, keep only the specific members
    UNLESS the question asks for categories.
(e) RELEVANCE: Does the entity answer THIS specific question, not a related but different question? If
    there's any mismatch in scope, REMOVE it.

Step 4: AGGRESSIVE PRUNING -- PRECISION MATTERS
- After validation, review your remaining list with a critical eye toward REMOVING borderline entries.
- Keep ONLY answers where you have HIGH CONFIDENCE (90%+) based on clear, direct snippet evidence.
- If an entity appears in only one snippet and the support is ambiguous or indirect, REMOVE it.
- If an entity is supported by multiple snippets with clear evidence, it is a strong answer -- keep it.
- Prefer a shorter, highly accurate list over a longer list with questionable entries.
- Ask yourself: "Would I bet on each answer being in the gold standard?" If not, remove it.

Step 5: DEDUPLICATION AND NAMING
- Merge synonyms/aliases into one canonical entry (e.g., keep "imatinib" not both "imatinib" and "
    Gleevec").
- Use standard biomedical nomenclature as it appears in the snippets.
- Use minimal precise names (e.g., "BRCA1" not "the BRCA1 gene product").
- Do NOT include parentheses characters in any answer string.
- Maximum 100 characters per answer string.

Step 6: FINAL VERIFICATION -- MANDATORY
Re-read the question. For each answer in your list:
1. Identify the specific snippet text supporting it.
2. Confirm the snippet DIRECTLY and EXPLICITLY links this entity to the asked relationship.
3. Confirm the entity type matches what was asked.
If ANY check fails, remove that answer immediately.

FORMATTING:
- Output ONLY a valid JSON list of strings. No explanations, no keys, no markdown, no extra text.
- Order answers by strength of evidence (strongest first).

Question: {question}
Snippets: {context}

Return ONLY the JSON list of strings.
```

D. Optimization Meta-Prompt Template

The following template was used at each iteration of the feedback-driven prompt optimization loop to generate an improved prompt. Placeholders in curly braces are filled at runtime with the corresponding values.

```
You are an expert prompt engineer optimizing a prompt for the BioASQ Phase B biomedical LIST question
    answering task.
```

```
The system uses an LLM to answer LIST questions: given a question and snippets from biomedical
    literature, it must output a JSON list of answer strings.

Here is the performance history of different prompt versions:
{history_summary}

Current prompt (Iteration {iteration}):
---
{current_prompt}
---

Official evaluation metrics from BioASQ:
- Precision: {precision} (fraction of predicted items that are correct)
- Recall: {recall} (fraction of gold items that were predicted)
- F1: {f1}

[If precision < recall]: Precision is low -> the model is listing too many wrong items. Focus on being
    more selective.
[If recall < precision]: Recall is low -> the model is missing correct items. Focus on being more
    thorough and extracting more entities.
[If balanced]: Both are relatively balanced. Try to improve both incrementally.

Generate an IMPROVED prompt that will achieve higher F1 score. The prompt MUST:
1. Contain {question} and {context} as placeholders (these will be replaced at runtime)
2. Instruct the model to output ONLY a valid JSON list of strings
3. Address the specific weakness identified above
4. Be self-contained (the model only sees this prompt + question + snippets)

Return ONLY the improved prompt text, nothing else. No markdown code blocks.
```