SynapFlow at BioASQ@CLEF 2026: Retrieval-Augmented Generation with Diversity-Enhanced Hybrid Retrieval*

CMU Team at CLEF 2026

John Dalton Gibson[1,*,†], Eric Nyberg[2,†]

[1]Carnegie Mellon University Africa, Regional ICT Center of Excellence Building, Kigali, Rwanda, BP 6150
[2]Language Technologies Institute, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213, USA

Abstract

Retrieval and selection of the most appropriate relevant documents are the primary challenges in biomedical question-solving (QA) using retrieval-augmented generation. Balancing diversity with relevance during retrieval ensures that two or more very similar documents are not among the top-ranked relevant documents, which could narrow the search scope and prevent other relevant documents from being selected. This paper describes our system that participated in the fourteenth edition of the BioASQ Task Synergy Challenge, as well as improvements made based on the challenge results. Our system employs query normalization and named entity extraction for query preprocessing; a hybrid information retriever that combines sparse BM25 and dense FAISS retrieval; document re-ranking using the MedCPT cross-encoder; maximal marginal relevance for diversity, and the OpenAI gpt-4o-mini-2024-07-18 model for answer generation. Our system participated in round 3 of Task Synergy, and the evaluation showed that it struggled with information retrieval, which affected answer quality. We optimized our system based on post-submission error analysis, and resolved four critical engineering failures – small corpus size of 549 documents used for Round 3, abstract-only encoding, max-only fusion-score normalization for hybrid retrieval, and malformed answer extraction – improving document MAP from 0.0078 to 0.1065, factoid MRR from 0.0455 to 0.3810, list F-measure from 0.2138 to 0.3241, and yes/no accuracy from 0.4545 to 1.0000 in our best ablation variant.

Keywords

Retrieval-Augmented Generation, Maximal Marginal Relivance, Dense Retrieval, Sparse Retrieval, Entity Extraction, Answer Generation, Large Language Models

1. Introduction

Because large language models (LLMs) generalize well in many tasks, prior to the introduction of retrieval-augmented generation, question answering involved the use of an LLM to directly provide answers to questions when prompted [1]. These LLMs are mainly successful in open-domain question answering, and this success usually depends on the use of very large datasets [1, 2]. Even with large datasets, these LLMs still struggle with domain-specific QA and hallucinate most of the time [3].

Retrieval-augmented generation solves this issue by exposing the relevant external literature to the LLM, providing it with context to generate an answer to a particular question [4, 2, 5, 6, 1]. However, finding the relevant literature for almost every query requires a very rich document corpus. PubMed provides a very large corpus of 40M biomedical literature documents. This exponentially growing corpus also makes exhaustive search infeasible, and effective retrieval remains a core challenge [6]. To consistently review and improve document retrieval, the BioASQ Task Synergy challenge uses an iterative retrieval setting in which systems refine their answers in multiple rounds based on expert feedback [7]. Task Synergy 14 consists of four rounds, as in Task Synergy 13, and the questions span four types of questions: `yes/no`, `factoid`, `list`, and `summary` [8, 9].

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

*Submitted to BioASQ Task Synergy 14 at CLEF 2026

*Corresponding author.

†These authors contributed equally.

jgibson2@andrew.cmu.edu (J. D. Gibson); en09@andrew.cmu.edu (E. Nyberg)

https://github.com/jdaltonll02/medicalragsys.git (J. D. Gibson); https://www.cs.cmu.edu/~ehn/ (E. Nyberg)

0009-0005-5977-0100 (J. D. Gibson); 0000-0001-8809-2728 (E. Nyberg)

Recent works have used a sparse retriever like BM25 to retrieve the relevant documents needed for the LLM to generate an answer to a query [10, 11], while others have used a dense retriever for the same task [12, 6, 5]. While a sparse retriever suffices most of the time when the target documents contain exact keywords from the query, a dense retriever can excel where a sparse retrieval fails by retrieving target documents that may not match exact query terms while containing terms that are considered semantically equivalent (having very similar word embeddings).

This paper describes SynapFlow, a multi-stage retrieval-augmented generation (RAG) pipeline that combines BM25 Elasticsearch retrieval with MedCPT-encoded FAISS dense retrieval, cross-encoder re-ranking, maximal marginal relevance (MMR) after re-ranking to ensure diversity, and gpt-4o-mini-2024-07-18 for answer generation. Our system participated in Round 3 of the BioASQ Task. A post-result error analysis reveals four key engineering failures that we resolved in building the first variant of our system. We performed four ablation studies on additional variants of our baseline system. It should be noted that several of the post-submission improvements address engineering failures that should have been addressed before submission. We report them transparently as lessons learned rather than novel research contributions, in the spirit of the negative-result reporting encouraged by the BioASQ community.

In Section 2, we review and discuss related work on sparse, dense, and hybrid retrieval, as well as on maximal marginal relevance for removing redundancy in re-ranked documents. In Section 3, we discuss our experiment setup, provide a complete description of our system, and our evaluation methodology. In Section 4, we discuss the competition results, the results of the different ablation studies, and our general findings. Section 6 provides conclusions, limitations, and directions for future work.

2. Related Work

The choice of information retrieval paradigm is a key design decision in retrieval-augmented generation (RAG). Recent advances in information retrieval have greatly improved biomedical question answering systems [13].

Several systems employ sparse retrieval by indexing PubMed Corpus documents using BM25 [10, 11]. Although sparse retrieval is effective in finding documents that contain exact keywords from the query, dense retrieval techniques also find semantically relevant documents that might be ignored by BM25. This has created a recent growing interest in dense retrieval, which uses a neural encoder to represent documents and queries and to find documents that are semantically related to these queries [12, 6, 5]. We built our system using two core approaches from information retrieval: hybrid information retrieval and maximal marginal relevance (MMR).

2.1. Hybrid and Two-Stage Information Retrieval

There has been a recent increase in interest in hybrid information retrieval that combines sparse retrieval, such as BM25, and dense retrieval, such as MedCPT. HU-WBI@BioASQ 12b Phase A system [14] uses BM25 for sparse first-stage retrieval and combines a variety of dense neural models via rank fusion for second-stage re-ranking, training neural models on question-answer pairs from previous BioASQ editions. RAG-BioQA [6] integrates BioBERT embeddings with FAISS indexing for retrieval and a LoRA fine-tuned FLAN-T5 for answer generation, comparing four retrieval strategies: dense retrieval (FAISS), BM25, ColBERT, and MonoT5, evaluated on PubMedQA, MedDialog, and MedQuAD.

While prior BioASQ tasks combined two retrieval techniques, they essentially did not compute a weighted sum of sparse and dense retrieval to form a hybrid score. They predominantly relied on two-stage retrieval pipelines, in which BM25 performs initial candidate selection, and a dense neural model subsequently re-ranks those candidates. Although effective, this architecture limits the recall ceiling to what BM25 retrieves in the first stage. In contrast, a more direct hybrid retrieval runs sparse (BM25) and dense (FAISS) retrievers in parallel and computes a hybrid score by adding the scores from both, weighted by their contribution to the fused score. This approach allows each to compensate for the other's weakness, with improved recall when compared to two-stage designs [2].

2.2. Maximal Marginal Relevance (MMR)

Re-ranking the documents found in a retrieval step can improve the likelihood that the most relevant documents appear at the top of the ranked list, which is important in retrieval-augmented generation (since there is usually a limit to the number of passages that can be included in the LLM prompt) [5] [2]. However, re-ranking without filtering out redundant documents can push other highly relevant documents further down in the ranked list, where they may be ignored by RAG selection. The introduction of maximal marginal relevance (MMR) as a criterion for reducing redundancy [15] while maintaining query relevance in re-ranking retrieved documents and in selecting appropriate passages has been a major advancement for text summarization. Several other works have used MMR to remove redundancy. DF-RAG [16] is based directly on the MMR framework to select information chunks that are both relevant to the query and maximally dissimilar from each other, applied to complex reasoning-intensive QA benchmarks. Sometimes, MMR is used directly as a retrieval step. In GeneRAG, the MMR algorithm is used to improve the recovery quality within an RAG pipeline to answer gene-related questions [17].

Many prior works have applied MMR as a standalone retrieval strategy or combined with a neural encoder for text summarization. However, no prior work in biomedical QA applies MMR as a post-reranking redundancy filter following a neural encoder (although Seeberger and Riedhammer [18] used a neural reranker followed by MMR to retrieve relevant documents and remove redundancy, they applied it to summarization rather than to the QA task). In our system, we adopt this sequential retrieval, re-ranker, and MMR within our RAG pipeline, where MMR is applied to the neural re-ranker's output to remove redundant documents and ensure diversity.

3. Methodology

In this section, we describe the datasets used, data preparation, experimental setup, an overview of the methodologies used by our system, and the key design choices. We first describe the system that participated in round 3 of the BioASQ Task Synergy 14, and then describe the changes made to improve it based on the Task Synergy results. For testing, we used the test data provided for round 3 of Synergy 14 on the BioASQ website. The test data set comprises 64 questions that span four question types: 12 yes/no questions, 11 factoids, 15 list questions, and 18 summary questions. There is no publicly available BioASQ evaluation dataset for Synergy 14. However, in each round, the feedback for the previous round is provided. Because round 4 feedback was based on round 3 submissions, the improved system was evaluated on the 64 answer-ready questions out of the 117 questions from round 4 feedback, which were curated into the golden data set.

3.1. Datasets and Resources

We used the PubMed-designated dataset for the Synergy task. Task Synergy 14 designates a snapshot version of each round, consisting of the annual baseline data files and updated files released up to the snapshot date. Since our system only participated in round 3, we used a sampled corpus from the annual baseline and all data files released for the round 3 snapshot date for submission. To improve the system after the release of Task Synergy results, we used the full PubMed Corpus, consisting of forty million documents.

3.1.1. Data Download

We design a PubMed fetcher that fetches raw data from the PubMed database using email and the PubMed API. The PubMed fetcher retrieves abstracts and their PubMed Identifications (PMIDs) in batches of 200. The raw PubMed articles are downloaded as raw Extended Markup Language (XML) files.

3.1.2. XML Parsing

The raw XML files are then passed into the PubMed Parser. The PubMed parser extracts the PMID, article title, abstract, publication date, author, and digital object identifier (DOI). The publication month for each article is given in the text. Therefore, the parser converts each month to its corresponding numerical representation. The parsed XML files are combined into a single large JSON file.

3.1.3. Data Preparation Per Round

We prepared a sample of data specifically for round 3. The per-round data preparation includes the same data download and XML parsing steps, but only a subset of the 40 million-document corpus is prepared at this stage. The data is sampled from the round's snapshot. This data includes a sampled corpus from the annual baseline, files released from previous rounds, and snapshot dates up to the round 3 snapshot date. This is done so that when round 3 test questions are run through the pipeline, the relevant documents for these questions will be available in the corpus. The same PubMed fetcher is used, but this time it fetches based on the snapshot date rather than the full corpus. We extracted 640 documents from the PubMed database using the PubMed API, of which 549 were unique. The pipeline was run on 64 questions. The same XML parser is also used to parse the raw round 3 XML files, and the data loader loads the round 3 JSON corpus into the pipeline.

3.1.4. Data Loading

We design a BioASQ Loader to load data into the pipeline. The data loader reads and parses the test and evaluation data. It loads the test set and golden dataset, filters questions by type, retrieves questions by ID, extracts document IDs, and extracts snippets. The purpose of the data loader is to allow the rest of the pipeline to access the structured BioASQ data for retrieval, answer generation, and evaluation.

3.1.5. Computing Resources

The complete QA pipeline was run on a NVIDIA A100 GPU, 160GB RAM, and 32 CPU cores. To index the PubMed corpus of 40M documents using BM25, we used Elasticsearch 8. We built a FAISS index of 122GB of disk storage.

3.1.6. Evaluation Protocol

The Round 3 submission was evaluated by the BioASQ Task Synergy Team based on expert feedback. Following error analysis based on the results of the round 3 evaluation, we also evaluated using the official BioASQ evaluation [19] to obtain realistic performance in accordance with the BioASQ standard. The evaluator performs the evaluation in two phases. Phase-A reports mean average precision (MAP), mean precision, recall, and F1 in top-5 for document and snippet retrieval. Phase-B reports yes/no accuracy, yes / no macro-F1, factoid straight accuracy, factoid MMR, list precision, list recall, and list F1. The summary evaluation is manual and is not reported using this evaluation.

3.2. System Overview

SynapFlow answers biomedical questions via a mobile app interface, passing them through a multi-step pipeline to generate reliable, accurate responses. The major phases in the pipeline are query processing, document retrieval, document re-ranking for diversity, snippet extraction, and answer generation (see Figure 1).

3.2.1. User Interface

We designed a Flutter mobile application where users can register and start asking questions. The mobile application makes an API request to the RAG system hosted in the cloud, which triggers the full

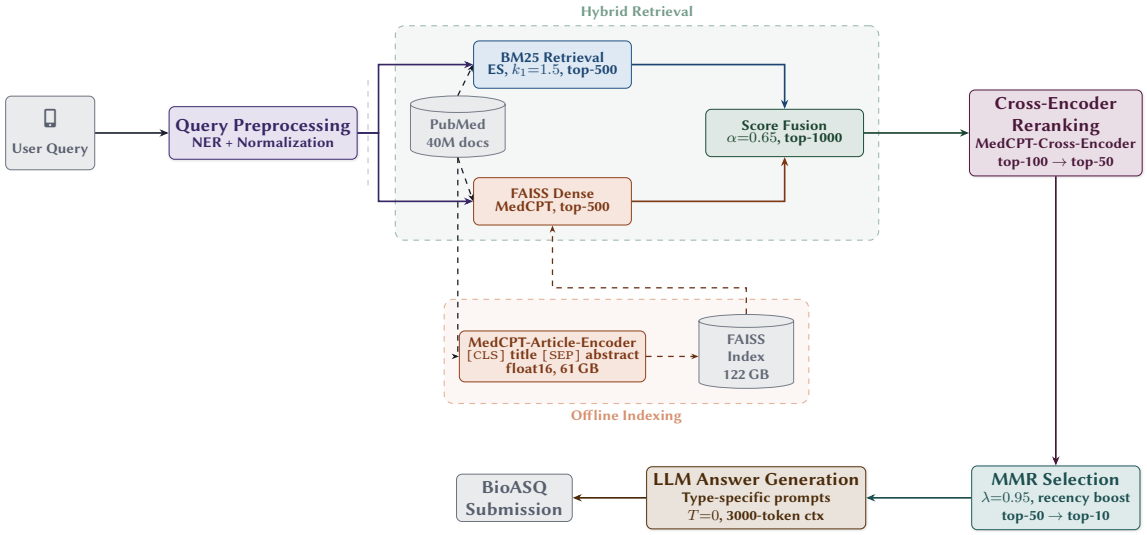

Figure 1: SynapFlow Full Architecture

pipeline to generate the answer. The answer is returned to the user interface.

3.3. Query Preprocessing

We used two key techniques to process the user's query: query normalization and named-entity recognition.

3.3.1. Query Normalizer

The pipeline takes the raw text and passes it through the query normalization. The query normalizes Unicode by decomposing combined characters into a base form plus a modifier, thereby reducing search/NER mismatches caused by different Unicode forms and making indexing and comparisons more consistent. It also converts all text to lowercase, removes punctuation, removes trailing whitespace, removes extra whitespace, and truncates text.

3.3.2. Named-Entity Recognition (NER)

The normalized query is fed into the named-entity recognition (NER) module. NER [1] is used to extract biomedical entities from the question (genes, diseases, drugs, proteins, etc.) so that the retrieval process can focus on medically relevant terms. The idea is that when using BM25 alone, too many words in the query might reduce the relevance score, since it relies on exact keyword matching. Therefore, for BM25, entity terms are promoted with a configurable weight to increase sparse recall for exact term matches. For FAISS, the entity surface forms are appended to the normalized query before encoding, improving the dense retrieval signal for rare biomedical terms. The NER model used in this pipeline is the d4data/biomedical-ner-all model from Huggingface.

3.4. Documents Retrieval

The information retrieval model retrieves relevant documents based on the query, providing context for the LLM to generate an accurate answer. We explore three retrieval techniques: sparse, dense, and hybrid (a combination of sparse and dense).

3.4.1. Facebook AI Similarity Search (FAISS)

We use FAISS as an indexing mechanism and search engine to store and query vectors, and the MedCPT encoder to create embedded images from both documents and queries [2]. MedCPT is a set of contrastive pre-trained transformers (query encoder and document encoder) [20]. As formulated by Jin, et al. [20], MedCPT encodes the query by taking in a query q, wrapping it in BERT special tokens ([CLS] at the beginning and [SEP] at the end, and running it through a Transformer (Trm). Document encoding is done by taking the document d, concatenating the abstract and title into a single long string, and passing it through the transformer. Mathematically, as shown in Jin et al. [20], we have

$$E(q) = Q\text{Enc}(q) = \text{Trm}([\text{CLS}]\ q\ [SEP])_{[\text{CLS}]} \in \mathbb{R}^h \tag{1}$$

and

$$E(d) = D\text{Enc}(d) = \text{Trm}([\text{CLS}]\ d^{title}\ [\text{SEP}]\ d^{abstract}\ [\text{SEP}])_{[\text{CLS}]} \in \mathbb{R}^h \tag{2}$$

We used the same ncbi/MedCPT-Query-Encoder and ncbi/MedCPT-Article-Encoder used by Jin et al. [20] for encoding queries and documents, respectively. The type of index used is IndexFlatIP (inner-product similarity over L2-normalized vectors) to provide the exact equivalent of Cosine Similarity while ensuring maximum retrieval precision.

3.4.2. Best Match 25 (BM25)

BM25 determines relevance using scores computed from how often a term appears in a document (term frequency), how rare the term is throughout the corpus (inverse document frequency or IDF) and how long the document is relative to the average document length. This is good for exact keyword matches [21]. We calculate the BM25 score using the formulation from [21]. For a document D and a query Q of words or terms $Q = w_1, w_2, ..., w_m$, the BM25 score is:

$$\text{BM25}(D, Q) = \sum_{w \in Q} \text{IDF}(w) \cdot \frac{f(w, D) \cdot (k_1 + 1)}{f(w, D) + k_1 \cdot B(D)} \tag{3}$$

Where

- $f(w, D)$ is the frequency of a word w in the document D
- IDF(t) is the inverse document frequency of the word t
- $B(D)$ is the length normalization factor

For our system, we use Elasticsearch for indexing and BM25 for retrieving the relevant documents. We use the parameter values $k_1 = 1.5$ and $b = 0.75$ to adjust how the algorithm computes the score. When k_1 is higher, it means that the score will increase rapidly the more times a word appears in a document. The b parameter is used for length normalization. The reason is that longer documents can unfairly receive higher scores than shorter ones. The b parameter penalizes longer documents, ensuring that shorter documents are not completely ignored.

3.4.3. Issues with BM25-Only and MedCPT-Only Retrievers

One big limitation of MedCPT is biased retrieval [20]. MedCPT uses a transformer and, like many other transformer models, learns the semantic relationships between queries and documents via its attention heads. The problem is that not all semantically similar documents are clinically relevant to the query, leading to false positives. On the other hand, BM25 does not search for a semantic relationship. Instead, it looks for documents that contain exactly the words present in the query. However, one challenge of BM25 is that a document may be contextually similar to a query but not contain the exact same words, and BM25 will consider it irrelevant (leading to false negatives).

3.4.4. Hybrid Retrieval

We use a hybrid retrieval technique, dense semantic retrieval (FAISS) plus sparse lexical retrieval (BM25) [2]. The rationale behind using both search techniques is that, for some questions, the key entities may appear heavily in many documents, facilitating an exact match, but for others, the exact-matched document might exist but may not contain many of the query's keywords. With BM25 alone, these documents will be discarded, but with FAISS plus BM25, both semantically related documents and documents with exact keywords from the query are considered. The final score is the alpha-weighted sum of the scores from FAISS and BM25, as shown in the equation below.

$$\alpha \cdot s_{\text{dense}}(d) + (1 - \alpha) \cdot s_{\text{sparse}}(d) \tag{4}$$

In this system, we retrieve the top 100 documents using FAISS and the top 100 using BM25. After combining both, we select the top 50 documents.

3.5. Documents re-ranking

Re-ranking is implemented by a cross-encoder model (pritamdeka/S-PubMedBert-MS-MARCO) [2]. It takes the input query and all 50 candidate documents, jointly scores each query-document pair, replaces each document's score with the re-ranking score, and sorts the documents in descending order. The top 10 documents are selected for the extraction of snippets.

3.6. Maximal Marginal Relevance (MMR)

To ensure diversity while maintaining relevance, we implement MMR [15], formulated as:

$$MMR \stackrel{\text{def}}{=} \text{Arg} \max_{D_i \in R \setminus S} \left[\lambda \left(Sim_1(D_i, Q) - (1 - \lambda) \max_{D_j \in S} Sim_2(D_i, D_j) \right) \right] \tag{5}$$

This prevents very similar documents from occupying the top-ranked documents. It takes three inputs: re-ranked documents, their embeddings, and the query embedding. It ensures diversity by first selecting the document with the highest relevance score and then selecting further documents to maximize relevance while penalizing similarity to already selected documents [15].

3.7. Snippets Extraction

After retrieving the relevant documents, the system splits the questions into key terms, then scans each retrieved document, and selects the sentence that best matches those terms. The selected sentence is returned as a snippet with character offsets and section labels, truncating long snippets to keep them concise.

3.8. Answer Generation

The final answer is generated by making an API request to an LLM. The request payload consists of the contents of the finally selected documents, the original query, the question type, and a strong few-shot system prompt. The LLM uses this extra payload information for context. For this system, we use the gpt-4o-mini-2024-07-18 from OpenAI.

3.8.1. Prompting

Alongside the retrieved documents, the instruction-tuned LLM needs a structured prompt that provides instructions on how the answers should be generated. Two prompts are provided: a role-based prompt or system prompt and a user prompt. The role-based prompt, also known as the system prompt, guides the LLM on how to answer the questions. It sets boundaries and constraints. The user prompt is the prompt that presents the question to the LLM. This prompt consists of the actual question from the

user, the context consisting of the retrieved top-ranked documents, and some additional instructions.

Listing 1: An Example of System and User Prompt

```
[
 {
  "role": "System"
  "content": "You are a biomedical expert answering BioASQ questions.
  Use the provided documents as your primary source. If insufficient,
  use your expertise. Never say information is unavailable. Always
  provide your best answer.

  FORMAT RULES (follow exactly):

  YES/NO questions:
  - Line 1: only "yes" or "no"
  - Line 2+: 2-3 sentence explanation

  FACTOID questions:
  - Line 1: the answer entity only such as bare name, number, or short
  phrase, no sentence, no prefix
  - Line 2+: explanation with evidence
  - If multiple answers, one per line (max 5), most specific first

  LIST questions:
  - One item per line, numbered, short phrases only (no sentences)
  - 1. First item
  - 2. Second item
  - After the list: one sentence summary

  SUMMARY questions:
  - A single paragraph, 2-4 sentences, max 200 words"
  },
  {
  "role": "User"
  "content": "Question: {question.get('body')}

  Retrieved documents:
  {context}

  Generate a concise answer (1-3 sentences)."
  }
]
```

3.8.2. Answer Post-Processing

To ensure that answers strictly follow the BioASQ submission format, we design a BioASQ formatter that converts raw retrievals and generated text into a BioASQ dictionary format. The formatter builds the required fields, including id, body, type, documents, and snippets. It then applies answer gating, so that non-answer-ready items get an empty exact answer and an ideal answer. Additionally, it normalizes snippets with section labels and character offsets. It also enforces question-type answer shapes. That is, yes/no answers are presented as strings, factoid/list answers are presented as nested arrays, and summary answers are presented with only ideal text. Lastly, before submission, a validation check ensures that all required fields are present, that per-type answers are in the correct format, document/snippet limits are met, and that the 200-word cap for ideal answers is met.

4. Results and Analysis

In this section, we analyze and discuss the performance of our system that participated in Task Synergy Round 3, as well as variants of this system, based on error analysis and ablation studies conducted after the Round 3 results. The official Synergy 14 Results are on BioASQ's results page. Since we only submitted for round 3 due to resource issues, results are only available for round 3 in our system, with the description (MedRAG Q/A System). For this study, we have six different systems. We have the baseline system that participated in the competition, along with five variants derived from four ablation studies and error analysis. We begin by analyzing the official BioASQ evaluation results. A critical analysis of the system's errors is discussed but is not included in this section. It is included in Appendix A and Appendix B. Next, we discuss the different variants of the system and their results.

4.1. Evaluation Metrics

The baseline system that participated in the competition and its variants used the official BioASQ evaluation metrics [22]. The metrics are classified into two phases: Phase A and Phase B. Phase A focuses on document retrieval and snippet extraction, while Phase B focuses on answer generation.

4.1.1. Phase A Evaluation Metrics

In phase A, we evaluate document retrieval and snippet extraction. For document retrieval and snippet extraction, we use mean precision, recall, F-Measure, Mean Average Precision (MAP), and Geometric Mean Average Precision (GMAP).

4.1.2. Answer Generation

: In Phase B, we evaluate the quality of the generated answers. For yes/no questions, we measure the answer accuracy, F1, and Macro F1. For factoid questions, we use strict and lenient precision. For list answers, we use mean precision, recall, and F-Measure. Automatic scores (Rouge-R) and manual scores (readability, recall, precision, and repetition) were used for ideal answers and are reported only for the system that participated in the competition (baseline system).

4.2. Information Retrieval Results

Round 3 Submission shows our system's performance on the fourteenth task, synergy, as indicated by our MAP score of 0.0078 and GMAP of 0.001 in Table 1, ranking our system eighth out of ten for document retrieval. Table 1 also shows that our system achieved a mean precision of 0.0272, which means that, on average, only 0.272 or 2.72% of the documents retrieved per question were actually relevant. This clearly shows that our system struggled to correctly find and rank relevant documents, which in turn affects the answer quality, which we discuss later in this paper. The recall value of 0.0255 in Table 1 also reinforced our findings that the system struggled to find relevant documents. A recall of 0.0255 means that, for each question, the system finds only 2.55% of the relevant documents. The low MAP also indicates that, across all questions, the system struggled to correctly rank the relevant documents we occasionally retrieved.

Table 2 shows that the snippet's scores were even lower, with a mean precision of 0.00245, a MAP of 0.0049 and a GMAP value of 0.0000. The mean precision of 0.0245 indicates that among the text characters our system extracted, only 2.45% matched the text in the expert's golden retrieved snippets. The low MAP of 0.0049 indicates that for the vast majority of the 64 questions, the most relevant snippets do not rank highly in the final top snippets, as in the document's case.

4.3. Answer Generation

In this sub-section of results and analysis, we focus on the performance of our system at Synergy 14. The quality of the generated answers depends on both the LLM and the retrieved documents.

Table 1
Document Retrieval: Performance of SynapFlow@BioASQ Task Synergy 14 other variants of SynapFlow. Bold values represent the system with the best result.

System	Description	Mprec	MRec	MF1	MAP	GMAP
SynapFlow	Round 3 Submission	0.0272	0.0255	0.0249	0.0078	0.0001
SynapFlow_v1	First Improved	0.1282	0.0998	0.0887	0.0894	0.0013
SynapFlow_v2	BM25-Only	0.1385	0.1025	0.0941	0.0978	**0.0018**
SynapFlow_v3	No Lowercase_Norm	0.1282	0.0998	0.0887	0.0894	0.0013
SynapFlow_v4	No Normalizer	**0.1496**	**0.1056**	**0.0995**	**0.1065**	0.0014
SynapFlow_v5	No MMR	0.1282	0.0998	0.0887	0.0894	0.0013

Table 2
Snippet Extraction: Performance of SynapFlow@BioASQ Task Synergy 14 and other variants of SynapFlow. Bold values represent the system with the best result.

System	Description	MPrec	MRec	MF1	MAP	GMAP
SynapFlow	Submitted System	0.0245	0.0073	0.0107	0.0049	0.0000
SynapFlow_v1	First Improved	0.0667	0.0209	0.0286	0.0502	0.000153
SynapFlow_v2	BM25-Only	0.0716	**0.0231**	0.0308	0.0523	**0.000206**
SynapFlow_v3	No Lowercase_Norm	0.0669	0.0209	0.0287	0.0502	0.000153
SynapFlow_v4	No Normalizer	**0.0735**	0.0229	**0.0312**	**0.0541**	0.000153
SynapFlow_v5	No MMR	0.0669	0.0209	0.0287	0.0502	0.000153

Table 3
3 Yes/No Answer Performance of SynapFlow@BioASQ Task Synergy 14 and other variants of SynapFlow. Bold values represent the system with the best result.

System	Description	Accuracy	MacroF1	F1-yes	F1-no
SynapFlow	Round 3 Submission	0.4545	0.4500	0.5000	0.4000
SynapFlow_v1	First Improved	0.9545	0.9494	0.9655	0.9333
SynapFlow_v2	BM25-Only	0.8636	0.8610	0.8800	0.8421
SynapFlow_v3	No Lowercase_Norm	0.9091	0.9018	0.9286	0.8750
SynapFlow_v4	No Normalizer	**1.0000**	**1.0000**	**1.0000**	**1.0000**
SynapFlow_v5	No MMR	0.9091	0.9018	0.9286	0.8750

For yes/no questions, Table 3 shows an accuracy score of 0.4545, a MacroF1 score of 0.4500, an F1-yes score of 0.5000, and an F1-no score of 0.40000. For the list question type, Table 4 shows a mean precision of 0.3378, a recall of 0.2136, and an F-measure of 0.2138. This shows that, compared to other question types, the system performed better on yes/no questions and then on list questions.

For factoid questions, Table 5 shows that the system achieved 0.0000 straight accuracy, 0.0909 lenient accuracy, and 0.0455 MMR. This shows that the system struggled with factoid questions and ideal answers.

Table 6 shows the automatic ROUGE scores, with an R-2 Recall of 0.0548 and an R-2 F1 of 0.0626. ROUGE-2 measures the bigram overlap between the generated ideal answers and the golden ideal answers. The strongest systems for ideal answers in the competition had ROUGE-2 F1 scores in the range of 0.17-0.20. An R-2 F1 score of 0.0626 is low compared to these systems in the competition, but

since biomedical QA is typically low, the score is not worse than the scores for factoid questions. The R-SUA metric is a variant of ROUGE that uses unigrams and skip-bigrams, awarding partial credit for word overlap. The R-SUA score is slightly higher than the R-2 score. What this means is that even when the exact phrases of the ideal answers do not match the ideal answers in the golden file, some answers in the submission shared some individual medical terms with the ideal answers in the golden file. The key finding here is that LLMs are generally good at ideal answers, and since there is no perfect gold standard for ideal answers, even without good retrieval, the system still managed to obtain many correct ideal answers compared to other question types.

Table 4
List Answer Performance of SynapFlow@BioASQ Task Synergy 14 and other variants of SynapFlow. Bold values represent the system with the best result.

System	Description	MPrec	Recall	F-Measure
SynapFlow	Round 3 Submission	0.3378	0.2136	0.2138
SynapFlow_v1	First Improved	**0.3212**	**0.4520**	0.3204
SynapFlow_v2	BM25-Only	0.3173	0.4502	**0.3241**
SynapFlow_v3	No Lowercase_Norm	0.2817	0.4237	0.2870
SynapFlow_v4	No Normalizer	0.3104	0.4388	0.3114
SynapFlow_v5	No MMR	0.2434	0.4006	0.2571

Table 5
Factoid Answer Performance of SynapFlow@BioASQ Task Synergy 14 and other variants of SynapFlow. Bold values represent the system with the best result.

System	Description	Strict Acc	Lenient Acc	MRR
SynapFlow	Round 3 Submission	0.0000	0.0909	0.0455
SynapFlow_v1	First Improved	**0.3810**	**0.3810**	**0.3810**
SynapFlow_v2	BM25-Only	0.2381	0.2381	0.2381
SynapFlow_v3	No Lowercase_Norm	0.1905	0.1905	0.1905
SynapFlow_v4	No Normalizer	0.2857	0.2857	0.2857
SynapFlow_v5	No MMR	0.2381	0.2381	0.2381

Table 6
Ideal Answer Performance (Automatic Scores) of SynapFlow@BioASQ Task Synergy 14. The BioASQ evaluator doesn't score ideal answers. Therefore, we only present scores for the Round 3 Submission.

System	Description	R-2 (Rec)	R-2 (F1)	R-SUA (Rec)	R-SUA (F1)
SynapFlow	Round 3 Submission	0.0548	0.0626	0.0578	0.0644

Table 7
Ideal Answer Performance (Manual Scores) of SynapFlow@BioASQ Task Synergy 14.

System	Description	Readability	Recall	Precision	Repetition
SynapFlow	Round 3 Submission	4.11	3.69	3.82	4.38

4.4. Post-Error Analysis Improvements

To better understand why our system performed poorly in document and snippet retrieval, we created an error analyzer that loads the submitted file for Task Synergy Round 3 and the Round 4 feedback file, which is based on Round 3 submissions. The analyzer first runs a corpus coverage check. Basically, for each question, this analysis computes the fraction of the documents found in the expert feedback that are actually in the initial 549-document index. This is because a retrieval system is highly dependent on the availability of documents in the document store. Therefore, even with highly functional information retrieval, when relevant documents are not available in the document store, the information retrieval module will perform poorly [23]. The analyzer also checked for LLM failures on some questions. It looks for fallback phrases like "information not available", and answer formats like the length of factoid answers. The analyzer calculated metrics and rendered diagnostic plots. A detailed error analysis for information retrieval can be found in Appendix A.

Generally, the system did not perform well in answer generation, since information retrieval also performed poorly. A detailed error analysis for answer generation can be found in Appendix B.

4.4.1. Improvements for Information Retrieval

The first variant of our system was a direct improvement on the system submitted for Round 3. The biggest change was that, instead of the 549-document corpus, we encoded and re-ingested the entire 40M-document PubMed corpus into our index. The improved system now runs on the whole pipeline. The second change was to adjust the scoring fusion alpha weighting from 0.5 for both BM25 and FAISS to 0.65 for FAISS and 0.35 for BM25. We also changed the encoder model from S-PubMedBert-MS-MARCO to ncbi/MedCPT-Article-Encoder for the article encoder and ncbi/MedCPT-Query-Encoder for the query encoder. We re-encoded the documents to include the title and not just the abstract. The new encoding captured both the title and abstract. Another improvement made was a change from max normalization to min-max normalization. Instead of dividing by the maximum only, the new system (SynapFlow_v1) first subtracts the minimum, ensuring the range is always 0-1. It also fixed the negative score issue. Negative scores no longer penalize documents. We used the golden file, curated from expert feedback, to evaluate SynapFlow_v1 performance on the newly generated submission file. The precision shown in Table 1 improved hugely from 0.0272 to 0.1282 by +0.101. MAP improved hugely from 0.0078 to 0.0894 by +0.0816. From almost 0 GMAP, GMAP increased by +0.0012. Table 2 also shows an improvement in snippet retrieval as MAP increased by +0.0453 from 0.0049 to 0.0502, and GMAP came from 0.000 to 0.000153. While the results in Tables 1 and 2 show that there is still room for improvement, they also revealed that the changes made had a huge impact on retrieval, and this impact will be reflected more in answer generation since answer generation also relies on retrieval quality.

4.4.2. How the Improvement in IR Affected Answer Quality for SynapFlow_v1

Table 3 shows that for yes/no, the improved system increased the precision from 0.4545 to 0.9545 by +0.500 and the recall from 0.4500 to 0.9494 by +0.4994. F1-yes went from 0.500 to 0.9655, and F1-no went from 0.4000 to 0.9333. This is the result of an improvement in information retrieval. Retrieving relevant documents improved the LLM's confidence when handling yes/no questions. The improvement in information retrieval also affected the other types of questions. Table 5 shows an extraordinary improvement for factoid questions. Table 5 also shows that the straight precision went from 0.0000 to 0.3810, the lenient precision went from 0.0909 to 0.3810, and the MMR went from 0.0455 to 0.3810. SynapFlow_v1 produced the best performance for factoid questions among all systems. Why information retrieval played a huge role in this improvement: the addition of an answer validator, improvements to the few-shot prompt, and improvements to the BioASQ formatter for factoid questions also contributed to this improvement. A slight drop in mean precision by 0.0166, but an improvement in mean recall from 0.2136 to 0.4520. Additionally, Table 8 shows the Phase B comparison between SynapFlow's best post-submission variant and the official Round 3 participants. The results show that yes/no from SynapFlow_v4 and list from SynapFlow_v2 would have outperformed RMC_2LM5KF_4, Fleming-1, and

Fleming-2 in answer generation for yes/no and list questions if we had submitted these variants for round 3.

Table 8
Phase B comparison of SynapFlow best post-submission variant against official Round 3 participants. Best value per metric shown in **bold**. SynapFlow results are post-submission ablation variants: yes/no from SynapFlow_v4, factoid from SynapFlow_v1, list from SynapFlow_v2.

	Yes/No				Factoid			List		
System	Acc.	F1-yes	F1-no	MacroF1	Strict	Lenient	MRR	Prec.	Rec.	F1
RMC_2	0.8182	0.8750	0.6667	0.7708	0.1818	0.1818	0.1818	0.3689	0.2218	0.2295
RMC_4	0.8182	0.8750	0.6667	0.7708	0.1818	0.1818	0.1818	0.3689	0.2218	0.2295
Fleming-1	0.8182	0.8750	0.6667	0.7708	0.2727	0.4545	0.3364	**0.5744**	**0.4705**	**0.4810**
Fleming-2	0.8182	0.8750	0.6667	0.7708	0.2727	0.4545	0.3364	**0.5744**	**0.4705**	**0.4810**
SynapFlow (best)	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**0.3810**	**0.3810**	**0.3810**	0.3173	0.4502	0.3241

4.5. Post-Submission Ablation Study

To understand the impact of different parts of the architecture, we conducted several ablation studies that yielded new variants of our systems: SynapFlow_v2, SynapFlow_v3, SynapFlow_v4, SynapFlow_v5, SynapFlow_v3, SynapFlow_v4, and SynapFlow_v5.

4.5.1. BM25-Only Ablation

Since we were using hybrid retrieval, it is important to determine whether either retrieval paradigm affects the hybrid score. We designed a new system variant (SynapFlow_v2) to test how BM25 would perform without FAISS. Since alpha is the weight of FAISS, we set alpha to 0.00, making the retrieval BM25-only. Table 1 shows that SynapFlow_v2 (BM25-only) achieves the second-highest Phase A document MAP (0.0978) and, surprisingly, the highest List F1 across all experiments (0.3241) from 4. This exceeds even the full pipeline's 0.3204, suggesting that BM25's strong keyword recall for list questions — which often contain multiple named entities — outweighs the advantage that dense retrieval brings for semantic diversity. However, the Yes/No accuracy (0.8636) and the Factoid MRR (0.2381) are substantially lower than those of the entire pipeline (0.9545 and 0.3810, respectively), confirming that BM25 alone is insufficient for answer types that require semantic understanding.

4.5.2. Removing Query Normalization

We maintained the SynapFlow_v1 configuration but completely removed query normalization, creating a new system called SynapFlow_v3. Removing the normalizer entirely produces the best Phase A MAP (0.1066), a 19% improvement over SynapFlow_v1, and perfect Yes/No accuracy. The raw query is more effective for both Elasticsearch BM25 (which uses its own standard analyzer) and the MedCPT query encoder (trained on naturally cased PubMed queries). The tradeoff is a 14-point drop in Factoid MRR (0.2857 vs 0.3810 for fullpipeline), as shown in Table 5, indicating that the normalizer's preprocessing benefits downstream factoid answer extraction even while hurting retrieval recall.

4.5.3. Maintain Normalizer but Remove Lowercase

We designed SynapFlow_v4 by retaining query normalization while removing uppercase-to-lowercase conversion. Preserving the case while still removing punctuation is the worst overall configuration. Phase A metrics are identical to Phase A metrics for SynapFlow_v1 as shown in Tables 1 and 2, but the factoid MRR of Phase B drops to 0.1905 — lower than all other configurations, including only BM25. The most likely cause is a mismatch in entity boosting: NER extracts entities in their original case form ("BRCA1", "COVID-19") and passes them to Elasticsearch as BM25 boost terms.

4.5.4. Disabling MMR

To assess the impact of introducing MMR into our system, we disabled it in an experiment. We maintain all configurations of SynapFlow_v1 except for the MMR, and we produce SynapFlow_v5. Disabling MMR is the most damaging single change for Phase B answer quality. Factoid MMR drops from 0.3810 to 0.2381 by 0.143 as shown in Table 5, and List F1 drops from 0.3204 to 0.2571 by 0.063 as shown in 4. The absence of the MMR caused the cross-encoder's top-10 documents to be highly similar to one another, resulting in redundancy. This affects the LLM, giving it insufficient coverage to answer multi-part factoid and list questions. The results in Tables 1 and 2 show that the Phase A metrics for SynapFlow_v5 are identical to the Phase A metrics for SynapFlow_v1, indicating that MMR affects only the document selection passed to the LLM, and not the retrieval pool used for the evaluation of Phase A. This is exactly the effect of the lack of diversity.

4.6. Key Findings

One major finding was that the quality of information retrieval depends not only on the retriever's quality or the effectiveness of the retrieval paradigm but also on the richness of the corpus. Our error analysis revealed that, due to a small corpus size of 549 documents during round 3, many questions in the test set lacked relevant documents in the index. We addressed this challenge by running the pipeline on the full document corpus. Initially, we hypothesize that introducing MMR in the pipeline will ensure diversity in the re-ranking of the retrieved documents. This hypothesis was validated by our ablation study that disabled MMR in the pipeline. The study showed that disabling MMR caused the top 10 documents in the re-ranker's output to be highly redundant and identical, resulting in a reduction in factoid MMR and list F1. A shocking finding was that the BM25-only retriever in SynapFlow_v2 outperformed the hybrid retriever in SynapFlow_v1. This finding revealed that FAISS was damaging the pipeline. It should be noted that these comparisons are based on 64 questions and have not been subjected to significant testing. Therefore, the differences should be interpreted as indicative rather than conclusive. Future work should focus on exploring different dense retrievals, first running the full pipeline on them, and identifying the best-performing dense retrieval before combining it with BM25. Finally, we discovered that our query normalization was too aggressive and was also affecting retrieval. The results in Table 1 show that SynapFlow_v4 (the non-normalizer variant) beats SynapFlow_v1 in information retrieval. Removing the query normalizer increased the MAP for the document retriever from 0.0894 to 0.1065.

5. Conclusion

Our paper presented a multi-stage retrieval-augmented generation (RAG) pipeline, which we designed to study and address biomedical challenges associated with question answering in the BioASQ Task Synergy. In this system, we design a hybrid retrieval module that combines the scores of dense FAISS and sparse BM25 retrievals into a single hybrid score. We also introduced a neural encoder for re-ranking and subsequently applied MMR for re-ranking. To prepare the query for retrieval, we also applied a two-stage query preprocessing. Our system participated only in Round 3 of Task Synergy. We first applied query normalization to clean the query, then applied NER to extract entities and key medical terms. While hybrid retrieval, which combines sparse and dense retrieval methods, is established in the broader information retrieval (IR) literature, its systematic ablation in the BioASQ Synergy setting – where questions are short, keyword-heavy, and iterative – provides new practical insights for future participants. While MMR is not new to information retrieval, our system was the first to apply MMR directly to the output of a cross-encoder re-ranker to filter out redundant documents in the retrieved documents for a QA task.

Addressing the challenges of biomedical question answering is one of BioASQ's objectives. The BioASQ evaluation on our system and our error analysis exposed four engineering failures, which reflected some of these challenges, which we discussed in our findings. We addressed these challenges

through four studies, yielding four variants of our system, plus an additional variant based on errors identified in post-error analysis discussed in Appendix A and Appendix B. We used our performance in Task Synergy 14, round 3, as our baseline, and our best-performing system for Phase A and yes/no questions from Phase B was the system that ran the full pipeline without a query normalizer. Although removing query normalization improved information retrieval, the entire pipeline used in SynapFlow_v1 still performed best on list and factoid questions.

In summary, SynapFlow results suggest that (i) re-ranking with MMR reduces redundancy among top-ranked documents, which improves coverage across the answer space, (ii) the raw query contains naturally-cased biomedical terms and punctuation that are better handled by Elasticsearch's own standard analyzer and MedCPT query encoder than our query normalizer, (iii) Yes/no and factoid questions favor a pipeline with query normalization, while list questions favor a BM25-only pipeline, and (iv) the FAISS retriever in the hybrid-retrieval module did not contribute to the retrieval, causing BM25 to suffice over hybrid retrieval which we initially proposed.

Our systems have several limitations. One of the biggest limitations of this work is that our system only participated in round 3 of Task Synergy 14, which prevents us from analyzing the behavior of the system throughout the iterative Synergy rounds. The current work did not establish a strong basis for determining if hybrid retrieval is better than dense or sparse. Future work should focus on conducting more experiments across different dense retrieval paradigms before combining them with BM25 to determine which dense retrieval technique performs best in biomedical QA tasks and which performs best with BM25. The current pipeline uses a fixed alpha to determine the weights of sparse and dense retrieval when computing the fusion score. Future work should focus on using an adaptive technique that adjusts alpha based on the query type. The current pipeline uses a fixed top-k value for all questions. Some questions require a broader scope of documents retrieved due to their type or nature, while others require a narrower scope. Future work should consider using a question-type-aware final top-k. BioASQ Synergy provides feedback from previous rounds. This work does not incorporate feedback via pseudo-relevance feedback for re-weighting or negative-example filtering. Considering this in future work could substantially improve the recall of questions that appeared in earlier rounds. The current pipeline uses IndexFlatIP (exact nearest neighbor search). For a 40M-vector corpus, future work should consider approximating nearest neighbor indices to reduce query latency and boost accuracy. Since our focus was on retrieval techniques, we used only a single GPT model as the LLM component in our pipeline; future work should also consider the effect of different LLMs on overall performance.

Acknowledgments

The authors thank the anonymous reviewers for their detailed comments on the submitted paper, which helped us to improve the final version. We also thank the Language Technologies Institute (LTI) at Carnegie Mellon University for providing computing resources through the LTI Babel Cluster.

Declaration on Generative AI

During the preparation of this work, the authors used Claude Sonnet for LaTeX formatting and syntax correction, and Grammarly for grammar and spelling checks. After using these tools, all AI-assisted content was carefully reviewed and edited by the authors as needed, and the authors assume full responsibility for its publication.

References

[1] F. Borazio, A. Shcherbakov, D. Croce, R. Basili, UniTor at BioASQ 2025: Modular biomedical QA with synthetic snippets and multiple task answer generation, in: G. Faggioli, N. Ferro, P. Rosso, D. Spina (Eds.), CLEF 2025 Working Notes, 2025, pp. 165–197. URL: https://ceur-ws.org/Vol-4038/paper_10.pdf.

[2] L. Stuhlmann, M. A. Saxer, J. Fürst, Efficient and reproducible biomedical question answering using retrieval augmented generation, in: 2025 IEEE Swiss Conference on Data Science (SDS), IEEE, Zürich, Switzerland, 2025, pp. 154–157. doi:10.1109/SDS66131.2025.00029.

[3] J. Ryan, A. I. Gumilang, R. Wiliam, D. Suhartono, Self-MedRAG: A self-reflective hybrid retrieval-augmented generation framework for reliable medical question answering, arXiv preprint arXiv:2601.04531, 2026. doi:10.48550/arXiv.2601.04531.

[4] B.-C. Chih, J.-C. Han, H.-C. Hung, R. T.-H. Tsai, NCU-IISR: Biomedical question answering via Gemini and GPT APIs in the BioASQ 13b phase B challenge, in: G. Faggioli, N. Ferro, P. Rosso, D. Spina (Eds.), CLEF 2025 Working Notes, 2025, pp. 206–213. URL: https://ceur-ws.org/Vol-4038/paper_12.pdf.

[5] S. Dueñas-Romero, L. A. Ureña-López, E. Martínez-Cámara, SINAI at BioASQ@CLEF 2025: A multi-stage RAG pipeline for biomedical semantic question answering, in: G. Faggioli, N. Ferro, P. Rosso, D. Spina (Eds.), CLEF 2025 Working Notes, 2025, pp. 243–265. URL: https://ceur-ws.org/Vol-4038/paper_15.pdf.

[6] L. Y. Panchumarthi, S. Saleti, S. P. Gudari, A. Negi, P. R. Budime, H. Upadhya, RAG-BioQA: A retrieval-augmented generation framework for long-form biomedical question answering, arXiv preprint arXiv:2510.01612, 2026. doi:10.48550/arXiv.2510.01612.

[7] 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.

[8] 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.

[9] A. Krithara, A. Nentidis, E. Vandorou, G. Katsimpras, Y. Almirantis, M. Arnal, A. Bunevicius, E. Farre-Maduell, M. Kassiss, V. Konstantakos, S. Matis-Mitchell, D. Polychronopoulos, J. Rodriguez-Pascual, E. G. Samaras, M. Samiotaki, D. Sanoudou, A. Vozi, G. Paliouras, BioASQ Synergy: a dialogue between question-answering systems and biomedical experts for promoting COVID-19 research, Journal of the American Medical Informatics Association (2024) ocae232. doi:10.1093/jamia/ocae232. arXiv:https://academic.oup.com/jamia/advance-article-pdf/doi/10.1093/jamia/ocae232/58

[10] J. H. Merker, A. Bondarenko, M. Hagen, A. Viehweger, MiBi at BioASQ 2024: Retrieval-augmented generation for answering biomedical questions, in: Working Notes of CLEF 2024 — Conference and Labs of the Evaluation Forum, volume 3740, CEUR-WS.org, 2024, pp. 176–187. URL: https://ceur-ws.org/Vol-3740/paper-16.pdf.

[11] J.-C. Han, B.-C. Chih, H.-C. Hung, R. T.-H. Tsai, NCU-IISR: A retrieval-augmented generation approach for BioASQ 13b phase A and A+, 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_19.pdf.

[12] S. Verma, F. Jiang, X. Xue, Beyond retrieval: Ensembling cross-encoders and GPT rerankers with LLMs for biomedical QA, arXiv preprint arXiv:2507.05577, 2025. doi:10.48550/arXiv.2507.05577.

[13] C. Hauff, et al. (Eds.), Advances in Information Retrieval: 47th European Conference on Information Retrieval, ECIR 2025, Lucca, Italy, April 6–10, 2025, Proceedings, Part V, volume 15576 of *Lecture Notes in Computer Science*, Springer Nature Switzerland, Cham, 2025. doi:10.1007/978-3-031-88720-8.

[14] O. Şerbetçi, X. D. Wang, U. Leser, HU-WBI at BioASQ 12b phase A: Exploring rank fusion of dense retrievers and re-rankers, in: Working Notes of CLEF 2024 — Conference and Labs of the Evaluation Forum, volume 3740, CEUR-WS.org, 2024. URL: https://ceur-ws.org/Vol-3740/paper-23.pdf.

[15] J. Goldstein, J. Carbonell, Summarization: (1) using MMR for diversity-based reranking and (2) evaluating summaries, in: Proceedings of the TIPSTER Text Program Phase III, Association for Computational Linguistics, Baltimore, Maryland, 1998, p. 181. doi:10.3115/1119089.1119120.
[16] S. H. Khan, S. Hong, J. Wu, K. Lybarger, Y. Yin, E. Babinsky, D. Liu, DF-RAG: Query-aware diversity for retrieval-augmented generation, arXiv preprint arXiv:2601.17212, 2026. doi:10.48550/arXiv.2601.17212, accepted to Findings of EACL 2026.
[17] M. Li, B. Baumgartner, Y. Zhang, W. Chen, GeneRAG: Enhancing large language models with gene-related tasks by retrieval-augmented generation, bioRxiv preprint, 2024. doi:10.1101/2024.06.24.600176.
[18] P. Seeberger, K. Riedhammer, Combining deep neural reranking and unsupervised extraction for multi-query focused summarization, arXiv preprint arXiv:2302.01148, 2023. doi:10.48550/arXiv.2302.01148, crisisFACTS Track, TREC 2022.
[19] P. Malakasiotis, I. Pavlopoulos, I. Androutsopoulos, A. Nentidis, Evaluation Measures for Task B, Technical Report, BioASQ, 2018. URL: http://participants-area.bioasq.org/Tasks/b/eval_meas_2018.
[20] Q. Jin, W. Kim, Q. Chen, D. C. Comeau, L. Yeganova, W. J. Wilbur, Z. Lu, MedCPT: Contrastive pre-trained transformers with large-scale PubMed search logs for zero-shot biomedical information retrieval, Bioinformatics 39 (2023) btad651. doi:10.1093/bioinformatics/btad651.
[21] S. Robertson, H. Zaragoza, The probabilistic relevance framework: BM25 and beyond, Foundations and Trends in Information Retrieval 4 (2009) 1–174. doi:10.1561/1500000019.
[22] P. Malakasiotis, I. Pavlopoulos, Evaluation measures for task B, BioASQ Participants Area, 2021. URL: https://participants-area.bioasq.org/Tasks/synergy/eval_measures_2021/.
[23] S. Barnett, S. Kurniawan, S. Thudumu, Z. Brannelly, M. Abdelrazek, Seven failure points when engineering a retrieval augmented generation system, arXiv preprint arXiv:2401.05856, 2024. doi:10.48550/arXiv.2401.05856.

A. Information Retrieval Error Analysis

Based on the analyzer's results, we identified the following factors that contributed to the system's poor performance in information retrieval.

A.1. Small Corpus Size

We mentioned earlier that in Round 3, the system indexed a corpus of 549 documents. There are 40M documents in the full PubMed Corpus, and only 549 were used. The error analysis revealed that the total number of documents in the golden dataset, curated from expert feedback, was 1,537. Of these, only 13 were actually present in our 549-document corpus (0.85%).

Figure 2 shows two plots: corpus coverage per question versus fraction of golden documents in the corpus, and golden documents reachable versus golden documents that are not indexed in the corpus of 549 documents. The plot shows that 55 out of 64 test questions had zero coverage in the corpus. This is indicated in the plot by the spike at 0, showing that 55 of 64 questions had zero overlap between our index and the required documents in the experts' feedback. The plot on the right of Figure 2 shows that of 1,537 documents in the golden file curated from expert feedback, 1,524 were never indexed. We concluded that one reason for poor retrieval is that most relevant documents were not available in the corpus due to its small size, which explains the low mean precision in 1.

Figure 3 shows the score distribution by question type for document retrieval. The plot on the left shows how many relevant golden documents there are per question, based on the experts' feedback. The plot on the left also shows that the summary question had the most relevant golden documents available. The factor questions had the fewest relevant golden documents. The middle plot shows the number of relevant documents retrieved compared with the number available in the feedback. This plot shows that only a few dots intersected the diagonal, indicating that very few of the retrieved documents were actually in the expert's feedback. The right panel shows the number of true positives per retrieval.

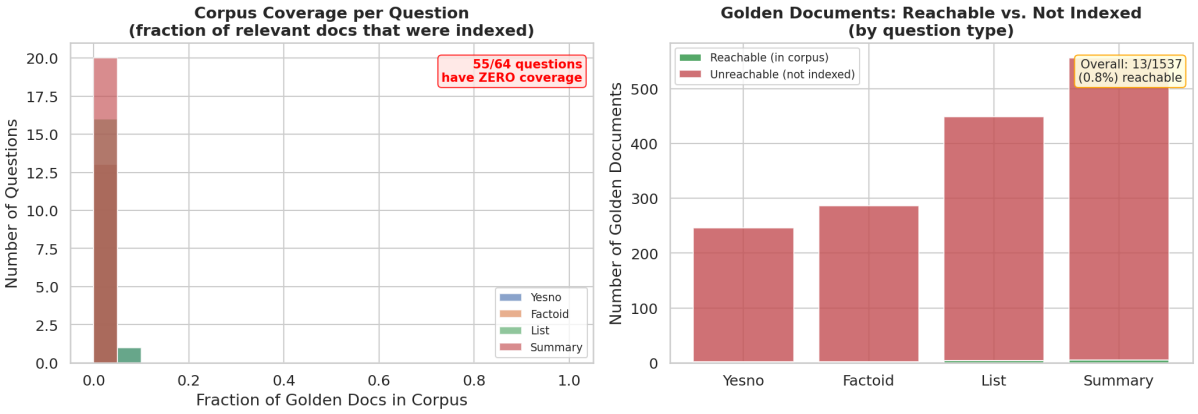

Figure 2: Corpus Coverage Per Question: The left panel shows a histogram of per-question corpus coverage. The right panel shows the stacked bar of reachable vs. unreachable golden docs per question type.

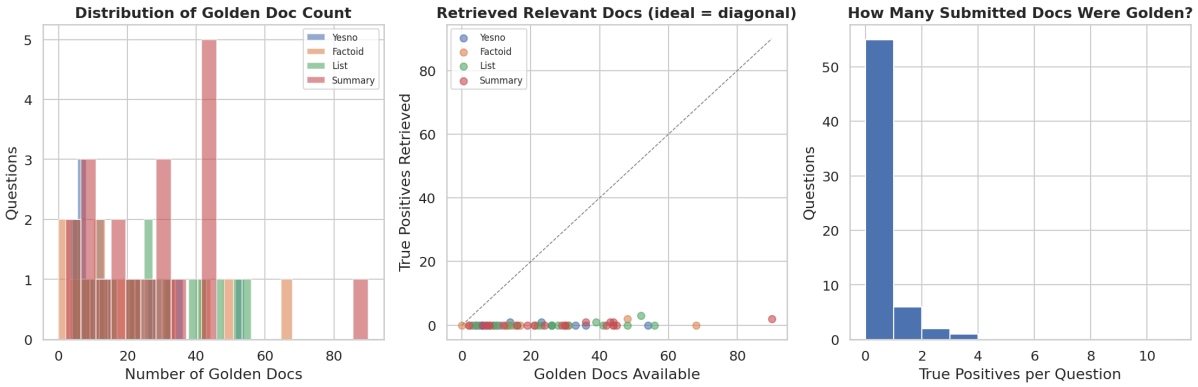

Figure 3: Distribution of Golden Doc Count, Retrieved Relevant Docs, and True Positives

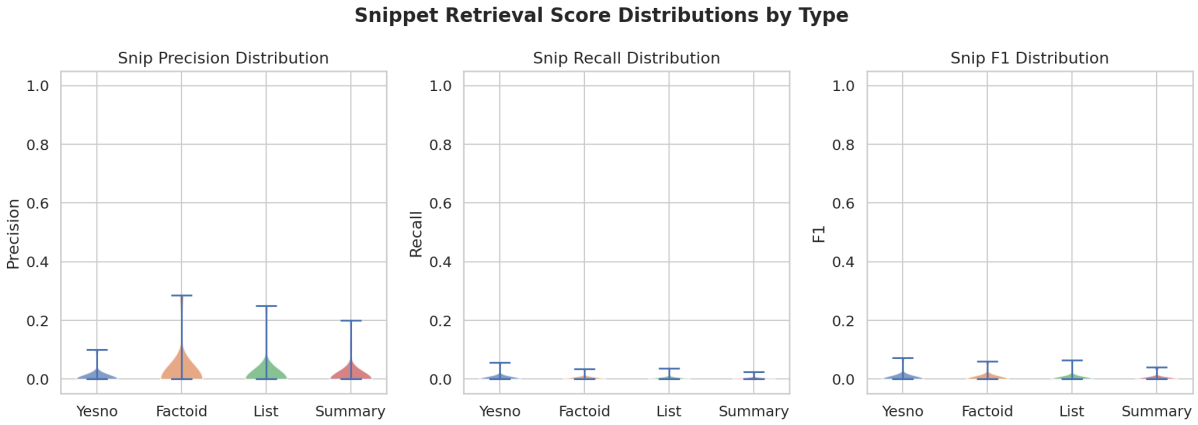

Figure 4: Snippets Retrieval Score Distribution by Question Type: These scores are different from the BioASQ official evaluator scores. They're the scores from the Error Analyzer.

Again, only a very few questions had a true probability up to 4. The vast majority of questions returned 0 matching documents.

Figure 4 shows the retrieval of the snippets. performance broken down by question type across three metrics, calculated by the error analyzer. A key observation from the plot on the left is that factoid and

list questions show slightly wider violin shapes than yes/no and summary questions. This means that a small number of factoid and list questions had non-zero precision scores, reaching up to 0.25. The recall and F1 for the middle and left appear to be uniformly flat across all question types. This near-zero recall confirms that even for partially matched snippets, they overlapped only a very small part of the golden snippet text.

A.2. Malfunctioning Reranker

Table 1 shows a very low MAP score of 0.0078 for document retrieval. One key finding was that even for relevant documents that were retrieved, the reranker model pushed them down the list. For the system that generated the Round 3 submission, we used the S-PubMedBert-MS-MARCO model from [2]. Later in this paper, we will show how changing this model improved the MAP and GMAP scores.

A.3. Abstract-Only Encoding

The MedCPT-Article Encoder was trained on query-PubMed article pairs. During model training, it learned to capture the meaning of both the title and the abstract of each article [20]. In contrast, our encoder model encoded only the abstracts.

A.4. Max-Only Scoring Normalization

When combining the BM25 and FAISS scores into the final score, both scores must be normalized to the same scale. BM25 produces unbounded scores, while FAISS produces scores based on cosine similarities (ranging from 0 to 1). Adding the raw scores together is not effective; our normalizer divides each score by the maximum score in the result (so that all scores range from 0 to 1). Due to the scarcity of relevant documents in the corpus, most FAISS scores were near zero. An example is shown below:

Document	FAISS score	Note
doc_A	$+0.0023$	
doc_B	$+0.0019$	
doc_C	-0.0008	
doc_D	$+0.0031$	random FAISS “winner”
doc_E	$+0.0015$	

The scores for each of those documents show that none of them is actually relevant. The winner won because it is slightly above the others. After dividing by the maximum (0.0031), the normalization gave:

Document	Normalized score	Effect
doc_A	0.742	
doc_B	0.613	
doc_C	-0.258	penalized
doc_D	1.000	Appears as fully relevant
doc_E	0.484	

Table A.4 shows that a randomly selected irrelevant document received a normalized FAISS score of 1.0, making it difficult to distinguish a truly top-ranking result. The key observation here is that the normalization converted a meaningless numerical accident into a confident relevance signal, which is misleading. This corrupted the fused score. For fusing the round 3 scores, we used an alpha of 0.5, making both BM25 and FAISS equally relevant. Therefore, the combined score formula was:

$$s(d) = 0.65 \cdot s_{\text{dense}}(d) + 0.35 \cdot s_{\text{sparse}}(d) \tag{6}$$

From our example in A.4, let us examine doc_D (the random FAISS winner) against doc_X (a document BM25 correctly identified as highly relevant). We will have:

$$s(\text{doc\_D}) = 0.5 \times 1.000 + 0.5 \times 0.10 = 0.55$$
$$s(\text{doc\_X}) = 0.5 \times 0.613 + 0.5 \times 1.000 = 0.8065$$

From the above results, the only reason why doc _X survived is that BM25 was highly confident. If we look for any question where BM25 was less decisive, the random FAISS winner would displace the correct document entirely due to normalization. Additionally, documents with negative cosine similarities, which arise when two vectors point in slightly opposing directions, were actively penalized. For this reason, a document correctly ranked by BM25 will be pushed down when the final ranking is applied due to a negative contribution derived entirely from a meaningless near-zero cosine similarity. This affected BM25's contribution to the final score fusion.

B. Error Analysis for Answer Generation

To better understand why response generation failed for most question types, the same error analyzer was used to investigate system errors affecting response generation.

B.1. Yes/No Confusion Matrix

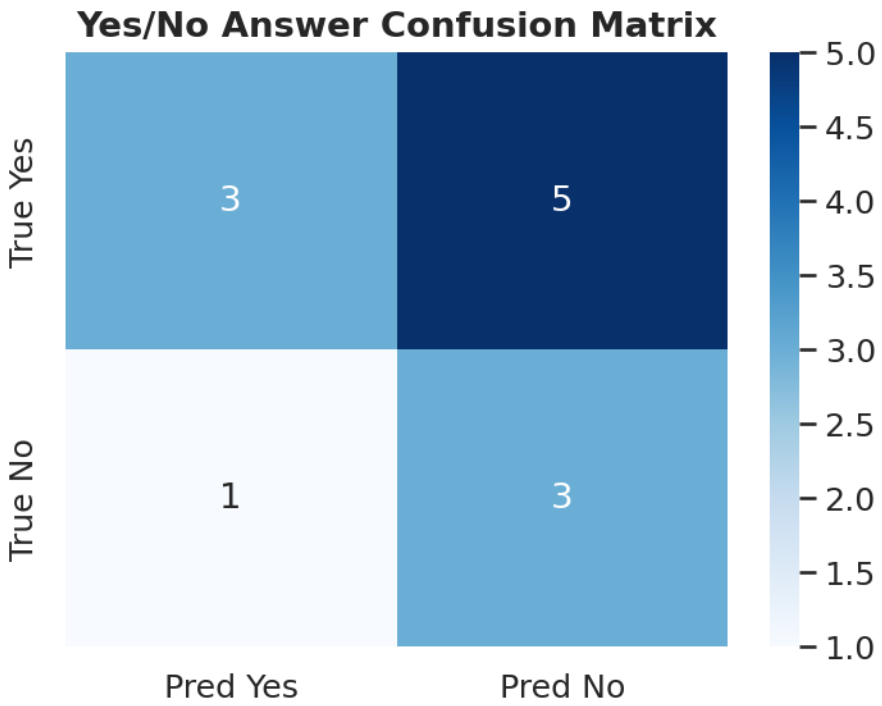

Figure 5: Yes/No Confusion Matrix

The yes/no accuracy of 0.4545 indicates that the LLM was influenced by irrelevant documents retrieved. If the LLM is not confident, as shown in the confusion matrix in 5, the system defaults to "no". In the confusion matrix, the system answered "no" to 5 of the 6 questions that had a correct answer of "yes". The confusion matrix shows that the system only incorrectly predicted "yes" when it should have predicted "no". This shows a strong bias towards "no". Also, from the confusion matrix, 5 of the 6 wrong answers were false negatives (predicted "no" when the answer was "yes"). The key finding

is that LLMs default to a negative/cautious answer when they receive documents that do not contain relevant content.

B.2. Factoid Format Compliance Failure

The error analysis showed that 31% of the factoid answers were formatted as prose rather than entity strings. Whenever the LLM does not receive relevant documents, there is no constraint that forces it to output in the [["EntityName"]] format. Instead, the LLM used fallback phrases like "Information not available" or "The main entity is...". A key observation was that there was no answer validator to constrain the LLM to return answers in the required format. The plot on the left of figure 6 shows that factoid F1 is bimodal. It is either 0 (no matched entity) or the score is partial, with the entity present but not informative.

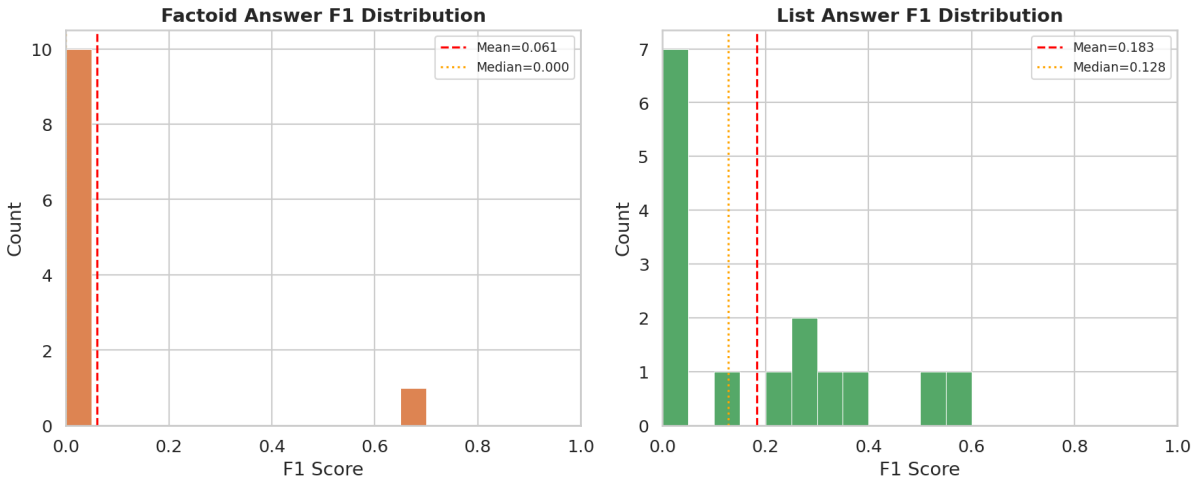

Figure 6: Factoid and List Answer F1 Distribution

Table 6 also shows that the list F1 is more spread compared to the factoid question. This means that even though many questions have very low F1 scores, some also have good scores.