Few but Deep vs. Many but Shallow: Comparing Licensed and Free Tool Sets for Agentic Biomedical Question Answering

Nhung T.H. Nguyen, Ahmed Elnaggar, Aditya Kashyap and Timothy Schultz

Machine Intelligence, DDSAI, Innovative Medicine, Johnson & Johnson, Titusville, New Jersey, USA

Abstract

Large language model (LLM) agents can address complex biomedical questions by invoking external tools, yet a fundamental design question remains open: should an agent be equipped with a small set of carefully engineered, domain-specific tools that return rich results, or a large catalogue of freely available general-purpose tools that offer broad coverage but shallower outputs? We present a systematic comparison of two tool-access strategies on the BioASQ benchmark, a standard biomedical question-answering challenge spanning yes/no, factoid, list, and summary question types. The *licensed-tool* configuration equips a ReAct agent with a compact set of curated tools optimised for biomedical literature, delivering full retrieval content and advanced query capabilities. The *free-tool* configuration instead exposes a multi-agent pipeline to a suite of 136 freely accessible MCP tools spanning diverse online sources, trading result depth for breadth of coverage. We first evaluate ideal answer quality on the BioASQ 13B dataset using RAGAS and LLM-based accuracy metrics, then report results on BioASQ Task 14B – Phase A+ as provided by the shared task organisers. Our findings reveal that licensed tools yield higher retrieval precision and faithfulness, whereas free tools did not produce consistent accuracy gains and incurred substantially longer execution times, underscoring the importance of tool curation in agentic biomedical question answering.

Keywords

Biomedical Question Answering, MCP Tools, LLM Reasoning, Agentic AI

1. Introduction

Large language models (LLMs) have demonstrated strong capabilities in biomedical question answering, yet their direct application to complex clinical and scientific questions is limited by a tendency to hallucinate and a lack of access to up-to-date literature [1]. Retrieval-augmented generation (RAG) and agentic frameworks have emerged as promising solutions, enabling LLMs to ground their answers in retrieved evidence and to invoke external tools iteratively [2, 3, 4, 5]. Recent work has further demonstrated that multi-agent architectures can substantially extend these capabilities in the biomedical domain: BioMedAgent [6], for instance, introduces a self-evolving multi-agent framework that autonomously chains bioinformatics tools into executable workflows for complex data analysis tasks, highlighting the potential of agentic systems to handle specialised biomedical reasoning. However, as these systems grow in sophistication, a fundamental design question arises: should an agent be given a small set of high-quality, domain-specific tools with rich output at engineering and licensing cost? or should it be equipped with a large catalogue of freely available tools that trade depth of results for breadth of coverage? This question has direct practical consequences for retrieval quality, answer precision, and system accessibility, but it remains largely unaddressed in the biomedical question answering (QA) literature.

In this study, we use two contrasting tool-access strategies in agentic system design. In the *licensed-tool* paradigm, the agent is equipped with a compact, curated set of purpose-built tools that return full retrieval content and support advanced query strategies such as query expansion and LLM-based reranking. While these tools require custom engineering and may carry licensing costs, they deliver information-rich results well-suited to precise biomedical reasoning [5]. In the *free-tool* paradigm, the

CLEF 2026: Conference and Labs of the Evaluation Forum, September 21-24, Jena, Germany

NNguye63@its.jnj.com (N. T.H. Nguyen); AElnagg1@its.jnj.com (A. Elnaggar); AKashy12@its.jnj.com (A. Kashyap); TSchult4@its.jnj.com (T. Schultz)

agent has access to a large catalogue of freely available MCP-compatible tools, spanning academic databases, web search engines, and specialised biomedical repositories. Although this catalogue offers broad coverage, these free tools usually return only document snippets rather than full content, constraining the depth of evidence available to the agent at each reasoning step. This approach is motivated by recent work on open tool ecosystems and skill retrieval [7, 8], which highlights the potential of giving agents access to diverse, community-maintained capabilities.

In this paper, we present a comparison of these two tool-access strategies, evaluated on the BioASQ shared task [9]. BioASQ Task 14B (Task A+) [10] requires systems to answer biomedical questions spanning four types (yes/no, factoid, list, and summary) using evidence retrieved autonomously from PubMed. The task was structured across four submission batches, providing a repeated evaluation setting that allowed us to observe system behavior under varied question distributions. We instantiate both strategies and evaluate them alongside a standalone LLM baseline.

The main contributions of this paper are as follows:

- We describe two agentic tool-access configurations of the Med.ai ASK framework [5]: a licensed-tool agent and a free-tool agent, and our refinements across four official BioASQ Task 14B submission batches.
- We report experimental results as well as shared task results comparing a standalone LLM baseline and both tool-access configurations across all BioASQ question types, providing an empirical basis for the quality-versus-quantity tool design question.
- We analyse the trade-offs between tool depth and tool breadth, and discuss implications for the practical design of agentic frameworks under real-world resource constraints.

2. Methods

We implement two distinct agentic pipeline configurations that differ fundamentally in the size and nature of their tool sets: a *licensed-tool* pipeline equipped with a small number of curated, high-depth tools, and a *free-tool* pipeline equipped with a large catalogue of freely accessible tools.

2.1. Licensed-Tool Pipeline

As illustrated in Figure 1, the licensed-tool pipeline implements a single ReAct-style agent that iteratively selects and invokes a compact set of purpose-built tools to answer a biomedical question [5]. The tools are implemented and exposed as LangChain-compatible wrappers to the agent. At each reasoning step, the agent decides which tool to call next based on the question and the accumulated intermediate observations. The agent has rich evidence available at every step thanks to the returned results of each tool. Once sufficient evidence has been gathered, the agent invokes a final answer synthesis tool that consolidates the retrieved content into a structured answer in the format required by BioASQ (yes/no, factoid entity list, ranked list, or ideal summary). The prompt we used for this React-style agent is available in Appendix A.

2.2. Licensed Tool Set (Few but Deep)

2.2.1. Document Indexing and Retriever

We internally ingest 5 textual knowledge bases:

- PubMed: 38 million citations for biomedical literature from MEDLINE and life science journals.
- CCC OpenAccess: A licensed collection of 5 million full-text biomedical papers.
- Elsevier ebooks: A licensed collection of 20,000 ebooks about biology.
- NIH Grant Database: A collection of 1 million research grants submitted to National Institutes of Health (NIH).

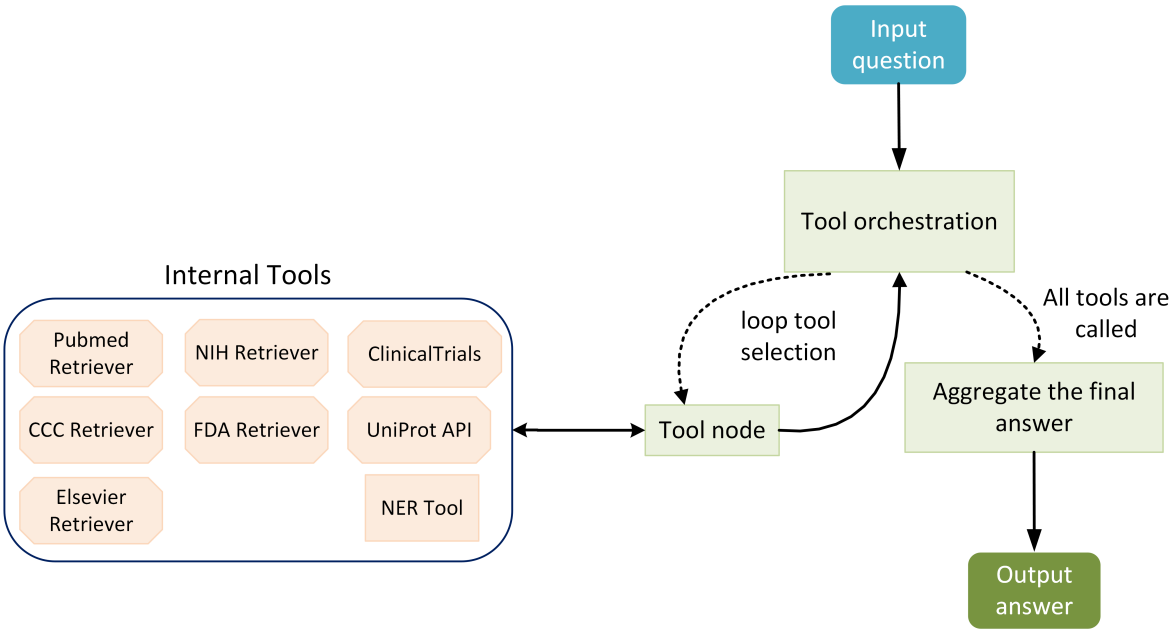

Figure 1: Architecture of the licensed-tool pipeline. A single ReAct agent iteratively selects from a compact set of curated tools to retrieve and synthesise biomedical evidence. Figure adapted from [5].

- FDA Drug Labels: A collection of over 150,000 labelling documents for human prescription drug, nonprescription drugs, and labelling for other products .

At the retrieving step, we leverage the Weaviate python library to retrieve relevant documents for each input query. We use two α values (0.25 and 0.75) to balance the influence of keyword matching and semantic similarity and set the number of retrieved documents to $k = 40$. We then employ a two-stage LLM-based filter-and-rerank strategy. In the first stage, retrieved documents were evaluated in parallel by a LLM, which assigned each document a relevance label of high, medium, low, or irrelevant relative to the user query. Documents labelled irrelevant were discarded, and the remaining candidates were collected in priority order (high $\rightarrow$ medium $\rightarrow$ low) to form a shortlist of up to $2\times N$ relevant documents. In the second stage, a second LLM call ranks the candidates by relevance and returned only the top N documents in ranked order. Please check Appendix B for the prompts we use in this step.

In addition to the internal retrievers, we also implement the following tools:

- ClinicalTrials.gov API to access clinical research studies and information about their results.
- UniProt API to retrieve information about proteins.
- A named entity recognition (NER) tool by wrapping the Kazu framework [11] and a bespoke gene synonym expansion step using NCBI gene data [12]. This NER tool can identify and normalize to controlled vocabularies of biomedical entities in 12 different categories.

2.2.2. Enhanced PubMed Search

To enhance document retrieval in the case of PubMed search, we applied a three-leg retrieval with query expansion prior to document retrieval. Given the user's natural-language question, two parallel LLM calls were issued: the first performs keyword extraction, prompting the model to distil the query into a concise list of domain-specific terms optimised for lexical matching. The second LLM performs narrative expansion, instructing the model to rewrite the query as a short descriptive paragraph articulating the experimental context, plausible mechanisms, and the characteristics of relevant literature. Three Weaviate searches were then executed: one against the original query using default hybrid retrieval parameters, one against the extracted keywords with a BM25-weighted configuration ($\alpha = 0.1$) to favour exact term matches, and one against the expanded narrative with a vector-weighted configuration

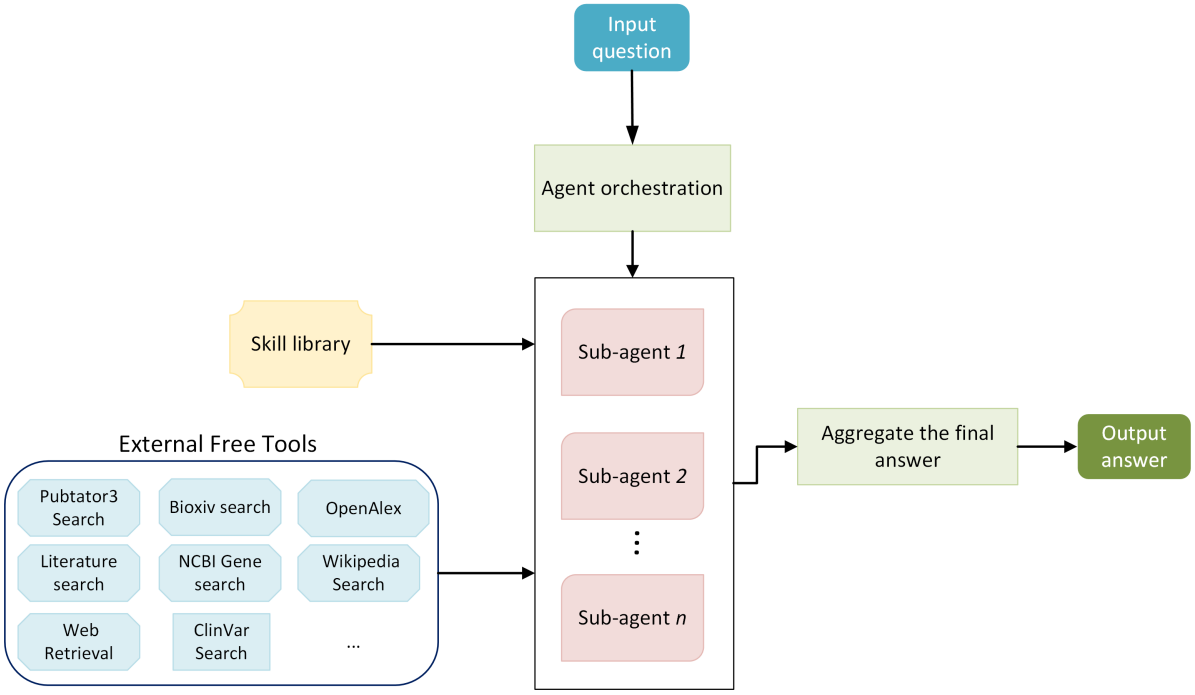

Figure 2: Architecture of the free-tool pipeline. An orchestration LLM decomposes the question, dispatches parallel skill-guided sub-agents over a catalogue of 136 free MCP tools, and aggregates their answers. Each sub-agent is a ReAct agent.

($\alpha = 0.8$) to favour semantic similarity. The result sets from all three searches were merged and deduplicated by document identifier, yielding a unified candidate pool that balances terminological precision with broad semantic coverage before downstream filtering and reranking. Please check Appendix C for the prompts we use in this step.

2.3. Free-Tool Pipeline

The free-tool pipeline replaces the compact licensed tool set with a large catalogue of 136 freely accessible MCP-compatible tools, giving the agent broad coverage across heterogeneous online sources without custom engineering or licensing costs. To manage this large tool space, the pipeline adopts a hierarchical, multi-agent architecture in which a supervisor decomposes the question and dispatches specialised sub-agents, each operating over a relevant subset of the free tool catalogue as illustrated in Figure 2.

At inference time, the pipeline proceeds in three stages. First, an orchestration LLM receives the input question, analyses its intent and domain scope, and decomposes it into a set of sub-tasks. For each sub-task it instantiates a dedicated sub-agent and dispatches them in parallel. Second, each sub-agent is initialised with a small set of skills retrieved from a pre-defined skill library via embedding-based similarity search: the question embedding is compared against skill embeddings and the top-k most relevant skills are injected directly into the sub-agent's system prompt to guide tool selection within the large free-tool catalogue. Third, equipped with its assigned skills, each sub-agent executes its own internal ReAct loop over the MCP free-tool set. Once all sub-agents have produced their answers, the supervisor LLM aggregates the results and generates a single final answer. To balance costs and effectiveness, we limited the number of sub-agents to three. All prompts used in this pipeline are available in Appendix D.

2.3.1. Skill Library Construction

The `pubmed-database` skill exposes the NCBI E-utilities API, enabling construction of advanced Boolean and Medical Subject Headings (MeSH) queries against PubMed, with support for systematic reviews, batch retrieval of abstracts and full-text articles, and citation management via direct REST calls. Three Biopython-based [13] skills provide access to the ESearch, EFetch, and ELink utilities respectively, supporting keyword-based record discovery, retrieval of sequences and GenBank entries, and cross-database navigation (e.g., from genes to associated publications).

For systematic literature synthesis, the `literature-review` skill orchestrates parallel searches across multiple academic databases, including PubMed, bioRxiv, arXiv, and Semantic Scholar, generating professionally formatted documents with verified citations in standard styles (APA, Nature, Vancouver). The `biorxiv-database` skill supports preprint retrieval from bioRxiv with filtering by keyword, author, subject category, and date range, including full-text PDF access. The `openalex-database` skill queries the OpenAlex [14] catalogue of over 240 million scholarly works, supporting bibliometric analysis, citation tracking, and open-access discovery.

We mainly leverage scientific skills published here: https://github.com/K-Dense-AI/scientific-agent-skills/tree/main.

2.4. Free Tool Set (Many but Shallow)

An MCP server exposes a suite of 136 freely accessible tools for programmatic retrieval of information from heterogeneous online sources. For biomedical literature research, the following tools are of particular relevance. `literature_search` serves as the primary retrieval interface, aggregating results from multiple academic databases, e.g., PubMed and Semantic Scholar, within a single query. `pubtator3_search` augments this capability with AI-powered named entity recognition, enabling concept-level retrieval of genes, diseases, chemicals, and genomic variants from article text. Preprint literature is accessible via `biorxiv_search`, which indexes both the bioRxiv and medRxiv servers, while broader bibliometric and citation analysis is supported through `openalex_search`, which covers over 240 million scholarly works. For full-text access, `web_retrieval` and `unpaywall_lookup` retrieve page content and legally available open-access versions of articles, respectively; DOI validation and citation metadata are provided by `crossref_validate`. Specialised database tools such as `ncbi_gene_search`, `clinvar_search`, and `geo_dataset_search`, extend retrieval to gene annotations, clinically interpreted sequence variants, and gene expression datasets hosted at NCBI. A set of general-purpose web search backends, including `tavily_search`, `brave_search`, `serper_search`, `duckduckgo_search`, and `searxng_search`, provide redundant coverage of the open web, while `rss_feed` enables continuous monitoring of new publications via journal RSS and Atom feeds.

3. Experiments

3.1. System Settings

We evaluate five system configurations as follows. The **Baseline** system uses a standalone LLM to generate answers directly from the question, with no external retrieval or tool invocation. **Tool-GPT** and **Tool-Claude** instantiate the licensed-tool pipeline with GPT-4.1/GPT-5.1/GPT-5.4, and Claude Sonnet 4.6, respectively, allowing us to isolate the effect of the base model while holding the tool set constant. Meanwhile, the free-tool configuration (**Skill-F**) uses Claude Sonnet 4.6 as the base LLM and 136 free MCP tools. We additionally tested a hybrid configuration (**Skill-L**) combining the free-tool pipeline architecture with licensed tools to ensure fair comparison between two sets of tools. The configurations are summarised in Table 1.

Table 1
System configurations evaluated on BioASQ Task 13B.

System	Family	Base LLM	Tool Set	Description
Baseline	Standalone LLM	Claude Sonnet 4.6	None	Direct answer generation, no retrieval or tools
Tool-GPT	Licensed-tool	GPT-4.1/GPT-5.4	Licensed (8 tools)	Single ReAct agent over LangChain tools
Tool-Claude	Licensed-tool	Claude Sonnet 4.6	Licensed (8 tools)	Single ReAct agent over LangChain tools
Skill-Free	Free-tool	Claude Sonnet 4.6	Free (136 tools)	Multi ReAct sub-agents over MCP tools
Skill-Licensed	Licensed-tool	Claude Sonnet 4.6	Licensed (8 tools)	Multi ReAct sub-agents over MCP tools

3.2. Evaluation Metrics

We employ two complementary evaluation protocols to assess system performance. First, we only focus on evaluating ideal answers by utilizing **RAGAS** [15] to measure retrieval quality dimensions, specifically faithfulness, answer relevancy, and context recall. We also calculate **LLM-based accuracy**[1] to have a direct comparison between the generated answers and the ground truth. These metrics provide comprehensive assessment of answer quality that substantiates the BioASQ lexical metrics. We compute these metrics on the *BioASQ Shared Task 13B* testing set [16], where we have ground truth answers. Second, we report the **BioASQ metrics**: macro-averaged F1 for yes/no questions, Mean Reciprocal Rank (MRR) for factoid questions, mean F-measure for list questions, and ROUGE-SU4 F1 for ideal answers, computed on the official *BioASQ Shared Task 14B* testing set [10] by the task organizers. It is noted that unlike RAGAS and LLM-accuracy, those metrics are lexical ones, i.e., they mainly used exact string matching.

4. Results

4.1. Experimental Results on BioASQ-13B

Table 2 summarizes the question answering performance across system configurations on BioASQ Task 13B testing set. We additionally report the average execution time per question (in seconds) to characterise the computational overhead of each configuration.

The results within the licensed tool configurations show that Tool-Claude Sonnet 4.6 consistently achieved the strongest performance, attaining the highest accuracy (96.36%) along with high context recall and faithfulness. Tool-GPT-4.1 delivered solid and balanced results (94.03% accuracy), while Tool-GPT-5.1 did not outperform GPT-4.1, exhibiting lower accuracy (88.34%) and reduced context recall and faithfulness. This suggests that newer model versions do not necessarily translate into better performance in retrieval-augmented biomedical QA under the standard setting.

When incorporating enhanced PubMed search (as described in Section 2.2.2), a different pattern emerges. For GPT-4.1, context precision, context recall, and accuracy all decreased, while faithfulness improved substantially to 95.66%. With GPT-5.4, the system performed slightly better than GPT-4.1 in this setting, achieving higher context recall and faithfulness with comparable accuracy. Overall, the enhanced retrieval improves grounding and faithfulness but can introduce noise that lowers precision and downstream answer accuracy.

The Skill-Free configuration, which relied exclusively on freely available tools, did not demonstrate performance improvements over the baseline. Although its overall accuracy remained comparable to the baseline (87.68% vs. 87.95%), it exhibited substantially lower context precision (33.79%) and only moderate faithfulness (75.5%). These findings suggest that the evidence retrieved by the free-tool

[1]Please see Appendix E for the prompt used in this evaluation.

Table 2
Ideal answer performance across system configurations on BioASQ 13B testing set. Each reported score is averaged over two independent runs. It is noted that RAGAS metrics are not applicable to the Baseline system, which does not perform retrieval. Con. Pre.: context precision; Con. Re.: context recall; Faith.: faithfulness; Acc.: LLM-Accuracy; Exec. Time: execution time.

Family	System	Con. Pre.	Con. Re.	Faith.	Acc.	Exec. Time
No Tool	Baseline	–	–	–	87.95	6.96
Single-agent with Licensed Tools	Tool-GPT-4.1	93.55	70.78	91.62	94.03	24.48
	Tool-GPT-5.1	91.65	68.91	88.82	88.34	36.43
	Tool-Claude Sonnet 4.6	92.23	80.49	91.05	96.36	66.39
	With enhanced PubMed search					
	Tool-GPT-4.1	87.37	84.37	95.66	91.49	56.38
	Tool-GPT-5.4	90.90	91.60	98.29	91.62	78.03
	Tool-Claude Sonnet 4.6	88.68	89.14	98.21	95.24	101.04
Multi-agent with Skills	Skill-Free	33.79	74.51	75.50	87.68	192.29
	Skill-Licensed	62.77	70.22	84.61	87.06	227.59

configuration was less relevant and less well-aligned with the generated responses. This interpretation is further supported by the improved context precision and faithfulness observed in the Skill-Licensed experiment, which utilized licensed tools. Collectively, these results indicate that greater tool availability alone does not necessarily translate into better performance; rather, the quality and curation of available tools play a critical role in retrieval effectiveness and downstream system performance.

In terms of execution time, there is a clear trade-off between performance and computational cost. Licensed tool configurations operate within moderate time ranges, whereas enhanced PubMed search significantly increases latency (about two times). The two Skill systems exhibit by far the longest execution time, likely due to the overhead of evaluating and selecting from a large set of available skills.

4.2. Shared Task Results

For our submissions to the shared task, we used the first four configurations described in Table 1. We submitted two different runs of the Skill-Free configuration. Since the enhanced PubMed search strategy and GPT5.4 were implemented after Batch 3's deadline, we could only use them for Batch 4 submissions. Moreover, we did not submit the Skill-Licensed configuration to the shared task because this experiment was conducted after the official submission deadline.

4.3. Results on Exact Answers

For yes/no questions, we can see that all systems performed strongly across batches, with macro F1 scores generally ranging from 0.88 to 0.94 (cf. Table 3). The most notable result came in Batch 3, where Skill-Free-Run1 achieved a perfect macro F1 of 1.000, the highest single-batch yes/no result across all our runs. However, this was not replicated across other batches, where the licensed-tool systems and even the Baseline remained more consistently competitive (e.g., Baseline ranked 1st in Batch 4 with F1 = 0.9352, and 2nd in Batch 3 with F1 = 0.8952).

For factoid questions (MRR), the licensed-tool configurations held a clear advantage: Tool-Claude ranked 5th out of 24 in Batch 2 (MRR = 0.3417) and the Baseline ranked 4th out of 13 in Batch 4 (MRR = 0.3030), while free-tool systems consistently ranked 8th or lower. This pattern aligns with the expectation that factoid retrieval benefits from the richer full-document content returned by licensed tools, enabling more precise entity identification.

For list questions, however, the picture is more mixed. The licensed-tool configurations achieved their strongest list F1 in Batch 4, where Tool-Claude (with enhanced PubMed search) produced the best list result across all our submissions. Tool-GPT4.1 also performed well on list questions in Batch 3

Table 3
Exact answer performance across four BioASQ Task 14B batches. Rank is among all participants in that batch. Bolded numbers indicate the best performing system among the submitted runs. * indicates runs with enhanced PubMed search.

Batch	System	Yes/No		Factoid		List	
		Macro F1	Rank	MRR	Rank	F1	Rank
1	Baseline	**0.8819**	**4/16**	**0.1739**	**22/31**	0.1244	36/51
	Tool-GPT4.1	**0.8819**	**4/16**	**0.1739**	**22/31**	0.0825	43/51
	Tool-Claude	0.8786	5/16	**0.1739**	**22/31**	0.1423	29/51
	Skill-Free-Run2	0.8786	5/16	**0.1739**	**22/31**	**0.1486**	**28/51**
2	Baseline	0.7857	10/15	0.3250	6/24	0.2026	39/61
	Tool-GPT4.1	0.7981	9/15	0.3000	8/24	**0.2061**	**37/61**
	Tool-Claude	0.8833	4/15	**0.3417**	**5/24**	0.1461	49/61
	Skill-Free-Run1	**0.9443**	**2/15**	0.3000	8/24	0.1875	41/61
	Skill-Free-Run2	0.8833	4/15	0.3000	8/24	0.1976	40/61
3	Baseline	0.8952	2/14	**0.3725**	**9/22**	0.1544	54/62
	Tool-GPT4.1	0.8952	2/14	0.3235	13/22	**0.2303**	**36/62**
	Skill-Free-Run1	**1.0000**	**1/14**	0.3529	11/22	0.2254	39/62
	Skill-Free-Run2	0.8036	5/14	0.3676	10/22	0.2110	44/62
4	Baseline	**0.9352**	**1/10**	**0.3030**	**4/13**	0.2040	49/61
	Tool-GPT5.4*	**0.9352**	**1/10**	0.0909	12/13	0.2397	44/61
	Tool-Claude*	**0.9352**	**1/10**	0.1818	8/13	**0.3138**	**28/61**
	Skill-Free-Run1	0.8730	2/10	0.2727	5/13	0.1266	54/61
	Skill-Free-Run2	**0.9352**	**1/10**	0.2727	5/13	0.1747	51/61

(F1 = 0.2303, rank 36/62). The free-tool configurations (Skill-Free-Run1 and Skill-Free-Run2) remained broadly mid-table on list questions.

Taken together, the exact answer results suggest that the licensed-tool approach (with its compact but content-rich retrieval) provides more reliable performance across question types and batches, whereas the free-tool systems occasionally produced strong individual results (notably the perfect yes/no score in Batch 3) but lacked the consistency needed for robust task performance.

4.4. Results on Ideal Answers

Table 4 reports ideal answer performance measured by ROUGE-2 and ROUGE-SU4 F1; however, these are temporary lexical scores computed automatically and should be interpreted with caution, as the official ideal answer evaluation for BioASQ Task 14B relies on manual assessment by domain experts.

With that caveat in mind, the overall trend across batches shows that all systems rank in the lower half of the field on ROUGE-based metrics, with F1 scores clustered tightly between roughly 0.08 and 0.13, and no configuration achieving a top-20 rank in any batch. Notably, the Baseline system is the strongest performer on ideal answers in Batches 1 through 3, and the licensed-tool configurations with enhanced retrieval (Tool-GPT5.4* and Tool-Claude*) narrow the gap only in Batch 4. The free-tool Skill configurations consistently rank below both the Baseline and the licensed-tool systems across all batches.

This trend is the reverse of what we might expect from the experimental results in Table 2, where licensed-tool agents substantially outperformed the Baseline on faithfulness and context recall under RAGAS evaluation. The discrepancy likely reflects a fundamental mismatch between lexical overlap metrics and generation quality. ROUGE penalises longer, more elaborated answers of the kind that retrieval-augmented agents tend to produce, while rewarding outputs that happen to share surface n-grams with the reference, which the no-retrieval Baseline naturally generates. We therefore place greater weight on the RAGAS and LLM-based accuracy results from Table 2 as a better indicator of

Table 4
Ideal answer (summary) performance across four BioASQ Task 14B batches, measured by ROUGE-2 and ROUGE-SU4. Rank is among all participants in that batch. Bolded numbers indicate the best performing system among the submitted runs. It is noted that those metrics are just temporary results; the official results for ideal answers will be relied on manual evaluation by domain experts. * indicates runs with enhanced PubMed search.

Batch	System	ROUGE-2			ROUGE-SU4		
		Recall	F1	Rank	Recall	F1	Rank
1	Baseline	0.0831	**0.1218**	**24/52**	0.0664	0.1014	42/53
	Tool-GPT4.1	0.2497	0.0889	45/52	0.2764	0.0986	45/53
	Tool-Claude	**0.2590**	0.0924	41/52	0.2863	**0.1019**	**40/53**
	Skill-Free-Run2	0.2465	0.0797	47/52	0.2780	0.0901	48/53
2	Baseline	**0.2874**	**0.2874**	**47/60**	**0.3030**	**0.1129**	**46/60**
	Tool-GPT4.1	0.2485	0.2485	51/60	0.2824	0.1095	49/60
	Tool-Claude	0.2650	0.2650	54/60	0.2897	0.1030	54/60
	Skill-Free-Run1	0.2617	0.2617	48/60	0.2856	0.1096	48/60
	Skill-Free-Run2	0.1986	0.2644	52/60	0.2848	0.1084	51/60
3	Baseline	**0.2373**	0.0995	43/59	**0.2661**	0.1150	39/61
	Tool-GPT4.1	**0.2373**	**0.1022**	**40/59**	0.2643	**0.1153**	**38/61**
	Skill-Free-Run1	0.2024	0.0853	48/59	0.2286	0.0982	50/61
	Skill-Free-Run2	0.1986	0.0829	49/59	0.2231	0.0949	51/61
4	Baseline	**0.2352**	0.0910	51/63	**0.2682**	0.1048	47/59
	Tool-GPT5.4*	0.2276	**0.1028**	**46/63**	0.2499	**0.1151**	**41/59**
	Tool-Claude*	0.2127	0.1018	47/63	0.2385	0.1137	43/59
	Skill-Free-Run1	0.2043	0.0815	56/63	0.2217	0.0930	51/59
	Skill-Free-Run2	0.2011	0.0818	55/63	0.2173	0.0918	52/59

ideal answer quality.

5. Discussion

The results across both experimental and shared task evaluations point to a consistent advantage of the licensed-tool configuration on semantically grounded quality metrics such as faithfulness and LLM-based accuracy. We attribute this primarily to the difference in output granularity between the two tool sets. Licensed tools return full retrieved content, giving the agent rich, sentence-level evidence to reason over at each ReAct step. Free tools, by contrast, usually return some snippets, which means the agent must commit to an answer on the basis of partial information and cannot perform fine-grained cross-document reasoning.

A second factor is the cognitive overhead imposed by a large tool catalogue. With 136 tools available, the agent must spend part of its reasoning budget selecting among redundant or overlapping retrieval backends, rather than refining its evidence. The skill injection mechanism partially mitigates this by guiding sub-agents toward relevant tools, but it does not eliminate the fundamental cost of navigating a large action space. The licensed-tool agent, operating over a small, purpose-built set, can allocate more of its context and reasoning steps to evidence synthesis rather than tool selection.

Despite this, the free-tool approach demonstrated one important strength: the perfect yes/no macro F1 achieved by Skill-Free-Run1 in Batch 3 shows that broad tool coverage can occasionally surface the right signal for binary classification questions. This suggests that the two strategies may be complementary rather than strictly ordered, and that a hybrid approach combining a curated core tool set with selective access to a broader free catalogue could be a promising direction for future work. Furthermore, the free-tool pipeline's multi-agent architecture may offer advantages in scalability and modularity. The current licensed tool set are limited to literature retrieval only, meanwhile with the free tool set, an

Table 5
Illustrative comparison of system outputs on three selected questions. ✓ = correct (LLM-based accuracy score = 1.0); × = incorrect (score = 0.0). Predicted answers are truncated for brevity.

Question	System	Answer	Used Tools
Which missense variant in SPDL1 is associated with IPF?	Ground truth	p.Arg20Gln (NM_017785.5:g.169588475 G>A)	
	Tool-GPT-4.1	✓ NM_017785.5:g.169588475 G>A, p.Arg20Gln (R20Q), identified through exome-wide association studies.	PubMed, NER
	Tool-Claude	✓ p.Arg20Gln (NM_017785.5:g.169588475 G>A), identified through exome-wide association study in 752 individuals, replicated in FinnGen.	PubMed, NER
	Skill-Free	× No relevant data found. Based on prior knowledge, the variant rs201371495 (p.Arg399Cys) has been reported — an incorrect variant.	None
How many primary genetic associations were identified through pQTL mapping within the Pharma Proteomics Project?	Ground truth	14,287 primary genetic associations from mapping of 2,923 proteins, of which 81% are previously undescribed.	
	Tool-GPT-4.1	✓ 14,287 primary genetic associations across 2,923 proteins; 81% previously undescribed.	PubMed, CCC
	Tool-Claude	✓ 14,287 primary genetic associations (Sun et al., *Nature*, 2023).	PubMed, CCC
	Skill-Free	× 54,306 primary genetic associations — an incorrect figure conflating a different statistic.	Literature Search, PubTator3, Web Search, Wikipedia
What are the lineage transcription factors of oligodendrocyte progenitor cells (OPCs)?	Ground truth	OLIG1/2, SOX10, NKX2.2, and ASCL1.	
	Tool-GPT-4.1	× Retrieved no results from PubMed or full-text search; returned an empty response.	PubMed, CCC, NER
	Tool-Claude	× Describes OLIG2, SOX10, and NKX2.2 in detail but omits ASCL1, leading to an incomplete answer.	PubMed, CCC
	Skill-Free	✓ OLIG2, SOX10, OLIG1, NKX2.2, NKX6.2, SOX17, ASCL1, and GSX2 — correctly includes ASCL1, benefiting from broader cross-database retrieval.	Literature Search, Wiki Pathway, Wikipedia, NCBI, Human Protein Atlas, GeoDataset, BioXiv, Web Search

agent can do other tasks such as data analysis and graph plotting, which can be useful for other tasks beyond biomedical QA.

5.1. Qualitative Analysis

Table 5 presents three illustrative cases that partially demonstrate the failure modes discussed above. Cases 1 and 2 show where the licensed-tool configuration succeeded and the free-tool agent failed due to snippet-level evidence limitations. Case 3 shows the reverse: the free-tool agent retrieved the correct answer by drawing on a broader range of sources, while both licensed-tool agents missed a required factor.

6. Conclusion

We presented a comparative study of three QA systems on BioASQ shared tasks: a prompt-based baseline, a licensed-tool ReAct agent with deep content retrieval, and a free-tool multi-agent system with broad but shallow coverage. Our internal experiments on ideal answers indicated that the licensed-tool configurations consistently outperformed the free-tool systems in terms of answer quality, retrieval quality and latency. Meanwhile, the official results of our submissions showed mixed trends between the two approaches: sometimes the licensed-tool configurations achieved the best performance, while in other cases the free-tool systems were competitive or even superior on certain question types. These findings suggest that the two approaches are not mutually exclusive: a hybrid design combining a curated core tool set with selective access to a broader catalogue is a promising direction for future work. More broadly, our results highlight the importance of iterative prompt and retrieval refinement within an agentic framework, and of deliberate tool curation over unrestricted tool proliferation when building retrieval-augmented systems for specialised domains.

Acknowledgments

We thank the anonymous reviewers for their fruitful comments, which help improve the paper.

Declaration on Generative AI

During the preparation of this work, the author(s) used Microsoft Co-pilot in order to: modify writing and grammar and spelling check. Further, the author(s) used Claude code to vibe code parts of the free-tool pipeline. After using these tool(s)/service(s), the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

References

[1] A. J. Thirunavukarasu, D. S. J. Ting, K. Elangovan, L. Gutierrez, T. F. Tan, D. S. W. Ting, Large language models in medicine., Nature Medicine 29 (2023). doi:10.1038/s41591-023-02448-8.

[2] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, S. Riedel, D. Kiela, Retrieval-augmented generation for knowledge-intensive nlp tasks, NIPS '20, Curran Associates Inc., Red Hook, NY, USA, 2020.

[3] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, Y. Cao, ReAct: Synergizing Reasoning and Acting in Language Models, in: International Conference on Learning Representations (ICLR), 2023.

[4] S. Ateia, U. Kruschwitz, BioRAGent: A retrieval-augmented generation system for showcasing generative query expansion and domain-specific search for scientific Q&A, in: Proceedings of ECIR 2025 Demo Track, 2024. arXiv:2412.12358.

[5] N. T. Nguyen, D. S. Lituiev, Z. Liu, A. Kashyap, G. Jenkinson, K. Kuhl, C. Corrado, N. M. Patel, K. Snigdha, S. Saeedi, et al., Med. AI ASK: an agentic system for biomedical question answering, Journal of the American Medical Informatics Association (2026) ocag038.

[6] D. Bu, J. Sun, et al., Empowering AI data scientists using a multi-agent LLM framework with self-evolving capabilities for autonomous, tool-aware biomedical data analyses, Nature Biomedical Engineering (2026). doi:10.1038/s41551-026-01634-6.

[7] H. Cho, R. Kang, Y. Kim, SkillRet: A large-scale benchmark for skill retrieval in LLM agents, arXiv preprint arXiv:2605.05726 (2026).

[8] Z. Li, L. Li, S. Lin, Y. Zhang, Know the ropes: A heuristic strategy for LLM-based multi-agent system design, arXiv preprint arXiv:2505.16979 (2025).

[9] 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. Gi-

annakoulas, 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.

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

[11] W. Yoon, R. Jackson, E. Ford, A. Grover, P. Gromkowski, T. Gamble, M. Jeong, J. Kang, Biomedical NER for the enterprise with distillated BERN2 and the Kazu framework, in: Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: Industry Track, Association for Computational Linguistics, Abu Dhabi, UAE, 2022, pp. 258–265. URL: https://aclanthology.org. doi:10.18653/v1/2022.emnlp-industry.26.

[12] E. W. Sayers, J. Beck, E. E. Bolton, J. R. Brister, J. Chan, R. Connor, M. Feldgarden, A. M. Fine, K. Funk, J. Hoffman, S. Kannan, C. Kelly, W. Klimke, S. Kim, S. Lathrop, A. Marchler-Bauer, T. D. Murphy, C. O'Sullivan, R. Schmieder, Y. Skripchenko, A. Stine, F. Thibaud-Nissen, J. Wang, J. Ye, E. Zellers, V. Schneider, K. D. Pruitt, Database resources of the national center for biotechnology information in 2025, Nucleic Acids Research 53 (2025) D20–D29. URL: https://doi.org/10.1093/nar/gkae979. doi:10.1093/nar/gkae979.

[13] P. J. A. Cock, T. Antao, J. T. Chang, B. A. Chapman, C. J. Cox, A. Dalke, I. Friedberg, T. Hamelryck, F. Kauff, B. Wilczynski, M. J. L. de Hoon, Biopython: freely available python tools for computational molecular biology and bioinformatics, Bioinformatics 25 (2009) 1422–1423. URL: https://doi.org/10.1093/bioinformatics/btp163. doi:10.1093/bioinformatics/btp163.

[14] J. Priem, H. Piwowar, R. Orr, OpenAlex: A fully-open index of scholarly works, authors, venues, institutions, and concepts, in: Proceedings of the 26th International Conference on Science, Technology and Innovation Indicators (STI 2022), Springer Nature, 2022.

[15] S. Es, J. James, L. Espinosa Anke, S. Schockaert, RAGAs: Automated evaluation of retrieval augmented generation, in: N. Aletras, O. De Clercq (Eds.), Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations, Association for Computational Linguistics, St. Julians, Malta, 2024, pp. 150–158. URL: https://aclanthology.org/2024.eacl-demo.16/. doi:10.18653/v1/2024.eacl-demo.16.

[16] A. Nentidis, G. Katsimpras, A. Krithara, G. Paliouras, Overview of BioASQ tasks 13b and Synergy13 in CLEF2025, in: Working Notes of CLEF 2025—Conference and Labs of the Evaluation Forum, CEUR-WS.org, 2025. URL: https://ceur-ws.org/Vol-4038/paper_1.pdf.

A. Prompt for the ReAct agent in the licensed-tool pipeline

```
You are a biomedical question-answering agent with access to specialized tools for
    querying authoritative scientific literature and biological databases.

Your objective: plan and execute tool calls, critically evaluate the returned evidence,
    and synthesize a grounded, well-cited answer.

================================================
CORE PRINCIPLES
================================================

1. Tool-Grounded Evidence Only:
   - All factual claims must be supported by tool outputs. Do not introduce external
       knowledge or unsupported inferences.
   - Treat tool outputs as the primary evidence layer.
```

```
2. Precision Over Exhaustiveness:
   - Prioritize high-relevance, high-confidence tool calls over broad but shallow
     exploration.
   - Every tool call should have a clear intent tied to answering the question.

================================================
TOOL USAGE STRATEGY
================================================

Planning:
- Decompose the question into discrete biomedical concepts (e.g., genes, proteins,
  diseases, pathways, drugs, clinical endpoints).
- Map each concept to the most appropriate tool(s).
- When multiple concepts can be explored independently, issue tool calls in parallel
  where possible.

Iterative Discovery:
- If a tool result reveals a new entity or relationship (e.g., a gene interaction, an
  alternative protein name, a related pathway), evaluate whether exploring it adds
  meaningful evidence to the answer.
- Cross-validate findings across different tools when the question demands high
  confidence (e.g., confirming a gene-disease association found in literature with a
  database annotation).

Evidence Quality:
- When tool results conflict, prefer:
  (a) Primary database annotations over free-text literature extractions
  (b) More recent publications over older ones
  (c) Results with explicit experimental evidence over computational predictions
- Note any conflicts or uncertainty in your final answer.

Stopping Criteria:
- Stop calling tools when:
  - The core question components have been addressed with converging evidence
  - Marginal tool calls are unlikely to change the conclusion
  - Tool results begin repeating previously obtained information

================================================
ANSWER CONSTRUCTION
================================================

Structure:
- Open with a direct answer to the question.
- Follow with supporting evidence organized by theme or concept.
- Close with any caveats, limitations, or areas of uncertainty.

Citations:
- Preserve all reference formats exactly as returned by the tools (e.g., "Source 1 -
  Paper Name - Section Name").
- Embed references as inline citations at the point of use.
- Do NOT add, modify, or fabricate any references beyond what the tools provide.

Handling Incomplete Evidence:
- If the evidence is insufficient for a definitive answer:
  - State which tools were called and what each returned
  - Identify specific gaps in the evidence
  - Suggest concrete next steps the user could pursue
```

```
  - Do NOT speculate beyond what the evidence supports

================================================
CONSTRAINTS
================================================

- Do not fabricate data, citations, or causal claims.
- Do not expose internal reasoning steps or tool selection logic.
- Do not alter tool-provided reference formats.
- Do not add external links or references not returned by the tools.
```

B. Prompts used to filter and rerank documents

B.1. Filter Prompt

```
You are a helpful assistant to biomedical scientists. You have been
given a list of documents and a user query.
Your job is to judge how relevant each document is to the user query
and return a structured output indicating the relevance of each
document along with your reasoning and the id of the document.
Return all documents and their relevance.

When assessing relevance, consider:
- Direct relevance: documents that explicitly discuss the topic in
  the query
- Mechanistic relevance: documents about underlying causes,
  mechanisms, or contributing factors that could explain the
  phenomenon described in the query
- Methodological relevance: documents about techniques, materials,
  or equipment mentioned or implied by the query

A document about a root cause of a described problem is highly
relevant even if it doesn't use the same terminology as the query.

Query:
{query}

Documents:
{document_context}
```

B.2. Reranker Prompt

```
You are a helpful assistant to biomedical scientists.
You have been given a list of documents and a query.
Your job is to rank the documents based on their relevance to the
query. Return only the top {limit} most relevant documents with their
rank and your reasoning for giving them that rank.
Think very carefully about these rankings.

Query:
{query}

Documents:
{document_context}
```

C. Prompts used in the three-leg retrieval step

C.1. Keyword Extraction Prompt

```
You are helping a scientist search a biomedical literature database.
Extract the most important search keywords and phrases from their
question.

Focus on: specific gene/protein names, drug names, disease names,
organism names, technique names, and other domain-specific terms that
would appear in paper titles and abstracts.

Return ONLY a comma-separated list of keywords and short phrases.
No explanation, no numbering. Keep it under 30 words.

Input: Question: {query}
```

C.2. Query Expansion Prompt

```
You are helping a scientist search PubMed. Rewrite their question as
a brief narrative paragraph that a search engine can match against
paper abstracts.

Write as if you are describing the problem to a colleague -- natural
language, not keywords. Include the specific type of experiment, what
might be going wrong (think broadly about all possible contributing
factors, mechanisms, and interactions), and what kind of papers would
be relevant.

Keep it under 60 words. Return ONLY the paragraph.

Input: Query: {query}
```

D. Prompt used in the free-tool pipeline

D.1. Agent orchestration prompt

```
You are a research strategist. Your task is to organize a research goal
into focused sub-agents, each handling a specific aspect of the investigation.

## Research Goal

{input_query}

## Available Tools

{tool_catalog}

## Instructions

Divide this research goal into 1-4 focused sub-agents. Each sub-agent should:
- Have a clear, specific research goal (a sub-task of the main goal)
- Be assigned 2-8 tools from the available set
- Include skill keywords for matching relevant domain knowledge
- Be self-contained for its research task
```

```
**Constraints:**
- Each tool may only appear in ONE sub-agent
- Every assigned tool must come from the Available Tools list above
- Create fewer sub-agents for narrow goals, more for broad multi-domain goals
- Keep goal descriptions to ONE sentence each
- Use empty params `{{}}` unless a specific query is needed
- Limit to 2-5 tools per sub-agent
- **Important:** Domain skills are automatically provided to each sub-agent based on
  your `skill_keywords`. Focus skill_keywords on domain topics (e.g., "rdkit",
  "molecular docking", "CRISPR").

Respond with ONLY a JSON object (no markdown, no code fences, no extra text):
{{
  "sub_agents": [
    {{
      "name": "short-kebab-case-name",
      "goal": "One sentence research goal",
      "tools": [{{"tool": "tool_name", "params": {{}}}}],
      "skill_keywords": ["keyword1", "keyword2"]
    }}
  ]
}}
```

D.2. Sub-agent Prompt

```
You are a specialized research sub-agent: {sa_name}.

## Input query

{query}

## Your Focus

{document_context}
## Instructions

1. Execute the most relevant tools to gather data for your focus area.
2. If domain skills are available, read their SKILL.md for methodology guidance.
3. Use skill scripts via `execute` when computational analysis is needed.
4. After each tool result, decide if you need more data or have enough.
5. Use tool results to inform subsequent tool calls.
6. When you have sufficient data, stop calling tools and provide a brief summary.
{skill_info}
```

E. Prompt used in the LLM-based accuracy metrics

```
You are provided an Answer to a Question as well as an array of
Reference Answers. Your job is to score the Answer by comparing it
against the Reference Answers using the given Grading Criteria.
Think about how to grade the Answer, reason it according to the given
Grading Criteria and finally give it a score between 0 or 1.
Respond with the reasoning_steps and the score.

---------
```

```
Question:
{question}

--
Answer:
{predicted_answer}

--
Reference Answers:
{reference_answers}

--
Grading Criteria:
    "yesno":   "The answer should be yes or no, the answer should
                match the References",
    "list":    "The answer should contain all items from the
                References list and should not contain any extra items",
    "factoid": "The fact conveyed by the answer should be the same
                as that of the References",
    "summary": "The answer should accurately summarize any of the
                References",
    "not_specified": """The Answer should convey the exact same
        information as the Reference Answers.
        If the Answer is a list, it must contain all the items in the
        Reference Answers list. Deduct 0.2 points for each missing
        answer and deduct 0.3 points for each answer that was not in
        the reference answer.
        If the Answer is a boolean answer, it must be the same as
        the Reference.""",
---------
```