Can LLMs Refine Their Own Keyword Search Strategies for Biomedical RAG?

BioASQ at CLEF 2026

Samy Ateia[1], Udo Kruschwitz[1]

[1]Information Science, University of Regensburg, Universitätsstraße 31, 93053, Regensburg, Germany

Abstract

In biomedical contexts, transparent and reproducible retrieval of evidence often matters for clinical question answering or conducting systematic reviews. But purely keyword-based generated Boolean search queries often do not reach the same retrieval performance as dense embedding-based retrieval and reranking. We explored the ability of current Large Language Models (LLMs) to iterate and refine their own transparent keyword search strategies. We used a small subset (24 questions) of the BioASQ training data, which is a biomedical retrieval augmented generation (RAG) based biomedical question-answering challenge. We demonstrate that Reciprocal Rank Fusion (RRF) of three different search strategies can achieve significantly higher recall@50 after refinement and also beat our prior single query baseline.

Keywords

Retrieval Augmented Generation, Large Language Models, Biomedical Question Answering, Professional Search, Query Refinement, Query Expansion, BioASQ

1. Introduction

LLM-based and generative search interfaces are increasingly used for information seeking, including practical search, learning-oriented search, and knowledge-work tasks [1].[1] In biomedical question answering, systematic reviews, and evidence synthesis, retrieval quality is especially important: if relevant publications are not found, even strong answer-generation models cannot produce well-grounded answers.

At the same time, biomedical professional search, as used in systematic reviews, often requires transparency and reproducibility [2]. Dense retrieval and neural reranking can improve effectiveness, but explicit keyword and Boolean strategies remain easier to inspect, report, and rerun. However, good Boolean queries are difficult to formulate because they require appropriate biomedical terminology, synonyms, abbreviations, controlled vocabulary, and relation terms.

We therefore investigate whether LLMs can iteratively refine their own transparent keyword-search strategies. Rather than optimizing individual queries or model weights, our approach refines reusable textual instructions for generating Boolean queries. These strategies remain inspectable while still allowing the model to adapt how it uses synonyms, MeSH terms, rare anchor terms, and relation phrases.

We evaluate this idea in BioASQ Task 14B [3], focusing on Phase A retrieval. We compare single-query LLM-generated Boolean baselines with a strategy-refined system that generates three complementary queries and combines them using reciprocal rank fusion (RRF). Our results suggest that LLM-guided textual strategy refinement can improve document recall@50 over both a strong few-shot single-query baseline and an unrefined multi-query RRF setup while preserving a transparent retrieval process.

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

*Corresponding author.

†These authors contributed equally.

Samy.Ateia@ur.de (S. Ateia); udo.kruschwitz@ur.de (U. Kruschwitz)

0009-0000-2622-9194 (S. Ateia); 0000-0002-5503-0341 (U. Kruschwitz)

[1]https://www.pewresearch.org/social-trends/2025/02/25/workers-experience-with-ai-chatbots-in-their-jobs/ https://www.pewresearch.org/internet/2026/02/24/how-teens-use-and-view-ai/

1.1. BioASQ Challenge

This is our fourth participation at BioASQ. BioASQ is an annually running lab and shared task at CLEF focusing on the intersection of natural language processing (NLP) and information retrieval (IR) for biomedical use cases [4]. Task B on biomedical semantic QA is a retrieval-augmented generation (RAG) task where, given a biomedical question, a system has to first find all relevant citations in PubMed, extract relevant snippets from abstracts and titles, and then, given this context, generate an accurate answer.

This year we focused on the retrieval aspect of our systems and explored if and how LLMs can optimize the generation of transparent Boolean keyword queries for the task.

2. Related Work

Transparent and reproducible search strategies are important in biomedical evidence synthesis. PRISMA [5], a widely adopted reporting guideline for systematic reviews, was extended by PRISMA-S [2] that aims to improve the reporting and reproducibility of search methods. But reproducible search remains an important challenge in evidence synthesis [6]. With the emergence of LLM-aided discovery and dense retrieval methods, we explore how the semantic knowledge of LLMs can be leveraged to create auditable and reproducible Boolean keyword search queries that can be used in these rigorous processes.

Prior work has explored automatic Boolean query formulation and refinement for systematic-review search. Existing approaches include transforming and refining Boolean queries through expansion and reduction [7, 8] as well as generating queries directly from review protocols using concept extraction and terminology expansion [9]. Our work focuses on refining reusable LLM-generated search strategies rather than modifying a single existing query.

Biomedical retrieval often combines free-text search with controlled vocabularies such as MeSH. Existing research has investigated automated MeSH term suggestion to support query formulation [10]. Similarly, our approach can incorporate MeSH terms but treats them as one component within broader retrieval strategies that also leverage title and abstract terminology.

Recent studies have evaluated LLMs for Boolean query generation in systematic-review search [11, 12]. Subsequent work highlighted the importance of factors such as prompt design, query validation, and model selection for retrieval effectiveness [13]. Unlike these studies, we focus on refining reusable query-generation strategies and evaluating them in a biomedical question-answering setting.

Our approach is also related to automatic prompt optimization. Methods such as Automatic Prompt Engineer [14], ProTeGi [15], PromptAgent [16], and DSPy [17] iteratively improve textual instructions using task feedback. We apply a similar idea to retrieval, optimizing search strategies that generate transparent Boolean queries rather than general-purpose task prompts.

Reciprocal rank fusion (RRF) is a widely used method for combining ranked retrieval results [18]. We use RRF to combine results from multiple LLM-generated query strategies and evaluate the resulting retrieval pipeline in BioASQ, a biomedical question-answering benchmark that combines document retrieval, snippet retrieval, and answer generation [19].

3. Methodology

We evaluated five LLM-based systems for BioASQ Task 14B. The task was structured into three phases. Phase A required retrieval of relevant PubMed documents and snippets. Phase A+ generated answers from the snippets retrieved by our Phase A systems. Phase B generated answers from the organizer-provided evidence as well as our own retrieved snippets. Compared to our previous year's participation at BioASQ13, the baseline and most of the pipeline stayed the same. We changed the base models used and added new refined search strategies for generating multiple diverse Boolean keyword queries to the Phase A retrieval step.

Table 1
Submitted system configurations, grouped by batch pair. "10-s" denotes 10-shot prompting and "0-s" denotes zero-shot prompting. "3-RRF" denotes three strategy-specific retrieval queries fused with reciprocal rank fusion (k=60).

Batches	System	Phase A	Phase A+	Phase B
1–2	UR-IW-1	Gemini 3 Flash, 10-s	Gemini 3 Flash, 10-s	Gemini 3 Flash, 10-s
1–2	UR-IW-2	Gemini 3.1 Flash Lite, 10-s	Gemini 3.1 Flash Lite, 10-s	Gemini 3.1 Flash Lite, 10-s
1–2	UR-IW-3	GPT-5.4, 0-s	GPT-5.4, 0-s[†]	GPT-5.4, 0-s
1–2	UR-IW-4	GPT-5.4-mini, 10-s	GPT-5.4-mini, 10-s	GPT-5.4-mini, 10-s
1–2	UR-IW-5	GPT-5.4-nano, 10-s	GPT-5.4-nano, 10-s	GPT-5.4-nano, 10-s
3–4	UR-IW-1	GPT-5.4, 0-s	GPT-5.4, 10-s	GPT-5.4, 10-s
3–4	UR-IW-2	Gemini 3 Flash, 10-s	Gemini 3 Flash, 10-s	Gemini 3 Flash, 10-s
3–4	UR-IW-3	GPT-5.4-mini, 10-s	GPT-5.4-mini, 10-s[†]	GPT-5.4-mini, 0-s
3–4	UR-IW-4	GPT-5.4-mini, 10-s, 3-RRF	GPT-5.4-mini, 10-s	GPT-5.4-mini, 10-s
3–4	UR-IW-5	Gemini 3 Flash, 10-s, 3-RRF	Gemini 3 Flash, 10-s	Gemini 3 Flash, 10-s

[†] In Phase A+, `UR-IW-3` reused the evidence retrieved by `UR-IW-4`; we therefore do not use this setting as an independent retrieval comparison.

3.1. Submitted Systems

The submitted systems used Gemini 3 Flash Preview, Gemini 3.1 Flash Lite Preview, GPT-5.4, GPT-5.4-mini, and GPT-5.4-nano. System configurations were identical in Batches 1–2 and changed in Batches 3–4, where strategy-refined retrieval was introduced. Table 1 summarises both groups.

3.2. Phase A Retrieval

The baseline Phase A pipeline generated one OpenSearch/Elasticsearch `query_string` query per biomedical question. Queries were executed against a PubMed-style index containing title, abstract, and MeSH fields. Retrieved documents were then passed to an LLM-based snippet extraction and reranking step. The final Phase A submission contained ranked documents and snippets.

Batches 1–2 used only single-query retrieval baselines. Batches 3–4 introduced strategy-refined retrieval in `UR-IW-4` and `UR-IW-5`. The cleanest submitted comparison is the Gemini pair, `UR-IW-2` versus `UR-IW-5`: both used Gemini 3 Flash with 10-shot prompting, while `UR-IW-2` used the single-query baseline and `UR-IW-5` used the three-query RRF system. The GPT-5.4-mini pair accidentally used 0-shot prompting in Phase B for `UR-IW-3` and the retrieved evidence of `UR-IW-4` in Phase A+ and therefore offers no clear comparison insight into the downstream effect of the retrieval approach for question answering.

3.3. Strategy-Refined Retrieval

The strategy-refinement loop was designed to learn reusable query-generation instructions rather than tune individual queries. GPT-5.4 was used as the strategy-reasoning model, while GPT-5.4-mini was used for high-volume query generation during evaluation. Initial strategies were also generated by GPT-5.4.

The main refinement run was executed for 10 iterations with the same 24 training questions per iteration. Training questions were selected from the latest 500 BioASQ training questions and sampled in a round-robin fashion across question types. In each iteration, every active strategy generated one Elasticsearch `query_string` query per question that retrieved up to 50 documents. During refinement, strategy performance was evaluated on the top 10 retrieved documents using precision, recall, F1, non-zero recall rate, and MAP; MAP was used as the strategy-ranking metric, as this is the official measure with which systems are ranked at BioASQ.

For each strategy, the loop selected up to three successful and three failed examples. Successful examples had non-zero recall, while failed examples had zero recall or no relevant document in the

evaluated ranking. The code additionally queried the Elasticsearch `_explain`[2] endpoint for selected relevant, missed gold, and misleading retrieved documents. These explanations were compressed into matched query terms, missing terms, dominant scoring fields, and dominant scoring terms. GPT-5.4 received the aggregate metrics, example queries, retrieval summaries, and compact explain summaries, then proposed one small strategy-level revision for the next iteration.

As a concrete example, in the first iteration the MeSH-led strategy generated the query `(mesh.terms:"Pamrevlumab"^4 OR mesh.flat:(pamrevlumab)) AND (title:(endpoint* OR ...) OR abstract:(...))` for the question "What are the primary and secondary endpoints of Pamrevlumab?", which retrieved no documents. No document in the index carries that exact controlled heading so hard-ANDing the exact-match `mesh.terms` keyword anchor excluded every candidate regardless of its title or abstract content. The compact explain summary for the missed gold document, correspondingly reported `matched terms: none; ...; note: document did not satisfy query match`. From this and the other examples, GPT-5.4 diagnosed "hard AND-gating on MeSH concept anchors" and revised the strategy to "add a low-boost title/abstract fallback for each concept anchor inside the same clause", the change that produced the title/abstract back-off used in the final MeSH-Guided strategy.

For submission, we used the best-performing strategy set observed in this run, which occurred at iteration 8. The submitted 3-RRF systems used three complementary strategies:

- **Synonym Expansion**: expands aliases, abbreviations, spelling variants, and paraphrases while preserving rare anchor terms.
- **MeSH-Guided Retrieval**: uses MeSH fields for established biomedical concepts and backs off to title and abstract text for newer or less standardized terms.
- **Literal Claim Core**: emphasizes the main entity, the biomedical relation expressed by the question, and optional answer-type constraints.

For each final strategy, we selected 10 few-shot examples from development questions by first generating zero-shot queries, executing them at retrieval depth 50, and ranking candidate examples by Recall@50, with average precision used as a tie-breaker.

At inference time, each strategy generated one query for the same question. The three ranked document lists were fused using reciprocal rank fusion with k=60. The fused ranking was then processed by the same snippet extraction and reranking pipeline used by the single-query baselines.

3.4. Phase A+ and Phase B Answer Generation

Phase A+ generated answers from snippets retrieved by the submitted Phase A systems. Phase B generated answers from the organizer-provided evidence as well as our top 20 retrieved snippets. Separate prompts were used for yes/no, factoid, list, and ideal-answer questions. The exact prompting approach for question answering was reused from our last year's submission [20].

The Phase A+ and Phase B systems followed the model and shot settings shown in Table 1. Because `UR-IW-3` in Phase A+ reused `UR-IW-4` evidence, it is not suitable for isolating retrieval differences. In Phase B Batches 3–4, `UR-IW-3` and `UR-IW-4` also differ in answer-generation shot count, so this comparison mixes retrieval and answer-generation effects.

3.5. Additional Ablation Study

To isolate the effect of strategy-refined retrieval and report validated metrics beyond the 14B preliminary results, we performed an additional ablation study on older BioASQ data. We sampled 200 questions from BioASQ 11b, 12b, and 13b, excluding questions used as few-shot examples or as strategy-refinement development data.

All ablation arms used GPT-5.4-mini:

[2] https://www.elastic.co/docs/api/doc/elasticsearch/operation/operation-explain

- **Old baseline**: the 10-shot single-query retrieval prompt.
- **Initial 3-RRF**: the unrefined three-strategy 10-shot RRF system.
- **Refined 3-RRF**: the submitted strategy-refined three-query 10-shot RRF system.

The primary retrieval metric was document recall@50. We also measured downstream impact after snippet extraction and reranking using document MAP@10 and snippet F1@10.

We additionally tested whether gains could be explained by exposing MeSH fields to the baseline prompt alone, because our few-shot baseline was sampled from prior runs where our index did not contain MeSH terms. For this ablation, we compared the original few-shot baseline with a minimally MeSH-aware variant, keeping the evaluation protocol unchanged.

3.5.1. Statistical analysis.

Because the same 200 validation questions were reused to compare three systems, each metric involved three pairwise system comparisons, and testing all of them raises the chance that at least one difference looks significant purely by chance. Uncorrected p-values were obtained from a two-sided paired permutation test on the per-question metric differences: for each system pair we used the absolute mean paired difference as the test statistic and estimated its p-value from $10{,}000$ random sign-flips of those differences (add-one estimator, fixed seed). Confidence intervals were estimated with a paired non-parametric bootstrap over the same per-question differences using $10{,}000$ resamples.

To control the family-wise error rate we then applied Holm's step-down correction, a standard multiple-comparison procedure that is uniformly more powerful than Bonferroni while giving the same protection against any false claim within a family. Within a metric, the three raw p-values are ranked from smallest to largest and multiplied by 3, 2, and 1 respectively, after which a running-maximum ("step-down") rule enforces that the adjusted values are non-decreasing down the list and all values are capped at 1. A resulting Holm-adjusted p-value is interpreted like an ordinary p-value against $\alpha = 0.05$, but with the assurance that the difference survives after accounting for the other comparisons on the same metric. For example, the smallest recall@50 p-value (refined vs. initial 3-RRF, raw $p = 0.0021$) becomes 0.0063 after multiplication by 3, and thus remains significant, whereas the initial-vs-baseline comparison (raw $p = 0.7716$) is unaffected.

3.6. Technical Implementation

All systems were implemented in Python notebooks using OpenAI and Google model APIs. Retrieval used OpenSearch/Elasticsearch `query_string` syntax over a PubMed-style index whose `title` and `abstract` are analyzed `text` fields (a custom `english_html` analyzer applying HTML stripping, standard tokenization, lowercasing, and English stop-word removal and stemming), whereas the MeSH annotations (`mesh.terms`, `mesh.flat`) are unanalyzed exact-match `keyword` fields. Snippet extraction and reranking were implemented with LLM prompts and applied consistently after retrieval.

4. Results

The official BioASQ results reported here are preliminary leaderboard results; final manual judgements were still pending at the time of writing. We therefore interpret the submitted runs descriptively and use the separate held-out ablation study for the main controlled retrieval claim.

4.1. Submitted Phase A Retrieval Results

The clearest signal in the submitted runs appeared in Phase A retrieval. Batches 1–2 used only single-query baselines and mainly serve as a reference for model behavior. In these batches, the best submitted document MAP was obtained by `UR-IW-1` in Batch 1 (MAP = 0.1473, rank 35/55) and by `UR-IW-2` in Batch 2 (MAP = 0.1474, rank 31/68). For snippet retrieval, `UR-IW-5` ranked highest among our systems

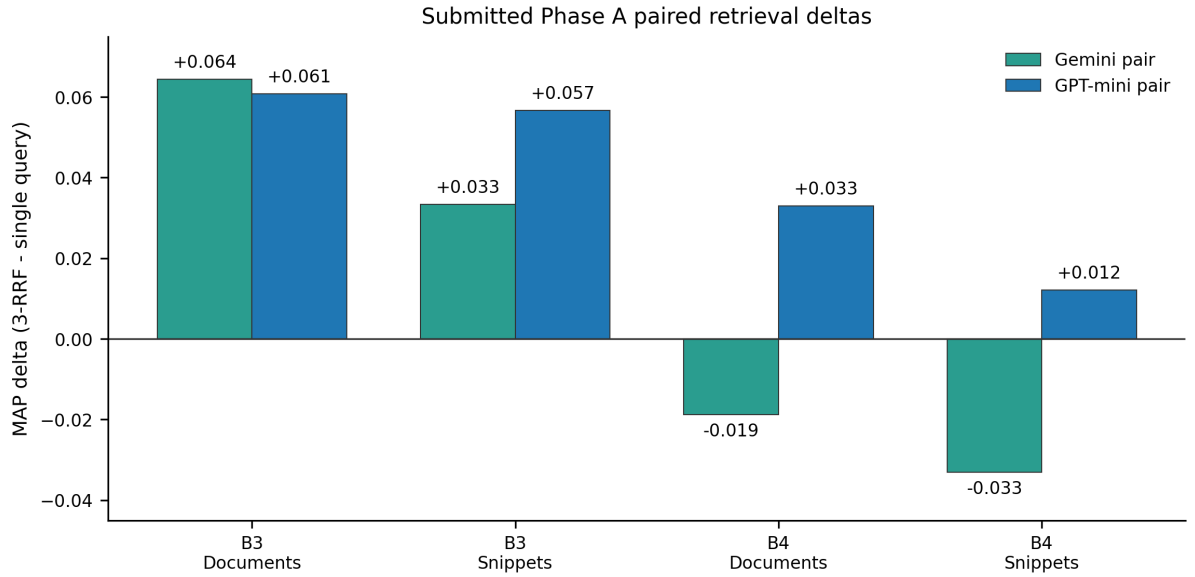

Figure 1: Submitted Phase A MAP deltas for the paired single-query and 3-RRF systems in Batches 3–4. Positive values indicate that the 3-RRF system outperformed the corresponding single-query baseline.

in Batch 1 (MAP = 0.1234, rank 9/55), while `UR-IW-3` was strongest in Batch 2 (MAP = 0.1375, rank 10/68).

Batches 3–4 introduced the strategy-refined 3-RRF retrieval systems. The strongest submitted retrieval result was `UR-IW-4`, which used GPT-5.4-mini with refined 3-RRF retrieval. In Batch 3, `UR-IW-4` reached document MAP = 0.1773 (rank 8/62) and ranked first overall for snippet MAP (MAP = 0.1440, rank 1/62). In Batch 4, `UR-IW-4` again performed strongly, with document MAP = 0.2019 (rank 6/76) and snippet MAP = 0.1508 (rank 2/76).

Figure 1 summarises the paired retrieval differences in Batches 3–4. In Batch 3, both 3-RRF systems improved over their single-query counterparts. The clean Gemini pair improved from document MAP = 0.0971 to 0.1615 and from snippet MAP = 0.0813 to 0.1146. The GPT-5.4-mini pair showed a similar direction, improving from document MAP = 0.1165 to 0.1773 and from snippet MAP = 0.0873 to 0.1440. In Batch 4, the GPT-5.4-mini 3-RRF system remained better than its single-query counterpart, but the Gemini 3-RRF system did not improve over the Gemini single-query baseline. Thus, the preliminary Phase A results are directionally consistent with the strategy-refinement hypothesis, especially in Batch 3, but are not uniform across batches.

4.2. Submitted Answer Generation Results

Downstream answer generation showed a less consistent relationship to the retrieval improvements. Yes/no questions were largely saturated: in Phase B Batch 3, all five submitted systems reached Macro F1 = 1.0000. Similar saturation was observed in Phase A+ Batch 3, where all submitted systems again reached Macro F1 = 1.0000.

Factoid and list questions showed more variation. In Phase B, `UR-IW-1` achieved the best submitted factoid result in Batch 1 (MRR = 0.5217, rank 3/72), while `UR-IW-5` was the strongest submitted factoid system in Batch 3 (MRR = 0.5000, rank 16/77). List questions showed some of the strongest downstream placements. In Phase A+, `UR-IW-4` ranked first overall in Batch 1 (F-measure = 0.2345, rank 1/54) and again ranked first overall in Batch 3 (F-measure = 0.3258, rank 1/63).

The clean Gemini downstream comparison was mixed (Figure 2). In Phase A+ Batch 3, `UR-IW-5` improved list F-measure over `UR-IW-2` (0.2717 vs. 0.2128), but had lower factoid MRR (0.3431 vs. 0.4216). In Phase B Batch 3, `UR-IW-5` improved factoid MRR over `UR-IW-2` (0.5000 vs. 0.4608), but had lower list F-measure (0.2629 vs. 0.2989). These results indicate that the submitted retrieval gains were more

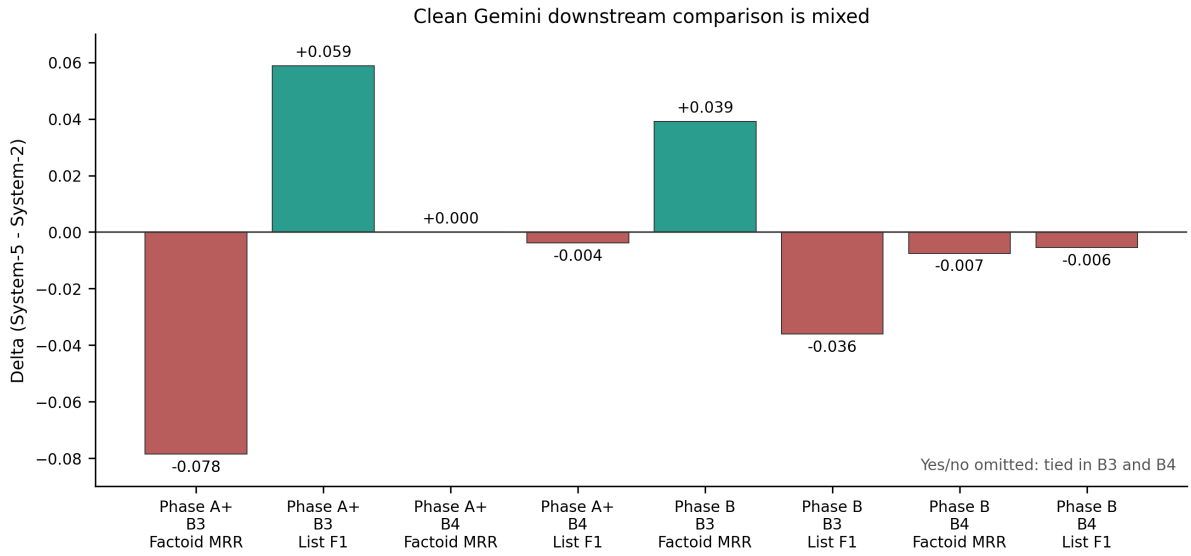

Figure 2: Downstream answer-generation deltas for the clean Gemini comparison in Batches 3–4. Values show `UR-IW-5` minus `UR-IW-2`; positive values favour the 3-RRF evidence setting.

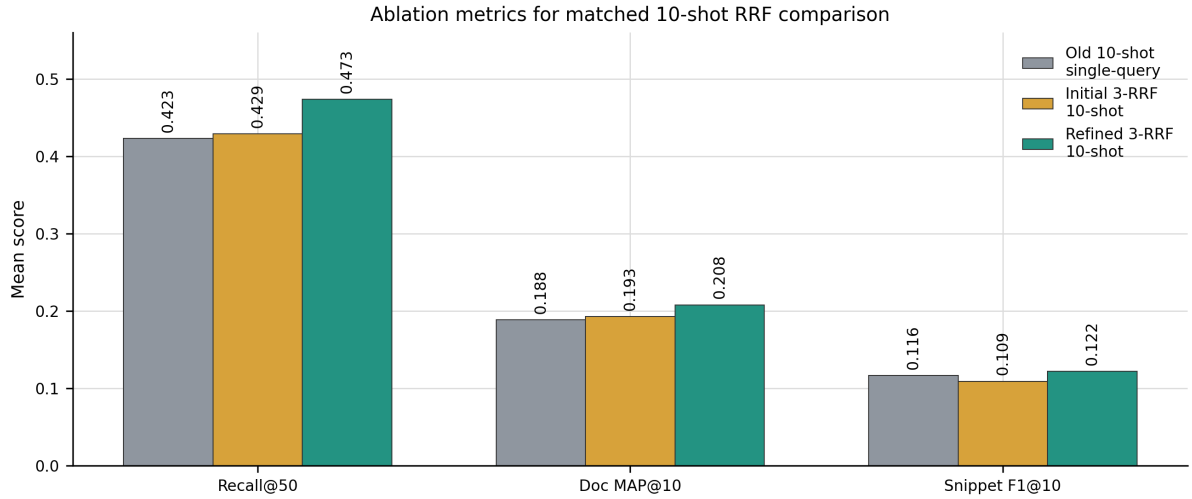

Figure 3: Mean retrieval and downstream Phase A metrics in the ablation study. Refined 3-RRF achieved the highest mean recall@50, document MAP@10, and snippet F1@10.

visible in Phase A metrics than in preliminary downstream answer metrics.

4.3. Ablation Study

The controlled 200-question ablation study provided stronger evidence for the retrieval effect of strategy refinement. All ablation arms used GPT-5.4-mini and were evaluated on older BioASQ 11b, 12b, and 13b questions that were excluded from the few-shot examples and the strategy-refinement development set. The primary fair comparison is between the refined 3-RRF system and the initial 3-RRF system, both using 10 strategy-specific examples.

The refined 10-shot 3-RRF system achieved the highest document recall@50: 0.4733, compared with 0.4291 for the matched initial 3-RRF 10-shot control and 0.4231 for the old 10-shot single-query baseline (Figure 3). Downstream scores showed the same ordering, but with smaller margins. Document MAP@10 was 0.2078 for refined 3-RRF, compared with 0.1929 for initial 3-RRF 10-shot and 0.1883 for the old baseline. Snippet F1@10 was 0.1220 for refined 3-RRF, compared with 0.1090 for initial 3-RRF 10-shot and 0.1163 for the old baseline.

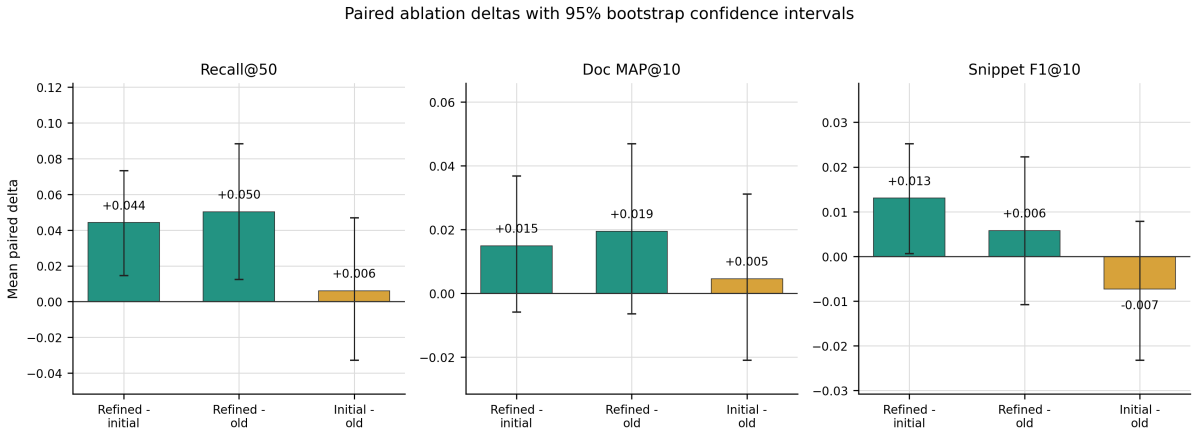

Figure 4: Paired ablation deltas for refined 3-RRF against the matched initial 3-RRF control and the old single-query baseline. Error bars show 95% paired bootstrap confidence intervals. Holm-adjusted p-values were computed separately within each metric across the three reported pairwise system comparisons.

The paired analysis confirmed that the strongest and most stable effect was at the retrieval stage (Figure 4). Compared with the matched initial 3-RRF control, refined 3-RRF improved recall@50 by +0.0442 (95% CI: +0.0146 to +0.0732; Holm-adjusted $p = 0.0063$). The improvement over the old single-query baseline was similar, +0.0502 recall@50 (95% CI: +0.0123 to +0.0883; Holm-adjusted $p = 0.0176$). The matched initial 3-RRF control did not improve meaningfully over the old baseline ($\Delta = +0.0060$, 95% CI: -0.0330 to +0.0468; Holm-adjusted $p = 0.7716$).

For downstream metrics, refined 3-RRF also had positive mean deltas, but these effects were not statistically stable after correction. Against the matched initial 3-RRF control, document MAP@10 improved by +0.0149 (95% CI: -0.0059 to +0.0368; Holm-adjusted $p = 0.4656$), and snippet F1@10 improved by +0.0130 (95% CI: +0.0006 to +0.0252; Holm-adjusted $p = 0.1314$). Against the old baseline, document MAP@10 improved by +0.0194 (95% CI: -0.0064 to +0.0469; Holm-adjusted $p = 0.4656$), while snippet F1@10 improved by +0.0058 (95% CI: -0.0108 to +0.0223; Holm-adjusted $p = 0.7261$). Thus, the downstream metrics were directionally consistent with the retrieval gain, but the clearest statistically stable effect remained recall@50.

The per-question analysis shows that the recall gain was not driven by only a few outliers (Figure 5). Against the matched initial 3-RRF 10-shot control, refined 3-RRF improved 65 questions, underperformed on 24, and tied on 111. Against the old 10-shot baseline, it improved 66 questions, underperformed on 40, and tied on 94. This supports the interpretation that refinement improved the RRF strategy beyond both a previous single-query baseline and a matched unrefined multi-query setup.

Finally, the MeSH prompt ablation did not explain the retrieval gains by itself. Adding MeSH field information to the baseline prompt produced only a small and unstable improvement in the few-shot comparison (recall@50 delta +0.0086) and did not reproduce the gains observed for the refined 3-RRF system. This suggests that the main benefit came from the learned strategy-refinement process and the resulting complementary query strategies, rather than simply exposing the model to MeSH-index fields.

5. Discussion

5.1. Effect of Strategy Refinement on Individual Query Strategies

To better understand where the 3-RRF improvement came from, we also evaluated each strategy branch independently on the 200-question validation set. This analysis showed that the largest gain came from the MeSH-based strategy, while the other two strategies provided smaller but complementary improvements. To make the effect of refinement concrete, we also show for each strategy a representative query pair generated by GPT-5.4-mini on the same question, matched between the initial and refined

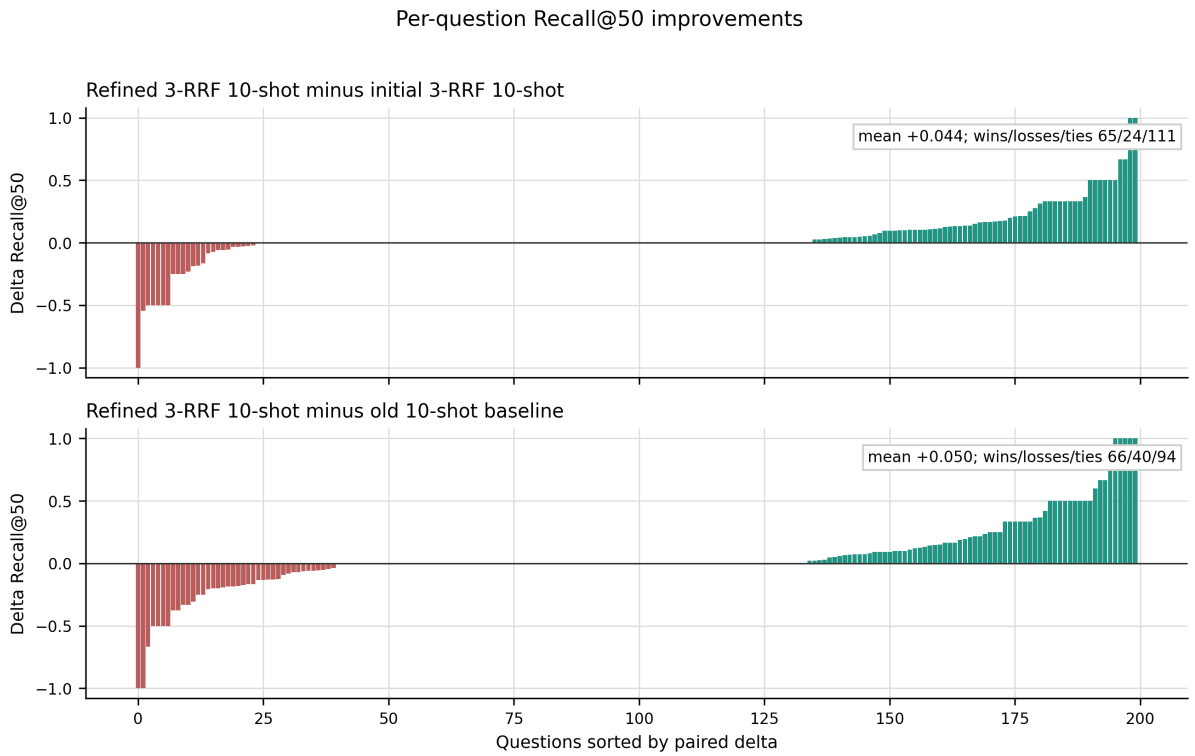

Figure 5: Sorted per-question recall@50 differences for refined 3-RRF against the matched initial 3-RRF control and the old baseline. Green bars indicate improved questions, red bars indicate regressions, and zero-height bars indicate ties.

strategy. In every case shown, the initial query returned no documents while the refined query recovered a full result set, illustrating the drop in zero-result queries reported below.

Literal Claim Core. The initial version of this strategy emphasized exact phrase matching and required all core concepts and relation terms to appear:

```
Extract 2-3 indispensable concepts from the question; make each a
quoted phrase in title and abstract, boost title heavily, require all
concepts with AND, and add only 1-2 high-value relation terms/phrases.
Keep synonym expansion minimal.
```

The refined version kept the literal-claim idea, but made relation terms optional and added syntax-safe variant handling:

```
Search for the smallest plausible literal claim. Make the core
entity/concept blocks mandatory, each as an OR over exact phrase,
full/abbr, and hyphen/spacing variants. Add the answer relation only as
a high-boost optional literal phrase, with at most one tight fielded AND
fallback. Use only ES-safe field:"phrase" or field:(term AND term).
```

This strategy improved from 0.3724 to 0.4057 recall@50 ($\Delta = +0.0333$), with 67 improved questions, 43 regressions, and 90 ties. AP@50 increased by +0.0328 and F1@10 by +0.0310. Search errors fell from 10 to 2, and zero-result queries fell from 13 to 5. For the question “Plaques en Prairie Fauchée are specific to which disease?”, this change turned an empty query into a productive one:

```
INITIAL (0 docs):
(title:"Prairie Fauchee"^5 OR abstract:"Prairie Fauchee")
```

```
AND (title:plaques^5 OR abstract:plaques OR mesh.terms:Plaques)
AND (title:(disease OR disorder OR syndrome) OR abstract:(...))

REFINED (50 docs):
("Plaques en Prairie Fauchee" OR "maltese cross" OR "mycosis fungoides"
 OR "cutaneous T-cell lymphoma" OR "CTCL" OR ...)
```

Forcing the rare foreign phrase and a generic "disease" block as mandatory AND clauses left no matches, so refinement followed the strategy's "answer relation only as a high-boost *optional* literal phrase" rule, dropping the mandatory relation and expressing the entity as an OR over literal variants, including the true answer (CTCL).

Synonym Expansion. The initial version treated retrieval as broad terminology expansion:

```
Break the question into buckets--disease, intervention, molecular
entity, phenotype/outcome, relation--and build one OR group per bucket;
combine buckets with AND. Add mesh.flat versions of major biomedical
concepts as extra coverage. Do not free-float ambiguous abbreviations.
```

The refined version made this expansion more parser-safe and reduced the number of broad mandatory anchors:

```
Treat retrieval as terminology coverage: build parser-safe OR clusters
per entity/relation from canonical terms, aliases/acronyms, variants,
subtypes, and paraphrases. AND only specific disease/subtype/population
/alteration anchors in title^3, abstract, and mesh.flat. Keep broader
class-label terms low-boost unless they are named entities; emit only
valid ES field clauses.
```

This strategy changed less than the MeSH branch. Recall@50 increased from 0.3325 to 0.3662 ($\Delta = +0.0337$), with 63 wins, 41 losses, and 96 ties. AP@50 increased by +0.0090 and F1@10 by +0.0104. Search errors fell from 15 to 2, and zero-result queries fell from 15 to 2. The query pair for "Estimated prevalence of postpartum psychosis." shows the parser-safety effect directly:

```
INITIAL (0 docs):
(title:(postpartum psychosis OR ...)^{3} OR mesh.flat:("Psychosis, Postpartum"))
AND (title:(prevalence OR incidence OR ...)^{3} OR abstract:(...))

REFINED (50 docs):
("postpartum psychosis" OR "puerperal psychosis" OR "postnatal psychosis")
AND (prevalence OR incidence OR epidemiology OR rate* OR frequency)
```

The initial query stacked three mandatory blocks and used the invalid {3} boost syntax, matching nothing; refinement enacted the strategy's "parser-safe OR clusters" and "emit only valid ES field clauses" directives, relaxing to two coordinated clusters with clean, valid syntax.

MeSH-Guided Retrieval. The initial MeSH-Guided strategy attempted to lock the biomedical topic using controlled vocabulary and then add relation terms:

```
Map the main entities to exact MeSH headings in mesh.terms; back them
off with looser lexical support in mesh.flat; then AND a question-type
relation lexicon in title/abstract.
```

The refined version made MeSH anchoring more selective and added syntax-safe title/abstract fallbacks:

```
Use a MeSH-first query: anchor concrete biomedical entities/mechanisms
in ANDed concept groups with ORed mesh.terms:"MH"^4, mesh.flat synonyms,
and low-boost syntax-safe title/abstract fallbacks, including standard
acronyms. Add a compact title/abstract relation frame. Restrict
title/abstract fallback for anchor concepts to syntax-safe quoted
phrases plus true acronyms/exact labels, and do not let broad surrogate
terms alone satisfy the anchor group.
```

This was the main driver of the overall improvement. Recall@50 increased from 0.2506 to 0.4022 ($\Delta = +0.1516$), with 94 improved questions, 22 regressions, and 84 ties. AP@50 increased by +0.0938, F1@10 by +0.0498, and hit@10 by +0.2300. Search errors fell from 4 to 1, and zero-result queries fell from 48 to 3, indicating that refinement made the MeSH-led strategy substantially less brittle. The query pair for "What is IL-35?" illustrates this brittleness and its fix:

```
INITIAL (0 docs):
(mesh.terms:"Interleukin-35"^4 OR mesh.flat:(interleukin-35 OR IL-35 OR IL35))
AND (title:("IL-35" OR IL35 OR "interleukin-35")^2 OR abstract:(...))

REFINED (50 docs):
("IL-35" OR IL35 OR "interleukin-35" OR "interleukin 35"
 OR (interleukin AND 35))
```

The initial query demanded an exact `mesh.terms` heading match alongside a second required block, and because no indexed article carried that heading it returned nothing; refinement applied the strategy's "low-boost, syntax-safe title/abstract fallbacks, including standard acronyms" rule, recovering the concept from free text.

The combined final refined 3-RRF system improved recall@50 from 0.4291 to 0.4733. The individual-strategy analysis suggests that this gain was mainly caused by turning the MeSH-Guided Retrieval strategy into a reliable contributor, while the Literal Claim Core and Synonym Expansion strategies preserved complementary lexical coverage. The downstream Phase A metrics followed the same direction but were less statistically stable than recall@50, suggesting that the refinement primarily improved candidate coverage.

Overall, the explored strategy-refinement approach for query generation with LLMs seems promising. Future work could investigate what upper bounds for this refinement exist and if there are optimal parameters, such as the number of initial questions or rounds that maximize the potential recall gains efficiently.

6. Conclusion

We presented a strategy-refinement approach in which LLMs iteratively improve reusable, textual keyword-search instructions for biomedical retrieval. Rather than fine-tuning model weights or optimizing individual queries, the method optimizes prompt-level search strategies that can be inspected and reused to generate Boolean queries as transparent keyword searches. In the preliminary BioASQ14 results[3], the refined 3-RRF systems achieved our strongest Phase A retrieval results and high-ranking snippet MAP scores. However, improvements in retrieval did not consistently lead to better Phase A+ or Phase B answer metrics, indicating that further refinements for our snippet reranking and question-answering approaches are needed.

[3] https://participants-area.bioasq.org/results/14b/phaseA/

In a separate ablation, refined 3-RRF produced a significant recall@50 improvement over both a strong single-query baseline and an unrefined multi-query RRF setup. This is notable because the strategies were refined from only a small gold-standard development set, yet generalized to older held-out BioASQ questions. Overall, LLM-guided textual strategy refinement appears to be a practical way to improve transparent biomedical retrieval, especially where reproducibility and auditability remain important.

Acknowledgments

We thank the BioASQ organizers for their continued work in running the challenge. This work is supported by the German Research Foundation (DFG) as part of the NFDIxCS consortium (Grant number: 501930651).

Declaration on Generative AI

The authors used the following generative-AI tools and services while preparing this paper[4]:

- **OpenAI ChatGPT** (GPT-5.5 and GPT-5.4) for drafting support, LaTeX formatting, paraphrasing, and rewording.
- **OpenAI Codex** (GPT-5.5) for coding assistance, code review, and plotting.
- **LanguageTool** for spellchecking, grammar suggestions, paraphrasing, and rewording.

After using these tools and services, the authors reviewed, edited, and verified the generated or tool-assisted content as needed. The authors take full responsibility for the final content of the publication.

References

[1] C. Kaiser, J. Kaiser, R. Schallner, S. Schneider, A New Era of Online Search? A Large-Scale Study of User Behavior and Personal Preferences during Practical Search Tasks with Generative AI versus Traditional Search Engines, in: Extended Abstracts of the CHI Conference on Human Factors in Computing Systems, CHI EA '25, Association for Computing Machinery, New York, NY, USA, 2025, pp. 1–7. doi:10.1145/3706599.3720123.

[2] M. Rethlefsen, S. Kirtley, S. Waffenschmidt, A. P. Ayala, D. Moher, M. Page, J. Koffel, PRISMA-S: an extension to the PRISMA Statement for Reporting Literature Searches in Systematic Reviews, Systematic Reviews 10 (2021). doi:10.1186/s13643-020-01542-z.

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

[4] A. Nentidis, G. Katsimpras, A. Krithara, M. Krallinger, M. Rodríguez-Ortega, E. Rodriguez-López, N. Loukachevitch, I. Rozhkov, E. Tutubalina, D. Dimitriadis, V. Patsiou, G. Tsoumakas, G. Giannakoulas, A. Bekiaridou, A. Samaras, G. M. Di Nunzio, N. Ferro, S. Marchesin, M. Martinelli, G. Silvello, G. Paliouras, Overview of BioASQ 2026: The fourteenth BioASQ Challenge on Large-Scale Biomedical Semantic Indexing and Question Answering, volume Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Seventeenth International Conference of the CLEF Association (CLEF 2026) of *Lecture Notes in Computer Science*, Springer, 2026.

[5] A. Liberati, D. Altman, J. Tetzlaff, C. Mulrow, P. Gøtzsche, J. Ioannidis, M. Clarke, P. Devereaux, J. Kleijnen, D. Moher, The PRISMA Statement for Reporting Systematic Reviews and Meta-Analyses of Studies That Evaluate Health Care Interventions: Explanation and Elaboration, Journal of clinical epidemiology 62 (2009) e1–34. doi:10.1016/j.jclinepi.2009.06.006.

[4] https://ceur-ws.org/GenAI/Policy.html

[6] M. L. Rethlefsen, T. J. Brigham, C. Price, D. Moher, L. M. Bouter, J. J. Kirkham, S. Schroter, M. P. Zeegers, Systematic review search strategies are poorly reported and not reproducible: a cross-sectional metaresearch study, Journal of Clinical Epidemiology 166 (2024) 111229. doi:10.1016/j.jclinepi.2023.111229.

[7] H. Scells, G. Zuccon, Generating Better Queries for Systematic Reviews, in: Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR '18, Association for Computing Machinery, New York, NY, USA, 2018, pp. 475–484. doi:10.1145/3209978.3210020.

[8] H. Scells, G. Zuccon, B. Koopman, Automatic Boolean Query Refinement for Systematic Review Literature Search, WWW '19, Association for Computing Machinery, New York, NY, USA, 2019, p. 1646–1656. doi:10.1145/3308558.3313544.

[9] H. Scells, G. Zuccon, B. Koopman, A comparison of automatic Boolean query formulation for systematic reviews, Inf. Retr. J. 24 (2021) 3–28. doi:10.1007/S10791-020-09381-1.

[10] S. Wang, H. Scells, B. Koopman, G. Zuccon, Automated MeSH Term Suggestion for Effective Query Formulation in Systematic Reviews Literature Search, Intelligent Systems with Applications 16 (2022) 200141. doi:10.1016/j.iswa.2022.200141.

[11] S. Wang, H. Scells, B. Koopman, G. Zuccon, Can ChatGPT Write a Good Boolean Query for Systematic Review Literature Search?, in: Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '23, Association for Computing Machinery, New York, NY, USA, 2023, pp. 1426–1436. doi:10.1145/3539618.3591703.

[12] M. Staudinger, W. Kusa, F. Piroi, A. Lipani, A. Hanbury, A Reproducibility and Generalizability Study of Large Language Models for Query Generation, in: Proceedings of the 2024 Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region, SIGIR-AP '24, Association for Computing Machinery, New York, NY, USA, 2024, pp. 186–196. doi:10.1145/3673791.3698432.

[13] S. Wang, H. Scells, B. Koopman, G. Zuccon, Reassessing Large Language Model Boolean Query Generation for Systematic Reviews, in: Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '25, Association for Computing Machinery, New York, NY, USA, 2025, pp. 3296–3305. doi:10.1145/3726302.3730329.

[14] Y. Zhou, A. I. Muresanu, Z. Han, K. Paster, S. Pitis, H. Chan, J. Ba, Large Language Models are Human-Level Prompt Engineers, in: The Eleventh International Conference on Learning Representations, 2023. URL: https://openreview.net/forum?id=92gvk82DE-.

[15] R. Pryzant, D. Iter, J. Li, Y. Lee, C. Zhu, M. Zeng, Automatic prompt optimization with "gradient descent" and beam search, in: H. Bouamor, J. Pino, K. Bali (Eds.), Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Singapore, 2023, pp. 7957–7968. doi:10.18653/v1/2023.emnlp-main.494.

[16] X. Wang, C. Li, Z. Wang, F. Bai, H. Luo, J. Zhang, N. Jojic, E. Xing, Z. Hu, PromptAgent: Strategic Planning with Language Models Enables Expert-level Prompt Optimization, in: The Twelfth International Conference on Learning Representations, 2024. URL: https://openreview.net/forum?id=22pyNMuIoa.

[17] O. Khattab, A. Singhvi, P. Maheshwari, Z. Zhang, K. Santhanam, S. V. A, S. Haq, A. Sharma, T. T. Joshi, H. Moazam, H. Miller, M. Zaharia, C. Potts, DSPy: Compiling Declarative Language Model Calls into State-of-the-Art Pipelines, in: The Twelfth International Conference on Learning Representations, ICLR 2024, 2024. URL: https://openreview.net/forum?id=sY5N0zY5Od.

[18] G. V. Cormack, C. L. A. Clarke, S. Büttcher, Reciprocal Rank Fusion Outperforms Condorcet and Individual Rank Learning Methods, in: Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '09, Association for Computing Machinery, New York, NY, USA, 2009, pp. 758–759. doi:10.1145/1571941.1572114.

[19] A. Krithara, A. Nentidis, K. Bougiatiotis, G. Paliouras, BioASQ-QA: A Manually Curated Corpus for Biomedical Question Answering, Scientific Data 10 (2023) 170. doi:10.1038/s41597-023-02068-4.

[20] S. Ateia, U. Kruschwitz, Can Language Models Critique Themselves? Investigating Self-Feedback

for Retrieval Augmented Generation at BioASQ 2025, in: G. Faggioli, N. Ferro, P. Rosso, D. Spina (Eds.), Working Notes of the Conference and Labs of the Evaluation Forum, CLEF 2025, Madrid, Spain, 9–12 September 2025, volume 4038 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2025, pp. 148–164. URL: https://ceur-ws.org/Vol-4038/paper_9.pdf.