lasigeBioTM: A Lean Biomedical QA System Empowered by Structured Knowledge

Notebook for the BioASQ Task 1-b: Biomedical Semantic Question Answering Lab at CLEF 2025

Paulo R. C. Lopes[1,*,†], Sofia I. R. Conceição[1,†], Maria Fernandes[2] and Francisco M. Couto[1]

[1] *LASIGE, Departamento de Informática, Faculdade de Ciências, Universidade de Lisboa, 1749–016 Lisboa, Portugal*
[2] *Novo Nordisk Foundation Center for Basic Metabolic Research, University of Copenhagen, Copenhagen, Denmark*

Abstract

Biomedical Question Answering (BQA) presents unique challenges due to the vast and complex nature of biomedical literature and the highly specialized terminology involved. In the BioASQ13 challenge, we participated in all phases (A, A^+, and B), proposing a hybrid system that integrates large language models (LLMs) with structured knowledge sources, namely ontologies and knowledge graphs. Our approach combines retrieval-augmented generation for document and snippet selection with graph-constrained decoding, which improves factual accuracy and supports reasoning grounded in biomedical knowledge. Its "lean" aspect arises from the use of a lightweight LLM and efficient techniques that minimize dependence on the extensive computational resources often associated with larger models. While our baseline system, lasigeBioTM, consistently performed well on exact answers across most batches, the integration of external information occasionally introduced bottlenecks or noise, highlighting how challenging it is to balance external knowledge and model fluency. Our results show that the system performance was closely tied to the quality of the retrieved snippets, with Phase B generally outperforming A^+ due to better snippet quality. Our findings emphasize the value of high-quality retrieval and controlled knowledge integration for improving BQA performance. The code is publicly available at https://github.com/lasigeBioTM/BioASQ13_2025.

Keywords

Constrained Decoding, Knowledge Graphs, LLMs, Ontologies, RAG

1. Introduction

Biomedical Question Answering (BQA) is a critical challenge within the demanding domain of biomedicine, addressing the hurdle of efficiently extracting insights from the vast and ever-growing biomedical literature. Such challenges are encompassed and advanced through initiatives like the BioASQ challenge [1], which promotes the generation of accurate and comprehensive answers to expert curated biomedical questions. BioASQ fosters innovation and enables comparative assessmentt through gold-standard snippets, reference answers, and a standardized evaluation framework [1, 2]. In particular, BioASQ13 Task b [3] addresses two primary challenges: Phase A, which focuses on managing the volume and complexity of biomedical literature, and Phase B, which involves generating precise and contextually appropriate answers. Our participation encompassed both tasks and the combined challenge (Phase A+).

Effectively addressing these tasks of extracting relevant information and producing reliable answers from the vast landscape of scientific literature demands powerful models capable of understanding and reasoning. For many years, transformer-based models fine-tuned on domain-specific corpora have enhanced BQA performance [4, 5], although at the cost of extensive computational resources and large

CLEF 2025 Working Notes, 9 – 12 September 2025, Madrid, Spain

[*] Corresponding author.

[†] These authors contributed equally.

prlopes@lasige.di.fc.ul.pt (P. R. C. Lopes); sconceicao@lasige.di.fc.ul.pt (S. I. R. Conceição); maria.fernandes@sund.ku.dk (M. Fernandes); fjcouto@ciencias.ulisboa.pt (F. M. Couto)

0009-0001-8599-4432 (P. R. C. Lopes); 0000-0002-8891-3546 (S. I. R. Conceição); 0000-0002-8733-1018 (M. Fernandes); 0000-0003-0627-1496 (F. M. Couto)

[1] https://www.bioasq.org/

labeled datasets.
More recently, Large language models (LLMs) have transformed natural language processing, offering unprecedented capabilities in understanding, generating, and synthesizing information. Despite these efforts, these systems still encounter limitations in addressing complex, domain-specific questions, such as hallucinations or the static nature of its intrinsic knowledge. These shortcomings can be mitigated by integrating structured information representations such as ontologies and knowledge graphs (KGs), that provide domain knowledge. Ontologies offer a formal representation of knowledge, providing a structured and standardized vocabulary for biomedical concepts and their relationships (e.g., Gene Ontology (GO) [6] and Medical Subject Headings (MeSH)), while KGs represent information as a network of entities and their relationships.

Retrieval-augmented generation (RAG) is a technique that empowers LLMs to go beyond their internal knowledge by incorporating up-to-date and domain-specific external information. This serves as a dynamic basis for generating answers, especially for very specific or recently published details. Combining RAG and LLMs offers several key benefits: (i) improvement of factual accuracy, by retrieving reliable external knowledge to support the answers and increase their verifiability; (ii) integration of recently published information enabling updates with the latest knowledge; and (iii) handling of specialized terminology by retrieving snippets from the literature or knowledge graphs, improving the interpretation of complex terms [7].

To address domain-specific challenges, we present lasigeBioTM a hybrid approach that focuses on open-source resources, combining LLM strengths with structured knowledge integration of KGs and ontologies. By integrating structured knowledge, lasigeBioTM aims to enhance factual accuracy, semantic understanding, reasoning capabilities, and answer explainability. Nevertheless, integrating ontologies and KGs with LLMs remains an ongoing research challenge [8].
The main consideration for lasigeBioTM's development is resource efficiency. Our system is developed to run without computationally intensive fine-tuning, facilitating adoption by institutions with limited resources. This "lean" design utilizes a lightweight LLM (Mistral-7B) with efficient 4-bit quantization and employs RAG to selectively retrieve relevant information from the vast corpus of biomedical literature found in PubMed. Participation in the BioASQ challenge provides a rigorous benchmark for evaluating lasigeBioTM 's performance.

2. Related Work

2.1. BioASQ challenge

The BioASQ13 challenge provided the training dataset 13b, which contains around 5,380 biomedical questions with corresponding answers ("exact" and "ideal") and supporting evidence (documents and snippets). The testing dataset, similarly to previous years, contains around 300 new questions [1].

In **Phase A**, participants address the challenge of handling the large volume and complexity of Biomedical literature, in our case, PubMed. The aim in this task is to efficiently and accurately retrieve the most relevant snippets from the literature to answer each given question. This becomes challenging due to domain-specific language and the contextual interpretation required to assess relevance.

In **Phase B**, the challenge is to combine contextual resources (i.e., information snippets) and potentially domain-specific external knowledge to reason and produce a precise and appropriate answer. This task requires the understanding of biomedical concepts, the reasoning over the provided support information and the generation of a tailored answer in line with each question.

The competition includes four types of questions: yes/no, list, factoid, and summary. For all types except summary questions, the participants provide an exact and an ideal answer, where the second is a more comprehensive, human-like answer providing support information for the given answer.

2.2. Biomedical QA systems

BQA systems typically work as a multistep pipeline addressing two primary challenges: (i) retrieving relevant information (documents or snippets) from vast literature, and (ii) providing precise answers based on the retrieved context [9].

Document and snippets retrieval: Traditional information retrieval models (e.g., BM25) rely on lexical matching, term frequency, and inverse document frequency to rank the documents and snippets, which are extremely efficient and effective for queries with clear keyword overlap. However, they are highly affected by vocabulary mismatch (e.g., use of synonyms and different phrasing) [10]. On the other hand, dense retrieval models (e.g., BERT-based models) encode questions and documents/snippets into embeddings and then compute the relevance based on the similarity between the embeddings. This approach better captures the semantic and contextual similarity but comes at the cost of increased computational overhead and remains vulnerable to irrelevant or sparse vocabulary. Finally, hybrid retrieval approaches combine both lexical and dense retrieval models, aiming to balance efficiency and semantic understanding while addressing the individual limitations of each model. However, combining them optimally is an open challenge.

Answer generation: Previous works used transformers-based models (e.g., BioBERT [11]) with fine-tuning on labeled datasets with relevant document text. This provides a high-confident text source and ensures factual correctness, which increases answer validation and reduces the risk of model hallucinations. However, these models offer limited information integration (i.e., restricted to the information used for training and fine-tuning) and struggle with questions requiring information aggregation and reasoning. More recently, encoder-decoder transformers and LLMs enabled the integration of information from multiple sources and showed the ability to reason to provide precise answers. Yet, those models are susceptible to hallucinations, require considerable computational power, and ensuring factual accuracy and explainability remains challenging, particularly in domains with very specific vocabulary, as is the case of biomedicine [9].

2.3. Integration of ontologies and knowledge graphs for biomedical QA

Ontologies provide a structured representation of the information, and are widely used within the biomedical field with the role of standardizing the highly specific vocabulary. In addition to the data structure, ontologies also contain a rich context and relations networks that can be used as input for fine-tuning LLMs to expand their knowledge [12].

We direct the reader to the summary provided by Pan et al. [8] for a detailed description of the individual strengths and limitations of LLMs and KGs, and how they are complementary. The main strengths of LLMs are: (i) the general knowledge - with the ability to cover vast and diverse knowledge; (ii) language processing - excelling at understanding and generating human language; and (iii) generalizability - having the capacity to adapt to a wide range of tasks and different domains. On the other hand, KGs provide: (i) structured knowledge - representing information in a structured, relational (and sometimes hierarchical) format; (ii) accuracy - high precision and verifiability based on curated information representations; (iii) contextual understanding; (iv) domain-specific and evolving knowledge - handling specialized vocabulary and allowing incremental updates to integrate new knowledge. Pan et al. conclude that combining both approaches enhances the overall representation and reasoning capabilities of the systems.

Different approaches have been proposed for the integration of LLMs and KGs. Graph-constrained decoding (GCD) [13] has been proposed to enhance LLMs' reasoning by integrating KGs and addressing the hallucination issue. This is done by validating the reasoning paths with the KG's factual structure. This way, the reasoning follows the path most supported by the KGs.

Despite not being well described in the BQA, there is potential to extend its application to this field, given the availability of domain-specific KGs and the improvement of QA systems in other domains. Li et al. [14] proposed an approach that takes advantage of the most relevant triplets from KGs and uses them to provide extra context to the question and refine the provided answer. On the other hand, Tian et al. [15] developed KG-Adapter to address the challenge of knowledge conflict and over-reliance on powerful LLMs, where it leverages information about entities and their relationships in the KGs to generate the final answers.

In addition, KGs have also been shown to improve the information retrieval phase [16].

Given the demonstrated potential of GCD in QA systems and the availability of open-source biomedical KGs such as The Monarch Initiative [17], we integrated this framework into our system.

3. Methodology

An overview of the lasigeBioTM system is shown in Figure 1. In Phase A, for retrieving the candidate documents, we combine BM25 and a re-ranking model.

In the pipeline for Phase A$^+$ and B, we perform Named-Entity Recognition (NER) and Linking (NEL) to extract biomedical entities, which are aligned with a structured knowledge source. These entities and their relations are then used to guide a constrained decoding process using an LLM to provide the final answer.

The source of documents and snippets differs in Phase A$^+$ and Phase B, as in Phase A$^+$, they come from our Phase A retrieval, while in Phase B, they are provided by the challenge organizers.

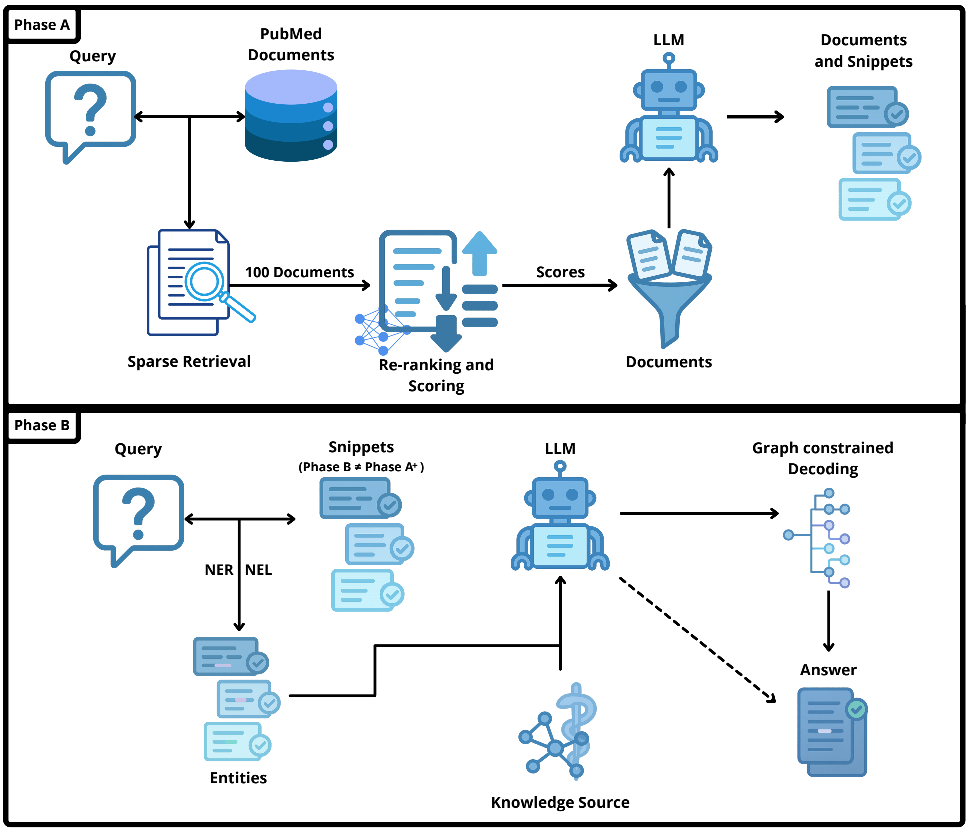

Figure 1: Overview of the BioASQ question answering pipeline. **A** - Pipeline for Phase A. **B** - Pipeline for Phase A$^+$ and B, where in Phase A$^+$ the snippets and documents come from Phase A, while in Phase B they are provided by the challenge organizers.

3.1. Data

3.1.1. BioASQ dataset

The BioASQ dataset is a collection of annotated articles indexed for MEDLINE accessed from PubMed, where the MeSH terms were assigned by PubMed experts. This is a publicly available dataset that was created with the actual information demands of biomedical specialists in mind and continues to expand yearly with each new BioASQ edition [18].

The training dataset for BioASQ Phase B (currently 13B) contains 5389 English questions of four types: yes/no, list, factoid, and summary [19]. For each question the respective gold standard concepts, related articles, snippets, RDF triples, "exact" and "ideal" answers in JSON format are provided, as described in [18].

The test set, similarly to previous years, was composed of 340 questions. The provided documents focused only on article titles and abstracts , divided into four evaluation batches.

3.1.2. External Knowledge - Knowledge Graph and Ontologies

In addition to the datasets provided by the BioASQ organizers, our system leverages external structured knowledge sources to enrich the information available for the question answering. Specifically we utilized the Monarch KG and biomedical ontologies detailed in Table 1.

The Monarch KG is a comprehensive resource that connects genotype, phenotype, disease, and other biomedical information across species. This resource was designed to support translational research and investigate human health and has two main components [17]. The first is the data ingestion and integration pipeline, which aggregates and harmonizes knowledge from 33 heterogeneous sources, including external databases such as the Panther Database [20], and holds the mappings between diseases annotations and standardized terms like Human Phenotype Ontology terms [21]. The second is the knowledge graph representation that captures the relationships among entities. This allows advancing querying, reasoning, and inference across species and data types.

The ontologies where employed for named-entity recognition (NER) and linking (NEL), to expand the Decoding on Graphs approach and to provide more context about the extracted entities.

To perform NER and NEL, we used BENT [22] (available at https://github.com/lasigeBioTM/BENT). BENT is a biomedical entity annotator that also deals with NIL entities (i.e., entities that do not link directly to any knowledge base) by associating them with the top-k relevant concepts within a knowledge base. This tool provides several pre-trained models fine-tuned from PubMeDBERT [23] with the respective knowledge organization system. Information regarding the knowledge bases and ontologies used in our systems is provided in Table 1.

Table 1
Knowledge Bases and Ontologies

Knowledge Base	Entity Type	Entity Example
Human Disease Ontology [24, 25][a]	Disease Entities	Glioblastoma DOID:3068
Chemical Entities of Biological Interest [26][b]	Drugs and Chemicals	Acyclovir CHEBI:2453
NCBI Gene	Genes and Gene Products	CSF1R NCBIGene:108306404
NCBI Taxon [27][c]	Organisms	Influenza virus NCBITaxon:11552
UBERON [28, 29][d]	Anatomical Entities: body parts, organs and tissues	Coronary artery UBERON:0001621
Cellosaurus [30, 31, 32][e]	Cell Line Information	HT-29 cells CVCL:6833
Gene Ontology - Biological Process and Cellular Component [6, 33][f]	Bioprocesses and Cellular Components	Glycolisis GO:0006096

All data accessed on March 19, 2025.
a: http://purl.obolibrary.org/obo/doid.obo; b: https://purl.obolibrary.org/obo/chebi.owl; c: https://purl.obolibrary.org/obo/ncbitaxon.obo; d: http://purl.obolibrary.org/obo/uberon/uberon-full.obo; e: https://ftp.expasy.org/databases/cellosaurus/cellosaurus.obo; f: https://purl.obolibrary.org/obo/go.obo

3.2. Workflow Phase A

To retrieve documents relevant to each question, we developed a two-stage hybrid approach. First, we indexed the articles and performed sparse retrieval to extract a set of candidate documents. Then, we applied an embedding model to score and rerank these candidates based on semantic similarity. Figure 1 A shows the pipeline used in this phase.

3.2.1. Sparse Retrieval

First, we indexed a subset comprising $23,285,041$ articles from the PubMed corpus, retrieved on 13 February 2025, using Elasticsearch 8.17.1 [34]. Then, we performed sparse retrieval based on the BM25 algorithm, with $k_1 = 1.2$ and $b = 0.75$, to obtain an initial set of 100 candidate documents per question. Due to computational resource constraints, we were unable to index the full PubMed collection and instead worked with a representative subset to balance coverage and efficiency.

3.2.2. Dense Retrieval

In the second stage, we employed an embedding model to score and re-rank the candidate documents, capturing semantic relevance beyond lexical overlap.

As a baseline for the first batch, we used the smaller model `gte-reranker-modernbert-base` [2] and selected the top 10 articles, scored directly by applying the model (cross-encoder). For all subsequent batches, we employed the larger and more capable `gte-Qwen2-1.5B-instruct` [3] to improve semantic matching performance, computing the score using the cosine similarity between the embedding vector of a candidate document and that of the corresponding question (bi-encoder).

A threshold of 70 was used for batch 2, while a more lenient threshold of 64.1 was adopted for batches 3 and 4, based on observations from earlier batches where certain questions failed to retrieve any documents under stricter filtering.

3.2.3. Snippet Extraction

For snippet extraction, a prompt was given to Mistral with the questions and the candidate document's title and abstract. In the initial batches (1 and 2) errors in this pipeline consisted of extracting irrelevant passages and not extracting word-for-word the exact copy of the passage of the input document. The model correctly extracted most of the time the beginning and the ending of the snippet, so we applied a boundary-based snippet extraction function that mapped the snippet's start and end positions in the original document for batch 3.

In batch 3, most of the previous errors were resolved, but the model began extracting substrings from larger passages, producing multiple fragmented snippets that were essentially repetitions of the same content, split into different segments, rather than distinct snippets.

In batch 4, the previous errors were minimized, and on this round the most prevalent error was the ones from the initial batches that the model failed to extract the text word-by-word from the original document.

3.3. Workflow Phase B/A$^+$

This section presents the components involved in the answering pipeline, including the graph-constrained decoding mechanism and the systems deployed by our team during the BioASQ13 Task B challenge.

It is worth noting that the same pipeline was used for both Phase B and A$^+$; however, the source of the snippets differed: in Phase B, snippets were provided directly by the BioASQ organizers, whereas in Phase A$^+$, they were extracted by our own retrieval pipeline developed for Phase A.

[2] https://huggingface.co/Alibaba-NLP/gte-reranker-modernbert-base
[3] https://huggingface.co/Alibaba-NLP/gte-Qwen2-1.5B-instruct

The steps used in this pipeline are illustrated in Figure 1 B.

3.3.1. Graph Constrained Decoding

This section outlines the methods used to implement graph-constrained decoding in our question answering pipeline. The implemented approach was based on the work Decoding on Graphs (DoG) from Li et al. [35].

To ensure that the model generates answers aligned with a structured knowledge source, we first perform NER and NEL to identify biomedical entities mentioned in the question. For each linked entity, we retrieve its corresponding subgraph from the knowledge graph, and then merge the subgraphs into a unified representation.

From this merged structure, we construct a prefix-constraining trie that encodes all valid reasoning paths. During answer generation, this trie is used to manipulate the model's output logits, restricting the decoding process to only allow sequences consistent with the graph structure.

3.3.2. Trie Data Structure

A trie is a tree-based data structure that is used to efficiently store and retrieve sequences, such as tokens, by organizing them based on common prefixes. In our case, each node represents a token, and each path from the root to a leaf represents a plausible sequence. This structure allows fast prefix matching and lookup, making it particularly useful for tasks like constrained decoding in language models, where only valid token sequences should be generated.

To enforce constrained decoding during answer generation, we used the trie implementation provided by Facebook Research as part of the GENRE framework [36]. This implementation constructs a prefix tree over valid token sequences, such as entity paths or answer candidates, and integrates with language models to restrict token generation to paths consistent with the trie structure.

This approach ensures that generated outputs remain faithful to a predefined set of valid completions, which is particularly important when reasoning over structured knowledge sources[4].

3.3.3. Decoding

For the decoding mechanism, the model was prompted to use the `<PATH>` and `</PATH>` special tags to mark the start and end of its reasoning path, or use `<T_BOS>` and `<T_EOS>` to delimit the start and end of valid triplets.

Upon entering one of these tags, constraints were enforced by modifying the output logits at each decoding step: the valid tokens are retrieved at the i^{th} iteration of token generation from the trie, and the logits of all other tokens are set to $-\infty$.

After applying the softmax, the altered logits result in a probability distribution that assigns non-zero probability only to the permitted tokens, ensuring that the generated sequence remains consistent with the predefined graph structure.

3.3.4. Systems

This section describes the four systems implemented by our team in Phases A, A^{+} and B, which are summarized in Table 2.

We used `Mistral-7B-Instruct-v0.3`[5] as the baseline LLM model for all our systems. We chose this model based on its balance of performance and resource efficiency, and since it also fits our purpose of achieving a "lean" system.

Due to varying GPU availability during the challenge, the inference was done either on an NVIDIA A30 or on a Tesla T4 GPU. The models were loaded in 4-bit precision using the BitsAndBytes library[6],

[4] Trie implementation utilized https://github.com/facebookresearch/GENRE/blob/main/genre/trie.py

[5] https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.3

[6] https://huggingface.co/docs/bitsandbytes/main/en/index.

with nested double quantization and NF4 (Normal Float 4) as quantization type, and the computation dtype was set to bfloat16.

Our team deployed four systems:

- lasigeBioTM : this system consisted of using the LLM without any additional resources to answer the questions using only the extracted (Phase A and A+) or the provided (Phase B) snippets.
- lasigeBioTM-onto-bl: until batch 3 this system consisted of using questions and snippets with named entities tagged by BENT tool. Starting batch 3, this system consisted of adding additional ontology information, namely definitions and synonyms, to help improve the answers.
- lasigeBioTM-onto-sm: until batch 3 this system consisted of adding additional ontology information, namely definitions and synonyms, to help improve the answers. Starting batch 3, this system consisted
- lasigeBioTM-onto-trie: this system [35]
- sp_lasigebiotm: this system is introduced on batch 3, and applies MarizaTrie

Table 2
Systems Summary

System	LLM	BENT Annotations	Ontology Info	Knowledge Graph
lasigeBioTM	✓	X	X	X
lasigeBioTM-onto-bl (before batch 3)	✓	✓	X	X
lasigeBioTM-onto-bl (from batch 3)	✓	✓	✓	X
lasigeBioTM-onto-sm (before batch 3)	✓	✓	✓	X
lasigeBioTM-onto-sm (from batch 3)	✓	✓	✓	✓
sp_lasigeBioTM (from batch 3)	✓	✓	X	✓

The first system **lasigeBioTM** is our baseline, and involves using the LLM for inference without any additional resources or text pre-processing to answer the questions, by using only the extracted (Phase A+) or provided (Phase B) snippets.

On the second system, **lasigeBioTM-onto-bl**, until the third batch, NER and NEL were applied to the question and the snippets using the BENT tool. Then having the LLM perform the inference by providing the annotated text as input. Starting the third batch, this system was improved by, besides the annotated text, adding additional ontology information such as definitions and synonyms of the annotated entities, to help the LLM perform inference.

For the **lasigeBioTM-onto-sm** system, we applied BENT to extract entities, then ontology definitions and synonyms of those entities were given as an input to the LLM prompt. Starting the third batch, the system was adapted to apply the Decoding on Graphs framework [35]. This methodology explores step-wise reasoning with the structure of knowledge graphs (KG), by forming a well-formed chain, which connect facts grounded in the KG.

Employing graph-aware constrained decoding, this approaches restrains the LLM's output based on the KG's topology. This way the LLM can reason directly on the KG without the need of external retrievers. We incorporated ontologies that matched those used in the named-entity task as additional graph nodes, alongside Monarch. Furthermore, we used the gene2go file[7] provided by the NCBI that maps gene IDs to GO terms, so we could map NCBI Genes to the Gene Ontology graph. For this system as KG's we used ontologies and a more broad KG, Monarch. Sections 3.3.1 and 3.1.2 contain more details about the used methodology and KG, respectively. Additionally, in lasigeBioTM-onto-bl and lasigeBioTM-onto-sm systems, the entities IDs were used to map the entities to the respective ontologies and extract definitions and synonyms information.

The **sp_lasigebiotm** system was first implemented during the third batch and implements the DoG framework (see 3.3.1) without ontologies. The graph constrained mechanism leverages a prefix-lookup data structure (trie) and the manipulation of the model logits to factually ground the LLM on the Knowledge Graph.

[7] Data accessed on April 08, 2025, from https://ftp.ncbi.nih.gov/gene/DATA/gene2go.gz

3.4. Metrics

The evaluation process is submission-based: we provide our system's answers to the BioASQ organizers, who then return the corresponding evaluation metrics for Phases A and B/A$^+$. For Phase A, standard information retrieval metrics are used, including Mean Precision, Recall, F-Measure, Mean Average Precision (MAP), and Geometric Mean Average Precision (GMAP).

While MAP reflects how well relevant documents are ranked near the top, GMAP is a stricter variant that penalizes poor performance on individual queries, offering a measure of robustness across questions. For Exact Answers in Phase B and A$^+$, accuracy is used for yes/no questions, and factoid questions are evaluated using Mean Reciprocal Rank (MRR), strict accuracy, and lenient accuracy.

The Ideal Answers of Phases B and A$^+$ are evaluated based on the ROUGE metric, specifically ROUGE-2 and ROUGE-SU4. These variants measure the overlap of bigrams and skip-bigrams, respectively, between the generated and reference answers—capturing both lexical similarity and fluency. This allows for a qualitative evaluation of the generated text in terms of content relevance and coherence [37].

4. Results

This section presents the results achieved by our systems in all phases. Phase A was divided in two main tasks: first, extracting relevant documents from the designated article repositories and second, extracting relevant snippets from those documents.

Regarding the task of identifying the most relevant documents, our highest score was achieved in batch 1 with an F-Measure of 0.1029 (Table 3). However, in the following batches, this score decreased, dropping to 0.037.

For the snippet extraction, the results are presented in Table 4, with our best F-Measure of 0.0629 achieved in the second batch.

Table 3
Phase A Documents

Batch	System	Mean Precision	Recall	F-Measure	MAP	GMAP
b1	lasigeBioTM	**0.0624**	**0.392**	**0.1029**	**0.3207**	**0.0023**
b2	lasigeBioTM	-	-	-	-	-
b3	lasigeBioTM	0.0597	0.133	0.0716	0.1089	0.0001
b4	lasigeBioTM	0.0248	0.0916	0.037	0.0452	0.0001

Table 4
Phase A Snippets

Batch	System	Mean Precision	Recall	F-Measure	MAP	GMAP
b1	lasigeBioTM	0.0385	0.056	0.0417	**0.1426**	**0.0006**
b2	lasigeBioTM	**0.0657**	**0.0942**	**0.0629**	0.1034	0.0002
b3	lasigeBioTM	0.0527	0.0475	0.0397	0.0518	0
b4	lasigeBioTM	0.0119	0.0251	0.0155	0.0265	0

The results for Phase A$^+$ regarding exact answers are presented in Table 5. For yes/no answer types, our systems demonstrated a better ability to identify "yes" responses. The best performing system in this category was lasigeBioTM, achieving a Macro F1 score of 1 in batch 1 and 0.8301 in batch 4.

With a score of 0.3636 for both the lasigeBioTM and lasigeBioTM-onto-bl systems, batch 4 had the highest mean reciprocal rank in the factoid category. The lasigeBioTM system achieved the highest mean precision and F-Measure for list-type answers in batch 2, with scores of 0.3684 and 0.1598, respectively. In later batches, however, the scores for this type of response declined.

Table 6 provides specific results for ideal responses in Phase A$^+$. Some of the systems do not appear in this table due to an incorrect JSON format submission. Despite a slight decrease in the matching F1 scores, these results show an increase in R-2 and R-SU4 recall for the lasigeBioTM system from batch 1 to batch 4.

Table 5
Phase A^+ Exact Answers

		Yes/No				Factoid			List		
Batch	System	Acc.	F1 Yes	F1 No	Macro F1	Strict Acc.	Lenient Acc.	MRR	Mean Prec.	Recall	F measure
b1	lasigeBioTM	**1**	**1**	**1**	**1**	0.1154	0.1154	0.1154	0.1138	0.1362	0.1203
b2	lasigeBioTM	0.4706	0.5263	0.4	0.4632	0.2593	0.2593	0.2593	**0.3684**	0.1058	**0.1598**
b3	lasigeBioTM-onto-sm	0.5909	0.6667	0.4706	0.5686	-	-	-	0.0455	0.0227	0.0303
b3	lasigeBioTM-onto-bl	0.5909	0.6667	0.4706	0.5686	0.1	0.1	0.1	0.0909	0.0341	0.0485
b3	lasigeBioTM	0.4091	0.381	0.4348	0.4079	0.05	0.05	0.05	-	-	-
b3	sp_lasigebiotm	0.2273	-	0.3704	0.1852	-	-	-	0.0455	0.0455	0.0455
b4	lasigeBioTM	0.8462	0.8824	0.7778	0.8301	**0.3636**	**0.3636**	**0.3636**	0.1482	**0.1396**	0.1363
b4	sp_lasigebiotm	0.8077	0.8485	0.7368	0.7927	0.3182	0.3182	0.3182	0.1947	0.1076	0.1347
b4	lasigeBioTM-onto-bl	0.8077	0.8571	0.7059	0.7815	**0.3636**	**0.3636**	**0.3636**	0.1518	0.1265	0.131
b4	lasigeBioTM-onto-sm	0.7692	0.8125	0.7	0.7563	0.2273	0.2273	0.2273	0.1228	0.0782	0.0892

Table 6
Phase A^+ Ideal Answers Automatic scores (Rouge - R)

Batch	System	R-2 (Rec)	R-2 (F1)	R-SU4 (Rec)	R-SU4 (F1)
b1	lasigeBioTM	0.1712	0.103	0.185	0.1082
b2	lasigeBioTM	0.1573	**0.1311**	0.1574	**0.1261**
b3	sp_lasigebiotm	0.0847	0.0726	0.0912	0.0722
b4	lasigeBioTM	**0.1935**	0.0867	**0.2249**	0.0988
b4	lasigeBioTM-onto-bl	0.1761	0.0848	0.2185	0.1005
b4	sp_lasigebiotm	0.1565	0.087	0.1861	0.1019
b4	lasigeBioTM-onto-sm	0.1041	0.0808	0.1141	0.0853

The results for Phase B, exact answers, are summarized in Table 7. For yes/no answer types, the highest score was a Macro F1 of 0.9328, achieved by the lasigeBioTM system in batch 1. In the factoid category, the sp_lasigebiotm system attained the best mean reciprocal rank of 0.5 in batch 4. For list-type answers, the top F-measure was 0.5592, also obtained by lasigeBioTM in batch 4.

In batch 3, all systems showed a decrease in precision but an increase in recall for list-type answers. Regarding ideal answers, the best R-2 and R-SU4 recall scores were 0.4309 and 0.4357, respectively, both achieved by the lasigeBioTM system in batch 3. The corresponding F1 scores for lasigeBioTM in batch 2 were 0.2749 and 0.2560, respectively.

These results are detailed in Table 8.

Table 7
Phase B Exact Answers

		Yes/No				Factoid			List		
Batch	System	Acc.	F1 Yes	F1 No	Macro F1	Strict Acc.	Lenient Acc.	MRR	Mean Prec.	Recall	F measure
b1	lasigeBioTM	**0.9412**	**0.9565**	**0.9091**	**0.9328**	0.1154	0.1154	0.1154	-	-	-
b2	lasigeBioTM	0.8235	0.8696	0.7273	0.7984	0.4815	0.4815	0.4815	**0.6842**	0.2308	0.3329
b2	lasigeBioTM-onto-sm	0.6471	0.6667	0.6250	0.6458	0.0741	0.1111	0.0926	0.0421	0.0301	0.0351
b2	lasigeBioTM-onto-bl	0.5294	0.5556	0.5000	0.5278	0.0741	0.1481	0.1111	0.0992	0.1266	0.1047
b3	lasigeBioTM-onto-bl	0.9091	0.9412	0.8000	0.8706	0.1000	0.1000	0.1000	0.5314	0.4916	0.5009
b3	sp_lasigebiotm	0.8636	0.9091	0.7273	0.8182	0.1500	0.1500	0.1500	0.5576	0.5149	0.5153
b3	lasigeBioTM	0.7727	0.8387	0.6154	0.7270	0.3000	0.3000	0.3000	0.5343	0.5196	0.5144
b3	lasigeBioTM-onto-sm	0.5000	0.5600	0.4211	0.4905	-	-	-	-	-	-
b4	sp_lasigebiotm	0.8462	0.8824	0.7778	0.8301	**0.5000**	**0.5000**	**0.5000**	0.4698	0.2960	0.3364
b4	lasigeBioTM	0.8077	0.8485	0.7368	0.7927	0.4091	0.4091	0.4091	0.4183	0.4092	0.3979
b4	lasigeBioTM-onto-bl	0.8077	0.8571	0.7059	0.7815	0.3636	0.4091	0.3864	0.6163	**0.5309**	**0.5592**
b4	lasigeBioTM-onto-sm	0.7308	0.7742	0.6667	0.7204	0.0455	0.1364	0.0833	0.3113	0.1887	0.2306

Table 8
Phase B Ideal Answers Automatic scores (Rouge - R)

Batch	System	R-2 (Rec)	R-2 (F1)	R-SU4 (Rec)	R-SU4 (F1)
b1	lasigeBioTM	0.1731	0.1859	0.1647	0.1768
b2	lasigeBioTM	0.3069	**0.2749**	0.2982	**0.2560**
b2	lasigeBioTM-onto-bl	0.1796	0.0819	0.2027	0.0880
b2	lasigeBioTM-onto-sm	0.1773	0.0885	0.1956	0.0945
b3	lasigeBioTM	**0.4309**	0.2236	**0.4357**	0.2113
b3	lasigeBioTM-onto-bl	0.3390	0.1846	0.3441	0.1783
b3	sp_lasigebiotm	0.3286	0.2142	0.3291	0.2031
b3	lasigeBioTM-onto-sm	0.0878	0.0754	0.1033	0.0829
b4	lasigeBioTM	0.4139	0.2373	0.3963	0.2187
b4	sp_lasigebiotm	0.3681	0.2285	0.3554	0.2120
b4	lasigeBioTM-onto-bl	0.3681	0.2158	0.3760	0.2091
b4	lasigeBioTM-onto-sm	0.1192	0.1105	0.1154	0.1078

5. Discussion and Conclusion

Our team participated in all batches for Phases A, A$^+$ and B and although some of the systems were introduced in later batches, identified errors between each batch were systematically addressed.

In the task for the most relevant articles in **Phase A**, was observed a decline in scores starting at batch 3. This decline may be attributed to switching from the smaller Cross Encoder reranking model to the LLM-based embedding model. The difference in effectiveness can be explained by the underlying mechanisms of each approach: cross-encoders output a similarity score by jointly encoding the query and document pairs, capturing deeper semantic interactions across tokens, in contrast to embedding-based methods that rely on comparing separate vector representations. Additionally, a threshold for document selection that was not flexible enough, potentially excluding relevant candidates, may have contributed to the decline.

A decrease in snippet performance was also noted starting from batch 3. Although the majority of errors identified in previous batches were corrected, the reduction in snippet scores is unlikely to be directly related to the snippets themselves. Instead, it is likely due to the extraction of irrelevant documents, which subsequently causes irrelevant snippets to be extracted.

Given that **Phase A$^+$** relied on the documents and snippets extracted during Phase A, the system's performance in Phase A$^+$ was correspondingly impacted by the quality of snippets, with degraded snippet quality resulting in reduced performance. Nonetheless, errors may occur independently of snippet quality, such as the errors mentioned at 3.2.3. There was a decrease in performance of our baseline system, lasigeBioTM, in batch 3, and the introduced sp_lasigebiotm model also exhibited suboptimal performance.

Notably, in batch 4, our models achieved the team's best scores across all question types. lasigeBioTM achieved the team's best scores for all the exact answers in different categories. However, in batch 4, lasigeBioTM-onto-bl performed equally well as lasigeBioTM for factoid answers.

In **Phase B** "Ideal Answers", in batches 3 and 4, lasigeBioTM achieved the highest automatic scores Rouge-R for R-2 (Rec) and R-SU4 (Rec). These results indicate that our system is capable of capturing relevant content and key points. However, comparatively low F1 scores indicate generation of extra or irrelevant information, reducing overall answer quality.

Performance across exact and ideal answer types was more heterogeneous. Specifically, lasigeBioTM dominated yes/no answers in the first three batches; however, in batch 4, sp_lasigebiotm outperformed lasigeBioTM by 0.0374 in macro F1.

A similar pattern was observed for factoid lists, with sp_lasigebiotm outperforming lasigeBioTM in batch 4. Across list-type questions, the top-performing models were lasigeBioTM-onto-sm (batch 2), sp_lasigebiotm (batch 3), and lasigeBioTM-onto-bl (batch 4). The results indicate that systems augmented with external knowledge succeed in list-type question answering.

Comparison of the models across Phase A^+ and Phase B, Tables 10 and 9, shows that improved snippet quality impacts system performance, resulting in higher scores on Phase B answers.

The lasigeBioTM-onto-sm system lags behind the other systems in the final batch, despite updates applied to all systems to address prior issues. This system employs ontologies to enforce a step-wise reasoning approach in the answering chain. The use of the OBO format, which only includes "is_a" relationship, likely constrained the reasoning paths, leading to bottlenecks. Moreover, this system occasionally generated reasoning paths rather than direct answers (e.g., "blinatumomab -> targets -> B-cell acute lymphoblastic leukemia").

Furthermore, the reliance on KG paths without sufficient filtering for relevance may have introduced noise, limiting answer quality, despite the structured knowledge integration. This highlights the inherent challenge in aligning structured KG reasoning paths with the generative nature of LLMs, requiring careful design to prevent hallucination or irrelevant path outputs.

Overall, our systems demonstrate better performance on ideal answers compared to the more structured exact answers. Moreover, when provided with high-quality snippets, as in Phase B, performance improves, particularly benefiting from external knowledge in generating list-type answers. Despite using a lightweight approach, the systems consistently delivered accurate outputs across various challenge batches, highlighting their prominent performance and reliability, even with limited resources.

One key takeaway from this work is the importance of carefully balancing the injection of structured knowledge with generative LLM capabilities. Although the incorporation of external knowledge provides explicit and verified context, its effectiveness depends on the quality of the entity recognition and linking, and the relevance of retrieved paths. It was observed that limited relation types, such as only "is_a" edges in OBO ontologies, may induce reasoning bottlenecks, thereby leading the model down uninformative and noisy paths and reducing final answer quality.

Another significant finding concerns the pipeline itself, as our results showed that errors in upstream document retrieval or snippet extraction propagate downstream and degrade system performance. Therefore, future iterations should prioritize designing pipelines where modules are loosely coupled but effectively informed by each other. Additionaly, external knowledge can introduce noise if not filtered appropriately or formatted to align with LLM expectations. Our KG-constrained models, despite their structured reasoning design, occasionally returned raw paths rather than direct answers. This emphasizes the need for improved prompt engineering, decoding constraints, and integration strategies to ensure that KG information enhances rather than overrides natural language generation capabilities. Finally, this work demonstrates that lean, efficient architectures can achieve competitive results when combined with targeted external knowledge and careful prompt design, suggesting promising directions for deploying lightweight, guided QA systems in resource-constrained environments.

Future work will focus on refining knowledge integration techniques, developing better error mitigation strategies, and continue exploring models that combine structured reasoning with fluent, accurate generation to support robust biomedical question answering.
Specifically, improvements will be made to the document selection process by implementing a more adaptive similarity threshold, informed by statistical measures such as the median or interquartile range. Additionally, we intend to investigate model performance without quantization and to incorporate beam search techniques to enhance generation quality.
Further directions include experimenting with alternative models and knowledge sources, testing different prompt designs, and investigating the sources of noise introduced by external knowledge in order to mitigate their impact on the final output.

Building on these experimental insights, we aim to leverage our system's "lean" design to operate efficiently in resource-constrained healthcare settings. This approach aims to support informed decision-making and enable secure, locally deployed models customized to specific needs, while ensuring sensitive patient data.

Acknowledgments

This work was supported by FCT (Fundacão para a Ciência e a Tecnologia) through funding of the PhD Scholarships with ref. UI/BD/153730/2022 and DOI identifier https://doi.org/10.54499/UI/BD/153730/2022 and DOI identifier https://doi.org/10.54499/UI/BD/153730/2022 attributed to SIRC, and the LASIGE Research Unit, ref. UID/00408/2025, and by a grant from the Novo Nordisk Foundation (NNF23SA0084103) through The CBMR International PhD & Postdoc Program.

Declaration on Generative AI

During the preparation of this work, the authors used GPT-4 and LanguageTool in order to: Grammar, spelling check and paraphrase. Further, the authors used GPT-4 for Figure 1 in order to: Generate icons. After using these tools, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] A. Nentidis, G. Katsimpras, A. Krithara, M. Krallinger, M. R. Ortega, N. Loukachevitch, A. Sakhovskiy, E. Tutubalina, G. Tsoumakas, G. Giannakoulas, A. Bekiaridou, A. Samaras, G. M. Di Nunzio, N. Ferro, S. Marchesin, L. Menotti, G. Silvello, G. Paliouras, BioASQ at CLEF2025: The thirteenth edition of the large-scale biomedical semantic indexing and question answering challenge, in: Advances in Information Retrieval, Springer Nature Switzerland, Springer Nature Switzerland, Cham, 2025. doi:10.1007/978-3-031-88720-8_61.

[2] 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, F. N. Maria Di Nunzio, Giorgio, 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: L. P. A. G. S. d. H. J. M. F. P. P. R. D. S. G. F. N. F. Jorge Carrillo-de Albornoz, Julio Gonzalo (Ed.), Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Sixteenth International Conference of the CLEF Association (CLEF 2025), 2025.

[3] A. Nentidis, G. Katsimpras, A. Krithara, G. Paliouras, Overview of BioASQ Tasks 13b and Synergy13 in CLEF2025, in: G. Faggioli, N. Ferro, P. Rosso, D. Spina (Eds.), CLEF 2025 Working Notes, 2025.

[4] A. Rawat, S. Singh Samant, Comparative analysis of transformer based models for question answering, in: 2022 2nd International Conference on Innovative Sustainable Computational Technologies (CISCT), 2022, pp. 1–6. doi:10.1109/CISCT55310.2022.10046525.

[5] A. Lamurias, F. M. Couto, LasigeBioTM at MEDIQA 2019: Biomedical question answering using bidirectional transformers and named entity recognition, in: D. Demner-Fushman, K. B. Cohen, S. Ananiadou, J. Tsujii (Eds.), Proceedings of the 18th BioNLP Workshop and Shared Task, Association for Computational Linguistics, Florence, Italy, 2019, pp. 523–527. doi:10.18653/v1/W19-5057.

[6] M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, et al., Gene ontology: tool for the unification of biology, Nature genetics 25 (2000) 25–29.

[7] Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, Y. Dai, J. Sun, M. Wang, H. Wang, Retrieval-augmented generation for large language models: A survey, 2024. arXiv:2312.10997.

[8] S. Pan, L. Luo, Y. Wang, C. Chen, J. Wang, X. Wu, Unifying large language models and knowledge graphs: A roadmap, IEEE Transactions on Knowledge and Data Engineering 36 (2024) 3580–3599. doi:10.1109/tkde.2024.3352100.

[9] Q. Jin, Z. Yuan, G. Xiong, Q. Yu, H. Ying, C. Tan, M. Chen, S. Huang, X. Liu, S. Yu, Biomedical

question answering: A survey of approaches and challenges, ACM Comput. Surv. 55 (2022). doi:10.1145/3490238.

[10] S. Robertson, H. Zaragoza, The probabilistic relevance framework: Bm25 and beyond, Foundations and Trends in Information Retrieval 3 (2009) 333–389. doi:10.1561/1500000019.

[11] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, J. Kang, Biobert: a pre-trained biomedical language representation model for biomedical text mining, Bioinformatics 36 (2020) 1234–1240. URL: https://doi.org/10.1093/bioinformatics/btz682. doi:10.1093/bioinformatics/btz682.

[12] X. Yao, A. Sannabhadti, H. Wiberg, K. S. Shehadeh, R. Padman, Can llms support medical knowledge imputation? an evaluation-based perspective, 2025. arXiv:2503.22954.

[13] L. Luo, Z. Zhao, C. Gong, G. Haffari, S. Pan, Graph-constrained reasoning: Faithful reasoning on knowledge graphs with large language models, 2024. arXiv:2410.13080.

[14] S. Li, Y. Gao, H. Jiang, Q. Yin, Z. Li, X. Yan, C. Zhang, B. Yin, Graph reasoning for question answering with triplet retrieval, 2023. arXiv:2305.18742.

[15] S. Tian, Y. Luo, T. Xu, C. Yuan, H. Jiang, C. Wei, X. Wang, KG-adapter: Enabling knowledge graph integration in large language models through parameter-efficient fine-tuning, in: L.-W. Ku, A. Martins, V. Srikumar (Eds.), Findings of the Association for Computational Linguistics: ACL 2024, Association for Computational Linguistics, Bangkok, Thailand, 2024, pp. 3813–3828. doi:10.18653/v1/2024.findings-acl.229.

[16] K. Soman, P. W. Rose, J. H. Morris, R. E. Akbas, B. Smith, B. Peetoom, C. Villouta-Reyes, G. Cerono, Y. Shi, A. Rizk-Jackson, S. Israni, C. A. Nelson, S. Huang, S. E. Baranzini, Biomedical knowledge graph-optimized prompt generation for large language models, Bioinformatics 40 (2024) btae560. URL: https://doi.org/10.1093/bioinformatics/btae560. doi:10.1093/bioinformatics/btae560.

[17] T. E. Putman, K. Schaper, N. Matentzoglu, V. Rubinetti, F. Alquaddoomi, C. Cox, J. H. Caufield, G. Elsarboukh, S. Gehrke, H. Hegde, J. Reese, I. Braun, R. Bruskiewich, L. Cappelletti, S. Carbon, A. Caron, L. Chan, C. Chute, K. Cortes, V. De Souza, T. Fontana, N. Harris, E. Hartley, E. Hurwitz, J. B. Jacobsen, M. Krishnamurthy, B. Laraway, J. McLaughlin, J. McMurry, S. T. Moxon, K. Mullen, S. O'Neil, K. Shefchek, R. Stefancsik, S. Toro, N. Vasilevsky, R. Walls, P. Whetzel, D. Osumi-Sutherland, D. Smedley, P. Robinson, C. Mungall, M. Haendel, M. Munoz-Torres, The monarch initiative in 2024: an analytic platform integrating phenotypes, genes and diseases across species, Nucleic Acids Research 52 (2023) D938–D949. doi:10.1093/nar/gkad1082.

[18] A. Krithara, A. Nentidis, K. Bougiatiotis, G. Paliouras, BioASQ-QA: A manually curated corpus for biomedical question answering, Scientific Data 10 (2023) 170. URL: https://doi.org/10.1038/s41597-023-02068-4. doi:10.1038/s41597-023-02068-4.

[19] G. Tsatsaronis, G. Balikas, P. Malakasiotis, I. Partalas, M. Zschunke, M. R. Alvers, D. Weissenborn, A. Krithara, S. Petridis, D. Polychronopoulos, et al., An overview of the bioasq large-scale biomedical semantic indexing and question answering competition, BMC bioinformatics 16 (2015) 1–28.

[20] P. D. Thomas, D. Ebert, A. Muruganujan, T. Mushayahama, L.-P. Albou, H. Mi, Panther: Making genome-scale phylogenetics accessible to all, Protein Science 31 (2022) 8–22. doi:https://doi.org/10.1002/pro.4218.

[21] S. Köhler, M. Gargano, N. Matentzoglu, L. C. Carmody, D. Lewis-Smith, N. A. Vasilevsky, D. Danis, G. Balagura, G. Baynam, A. M. Brower, T. J. Callahan, C. G. Chute, J. L. Est, P. D. Galer, S. Ganesan, M. Griese, M. Haimel, J. Pazmandi, M. Hanauer, N. L. Harris, M. Hartnett, M. Hastreiter, F. Hauck, Y. He, T. Jeske, H. Kearney, G. Kindle, C. Klein, K. Knoflach, R. Krause, D. Lagorce, J. A. McMurry, J. A. Miller, M. Munoz-Torres, R. L. Peters, C. K. Rapp, A. M. Rath, S. A. Rind, A. Rosenberg, M. M. Segal, M. G. Seidel, D. Smedley, T. Talmy, Y. Thomas, S. A. Wiafe, J. Xian, Z. Yüksel, I. Helbig, C. J. Mungall, M. A. Haendel, P. N. Robinson, The human phenotype ontology in 2021, Nucleic Acids Research 49 (2020) D1207–D1217. doi:10.1093/nar/gkaa1043.

[22] P. Ruas, F. M. Couto, Nilinker: Attention-based approach to nil entity linking, Journal of Biomedical Informatics 132 (2022) 104137. doi:https://doi.org/10.1016/j.jbi.2022.104137.

[23] Y. Gu, R. Tinn, H. Cheng, M. Lucas, N. Usuyama, X. Liu, T. Naumann, J. Gao, H. Poon, Domain-specific language model pretraining for biomedical natural language processing, 2020.

arXiv:arXiv:2007.15779.

[24] W. A. Kibbe, C. Arze, V. Felix, E. Mitraka, E. Bolton, G. Fu, C. J. Mungall, J. X. Binder, J. Malone, D. Vasant, H. Parkinson, L. M. Schriml, Disease ontology 2015 update: an expanded and updated database of human diseases for linking biomedical knowledge through disease data, Nucleic Acids Research 43 (2014) D1071–D1078. URL: https://doi.org/10.1093/nar/gku1011. doi:10.1093/nar/gku1011.

[25] L. M. Schriml, J. B. Munro, M. Schor, D. Olley, C. McCracken, V. Felix, J. Baron, R. Jackson, S. Bello, C. Bearer, R. Lichenstein, K. Bisordi, N. C. Dialo, M. Giglio, C. Greene, The human disease ontology 2022 update, Nucleic Acids Research 50 (2021) D1255–D1261. doi:10.1093/nar/gkab1063.

[26] J. Hastings, G. Owen, A. Dekker, M. Ennis, N. Kale, V. Muthukrishnan, S. Turner, N. Swainston, P. Mendes, C. Steinbeck, Chebi in 2016: Improved services and an expanding collection of metabolites, Nucleic Acids Research 44 (2015) D1214–D1219. doi:10.1093/nar/gkv1031.

[27] S. Federhen, The ncbi taxonomy database, Nucleic Acids Research 40 (2011) D136–D143. doi:10.1093/nar/gkr1178.

[28] C. J. Mungall, C. Torniai, G. V. Gkoutos, S. E. Lewis, M. A. Haendel, Uberon, an integrative multi-species anatomy ontology, Genome biology 13 (2012) 1–20.

[29] M. A. Haendel, J. P. Balhoff, F. B. Bastian, D. C. Blackburn, J. A. Blake, Y. Bradford, A. Comte, W. M. Dahdul, T. A. Dececchi, R. E. Druzinsky, et al., Unification of multi-species vertebrate anatomy ontologies for comparative biology in uberon, Journal of biomedical semantics 5 (2014) 1–13.

[30] T. Robin, A. Capes-Davis, A. Bairoch, Clastr: the cellosaurus str similarity search tool-a precious help for cell line authentication, International Journal of Cancer 146 (2020) 1299–1306.

[31] A. Bairoch, The cellosaurus, a cell-line knowledge resource, Journal of biomolecular techniques: JBT 29 (2018) 25.

[32] A. Bairoch, Cellosaurus micro-review 1: cellonauts, spacefaring cell lines (2019).

[33] T. G. O. Consortium, The gene ontology resource: enriching a gold mine, Nucleic Acids Research 49 (2020) D325–D334. URL: https://doi.org/10.1093/nar/gkaa1113. doi:10.1093/nar/gkaa1113.

[34] E. NV, Elasticsearch version 8.17.1, 2025. URL: https://www.elastic.co/guide/en/elasticsearch/reference/8.17/release-notes-8.17.1.html, accessed: 2025-05-28.

[35] K. Li, T. Zhang, X. Wu, H. Luo, J. Glass, H. Meng, Decoding on graphs: Faithful and sound reasoning on knowledge graphs through generation of well-formed chains, 2024. arXiv:2410.18415.

[36] N. D. Cao, G. Izacard, S. Riedel, F. Petroni, Autoregressive entity retrieval, CoRR abs/2010.00904 (2020). URL: https://arxiv.org/abs/2010.00904. arXiv:2010.00904.

[37] G. Balikas, I. Partalas, A. Kosmopoulos, S. Petridis, P. Malakasiotis, I. Pavlopoulos, I. Androutsopoulos, N. Baskiotis, E. Gaussier, T. Artieres, P. Gallinari, Evaluation Framework Specifications, Project deliverable D4.1, 2013. URL: sites/default/files/PublicDocuments/BioASQ_D4.1-EvaluationFrameworkSpecification_final.pdf.

A. Prompts

In this section, we present the prompts used for lasigeBioTM, lasigeBioTM-onto-sm, and lasigeBioTM-onto-bl systems submitted in different batches of Phase B.

lasigeBioTM

You are an expert biomedical question answering helper. Given a question and some snippets extracted from related documents, you answer the question in two ways. Snippets are small key text fragments that provide context for answering the question.

{question type specific prompt}

The format MUST ALWAYS be a short Exact answer and in the next line a longer Ideal Answer as such:

Exact Answer: ...

Ideal Answer: ...

Based on the following question and the documents, extract AT MOST 10 snippets for answering it without additional information.

If the snippet or question contains any entity tag <e1>entity_name<e1> ignore it and just use the entity name if needed.

Question: {question}

Entities: {snippets}

lasigeBioTM-onto-sm

You are a helpful assistant that can analyse the knowledge graphs in the contexts and then answer the questions based on the knowledge graphs.
The answers should give the grounded reasoning chains and think step by step, and the reasoning chains should be logically complete but have as fewer steps as possible. Do not include information irrelvant to the question.

Guidelines:
- Provide two distinct answers on separate lines
{question type specific prompt} - The answer format MUST always be:
Exact Answer: ...
Ideal Answer: ...

Do NOT:
- Include any additional text, commentary, or reasoning in the Exact Answer beyond the required answer.
- Reference how you derived the answer or mention the context (snippets, ontology info) in the Exact Answer line.

Based on the following question and entities, perform the two tasks described but you will only output the last task:

1. Extract and output unique subject-predicate-object triplets in the format:
<T_BOS> entity1 -> relation -> entity2 <T_EOS>
Use these triples to think about a possible answer.

2. Output two answers on separate lines:
Exact Answer: ...
Ideal Answer: ...
Question: {question}
Entities: {entities}
Guidelines:
- Do not include explanations, headers, or additional comments beyond the two answers.
- Do not include information irrelevant to the question.

lasigeBioTM-onto-bl

You are an expert biomedical question answering helper. Your response must strictly adhere to the specific instructions provided under q_type. Follow these rules exactly:

1. **Response Format:**
Your response MUST consist of EXACTLY two lines:
- Line 1: "Exact Answer:" followed by the answer in the required format.
- Line 2: "Ideal Answer:" followed by a detailed answer (maximum 200 words).

2. **Exact Answer Requirements by q_type:**
- For yes/no questions: The Exact Answer MUST be either "yes" or "no" (only that word, no extra text).
- For factoid questions: The Exact Answer MUST be a comma-separated list of up to 5 entity names (or similar short expressions), sorted by decreasing confidence. Do not add any extra words or context.
- For list questions: The Exact Answer MUST be a comma-separated list of entity names (or similar short expressions) that represent a unified answer. Do not include more than the allowed number of entries.
- For summary questions: There is no Exact Answer; leave it empty.

3. **Ideal Answer Requirements:**
The Ideal Answer SHOULD be a single paragraph (up to 200 words) that provides a detailed explanation or summary. It MUST NOT include any meta commentary about your process or reference to the context (e.g., do not say "the snippets provided…").

4. **Do NOT:**
- Include any additional text, commentary, or reasoning in the Exact Answer beyond the required answer.
- Reference how you derived the answer or mention the context (snippets, ontology info) in the Exact Answer line.

Question: {q_body}
Context: {snippets}
Ontology info: {entity_context}
Additional Instructions for this type: {q_type}

B. Results Phase A$^+$

Tables 9 and 10 summarize the comparison between phases A$^+$ and B for the ideal and exact answers, respectively.

Table 9
Phase A$^+$ and B Ideal Answers Automatic scores (Rouge - R) Comparison

Batch	System	R-2 (Rec)	R-2 (F1)	R-SU4 (Rec)	R-SU4 (F1)
Ab1	lasigeBioTM	0.1712	0.103	0.185	0.1082
Bb1		0.1731	0.1859	0.1647	0.1768
Ab2	lasigeBioTM	0.1573	0.1311	0.1574	0.1261
Bb2		0.3069	0.2749	0.2982	0.2560
Ab3	sp_lasigebiotm	0.0847	0.0726	0.0912	0.0722
Bb3		0.3286	0.2142	0.3291	0.2031
Ab4	lasigeBioTM	0.1935	0.0867	0.2249	0.0988
Bb4		0.4139	0.2373	0.3963	0.2187
Ab4	lasigeBioTM-onto-bl	0.1761	0.0848	0.2185	0.1005
Bb4		0.3681	0.2158	0.3760	0.2091
Ab4	sp_lasigebiotm	0.1565	0.087	0.1861	0.1019
Bb4		0.3681	0.2285	0.3554	0.2120
Ab4	lasigeBioTM-onto-sm	0.1041	0.0808	0.1141	0.0853
Bb4		0.1192	0.1105	0.1154	0.1078

Table 10
Phase A$^+$ and B Exact Answers Comparison

		Yes/No				Factoid			List		
Batch	System	Acc.	F1 Yes	F1 No	Macro	Strict	Lenient	MRR	Mean	Recall	F
					F1	Acc.	Acc.		Prec.		measure
Ab1	lasigeBioTM	1	1	1	1	0.1154	0.1154	0.1154	0.1138	0.1362	0.1203
Bb1		0.9412	0.9565	0.9091	0.9328	0.1154	0.1154	0.1154	-	-	-
Ab2	lasigeBioTM	0.4706	0.5263	0.4	0.4632	0.2593	0.2593	0.2593	0.3684	0.1058	0.1598
Bb2		0.8235	0.8696	0.7273	0.7984	0.4815	0.4815	0.4815	0.6842	0.2308	0.3329
Ab3	sp_lasigebiotm	0.2273	-	0.3704	0.1852	-	-	-	0.0455	0.0455	0.0455
Bb3		0.8636	0.9091	0.7273	0.8182	0.1500	0.1500	0.1500	0.5576	0.5149	0.5153
Ab3	lasigeBioTM-onto-sm	0.5909	0.6667	0.4706	0.5686	-	-	-	0.0455	0.0227	0.0303
Bb3		0.5000	0.5600	0.4211	0.4905	-	-	-	-	-	-
Ab3	lasigeBioTM-onto-bl	0.5909	0.6667	0.4706	0.5686	0.1	0.1	0.1	0.0909	0.0341	0.0485
Bb3		0.9091	0.9412	0.8000	0.8706	0.1000	0.1000	0.1000	0.5314	0.4916	0.5009
Ab3	lasigeBioTM	0.4091	0.381	0.4348	0.4079	0.05	0.05	0.05	-	-	-
Bb3		0.7727	0.8387	0.6154	0.7270	0.3000	0.3000	0.3000	0.5343	0.5196	0.5144
Ab4	lasigeBioTM	0.8462	0.8824	0.7778	0.8301	0.3636	0.3636	0.3636	0.1482	0.1396	0.1363
Bb4		0.8077	0.8485	0.7368	0.7927	0.4091	0.4091	0.4091	0.4183	0.4092	0.3979
Ab4	sp_lasigebiotm	0.8077	0.8485	0.7368	0.7927	0.3182	0.3182	0.3182	0.1947	0.1076	0.1347
Bb4		0.8462	0.8824	0.7778	0.8301	0.5000	0.5000	0.5000	0.4698	0.2960	0.3364
Ab4	lasigeBioTM-onto-bl	0.8077	0.8571	0.7059	0.7815	0.3636	0.3636	0.3636	0.1518	0.1265	0.131
Bb4		0.8077	0.8571	0.7059	0.7815	0.3636	0.4091	0.3864	0.6163	0.5309	0.5592
Ab4	lasigeBioTM-onto-sm	0.7692	0.8125	0.7	0.7563	0.2273	0.2273	0.2273	0.1228	0.0782	0.0892
Bb4		0.7308	0.7742	0.6667	0.7204	0.0455	0.1364	0.0833	0.3113	0.1887	0.2306