MeSH for BioASQ Task 14B : A Lightweight Knowledge-Enhanced RAG Pipeline

Notebook for the BioASQ Lab at CLEF 2026

Liuya Sun[1], Hua Yang[1,*], Jie Tang[1,2], Paulo Quaresma[3], Yifan Jia[4] and Zhuowen Li[1]

[1]School of Artificial Intelligence, Zhongyuan University of Technology, Zhengzhou, 450007, Henan, China
[2]School of Computer Science, Zhongyuan University of Technology, Zhengzhou, 450007, Henan, China
[3]Department of Computer Science, University of Évora, 7000-671 Évora, Portugal
[4]Information Management School, NOVA University of Lisbon, 1070-312 Lisbon, Portugal

Abstract

This paper describes our Retrieval-Augmented Generation (RAG) system developed for the BioASQ biomedical question answering challenge. The system integrates MeSH-based knowledge expansion, multi-stage PubMed retrieval and reranking, sentence-level evidence snippet extraction, and a DSPy-based answer generation framework to generate both exact answers and ideal answers. For document retrieval, we first employ SciSpacy to extract biomedical entities and noun phrases from the input question. The extracted terms are then expanded using MeSH Descriptors and Supplementary Concept records to incorporate synonymous biomedical terminology. The expanded structured queries are submitted to PubMed through the NCBI E-utilities API, followed by a two-stage reranking process combining BM25 scores with heuristic retrieval features. For evidence extraction, retrieved abstracts are segmented into sentences, and highly relevant snippets are selected based on lexical overlap and question-type-specific cue phrases. For answer generation, we adopt DSPy, a framework for organizing and optimizing LLM pipelines, as the orchestration layer of our system. The framework supports multiple LLM backends, including GPT-4o-mini, DeepSeek-chat, and a LoRA fine-tuned BioMistral model. Experimental results reveal three overall trends. First, MeSH-based synonym expansion effectively alleviates the retrieval degradation caused by biomedical terminology variation and abbreviation ambiguity. Second, sentence-level evidence snippets generally provide more effective contextual grounding for answer generation than full abstracts. Third, combining lexical retrieval, knowledge expansion, and heuristic reranking yields more stable retrieval performance than relying solely on surface-level text matching. Overall, the results demonstrate that lightweight knowledge-enhanced retrieval can effectively improve biomedical literature retrieval and answer generation quality in the BioASQ setting.

Keywords

Biomedical Question Answering, Retrieval-Augmented Generation, MeSH, Snippet Retrieval, Large Language Models

1. Introduction

The widespread adoption of Large Language Models (LLMs) is transforming how biomedical knowledge is accessed. Compared with traditional literature retrieval and manual reading workflows, directly querying models such as ChatGPT enables users to obtain more immediate natural-language answers. However, even state-of-the-art large-scale LLMs may still generate factually incorrect or unverifiable content in biomedical question answering scenarios [1]. Retrieval-Augmented Generation (RAG), which incorporates external biomedical literature as contextual evidence for generation, has therefore become an important approach for mitigating hallucination and improving answer faithfulness and traceability [2].

BioASQ provides a standardized benchmark based on PubMed literature, including document retrieval, snippet extraction, and answer generation [3, 4], enabling systematic evaluation of RAG systems in biomedical question answering. BioASQ Task 14B consists of three evaluation phases: Phase A focuses

CLEF 2026 Working Notes, 21 – 24 September 2026, Jena, Germany
*Corresponding author.
2025120018@zut.edu.cn (L. Sun); huayang@zut.edu.cn (H. Yang); 2024007005@zut.edu.cn (J. Tang); pq@uevora.pt (P. Quaresma); 20251278@novaims.unl.pt (Y. Jia); 202308064723@zut.edu.cn (Z. Li)


on document and snippet retrieval, Phase A+ evaluates answer generation using official retrieval outputs, and Phase B performs end-to-end biomedical question answering including both retrieval and generation. Questions are categorized into yes/no, factoid, list, and summary types. Depending on the question category, systems are required to generate exact answers and/or ideal answers across multiple test batches. In recent years, participating systems [5] have generally adopted modular RAG pipelines: relevant abstracts are first retrieved from PubMed, followed by neural reranking to identify highly relevant documents, and finally used as contextual evidence for generating exact answers and ideal answers. Typical approaches combine multi-stage retrieval and reranking methods based on BM25 [6], MiniLM [7], and MPNet [8], while employing LLMs such as GPT-4 and Mixtral for answer generation. Some studies further replace full abstracts with sentence-level snippets to reduce contextual noise [9].

Despite these advances, existing RAG systems still rely heavily on lexical matching and neural semantic reranking during retrieval, with limited consideration of biomedical terminology variation, abbreviation ambiguity, and synonym expansion [10]. The same disease, gene, or drug may appear in the literature under multiple alternative expressions, making surface-level text matching insufficient for comprehensive document retrieval. Knowledge Graphs (KGs) provide a promising solution by leveraging entity relations and standardized concepts to expand the semantic space of retrieval queries, thereby improving literature recall. Recently, methods such as GraphRAG [11] have combined KGs with RAG frameworks through entity linking, subgraph expansion, and graph-structured contextual organization to enhance reasoning over complex questions [12, 13]. In the biomedical domain, MeSH (Medical Subject Headings, MeSH)[1] provides a rich hierarchy of standardized concepts and synonym mappings and has been widely adopted for PubMed retrieval and query expansion.

Motivated by these studies [9, 13, 12] , we propose a MeSH-enhanced biomedical RAG question answering system. Specifically, we employ SciSpacy [14] to extract biomedical entities and key phrases from user questions and perform controlled synonym expansion based on MeSH concepts. The expanded queries are then combined with multi-stage PubMed retrieval, BM25 scoring, and heuristic reranking to obtain candidate documents. Finally, sentence-level evidence snippets together with a DSPy-based framework, a declarative framework for composing and optimizing LLM pipelines [15], are used to generate both exact answers and ideal answers. Unlike GraphRAG approaches that emphasize complex graph reasoning, our work focuses on lightweight knowledge-enhanced retrieval, using structured query expansion to improve biomedical literature recall and downstream question answering performance.

2. Related Work

2.1. Biomedical Question Answering

BioASQ is one of the most representative biomedical QA benchmarks [16], which requiring systems to perform document retrieval, evidence snippet extraction, and answer generation based on PubMed literature [17]. Questions are categorized into four types: yes/no, factoid, list, and summary questions [16]. Systems are required to generate both exact answers and ideal answers under different evaluation settings, including accuracy, Mean Reciprocal Rank (MRR), F1-score, and ROUGE-based metrics [17, 16].

Recent advances in LLMs have significantly accelerated progress in biomedical question answering. Models such as BioGPT [18] and PubMedGPT [19] , which are pre-trained on large-scale biomedical corpora, have demonstrated strong performance on biomedical QA tasks. However, these approaches primarily rely on parametric knowledge and often lack verifiable supporting evidence, making them prone to hallucination. As a result, RAG has become a mainstream direction for biomedical question answering systems.

[1] MeSH (Medical Subject Headings) is the U.S. National Library of Medicine's controlled vocabulary thesaurus. It provides a hierarchical structure of standardized concepts and synonym mappings, and is widely used for indexing, cataloging, and searching biomedical literature in PubMed. For more information, see: https://www.nlm.nih.gov/mesh/introduction.html

2.2. RAG-based Biomedical QA

RAG has been widely adopted in BioASQ systems by combining external literature retrieval with LLM-based answer generation [20]. Existing systems commonly employ multi-stage retrieval and reranking pipelines [21]. In the first stage, BM25 [21] or PubMed API retrieval is used for high-recall document retrieval, while neural reranking models such as MiniLM [7] cross-encoders and MPNet [8] bi-encoders are applied in the second stage to improve document relevance.

Several studies further explore snippet-level evidence retrieval, where abstracts are segmented into sentences or sliding windows and only the most relevant snippets are selected as generation context [22]. This strategy reduces the influence of redundant information contained in complete abstracts and improves answer quality. A representative example is the MiBi system [9] proposed for BioASQ 2024, which adopts a modular RAG pipeline consisting of document retrieval, snippet extraction, and answer generation. The system combines BM25 with MiniLM and MPNet reranking models, explores both rule-based and LLM-based snippet extraction methods, and employs GPT-4, Mixtral, and the DSPy [15] framework for exact and ideal answer generation. MiBi further investigates three retrieval-generation paradigms, including Retrieval-Then-Generation (RtG) [23], Generation-Then-Retrieval (GtR) [23], and iterative RAG variants [24].

Despite these advances [9, 17], most existing RAG systems still rely heavily on lexical matching and neural semantic reranking, while limited attention has been paid to terminology variation, abbreviation ambiguity, and synonym expansion in biomedical retrieval.

2.3. Knowledge Graph Enhanced Retrieval

KG-enhanced retrieval has recently been introduced into RAG systems to improve entity disambiguation, semantic retrieval, and query expansion [11, 13, 25]. GraphRAG-based approaches integrate graph structures with document retrieval pipelines, where entity linking, candidate node retrieval, and subgraph expansion are used to organize contextual evidence for downstream reasoning [13, 25]. These methods demonstrate that structured entity relations can improve semantic retrieval and multi-hop reasoning capabilities.

In the biomedical domain, standardized medical knowledge resources such as MeSH and UMLS [26] have also been incorporated into PubMed retrieval and query expansion [1, 5]. Previous studies have shown that MeSH-based query expansion can alleviate terminology inconsistency and improve biomedical literature retrieval effectiveness [27]. However, existing GraphRAG approaches mainly focus on open-domain knowledge graphs and complex reasoning scenarios, while lightweight knowledge-enhanced retrieval for biomedical QA remains relatively underexplored.

Motivated by these studies, our work introduces MeSH-based knowledge enhancement into the BioASQ retrieval pipeline through biomedical entity extraction, synonym expansion, and structured query construction. Unlike GraphRAG approaches that emphasize complex graph reasoning, our method focuses on lightweight knowledge-enhanced retrieval for improving biomedical literature recall [11, 28].

3. Methodology

Our system follows the RAG paradigm and is designed as a modular pipeline for the BioASQ biomedical question answering task. The pipeline consists of four stages: (1) MeSH-based knowledge expansion, (2) document retrieval and two-stage reranking, (3) evidence snippet extraction, and (4) answer generation. Given an input question, the system first extracts biomedical entities and key phrases for MeSH-based synonym expansion. The expanded queries are then used in a multi-stage PubMed retrieval and reranking framework to obtain candidate documents, followed by sentence-level evidence extraction for snippet construction. Finally, a DSPy-based question-type-aware framework generates both exact answers and ideal answers. The overall architecture of the proposed system is illustrated in Figure 1.

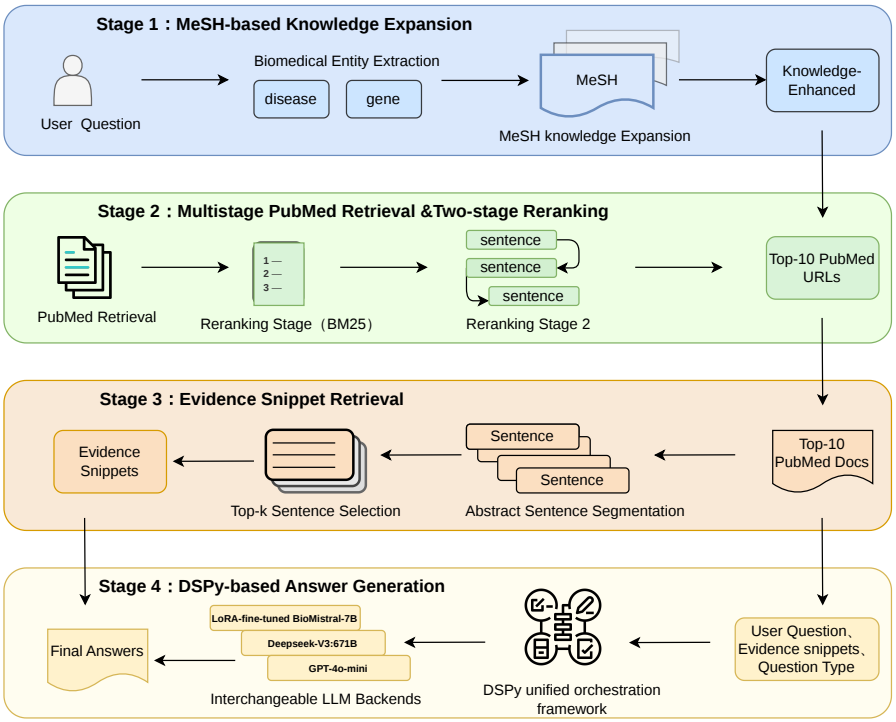

Figure 1: Overall framework of the MeSH-enhanced retrieval-augmented generation system.

3.1. MeSH-based Knowledge Expansion

The MeSH knowledge resource is constructed from the official Descriptor and Supplementary Concept files, which provide standardized biomedical concepts together with their synonymous expressions. Given an input question, the proposed module first identifies representative biomedical terms and then expands them using MeSH concepts to construct a structured Boolean query for PubMed retrieval. To improve retrieval recall while avoiding excessive query drift, the expansion process follows a controlled strategy that limits both the number of retained biomedical terms and the number of selected synonyms for each concept. Specifically, candidate biomedical terms are extracted from the input question, normalized, filtered, and matched against MeSH concepts. Exact concept matching is prioritized, while synonym matching is restricted to multi-word expressions. After MeSH matching, only up to two representative synonyms are retained for each concept, and terms containing overly generic expressions, such as "state" or "clinical trial", are discarded. Preliminary experiments on a development subset showed that larger expansion sizes introduced ambiguous concepts and reduced retrieval precision, motivating the use of controlled synonym expansion. Figure 2 illustrates the overall workflow of biomedical term extraction, MeSH synonym expansion, and structured Boolean query construction.

We employ the SciSpacy [14] biomedical English model `en_core_sci_sm` to process the input question and extract candidate biomedical terms. Candidate terms are obtained from two linguistic sources: named entity spans and noun phrase spans. Low-information task-description expressions, such as "describe", "role", and "clinical trials", are removed, followed by lowercase normalization and deduplication.

The expanded query is organized in a structured AND/OR form. Each core biomedical term is grouped together with its selected MeSH synonyms using OR operators, while these term-specific groups are combined using AND operators to enforce high-precision constraints while maintaining recall [13]. Retrieval is restricted to the PubMed Title/Abstract fields.

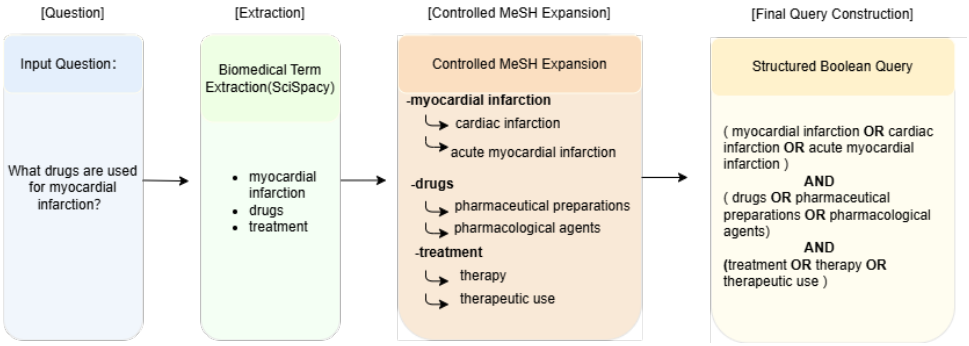

Figure 2: Figure 2: Workflow of biomedical term extraction, filtering and normalization, controlled MeSH synonym expansion, and structured Boolean query construction. The example shows term extraction, synonym expansion, and the final structured query using AND/OR operators.

$$Q_{exp} = \bigwedge_{t_i \in T} \left(t_i \ \vee \bigvee_{s \in Syn(t_i)} s \right)$$

where t_i ($i = 1, 2, \ldots$) are the core biomedical terms extracted from the input question, and $Syn(t_i)$ denotes the selected MeSH synonym set associated with term t_i. The operator $\wedge$ denotes logical AND and $\vee$ denotes logical OR. If no valid core terms are extracted, the original question text is directly used for retrieval. Algorithm 1 summarizes the overall procedure of biomedical term extraction, MeSH concept matching, synonym selection, and Boolean query construction. The retained terms are subsequently divided into anchor, focus, and constraint groups according to their extraction order, as described in Section 3.2.

3.2. Query Prioritization Strategy

Although multiple biomedical terms can be extracted from a question, their contribution to retrieval is not equally important. This motivates a prioritization strategy that organizes the extracted terms into different semantic roles before constructing the final retrieval query. Based on the extracted terms described in Section 3.1, only the first five retained terms are considered for downstream retrieval. This heuristic is based on preliminary observations on BioASQ development questions, where earlier extracted terms frequently corresponded to central biomedical concepts in the query. The retained terms are then organized into three functional groups according to their extraction order: the first two terms are treated as *anchor terms*, representing the primary semantic constraints; the following two terms are assigned as *focus terms*, capturing descriptive attributes or relational context; and any remaining term is considered a *constraint term*, providing additional filtering signals.

The prioritized term groups are subsequently used in the query construction process, where anchor terms define the core retrieval constraints, while focus and constraint terms modulate the specificity of the query. The controlled MeSH-based synonym expansion is applied as described in Section 3.1, and is not redefined here to avoid redundancy. The final PubMed query is constructed using the structured Boolean formulation introduced in Algorithm 1. In this formulation, anchor terms serve as primary constraints, while focus and constraint terms contribute to additional semantic coverage under the same Boolean framework. This prioritization scheme provides a lightweight mechanism to emphasize semantically central biomedical concepts without introducing additional model complexity.

3.3. Document Retrieval and Two-stage Reranking

The document retrieval stage aims to retrieve relevant PubMed abstracts for each biomedical question. Candidate documents are obtained through the NCBI E-utilities API, a web service that provides programmatic access to PubMed records, and ranking quality is further improved through a local two-stage

Algorithm 1 Biomedical term extraction with MeSH matching and synonym selection. T denotes the set of extracted terms, M_t the MeSH concept matched for term t, and $Syn(t)$ the selected synonyms for t.

Require: User question q, MeSH knowledge base $\mathcal{M}$, generic term list $\mathcal{G}$
Ensure: Expanded Boolean query Q_{exp}

$T \leftarrow$ SciSpacy.extract_entities(q)
$T \leftarrow T \cup$ SciSpacy.extract_noun_chunks(q)
$T \leftarrow$ lowercase_normalize(T)
$T \leftarrow T \setminus \mathcal{G}$ // remove generic biomedical or task-related terms
$T \leftarrow$ deduplicate(T)
retain up to 5 highest-salience terms in T according to their original extraction order from SciSpacy // Earlier extracted terms frequently corresponded to central biomedical entities in our development observations
for $t \in T$ **do**
  if t exactly matches a MeSH concept name **then**
    $M_t \leftarrow$ corresponding MeSH concept
  else if t appears in a MeSH concept synonym (entry term) list **then**
    $M_t \leftarrow$ corresponding MeSH concept
  else
    continue
  end if
  if M_t exists **then**
    $Syn(t) \leftarrow$ select up to 2 entry terms from the MeSH concept synonym list // synonyms are selected from MeSH entry terms (not ranked list), keeping up to two representative biomedical variants after filtering generic and ambiguous expressions
    query_clauses$[t] \leftarrow (t \vee \bigvee_{s \in Syn(t)} s)$
  end if
end for
$Q_{exp} \leftarrow \bigwedge_{t \in T} (t \vee \bigvee_{s \in Syn(t)} s)$
return Q_{exp}

reranking framework [9]. Before retrieval, abbreviation and alias normalization are applied to reduce terminology ambiguity. Ambiguous abbreviations are expanded according to question context (e.g., "CMS" to "consensus molecular subtype"), and word-boundary-based regular expression replacement is used to avoid incorrect substitutions.

Candidate retrieval follows a four-stage progressively relaxed strategy, as illustrated in Figure 3. Stage 1 employs the structured entity- and MeSH-based query introduced in Section 3.1. If the number of retrieved records is insufficient, query constraints are gradually relaxed in subsequent stages through simplified term combinations and progressively reduced filtering requirements. Results from all stages are merged and duplicate PubMed identifiers are removed while preserving their original retrieval order. To control computational cost while maintaining candidate coverage, the merged document set is truncated to a maximum of 300 PubMed records before abstract retrieval. This upper bound was selected empirically based on development experiments, where larger candidate pools increased computational overhead while providing limited improvements in downstream retrieval quality. Titles and abstracts of the retained documents are subsequently retrieved through the PubMed `efetch` API.

The first-stage reranking module is based on BM25Okapi. Instead of directly using the raw question text, a weighted query representation is constructed from the extracted anchor, focus, and constraint terms. Anchor terms are emphasized through repetition, focus terms receive moderate weighting, and constraint terms contribute lower weights. Original question tokens are additionally retained to preserve semantic completeness. Document titles are repeated three times before concatenation with abstracts in order to increase title influence during retrieval.

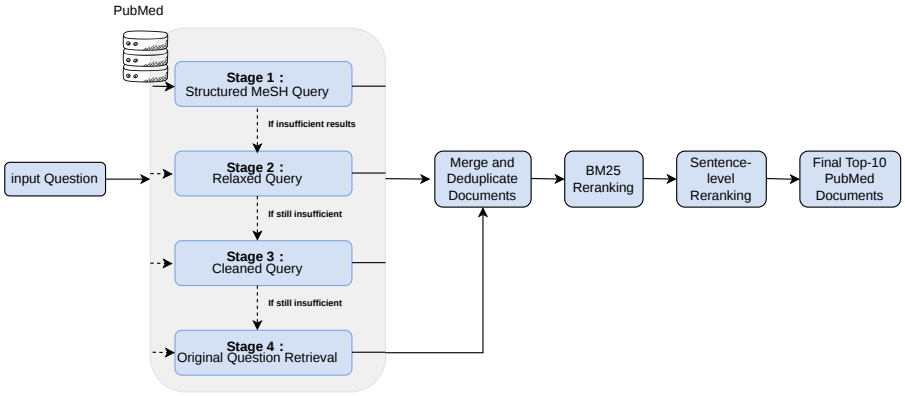

Figure 3: Four-stage progressively relaxed retrieval strategy and two-stage reranking pipeline.

In addition to BM25 scores, several heuristic retrieval features are incorporated, including anchor-term coverage, phrase matching, title matching, focus-term matching, constraint-term matching, negative-topic penalties, and question-type-specific answer cues. The document score is defined as:

$$S_d = S_{BM25} + S_{lex} - S_{neg} + S_{sent}$$

where S_{lex} represents lexical matching features and S_{sent} denotes the sentence-level evidence score.

For yes/no, factoid, and list questions, sentence-level reranking is additionally performed to emphasize local evidence relevance. Abstracts are first segmented into sentences using punctuation-based splitting, and each sentence is scored according to (1) anchor–focus term co-occurrence, (2) question-type-specific cue phrases, and (3) constraint-term matching.

Formally, for a sentence s:

$$S_s = \alpha S_{co}(s) + \beta S_{cue}(s) + \gamma S_{con}(s)$$

where S_{co} measures anchor–focus co-occurrence, S_{cue} captures question-type-specific cue phrases, and S_{con} measures constraint-term matching.

The sentence-level evidence score of a document is defined as:

$$S_{sent} = \max_{s \in \text{abstract}(d)} S_s$$

The final document ranking score combines BM25 retrieval and sentence-level evidence signals. The top 50 reranked documents are retained for evidence extraction, and the final top-10 PubMed URLs are returned as BioASQ document predictions. All weighting parameters are determined empirically on development data and are not optimized through model training.

3.4. Evidence Snippet Retrieval

The evidence snippet extraction subtask, corresponding to BioASQ Task A, aims to identify short text segments from retrieved PubMed abstracts that directly support answering the question. Rather than using structured snippets provided by PubMed, evidence selection is performed at the abstract level through sentence segmentation and relevance-based ranking. For each retrieved document, the title and abstract are obtained, and candidate sentences are generated using punctuation-based splitting at sentence boundaries. To capture sufficient contextual information, neighboring sentences are grouped into evidence units consisting of up to three consecutive sentences. Single sentences often omit important biomedical relations or supporting details, whereas larger text spans may introduce redundant information. Empirical observations on the development set indicated that a three-sentence window provided a reasonable trade-off between contextual completeness and noise reduction; shorter windows

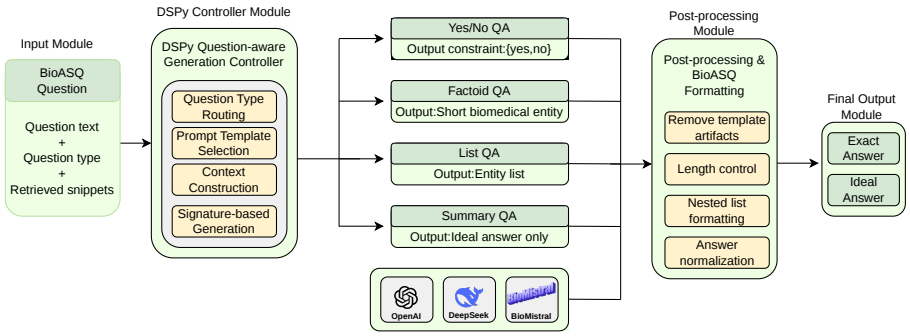

Figure 4: DSPy-based question-aware answer generation framework. Different BioASQ question types are routed to dedicated generation signatures and processed using interchangeable LLM backends, followed by unified post-processing and BioASQ-format normalization.

frequently omitted biomedical relations, whereas larger windows introduced redundant context. The resulting evidence units are ranked according to relevance scores. Relevance scoring combines three components: (1) lexical overlap between question terms and evidence terms after tokenization; (2) question-type-specific cue phrases; and (3) sentence-length preference. Very short tokens are removed only when they are unlikely to carry meaningful semantic information. Tokens recognized during abbreviation normalization or matched to biomedical concepts are retained, since many biomedical entities (e.g., gene symbols or disease abbreviations) are naturally represented by short expressions.

Question-type-specific cue phrases are manually curated from BioASQ development data and biomedical QA heuristics. Examples include enumeration indicators for list questions and relational expressions for yes/no questions. A mild sentence-length preference is also applied to favor sentences of moderate length, since very short sentences often lack sufficient information while excessively long sentences tend to introduce irrelevant content. The highest-ranked evidence units are retained as candidate snippets and concatenated in their original order to preserve readability and discourse coherence. During answer generation, only the top five retrieved snippets are used as contextual evidence to maintain stable input length across different LLM backends. Each snippet is truncated to at most 600 characters, and the total evidence context is limited to 3000 characters. These limits are introduced solely to maintain a stable input size during answer generation and do not affect snippet retrieval or ranking.

The resulting snippet is formatted according to BioASQ conventions, with both `beginSection` and `endSection` set to "abstract". Because each snippet may contain multiple sentences, character-level offsets are simplified: `beginOffset` is fixed at 0 and `endOffset` corresponds to the snippet length.

3.5. Answer Generation

The answer generation subtask (BioASQ Task B) requires producing both exact answers and ideal answers. Our answer generation module takes as input the question text, question type, and retrieved snippets, and employs a DSPy-based question-type-aware generation framework to produce exact and ideal answers. For exact answer generation, the top-ranked retrieved snippets (up to 5) are concatenated sequentially to serve as context. This threshold was selected empirically on development data to balance evidence completeness and input length across different LLM backends while maintaining a stable evidence context size. Optionally, for ideal answer generation, the previously generated exact answer can be provided as auxiliary input.

Dedicated DSPy [15] signatures are defined for different question types: yes/no questions constrain outputs to {"yes", "no"}; factoid questions output a short entity string; list questions output an entity list, formatted as a nested BioASQ list ([[item_1], [item_2], ...]); and summary questions assign the exact answer as "n/a". Answers are generated via `Predict(Signature)` and subjected to light post-processing to remove template remnants (e.g., "Answer:") and truncate overly long outputs. Figure 4 illustrates the overall DSPy-based question-aware answer generation framework, including question-

type routing, multi-backend LLM generation, and BioASQ-oriented post-processing.

Ideal answers consist of 1–3 evidence-based sentences generated using the same snippet context, optionally conditioned on the exact answer. For smaller or context-sensitive models, context budgets are enforced by limiting the number of entries and total character count. Generated results are cleaned to remove abnormal prefixes or repeated content. The framework provides a unified interface supporting multiple interchangeable LLM backends. In our experiments, three configurations were used: OpenAI GPT-4o-mini (via the OpenAI API, specified with `-llm` and `OPENAI_API_KEY`), DeepSeek-chat (via OpenAI-compatible endpoints), and locally deployed BioMistral-7B with LoRA adapters for supervised fine-tuning. All backends share identical DSPy signatures, prompt templates, and post-processing rules.

Model details. DeepSeek-V3 is a strong Mixture-Of-Experts (MoE) language model with 671B total parameters and 37B activated for each token. BioMistral-7B-DARE-distill is based on the BioMistral-7B-DARE model and is Supervised Fine-Tuned (SFT) using the official BioASQ Training 14b dataset to enhance its domain knowledge in biomedical question answering. The fine-tuning is implemented with the LLaMA-Factory framework using Low-Rank Adaptation(LORA) for parameter-efficient adaptation, updating only a small subset of attention module parameters. The training employs bf16 mixed precision, the AdamW optimizer, and a cosine learning rate schedule, with the input length extended to 3072 tokens to better model long biomedical contexts. The base model parameters remain frozen, reducing GPU memory overhead and training cost. The resulting model is used for all BioMistral-based runs in our experiments.

Finally, outputs are normalized to BioASQ format: yes/no answers are mapped to "yes"/"no"; list answers are converted to nested lists; the number of documents and snippets is truncated to official limits (10 each). This procedure reduces formatting errors and ensures reproducibility across different LLM backends.

3.6. Prompt Design and DSPy Signatures

The answer generation module is implemented using DSPy [15], a framework for declarative prompt-based LLM programming. Different BioASQ question types are associated with dedicated DSPy signatures, each defining a fixed prompt template and corresponding output constraints. These signatures remain unchanged during inference and are shared across all LLM backends.

Input Representation. Given a question q, the model input consists of the question body q_{body} and a context set $\mathcal{C} = \{c_1, c_2, \ldots, c_n\}$ derived from retrieved evidence. For exact answer generation, the context contains the top-ranked retrieved snippets. For ideal answer generation, the context may additionally include the generated exact answer as auxiliary information together with selected evidence snippets.

DSPy Prompt Templates. Each question type uses a dedicated prompt template embedded in the DSPy signature definition.

For yes/no questions, the model is instructed to determine whether the evidence supports or contradicts the statement and return only "yes" or "no".

For factoid questions, the model is instructed to return only a short biomedical entity without explanations or complete sentences.

For list questions, the model is instructed to output multiple valid entities, with each entry corresponding to a candidate answer.

For ideal answer generation, the model is instructed to generate a concise evidence-grounded response consisting of 1–3 sentences.

Output Constraints. The output format is constrained according to the corresponding question type. Yes/no questions are restricted to binary outputs $\{\text{yes}, \text{no}\}$, factoid questions require a single entity

Table 1
Implemented Runs. ✓ indicates that the run was submitted for the corresponding phase, while ○ indicates that the run was not submitted.

Submit Name	Run	Model and Method	Phase A	Phase A+	Phase B
IR_Y_1	first run	DeepSeek-V3:671B	✓	○	○
IR_Y_2	second run	GPT-4o-mini	✓	○	○
IR_Y_1	first run	BioMistral-7B-DARE -distill	○	✓	○
IR_Y_2	second run	BioMistral-7B-DARE -distill (random snippet)	○	✓	○
IR_Y_3	third run	DeepSeek-V3:671B	○	✓	○
IR_Y_4	fourth run	DeepSeek-V3:671B (random snippet)	○	✓	○
IR_Y_5	fifth run	GPT-4o-mini	○	✓	○
IR_Y_1	first run	BioMistral-7B-DARE -distill	○	○	✓
IR_Y_2	second run	BioMistral-7B-DARE -distill (random snippet)	○	○	✓
IR_Y_3	third run	DeepSeek-V3:671B	○	○	✓
IR_Y_4	fourth run	DeepSeek-V3:671B (random snippet)	○	○	✓
IR_Y_5	fifth run	GPT-4o-mini	○	○	✓

string, list questions require a sequence of entity outputs, and ideal answers are generated as short free-form responses. These constraints are enforced through DSPy signatures to ensure consistent output formatting across different LLM backends.

4. Results and Analysis

Table 1 summarizes all runs we submitted to the BioASQ 14 B task, covering Phase A, Phase A+, and Phase B. Each row corresponds to a specific run configuration. The "System Name" column indicates the run group, e.g., `IR_Y_1` to `IR_Y_5`. The "Submission Name" and "Run" columns represent the submission identifier and the order of runs within the system. The "Model and Method" column specifies the large model used and the fine-tuning or data manipulation strategy applied. The last three columns indicate whether the run was submitted to the corresponding phase (Phase A, Phase A+, or Phase B). For instance, `IR_Y_2` represents the second run in Phase B, which uses the BioMistral-7B-DARE -distill model and adopts the method of shuffling the order of official Phase B snippets. This run was submitted to Phase B.

Table 2 and Table 3 report the preliminary leaderboard results of our submitted systems on the BioASQ benchmark. These results correspond to the online evaluation available during the shared task and do not represent the final official evaluation after the competition. We adopt a subset of the official BioASQ evaluation metrics, including Recall and MAP for document retrieval, and Macro F1, MRR, F-measure, and ROUGE-SU4 F1 for answer generation. For reference, the tables also include the highest and lowest scores reported on the preliminary online leaderboard for each evaluation setting. Overall, our systems achieve competitive performance across multiple BioASQ phases and batches. In particular, the proposed system obtains the highest Macro F1 score for yes/no exact answers in Phase B of batches 3 and 4, as well as in Phase A+ of batch 4. In addition, our ideal answer generation system ranks second in Phase A+ of batch 1. However, the performance on factoid exact answers remains relatively limited, especially in Phase B settings. In the following, we highlight some interesting differences between our system variants with respect to the evaluation metrics. All comparisons with the best, median, and worst systems are based on the preliminary online leaderboard released during the BioASQ shared task and should not be interpreted as the final official evaluation results.

4.1. Task 14 B Phase A

Effectiveness of our retrieval and answer generation approaches according to the preliminary online leaderboard of the shared task. We report Recall and Mean Average Precision for the Phase A retrieval task. The highest, lowest, and median scores among all submissions are highlighted in gray, while above-median results are shown in bold. See Table 2 for details.

Table 2
Task 14B Phase A, Document Retrieval.

Batch	System	Abstracts		Snippets	
		Rec.	MAP	Rec.	MAP
1	best	0.4090	0.2835	0.2213	0.1646
	IR_Y_1	0.1685	0.1077	**0.1364**	0.0255
	IR_Y_2	0.1685	0.1077	**0.1364**	0.0255
	worst	0.0047	0.0228	0.0056	0.0048
	median	0.2703	0.1887	0.0858	0.0542
2	best	0.3357	0.2370	0.2082	0.1824
	IR_Y_1	0.1850	0.1177	**0.1114**	0.0318
	IR_Y_2	0.1768	0.1099	**0.1096**	0.0316
	worst	0.0323	0.0109	0.0102	0.0033
	median	0.2580	0.1399	0.0997	0.0539
3	best	0.3206	0.2114	0.1544	0.1226
	IR_Y_1	0.1385	0.1100	0.0745	0.0315
	IR_Y_2	0.1385	0.0946	0.0559	0.0203
	worst	0.0764	0.0416	0.0163	0.0000
	median	0.2327	0.1411	0.7450	0.0315
4	best	0.3698	0.2310	0.2055	0.1750
	IR_Y_1	0.1341	0.0710	0.0603	0.0146
	IR_Y_2	0.1327	0.0718	0.0670	0.0119
	worst	0.0374	0.0000	0.0015	0.0004
	median	0.2544	0.1503	0.0978	0.0296

We further compared the performance of DeepSeek-V3:671B and GPT-4o-mini on BioASQ Task 14 B Phase A. In Batch 1, the systems based on DeepSeek-V3:671B and GPT-4o-mini achieved comparable Recall and MAP scores for both document retrieval and snippet extraction. However, from Batches 2 to 4, the systems using DeepSeek-V3:671B consistently outperformed those based on GPT-4o-mini in terms of Recall and MAP for both document retrieval and snippet extraction.

For snippet extraction, the DeepSeek-V3:671B-based system achieved a Recall score of 0.0603, while GPT-4o-mini obtained 0.067, exceeding DeepSeek-V3:671B by only 0.0067. These results indicate that different large language models exhibit noticeable performance differences across retrieval and extraction tasks, which may be attributed to differences in task characteristics and model capabilities [29].

4.2. Task 14 B Phase A+

For Phase A+, we submitted five runs for each batch, with detailed system configurations summarized in Table 3. To assess the models' dependence on snippet order, we designed a *Random Snippet Shuffling* method: for each question, all snippets retrieved in Phase A are randomly permuted to form a new input context before being fed into the model. Comparing answer quality between original and shuffled snippet orders allows us to determine whether a model performs reasoning based on snippet content or relies excessively on local coherence or positional bias [9].

We evaluated three models: BioMistral-7B-DARE-distill ("BioMistral"), DeepSeek-V3:671B ("DeepSeek"), and GPT-4o-mini. The standard and randomly shuffled versions of BioMistral are denoted as IR_Y_1 and IR_Y_2, respectively; those of DeepSeek are IR_Y_3 and IR_Y_4; GPT-4o-mini is used as a reference system IR_Y_5. Additionally, we report the best, median, and worst scores from the preliminary leaderboard as performance baselines. In Batch 1, for exact-answer yes/no questions, all submitted systems scored below the median system. The best-performing system was GPT-4o-mini, approaching the median level. For ideal answers, both BioMistral variants and GPT-4o-mini exceeded the median, with IR_Y_2 (randomly shuffled BioMistral) ranking second among all submissions. This

Table 3
Task 14B Phase A+ and Phase B.

System	Phase A+				Phase B			
	Exact Answer			Ideal(R-SU4F1)	Exact Answer			Ideal(R-SU4F1)
	Yes-no (Ma.F1)	Fact (MRR)	List (F.M.)		Yes-no (Ma.F1)	Fact (MRR)	List (F.M.)	
Test batch 1								
best	1.0000	0.4783	0.2345	0.1821	1.0000	0.5217	0.3613	0.2381
IR_Y_1	0.3704	0.0109	0.0478	**0.1340**	0.5096	0.0522	0.1855	0.0739
IR_Y_2	0.3704	0.0217	0.0648	**0.1671**	0.5096	0.0304	0.1841	0.0502
IR_Y_3	0.8211	0.0870	0.0329	0.0050	0.9377	0.2391	0.1769	**0.1864**
IR_Y_4	0.7018	0.0435	0.0372	0.0132	0.9377	0.2935	0.1769	**0.1915**
IR_Y_5	0.8786	0.1087	0.0388	**0.1294**	0.9377	**0.3913**	0.1791	**0.2130**
worst	0.3462	0.0109	0.0063	0.0050	0.5096	0.0304	0.0473	0.0177
median	0.8786	0.2971	0.1500	0.1273	0.9377	0.3696	0.2891	0.1725
Test batch 2								
best	0.9481	0.4000	0.3909	0.1851	1.0000	0.4000	0.4720	0.2310
IR_Y_1	0.7667	0.0975	0.0507	0.0739	0.8929	0.1600	0.1919	0.0975
IR_Y_2	0.8329	0.0775	0.0277	0.0780	0.8929	**0.2200**	0.1986	0.1194
IR_Y_3	0.8329	**0.3000**	0.0543	0.1272	0.8929	**0.2750**	0.1634	0.1656
IR_Y_4	0.8329	**0.3250**	0.0543	0.1249	0.8929	**0.3000**	0.1567	0.1529
IR_Y_5	0.7857	0.1750	0.0627	0.1349	0.8929	**0.2417**	0.1773	0.1300
worst	0.4000	0.0775	0.0272	0.0739	0.4000	0.0500	0.0171	0.0257
median	0.8444	0.2000	0.1778	0.1349	0.8929	0.2000	0.3630	0.1816
Test batch 3								
best	1.0000	0.5882	0.3833	0.1537	1.0000	0.5882	0.4884	0.2547
IR_Y_1	0.6071	0.1294	**0.1370**	0.0480	**1.0000**	0.3431	**0.3895**	0.0970
IR_Y_2	0.6071	0.2941	**0.1128**	0.0671	0.6118	**0.4412**	0.3225	0.1063
IR_Y_3	0.6071	**0.3235**	0.1093	0.1022	0.8952	**0.4706**	0.3084	0.1600
IR_Y_4	0.6071	**0.3235**	0.1122	0.1064	0.8952	**0.4706**	**0.3846**	**0.1757**
IR_Y_5	0.6071	**0.3235**	0.1102	0.1043	0.8952	**0.4706**	**0.3559**	**0.2012**
worst	0.1538	0.0588	0.0588	0.0243	0.33889	0.0588	0.0860	0.0078
median	0.6071	0.3235	0.1122	0.1236	0.8952	0.4118	0.3244	0.1600
Test batch 4								
best	0.9352	0.4091	0.4840	0.1699	1.0000	0.5758	0.6543	0.2699
IR_Y_1	**0.9352**	0.0909	0.0448	0.0506	**1.0000**	0.1212	0.3661	0.0859
IR_Y_2	**0.9352**	0.0909	0.0254	0.0536	**1.0000**	0.1136	0.3412	0.0821
IR_Y_3	**0.9352**	0.1364	0.0275	**0.1406**	**0.9352**	**0.4091**	0.1606	**0.2002**
IR_Y_4	**0.9352**	**0.2727**	0.0172	0.1277	**0.9352**	0.3636	0.1757	**0.1947**
IR_Y_5	0.8057	0.0909	0.0363	**0.1379**	**0.9352**	**0.4545**	0.1690	**0.1890**
worst	0.6135	0.0909	0.0091	0.0048	0.5000	0.0909	0.1338	0.0126
median	0.8750	0.2273	0.2588	0.1319	0.8667	0.3864	0.3969	0.1525

Note: Ma.F1 = Macro F1, MRR = Mean Reciprocal Rank, F.M. = F-measure, R-SU4(F1) = ROUGE-SU4 F1.

suggests that snippet shuffling can occasionally improve performance on ideal-answer tasks, possibly by reducing model reliance on "shortcuts" present in the original snippet order. In Batch 2, for exact-answer yes/no questions, the DeepSeek systems outperformed BioMistral and GPT-4o-mini. Notably, IR_Y_4 (shuffled DeepSeek) achieved slightly higher MRR than IR_Y_3 (standard DeepSeek), indicating a minor positive effect of snippet shuffling for this model on this metric. Performance on list questions remained poor for all systems. For ideal answers, all systems except GPT-4o-mini scored below the median. In Batch 3, for exact-answer factoid questions, only the BioMistral systems exceeded the median; other metrics fluctuated near the median. The shuffled and standard DeepSeek runs showed comparable MRR for yes/no questions, indicating negligible impact of order perturbation in this batch. In Batch 4, for exact-answer yes/no questions, both BioMistral and DeepSeek systems achieve the best system baseline, demonstrating strong reasoning capability. However, for factoid and list questions, all our systems fell below the median. For ideal answers, DeepSeek systems and GPT-4o-mini outperformed the median.

Overall, across four batches, the effect of random snippet shuffling varied by model and task type.

Shuffling sometimes improved yes/no MRR for DeepSeek and ideal-answer performance for BioMistral, but had no positive effect on list questions. These results indicate that model dependence on snippet order is closely linked to question type and model architecture. Random permutation can mitigate harmful positional bias but may also disrupt necessary local coherence, suggesting its use should be context-specific.

4.3. Task 14 B Phase B

For Phase B, we employed the same model configurations and Random Snippet Shuffling method as in Phase A+, with system details summarized in Table 3. Compared with Phase A+, the overall performance distribution changed substantially, and in some batches our systems even exceeded the best score reported on the preliminary leaderboard, indicating that differences in retrieval context or question distribution between phases influenced model behavior. In Batch 1, for exact-answer yes/no questions, DeepSeek and GPT systems reached median-level performance, whereas BioMistral scored only 0.5096, substantially below median. For factoid questions, GPT-based systems slightly outperformed the median; DeepSeek's IR_Y_4 achieved MRR of 0.2935 versus IR_Y_3 at 0.2391, though both remained below the median. For list questions, all systems fell below the median. Regarding ideal answers, both DeepSeek systems exceeded the median, with IR_Y_4 slightly better. Overall, DeepSeek and GPT performed better than BioMistral, and random snippet shuffling had a mild positive effect on factoid and ideal-answer questions. In Batch 2, all systems scored similarly on exact-answer yes/no questions, approaching the best system baseline. BioMistral IR_Y_2 outperformed IR_Y_1. Ideal-answer performance was below median for all systems. In this batch, random shuffling improved factoid question ranking but slightly reduced ideal-answer performance. In Batch 3, exact-answer yes/no performance diverged: BioMistral IR_Y_1 achieved perfect score (1.0), while IR_Y_2 only scored 0.6118; DeepSeek and GPT systems were both 0.8952. For factoid questions, BioMistral IR_Y_2 substantially outperformed IR_Y_1, with DeepSeek and GPT slightly above median. For list questions, BioMistral IR_Y_2, DeepSeek IR_Y_4, and GPT IR_Y_5 exceeded median. Ideal-answer performance was above median for DeepSeek and GPT. In this batch, random snippet shuffling noticeably improved BioMistral factoid performance but substantially reduced yes/no performance, indicating that the effect of snippet permutation depends strongly on the interaction between model and question type. In Batch 4, exact-answer yes/no scores reached new highs: both BioMistral runs achieved perfect scores, and DeepSeek and GPT runs achieve the best system. For factoid questions, DeepSeek IR_Y_3 and GPT IR_Y_5 significantly outperformed median, while BioMistral systems underperformed. For ideal answers, DeepSeek and GPT systems exceeded median. Random snippet shuffling did not harm BioMistral yes/no performance, but slightly decreased DeepSeek factoid performance, confirming that the impact of shuffling varies by model and metric.

Across the four batches, random snippet shuffling in Phase B exhibited non-monotonic effects similar to Phase A+: it improved factoid question ranking in some scenarios (e.g., BioMistral in Batches 2 and 3), had positive or neutral effects on list and ideal-answer questions, but occasionally caused sharp drops (e.g., BioMistral yes/no in Batch 3). Overall, random snippet shuffling serves as a lightweight robustness diagnostic, effectively revealing model sensitivity to snippet order, which depends on question type, model architecture, and retrieval context.

5. Conclusions and Future Work

This paper presents a lightweight knowledge-enhanced retrieval-augmented generation system for the BioASQ 2026 Task 14b biomedical question answering challenge. Instead of relying solely on lexical retrieval and neural reranking, our approach incorporates MeSH-based controlled synonym expansion to improve retrieval robustness under biomedical terminology variation and abbreviation ambiguity. The proposed framework combines structured query expansion, multi-stage PubMed retrieval, heuristic reranking, sentence-level evidence extraction, and question-type-aware answer generation within a unified DSPy-based workflow. We evaluate the system across multiple BioASQ phases and batches using several large language model backbones, including GPT-4o-mini, DeepSeek-V3, and BioMistral-7B.

The experimental results reveal several trends of general significance. First, our observations suggest that MeSH-based controlled synonym expansion may help alleviate recall loss caused by terminology variation. Particularly for factoid and list questions, the AND/OR structure of the resulting Boolean queries yields stable and performance gains compared with retrieval using the raw question text. Second, Preliminary observations indicate that sentence-level evidence snippets may provide more focused contextual grounding than full abstracts. Removing redundant and distracting information from abstracts helps large language models identify relevant answers more precisely. Third, we analyze models' sensitivity to context ordering through random snippet permutation experiments. This sensitivity varies across question types, including yes/no, factoid, list, and summary questions, as well as across different model architectures. Random snippet shuffling occasionally improves performance, especially for ideal answer generation and certain factoid questions, but may also lead to noticeable performance degradation for some yes/no questions. These findings suggest that models may rely not only on positional bias but also on local logical coherence, implying that context organization strategies should be carefully designed for different tasks.

In terms of overall performance,our system achieved competitive preliminary leaderboard performance and obtained top-ranked results in several preliminary evaluation batches for exact yes/no answers in several batches. These comparisons are based on the preliminary online leaderboard rather than the final official evaluation, while ranking among the top two for ideal answer generation. However, performance on list questions and some factoid questions remains below the median, indicating clear bottlenecks in handling multi-entity enumeration, relational reasoning, and answer deduplication.

Future work can be pursued along two directions: (1) introducing the hierarchical structure of MeSH and drug–gene–disease associations as a lightweight knowledge graph into the retrieval stage, moving from flat synonym expansion to subgraph-guided semantic retrieval [30, 12, 25, 28]; and (2) designing dedicated snippet aggregation and redundancy suppression strategies for list-type questions, and enhancing model robustness for generating list-structured answers under fixed retrieval contexts.

Acknowledgments

This research work was funded by the Provincial International Exchange and Cooperation Projects, grant no. HN–2026045.

Declaration on Generative AI

During the preparation of this manuscript, the author used ChatGPT in order to: Text translation. After using this tool, the author reviewed and edited the content as needed and takes full responsibility for the accuracy and integrity of the final version.

References

[1] K. Singhal, T. Tu, J. Gottweis, R. Sayres, E. Wulczyn, L. Hou, K. Clark, S. R. Pfohl, H. J. Cole-Lewis, D. Neal, M. Schaekermann, A. Wang, M. Amin, S. Lachgar, P. A. Mansfield, S. Prakash, B. Green, E. Dominowska, B. A. y Arcas, N. Tomaev, Y. Liu, R. C. Wong, C. Semturs, S. S. Mahdavi, J. K. Barral, D. R. Webster, G. S. Corrado, Y. Matias, S. Azizi, A. Karthikesalingam, V. Natarajan, Towards expert-level medical question answering with large language models, ArXiv abs/2305.09617 (2023). URL: https://api.semanticscholar.org/CorpusID:258715226.

[2] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W. tau Yih, T. Rocktäschel, S. Riedel, D. Kiela, Retrieval-augmented generation for knowledge-intensive NLP tasks, in: Advances in Neural Information Processing Systems 33 (NeurIPS 2020), 2020, pp. 9459–9474. URL: https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html.

[3] 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.

[4] 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.

[5] Y. Li, J. Zhao, M. Li, Y. Dang, E. Yu, J. Li, Z. Sun, U. Hussein, J. Wen, A. M. Abdelhameed, J. Mai, S. Li, Y. Yu, X. Hu, D. Yang, J. Feng, Z. Li, J. He, W. Tao, T. Duan, Y. Lou, F. Li, C. Tao, Refai: a gpt-powered retrieval-augmented generative tool for biomedical literature recommendation and summarization, Journal of the American Medical Informatics Association 31 (2024) 2030–2039. URL: https://doi.org/10.1093/jamia/ocae129. doi:10.1093/jamia/ocae129.

[6] S. E. Robertson, S. Walker, S. Jones, M. Hancock-Beaulieu, M. Gatford, Okapi at trec-3, in: Proceedings of TREC 1994, volume 500-225 of *NIST Special Publication*, NIST, 1994, pp. 109–126. URL: http://trec.nist.gov/pubs/trec3/papers/city.ps.gz.

[7] W. Wang, F. Wei, L. Dong, H. Bao, N. Yang, M. Zhou, Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers, in: Proceedings of NeurIPS 2020, NeurIPS Foundation, 2020. URL: https://proceedings.neurips.cc/paper/2020/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

[8] K. Song, X. Tan, T. Qin, J. Lu, T. Liu, Mpnet: Masked and permuted pre-training for language understanding, in: Proceedings of NeurIPS 2020, NeurIPS Foundation, 2020. URL: https://proceedings.neurips.cc/paper/2020/hash/c3a690be93aa602ee2dc0ccab5b7b67e-Abstract.html.

[9] J. H. Merker, A. Bondarenko, M. Hagen, A. Viehweger, Mibi at bioasq 2024: Retrieval-augmented generation for answering biomedical questions, in: Conference and Labs of the Evaluation Forum (CLEF), 2024, pp. 176–187. URL: https://ceur-ws.org/Vol-3740/paper-16.pdf.

[10] Neha, et al., Retrieval-augmented generation (rag) in healthcare: A comprehensive review, AI 6 (2025) 1–25. URL: https://www.mdpi.com/journal/ai. doi:10.3390/ai6010001.

[11] D. Edge, H. Trinh, N. Cheng, J. Bradley, A. Chao, A. Mody, S. Truitt, D. Metropolitansky, R. O. Ness, J. Larson, From local to global: A graph rag approach to query-focused summarization, arXiv preprint abs/2404.16130 (2024). URL: https://arxiv.org/abs/2404.16130. doi:10.48550/arXiv.2404.16130.

[12] S. Xiang, H. Lin, F. Cai, Z. Jiang, Integrating knowledge graphs with ancient chinese medicine classics: Challenges and future prospects of multi-agent system convergence, Chinese Medicine 20 (2025) 168. URL: https://doi.org/10.1186/s13020-025-01226-7. doi:10.1186/s13020-025-01226-7.

[13] J. Zhang, L. Ma, H. Liu, B. An, Q. Sun, Irg-rag: an iterative reflective graph retrieval-augmented generation method for chinese medicine, in: 2025 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), 2025, pp. 4440–4445. URL: https://doi.org/10.1109/bibm66473.2025.11356941. doi:10.1109/BIBM66473.2025.11356941.

[14] M. Neumann, D. King, I. Beltagy, W. Ammar, scispaCy: Fast and robust models for biomedical natural language processing, in: Proceedings of the 18th BioNLP Workshop and Shared Task (BioNLP@ACL 2019), Association for Computational Linguistics, 2019, pp. 319–327. URL: https://aclanthology.org/W19-5034/. doi:10.18653/v1/W19-5034.

[15] O. Khattab, A. Singhvi, P. Maheshwari, Z. Zhang, K. Santhanam, S. Vardhamanan, S. Haq, A. Sharma, T. T. Joshi, H. Moazam, H. Miller, M. Zaharia, C. Potts, Dspy: Compiling declarative language model calls into self-improving pipelines, arXiv preprint arXiv:2310.03714 (2023). URL: https://arxiv.org/abs/2310.03714. doi:10.48550/arXiv.2310.03714.

[16] A. Krithara, A. Nentidis, K. Bougiatiotis, G. Paliouras, Bioasq-qa: A manually curated corpus for

biomedical question answering, Scientific Data 10 (2023) 170.

[17] A. Nentidis, G. Katsimpras, A. Krithara, M. Krallinger, M. Rodríguez-Ortega, E. Rodriguez-López, N. Loukachevitch, A. Sakhovskiy, E. Tutubalina, D. Dimitriadis, G. Tsoumakas, G. Giannakoulas, A. Bekiaridou, A. Samaras, G. M. D. Nunzio, N. Ferro, S. Marchesin, M. Martinelli, G. Silvello, G. Paliouras, Overview of bioasq 2025: The thirteenth bioasq challenge on large-scale biomedical semantic indexing and question answering, in: J. C. de Albornoz, J. Gonzalo, L. Plaza, A. G. S. de Herrera, J. Mothe, F. Piroi, P. Rosso, D. Spina, G. Faggioli, N. Ferro (Eds.), Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Sixteenth International Conference of the CLEF Association (CLEF 2025), 2025.

[18] R. Luo, L. Sun, Y. Xia, T. Qin, S. Zhang, H. Poon, T.-Y. Liu, BioGPT: Generative pre-trained transformer for biomedical text generation and mining, Briefings in Bioinformatics 23 (2022) bbac409. URL: https://doi.org/10.1093/bib/bbac409. doi:10.1093/bib/bbac409.

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

[20] M. E. Chatzimina, N. S. Tachos, D. I. Fotiadis, M. Tsiknakis, Hemarag: A retrieval-augmented generation system for medical question answering in hematologic malignancies, in: 2025 IEEE EMBS International Conference on Biomedical and Health Informatics (BHI), 2025, pp. 1–4. doi:10.1109/bhi67747.2025.11269561, conference held Oct. 26-29, 2025, in Atlanta, GA, USA.

[21] J. Guo, Y. Fan, L. Pang, L. Yang, Q. Ai, H. Zamani, C. Wu, W. B. Croft, X. Cheng, A deep look into neural ranking models for information retrieval, Information Processing & Management 57 (2020) 102067. URL: https://doi.org/10.1016/j.ipm.2019.102067. doi:10.1016/j.ipm.2019.102067.

[22] T. Hwang, S. Jeong, S. Cho, S. Han, J. C. Park, Dslr: Document refinement with sentence-level re-ranking and reconstruction to enhance retrieval-augmented generation, in: Proceedings of the 3rd Workshop on Knowledge Augmented Methods for NLP (KnowledgeNLP@ACL 2024), Association for Computational Linguistics, Bangkok, Thailand, 2024, pp. 73–92. URL: https://aclanthology.org/2024.knowledgenlp-1.6/. doi:10.18653/v1/2024.knowledgenlp-1.6. arXiv:2407.03627.

[23] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W. tau Yih, T. Rocktäschel, S. Riedel, D. Kiela, L. Zettlemoyer, Retrieval-augmented generation for knowledge-intensive NLP tasks, in: Advances in Neural Information Processing Systems 33 (NeurIPS 2020), 2020, pp. 9242–9253. URL: https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html.

[24] Z. Shao, Y. Gong, Y. Shen, M. Huang, N. Duan, W. Chen, Enhancing retrieval-augmented large language models with iterative retrieval-generation synergy, in: Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP 2023), Association for Computational Linguistics, Singapore, 2023, pp. 620–635.

[25] W. Lin, B. Xing, X. Liu, Y. Xie, Knowledge-grounded rag for diagnosing spleen-stomach disorders in tcm, in: 2025 IEEE 6th International Conference on Computer, Big Data, Artificial Intelligence (ICCBD+AI), 2025, pp. 1–5. URL: https://doi.org/10.1109/iccbdai66607.2025.11388485. doi:10.1109/ICCBDAI66607.2025.11388485.

[26] O. Bodenreider, The unified medical language system (umls): Integrating biomedical terminology, Nucleic Acids Research 32 (2004) D267–D270. URL: https://doi.org/10.1093/nar/gkh061. doi:10.1093/nar/gkh061.

[27] M. C. Diaz-Galiano, M. T. Martin-Valdivia, L. A. Urena-Lopez, Query expansion with a medical ontology to improve a multimodal information retrieval system, Computers in Biology and Medicine 39 (2009) 396–403. URL: https://doi.org/10.1016/j.compbiomed.2009.01.012. doi:10.1016/j.compbiomed.2009.01.012.

[28] Q. Chen, L. Ni, Tcm mlkg-rag: Traditional chinese medicine intelligent diagnosis based on multi-layer knowledge graph retrieval-augmented generation, in: 2024 4th International Conference on Electronic Information Engineering and Computer Communication (EIECC), 2024, pp. 958–

962. URL: https://doi.org/10.1109/eiecc64539.2024.10929529. doi:10.1109/EIECC64539.2024.10929529.

[29] J. Wei, Y. Tay, R. Bommasani, C. Raffel, B. Zoph, S. Borgeaud, D. Yogatama, M. Bosma, D. Zhou, D. Metzler, et al., Emergent abilities of large language models, arXiv preprint arXiv:2206.07682 (2022). URL: https://arxiv.org/abs/2206.07682.

[30] R. Huang, A. Abudukeranmu, C. Tu, Z. Yu, Z. Cui, B. Du, An intelligent tcm diagnosis system with localized llms and modular multimodal reasoning enhanced by rag, in: 2025 IEEE Cyber Science and Technology Congress (CyberSciTech), 2025, pp. 407–414. URL: https://doi.org/10.1109/cyberscitech68397.2025.00061. doi:10.1109/CyberSciTech68397.2025.00061.