Health-NLP at the BioASQ 2026 Task B on Biomedical Semantic QA and GutBrainIE on Gut-Brain Interplay Information Extraction Tasks

Sebastian Volz[1,†], Bileam Scheuvens[1,†], Shrestha Ghosh[1] and Harrisen Scells[1,*]

[1]University of Tübingen, Tübingen, Germany

Abstract

These working notes present the results of the submissions from team Health-NLP at the BioASQ 2026 challenge for the BioASQ Task B on Biomedical Semantic QA (Task B) and BioASQ Task GutBrainIE on Gut-Brain interplay Information Extraction (GutBrainIE) tasks. Question-answering and information extraction are two key tasks in biomedical information access. Our approach to tackling these tasks as part of BioASQ was to investigate recent state-of-the-art approaches that have seen little or no applicability to these tasks to understand how well they generalise. For Task B, we investigate a recent LLM-based Boolean query formulation approach and a recent approach to question-answering in the form of retrieval-augmented generation that uses document parameterisations instead of document text to represent answer contexts. For GutBrainIE, we investigate a recent named entity recognition model and propose a new approach to perform entity disambiguation. In investigating these relatively new approaches for the proposed tasks in BioASQ, we highlight the limitations of these approaches for biomedical tasks, the domain-specific adaptations that are needed to make these approaches more effective and the future work that can be done to further improve these approaches.

Keywords

biomedical information access, parametric retrieval-augmented generation, information extraction

1. Introduction

Information access in bio-medicine requires domain-specific solutions that address problems where a single model is not sufficient. Researchers and clinicians use biomedical information access systems as an interface to address complex tasks, such as question-answering (QA) and to automate data analysis. The BioASQ lab provides the basis for developing and evaluating such tasks. In these working notes, we describe the Health-NLP team's submissions to two tasks at BioASQ 2026 [1]: *BioASQ Task B on Biomedical Semantic QA* (Task B) [2] and *BioASQ Task GutBrainIE on Gut-Brain interplay Information Extraction* (GutBrainIE) [3].

Task B involved a retrieval phase and a question-answering stage. For the retrieval phase, we used the AutoBool model [4] to formulate Boolean queries that we then submitted to PubMed via the Entrez API [5]. We then re-ranked the results using MedCPT cross-encoder [6]. For the question-answering phase, we investigated the viability of parametric retrieval-augmented generation (PRAG) [7] for complex biomedical questions. We found that task-specific warm-ups and combining PRAG with in-context RAG led to the best effectiveness. However, we also highlight several issues with this method which we describe in greater detail in Section 2.

GutBrainIE involved four subtasks: named entity recognition (NER), named entity recognition and disambiguation (NERD), mention-level relation extraction (M-RE), and concept-level relation extraction (C-RE). For the entity and relation extraction tasks, we investigated the GLiNER2 model [8]. For the disambiguation task, we propose a new architecture that we call Graphlinker that embeds entities

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

*Corresponding author.

†Equal Contribution.

sebastian.volz@student.uni-tuebingen.de (S. Volz); bileam.scheuvens@student.uni-tuebingen.de (B. Scheuvens); shrestha.ghosh@uni-tuebingen.de (S. Ghosh); harrisen.scells@uni.tuebingen.de (H. Scells)

0009-0007-9364-8242 (S. Volz); 0009-0006-5632-9421 (B. Scheuvens); 0000-0002-1711-0500 (S. Ghosh); 0000-0001-9578-7157 (H. Scells)

and URIs into a shared vector space. A more detailed description of how we approached these four challenges and our novel entity disambiguation model can be found in Section 3.

The following two sections describe the two tasks we participated in greater detail, providing related work, a deeper description of the methodologies, and an analysis of the results.

2. Biomedical Semantic QA

The volume of biomedical literature is growing faster than researchers can keep up with, making automated information retrieval and question-answering increasingly important. Retrieval-augmented generation (RAG) has become the standard approach: retrieved documents are passed in the input context to help the model answer questions. Despite its wide adoption, in-context RAG has well-known limitations: effectiveness degrades when documents are placed in the middle of long contexts, results are sensitive to document ordering, context length constrains how many documents can be used, and inference cost grows quadratically with context length. Su et al. [7] propose Parametric RAG (PRAG), which injects documents into model weights rather than the input context, to address these issues. We compare PRAG against plain ICL and a combined approach.

Task B consists of two phases:

- **Phase A:** given a natural-language biomedical question, return up to 10 relevant PubMed articles.
- **Phase B:** given the question together with the gold relevant articles and snippets, produce both an exact answer (e.g. a named entity for factoid questions, a list of entities, or yes/no) and an ideal answer (a paragraph-sized summary).

Our code and the configurations to reproduce our submissions are available at https://github.com/health-nlp/bioasq26-14b-autobool and https://github.com/health-nlp/bioasq26-14b-prag.

Related Work Literature retrieval in the biomedical domain has traditionally relied on Boolean queries submitted to PubMed. Training models for Boolean query generation is challenging because existing datasets are small and expert-crafted queries are often inconsistent, making them unreliable as ground truth for supervised fine-tuning. Wang et al. [4] address this by using reinforcement learning (RL) to train an LLM to generate Boolean queries, optimising directly for retrieval measures without requiring target queries. AutoBool outperforms zero-shot and few-shot prompting and matches or exceeds much larger models such as GPT-4o. For reranking, we use MedCPT [6].

For question-answering, Su et al. [7] propose PRAG, which converts each retrieved document into a LoRA adapter through a document augmentation and fine-tuning process, then merges the adapters directly into the model weights at inference time. This approach aims to avoid the context-length overhead of passing documents in the input and to integrate knowledge more directly into the model's parameters. Qi et al. [9] show that combining parametric and in-context injection outperforms either method alone.

2.1. Methodology

This section describes our approach to Phase A and Phase B. For Phase A we use a two-stage pipeline: Boolean query generation for initial retrieval followed by document reranking. For Phase B we train LoRA adapters for each gold snippet provided by BioASQ following the PRAG framework, and evaluate it in three modes: parametric injection only (PRAG), in-context injection only (ICL), and a combination of both (Combine).

2.1.1. Phase A: Boolean Query Generation and Reranking

Boolean Query Generation We first generate a PubMed Boolean query from the natural-language question using AutoBool [4]. AutoBool is based on Qwen 4B. We chose the no-reasoning variant because it obtains the highest recall, which is essential in first-stage retrieval, where we are more concerned

with not missing relevant documents. We used the same system prompt as in the original paper. The Boolean query is executed on PubMed's eSearch endpoint to obtain the initial set of documents [5].

Reranking The documents obtained using the Boolean query are then reranked using MedCPT [6]. MedCPT is a cross-encoder trained on 18.3 million click-based, sentence-like query–article pairs from PubMed search logs (single-keyword queries were filtered out), using listwise contrastive loss with hard negatives sampled from the top results of the MedCPT retriever. It is specifically trained to handle natural-language questions, like those in this task.

2.1.2. Phase B: Answer Generation

We compare three approaches. (1) **ICL (in-context learning).** The gold snippets are passed directly in the prompt context. (2) **PRAG (Parametric RAG)** [7]**.** Document knowledge is parameterised into LoRA adapters and injected directly into the model weights. For each gold snippet, the snippet is rewritten once and three QA pairs are generated from its content. We then train the adapter model on top of the base LLM using (original passage, QA_1), (original passage, QA_2), (rewritten passage, QA_1), (rewritten passage, QA_2), and QA_3 without any passage in context, for 3 epochs. The loss is computed over the entire concatenated sequence—passage, question, and answer—so the model is also trained to regenerate the passage content, not only to predict the answer. The resulting LoRA adapter is intended to encode information to answer questions about the snippet. (3) **Combine** [9]**.** The gold snippets are passed in the input context *and* the LoRA adapters are added to the weights.

When more than one gold snippet is used, each snippet is parameterised by a pair of low-rank matrices $A_j \in \mathbb{R}^{h\times r}$ and $B_j \in \mathbb{R}^{k\times r}$, whose product $A_j B_j^\top$ approximates the weight update ΔW_j for that snippet. The original PRAG paper merges all snippets' updates by summing:

$$\Delta W_{\text{merge}} = \alpha \cdot \sum_{j=1}^{k} A_j B_j^\top \tag{1}$$

where α is the LoRA scaling factor and k is the number of gold snippets. When $k > 1$ this sum grows proportionally with k, making the total weight update larger the more snippets are used. To keep the update magnitude independent of k, we divide by k:

$$\Delta W_{\text{merge}} = \frac{\alpha}{k} \cdot \sum_{j=1}^{k} A_j B_j^\top \tag{2}$$

2.2. Experimental Setup

Experiments were conducted on NVIDIA A100 or 2080ti GPUs depending on the availability of the hardware in the cluster environment we used to perform our experiments.

2.2.1. Phase A

AutoBool was given the question and returned a Boolean query, which was sent to PubMed's eSearch endpoint. The result list was cut off at 8 000 IDs (4 000 for Batch 1). These IDs were then fetched from the eFetch endpoint. The number of results returned by each Boolean query was logged. From Batch 3 onwards, results were requested sorted by relevance.

2.2.2. Phase B

The base model we used for generating responses is `qwen2.5-1.5b-instruct` and the augmentation model is `qwen2.5-7b-instruct`.[1,2] LoRA rank 2 and $\alpha = 32$ were used as in the original PRAG paper.

[1] https://huggingface.co/Qwen/Qwen2.5-1.5B-Instruct
[2] https://huggingface.co/Qwen/Qwen2.5-7B-Instruct

yes/no `Given only the QUESTION, answer the QUESTION only with 'Yes' or 'No'.`

factoid `Extract key biomedical entities to answer the QUESTION. Answer with exactly one highly relevant biomedical entity. Do not provide a list. Just the entity. If no relevant entities exist, return 'None.'.`

list `Extract key biomedical entities to answer the QUESTION. Return your answer as a strict comma-separated list of entities (e.g. Entity1, Entity2, Entity3). Do not include any other text. If no relevant entities exist, return 'None'.`

summary `Answer the QUESTION by returning a single paragraph sized text (use max 50 words).`

Figure 1: Templates prepended to the system prompt from Batch 3 onward to enforce the correct output format per question type.

Training used a learning rate of 10^{-4} for 3 epochs. Task-specific prompt templates were introduced from Batch 3 onward to help enforce correct output formatting; this notably improved adherence to the required structure for factoid questions. The exact templates are listed in Figure 1.

For Phase B we only submitted ideal answers for summary questions; for yes/no, factoid, and list questions only exact answers were submitted. The configurations compared are: (1) ICL; (2) PRAG with 1 document; (3) PRAG with 3 documents, unscaled (Eq. 1); (4) PRAG with 3 documents, scaled (Eq. 2); and (5) Combine with 3 documents, scaled. We restrict ourselves to 3 documents because PRAG reports effectiveness only with that many; for a fair comparison, ICL also receives the same 3 documents.

Top-3 document selection. We observed that the training dataset contains many near-duplicate golden snippets. Selecting the three most relevant passages by cross-encoder score therefore tends to return redundant content. To counter this, we additionally compute embeddings and apply Maximal Marginal Relevance (MMR):

$$\mathrm{MMR}(p) = \lambda \cdot \mathrm{rel}(p) - (1 - \lambda) \cdot \max_{s \in S} \mathrm{sim}(p, s),$$

with $\lambda = 0.5$, using `cross-encoder/ms-marco-MiniLM-L2-v2` for relevance scoring and `sentence-transformers/all-MiniLM-L6-v2` for similarity.[3,4]

2.3. Results

2.3.1. Phase A

A pipeline of Boolean-query generation plus cross-encoder reranking performs reasonably well compared to other competition systems.

Figure 2 shows the number of PubMed IDs returned by the Boolean query per question type. The distribution is heavily right-skewed: some queries return only a few hundred documents, while others return more than 100k and, for very general questions, over a million. Cutting off at 8 000 is therefore likely to miss many relevant documents in the upper tail. At the same time, the long lower tail means that many queries return only a few hundred documents to begin with.

Tables 1 and 2 report our effectiveness as evaluated by the task organisers and the recall gap to the top-performing system per batch. Despite the aggressive 8 000-document cutoff and no LLM-based final reranking, the system obtains relatively good recall, indicating that most relevant documents were present within the top 8 000 results. The remaining gap is likely a reranking problem rather than a retrieval problem: increasing the cutoff from 4 000 to 8 000 documents in Batch 2 did not improve recall

[3] https://huggingface.co/cross-encoder/ms-marco-MiniLM-L2-v2
[4] https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2

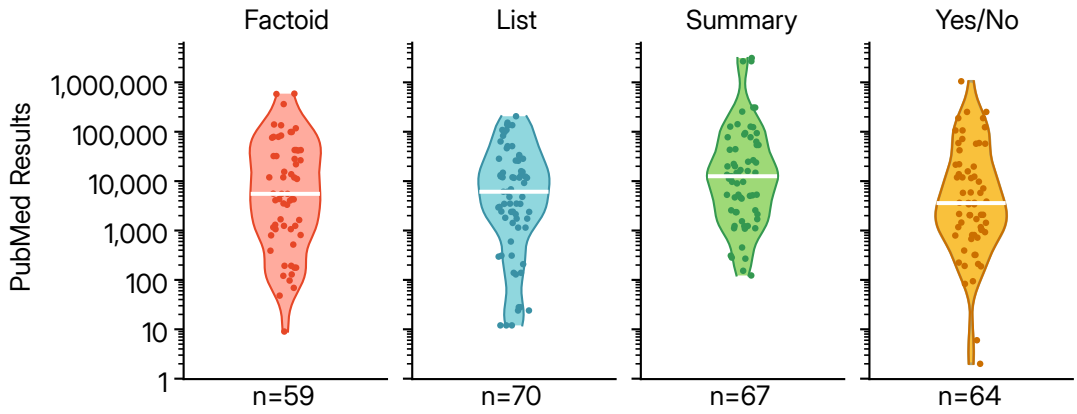

Figure 2: Distribution of PubMed total document counts per question type across 260 queries from Batches 2–4. Each violin shows a kernel density estimate (KDE) computed in $\log_{10}$ space using a Gaussian kernel plotted on a $\log_{10}$ y-axis. Individual queries are shown as jittered points; the white horizontal line marks the median.

Table 1
Phase A evaluation results (documents).

Batch	System	Prec	Recall	F	MAP	GMAP
1	autobool-medcpt-4000	0.0892	0.2769	0.1163	0.1696	0.0017
2	autobool-medcpt-8000	0.0721	0.2703	0.1076	0.1462	0.0008
3	autobool-medcpt-8000	0.0572	0.2263	0.0853	0.1072	0.0005
4	autobool-medcpt-8000	0.0567	0.2228	0.0853	0.1268	0.0003

Table 2
Recall of the top-performing system per batch versus ours.

Batch	Top System	Top Recall	Our Recall	Difference
1	dictycite-max-rew-sl	0.4090	0.2769	0.1321
2	dictycite-max-rew-si	0.3357	0.2703	0.0654
3	Fleming-2	0.3571	0.2263	0.1308
4	dictycite-max-rew-si	0.3698	0.2228	0.1470

(0.258 vs. 0.270), suggesting that the reranker, not AutoBool, is the bottleneck. In hindsight, using a lower α in AutoBool's reward to increase precision may have been more appropriate given that many documents were discarded by the cutoff.

One notable finding is that adding the "sort by relevance" eSearch parameter consistently worsened retrieval effectiveness. A plausible explanation is that BioASQ questions (and the answers to those questions) are biased toward recent topics (and results), so PubMed's default ordering (by most recent publication date) is a better prior than its match-count-based relevance ranking. Overall, AutoBool is effective given its minimal setup.

2.3.2. Phase B

We compare five configurations, all using task-specific system prompts: ICL, PRAG (n=1), PRAG (n=3, unscaled), PRAG (n=3, scaled), and Combine (n=3, scaled). Figure 3 reports results for Batches 3 and 4 across all question types.

Results are mixed across question types, and no single configuration is consistently effective. As

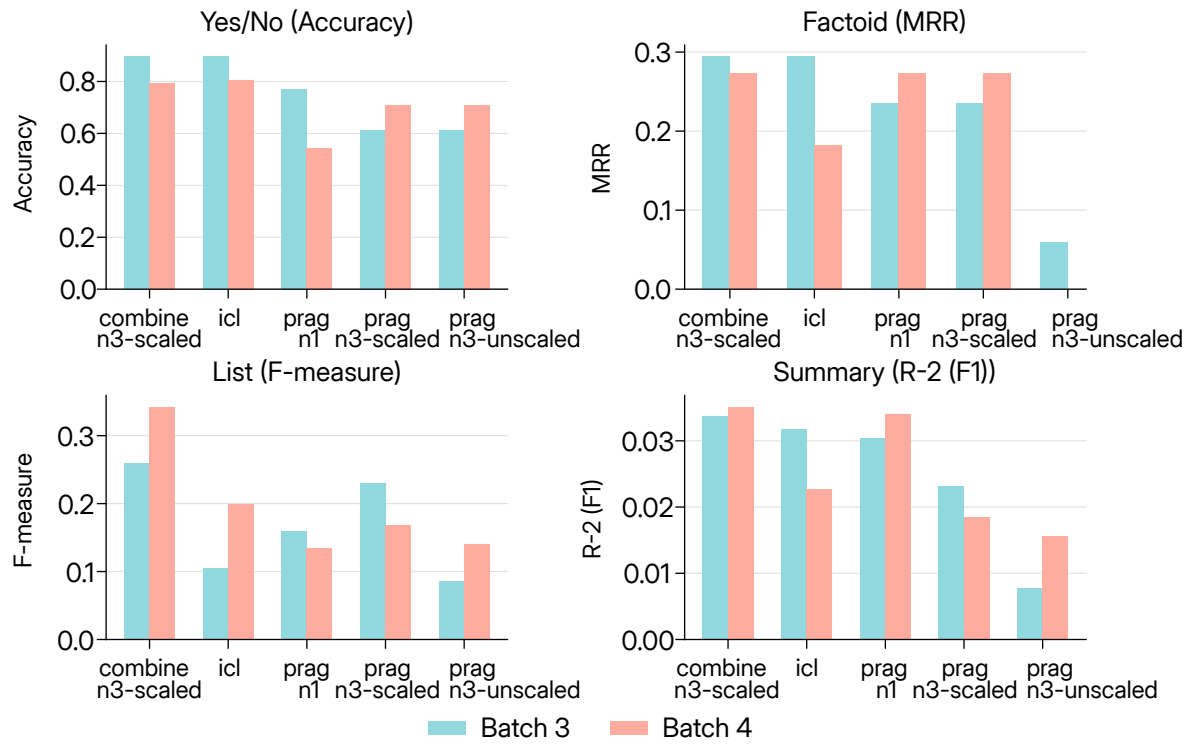

Figure 3: Phase B results for Batches 3 and 4 across all question types and configurations. Batches 3 and 4 are the most informative since all configurations used task-specific prompt templates from Batch 3 onward.

a general trend, Combine and ICL tend to be more effective than PRAG alone, which aligns with the finding in related work that parametric injection by itself is not sufficient [9]. PRAG unscaled is not effective across the board. Without the scaling factor, the summed adapters over-amplify the weight update in proportion to the number of documents, and the model often broke entirely and started repeating fragments of the system prompt. There is no clear effectiveness difference between PRAG $n{=}1$ and PRAG $n{=}3$ scaled, leaving open the question of whether PRAG actually incorporates knowledge from the retrieved documents or whether the adapter training merely changes the model's output behaviour in other ways.

Results are also subject to evaluation bias. PRAG's adapter training encourages short, concise outputs, which F1-based metrics reward. Part of PRAG's apparent advantage in factoid and list tasks may therefore reflect better output formatting rather than better understanding of the document content. This aligns with Qi et al. [9], who note that F1 scores in this setting are biased by output structure. These issues limit the conclusions that can be drawn from the results.

2.4. Future Work

Several directions remain for future work. For Phase A, AutoBool could be fine-tuned on BioASQ training data rather than relying on the off-the-shelf systematic-review checkpoint, and the hard 8 000-document cutoff could be replaced by a query-adaptive scheme. For example, combining intersecting and union Boolean queries or tuning AutoBool's reward parameter α toward higher precision to better match the reranker's effective input size. For Phase B, an open question is whether PRAG actually incorporates document knowledge, which could be investigated with controlled probing on synthetic facts. Additionally, alternative adapter merging strategies that reduce inter-adapter interference, such as TIES-Merging [10] or ARROW [11], are promising directions.

3. Gut-Brain Interplay Information Extraction

The corpus of scientific literature is growing at a pace that makes it difficult to keep up with developments. This amplifies the existing need for structured information extraction and prompted the creation of the GutBrainIE task at the BioASQ Lab [3]. The task specifically examines the connection between gut microbiome and brain health with the goal of improving the state-of-the-art methods in information extraction, semantic indexing and question-answering. In the second edition, the task poses four subtasks:

- Named Entity Recognition (NER): Extraction and classification of entity text spans into one of 13 categories.
- Named Entity Recognition and Disambiguation (NERD): Extraction and classification of entity text spans, with a link to one of over 3500 concept identifiers.
- Mention-Level M-RE: Identification of relationships between entity mentions.
- Concept-Level C-RE: Identification of relationships between entities linked to their concept identifier.

For training and evaluating methods to address these subtasks, the organizers provide four data collections of varying quality of annotation: gold (expert annotated), silver, silver2025, and bronze (model annotated). Our code and the configurations to reproduce our submissions are available at https://github.com/health-nlp/bioasq26-gutbrainie.

Related Work The dominant approach for NER are transformer-based models such as modifications of BERT [12] or GLiNER [13]. Due to the specialized vocabulary in the biomedical domain these models require fine-tuning to be effective, which lead to the development of models such as BioBERT [14] or BiomedBERT [15]. These models can also be applied to the problem of RE by framing it as a sequence classification problem [16, 8], though more specialized approaches such at ATLOP exist [17]. Both saw use in the previous edition of the Lab [18], supplemented with techniques such as ensembling or weighting training.

3.1. Methodology

This section explains our approach to the GutBrainIE task. We group the subtasks by extraction (NER and M-RE; Section 3.1.1) and disambiguation (NERD and C-RE; Section 3.1.2) and develop models to address them in isolation. The resulting extraction and linking models are chained to produce predictions of linked entities and their relations for a given paper. We employ loss weighting to counteract dataset quality and class imbalance. Finally, we ensemble by filtering to extractions that are predicted by two or more separate pipeline configurations.

3.1.1. Entity and Relation Extraction

To jointly extract entities and relations from the title and abstract, we adopt the GLiNER2 multi-task architecture [8]. By creating a schema that specifies the entity and relation types relevant for the task, optionally with descriptions of each, the model extracts and classifies the input regions of interest in a single forward pass. We train Low Rank Adaptation (LoRA) [19] weights of the model to increase domain specific effectiveness, while preserving the generalization ability gained in pretraining. In some cases and depending on the confidence thresholds, the model predicts text spans as in relation to one another, which it did not recognise as entities under the specified classes. In this case, we default to Disease Disorder or Finding (DDF) as the most common class.

To implement weighted training, we first compute a weight w_c for each class c, derived from the relative frequency $freq_c$, where c may be either an entity class or relation type, We min-max scale the resulting weights to the range $[w_{min}, w_{max}]$ to maintain stable gradients (Eq. 4), aggregate on

abstract level and factor in the collection quality weight w_{split} to get a weight per sample w_{sample} (Eq. 5). Multiplying the loss $\mathcal{L}(p)$ of a sample p by w_{sample} yields the final training objective $\mathcal{L}_{weighted}$ (Eq. 6).

$$w_c = \frac{|N_{split}|}{|C|} \cdot \frac{1}{freq_c} \tag{3}$$

$$\bar{w}_c = \frac{w_c - \min\limits_{c' \in N_{split}} (w'_c)}{\max\limits_{c' \in N_{split}} (w'_c) - \min\limits_{c' \in N_{split}} (w'_c)} \cdot (w_{max} - w_{min}) + w_{min} \tag{4}$$

$$w_{sample}(p) = \underset{c \in p}{agg}(\bar{w}_c) \tag{5}$$

$$\mathcal{L}_{weighted} = \sum_{split \in \mathcal{D}} \sum_{p \in split} \mathcal{L}(p) \cdot w_{sample}(p) \cdot w_{split} \tag{6}$$

Where N_{split} and $|N_{split}|$ are the set of all entities and relations, and the number of all entites and relations in the collection $split$, respectively, $|C|$ is the number of entity classes and relation types, agg is the sample weight aggregation function briefly discussed in 3.3, $\bar{w}_c$ is the scaled class weight and $\mathcal{D}$ is the entire training corpus.

3.1.2. Entity Disambiguation

Reasoning over extracted knowledge from text requires the mentioned entities to correspond to their real world counterpart. It is not sufficient to let entities be represented by their raw text span, as the span is neither necessarily unique nor unambiguous. A natural way to approach this problem is by linking each mention to an entry in a known knowledge base identified by a Uniform Resource Identifier (URI). This shifts the problem of ambiguity to a problem of extreme classification over the possible entities. Depending on the underlying knowledge base, this can quickly become intractable when framed as a classical supervised learning problem, especially if there exist entities with few or no training examples. For a general approach to this problem, we therefore require that it function with few examples per class, ideally by using entity semantics rather than the URI directly, which is often meaningless on its own.

Baseline Linking We begin by adopting the linking pipeline provided as the competition baseline. This pipeline consists of three steps to link entity mentions. First, it checks whether an exact match exists in the training data, that is a previous mention where both the text span and assigned class match the mention in question; if so, it assigns the same URI. Next, it checks for a partial match, that is an identical text span that got assigned a different entity class. If neither match, it encodes the mention using an embedding model and compares it to cached embeddings of URIs descriptions, assigning the URI with the highest similarity. By default, the implementation uses `pubmedbert-base-embeddings` as the embedding model,[5] though this is a degree of freedom as long as mentions and URI descriptions are embedded into the same vector space. The descriptions are obtained from the Unified Medical Language System (UMLS).

This approach works but has several limitations that we wish to address. **Firstly,** as noted, the mention text span is an imperfect representation in some cases. Consider a paper discussing E. coli and later referring to it as 'the bacterium'. From this span alone it is impossible to determine the correct entity URI. Regardless of whether the span has a match in the training set or is resolved by similarity, it essentially chooses at random. It follows that injecting a context representation is necessary.

Secondly, the URI representation is entirely dependent on description embeddings. Although great effort goes into maintaining UMLS, it is often less detailed for rare entities. Consider the entity identified

[5] https://huggingface.co/NeuML/pubmedbert-base-embeddings

by 2042320.[6] UMLS stores a single name: *Parabacteroides gordonii* and no definitions. Even for domain-specific language models like PubMedBert this is unlikely to have a well grounded embedding. Without a description to supplement, this approach may not hold enough semantic depth to be able to resolve aliases or abbreviations.

Finally, from an engineering standpoint it is inconvenient to require the full training set at inference. Although manageable for the approximately 150k entity mentions in the competition dataset, it becomes infeasible at scale.

We propose an alternative architecture, which we term **Graphlinker**, to resolve URIs in a semantically richer way. Similarly to the baseline, our method embeds entities and URIs into a shared vector space and assigns a label by proximity, but we differ in the components that influence this embedding. Figure 4 gives an overview of our proposed architecture.

Entity Embedding We aim to capture mention semantics beyond the text span. To that end, we feed not just the span but also the class label with its description and window around mention to a pretrained embedding model. The three vectors are concatenated and projected to a bottleneck. The result is then projected into the shared vector and normalized.

URI Embedding To generate richer representations for URIs we consider the known relationships between them. Recall our example of the bacterium *Parabacteroides gordonii*. Even though neither aliases nor a description are available, we can learn certain attributes from the relationships it has with other entities. We know, for example, that it changes the expression of APOEe2 alleles [20] and influences neuroinflammation [20], that Parkinson's disease changes its abundance [21] and that it is part of the gut microbiome [21]. This knowledge can be incorporated through the use of message passing in a Graph Neural Network (GNN).

We begin by constructing a concept-level graph of all relationships in the training set, where each vertex is a URI and each edge denotes a relationship, annotated with the relation type. Each vertex is initially positioned as the BERT embedding of its description, similarly to the baseline. We then train a two layer graph attention network [22] to aggregate neighborhood information in a two hop radius. To preserve the information from the initial description embedding, we add a skip connection before normalizing. In Section 3.2 we show that similar performance can be achieved by adding self loops to the graph. We retain the skip connections, such that the role of the GNN stays isolated to representing the relational information.

To train the model given (entity, URI) pairs, we compute the cosine similarity $s_{i,j}$ between an entity embedding e_i and all URI embeddings u_i and apply scaled CrossEntropy loss with a learned temperature τ (Eq. 8), inspired by Radford et al. [23]s use of the InfoNCE loss originally proposed by van den Oord et al. [24]. At inference we cache the URI embeddings and classify by largest similarity.

$$s_{i,j} = \frac{e_i \cdot u_j}{\tau} \tag{7}$$

$$\mathcal{L}_{\mathcal{B}} = \frac{1}{B} \sum_{i=1}^{B} \left[-log \frac{exp(s_{i,y_i})}{\sum_{j=1}^{J} exp(s_{i,j})} \right] \tag{8}$$

Where $\mathcal{L}_{\mathcal{B}}$ is the loss for batch B, J is the number of URIs and y_i is the correct URI for entity e_i.

3.2. Experimental Setup

This section describes the experimental setup. Experiments were conducted on A100 or 2080ti NVIDIA GPUs, depending on the computational demand. The development process is composed of isolated experiments to inform design decisions. We assume little interaction between extraction and linking model and optimize them separately.

[6] http://id.nlm.nih.gov/mesh/2042320

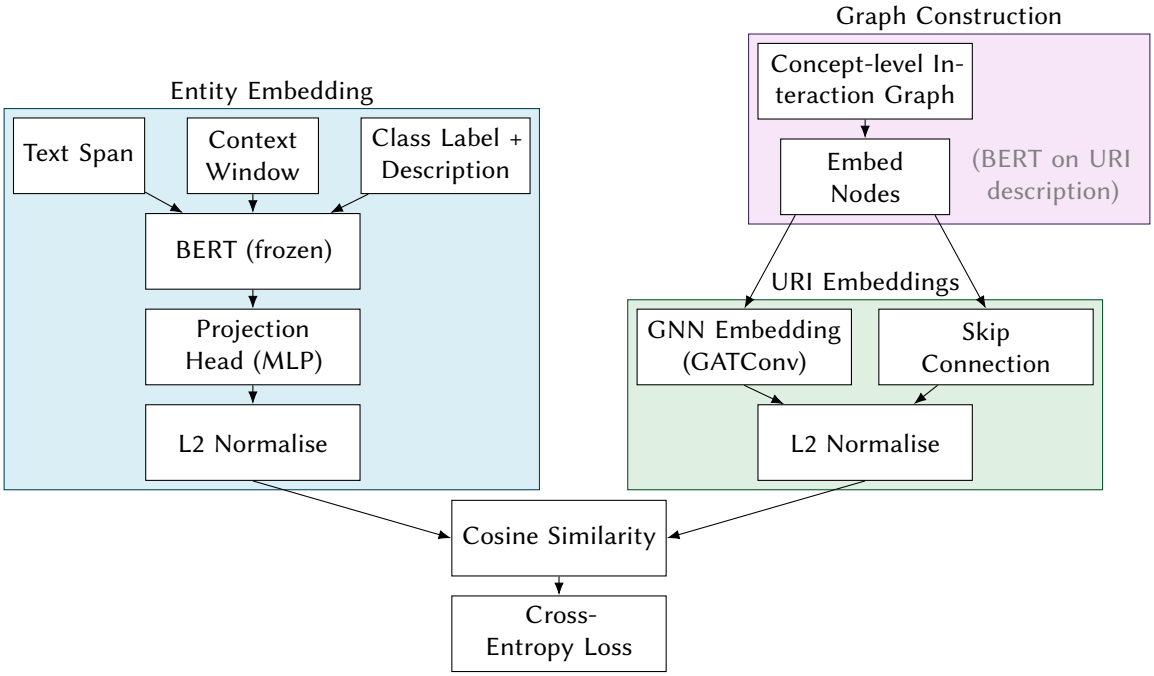

Figure 4: Graphlinker architecture. We embed entities and URIs into a shared vector space and employ CrossEntropy on their similarity as a contrastive loss for classification. The entity embedding is derived from the text span, a context window and the predicted class. The URI is embedded as its description enriched with relationships to other URIs through application of a GNN.

3.3. GLiNER2

We explore the impact of the following hyperparameters and report our findings:

LoRA Rank r and Scaling Factor α We explore a range of r and α between 4 and 32. With the exception of $r = 8, \alpha = 4$, which performs noticeably worse, we observe absolute differences no greater than 0.003 in F1. We conclude that this choice has limited impact and choose $r = 8, \alpha = 16$ as suggested by Zaratiana et al. [8].

Learning Rate We investigate learning rates between $1e^{-5}$ and $5e^{-4}$ and observe F1 scores between 0.704 and 0.760 for entities and 0.100 to 0.114 for relations. Both are maximised by a learning rate of $5e^{-4}$

Sample Weight Aggregation Strategy We compare the aggregation strategies of max pooling and product. Individual entity and relation weights are clamped to ranges of $[.5, 1], [1, 2], [1, 5]$ or $[1, 10]$. The result of this experiment is shown in Table 3. We conclude that max pooling is the best strategy, as it is more stable when faced with outliers. Preventing outliers all together by clamping to a narrow range appears to benefit both methods.

Data Quality Weight To study the effect of data collection weighting we compare unweighted to heavily weighted training with the gold collection at 4x and silver at 2x the impact of the bronze collection. We observe an absolute effectiveness gain of 0.04 in entity F1. For our submissions we follow Andersen et al. [25] and adopt mild weighting of bronze at 0.75x, and gold at 1.25x the loss per sample.

3.4. Graphlinker

We follow a similar approach for tuning the graphlinker architecture.

Table 3: Evaluating sample weight aggregation schemes.

Aggregation Scheme	Clamp Min	Clamp Max	F1 Entity	F1 Relation
Product	1	10	0.489	0.032
	1	5	0.726	0.087
	1	2	0.759	0.112
	0.5	1.5	0.701	0.067
Max	1	10	0.754	0.114
	1	5	0.758	0.117
	1	2	0.758	0.117
	0.5	1.5	**0.760**	**0.125**

Table 4: Evaluating learning rates.

Entity LR	URI LR	F1
e^{-2}	e^{-2}	0.492
e^{-2}	e^{-3}	0.478
e^{-3}	e^{-2}	0.556
e^{-3}	e^{-3}	0.584
e^{-4}	e^{-2}	**0.648**
e^{-4}	e^{-3}	0.647

Table 5
Description and scores of the submitted runs. Highest F1 on dev set is highlighted in bold, ignoring runs trained on dev set. For full configuration of individual components see Table 6.

RunID	Description	Extractor	Linker	F1 (NER)	F1 (NERD)	F1 (M-RE)	F1 (C-RE)
0	both best runs during exploration	PREV	PREV	0.6701	0.4613	0.1335	0.0725
1	both trained on dev set	DEV	DEV	0.6857	0.4626	0.1554	0.0840
2	extractor full finetune	FULL	PREV	0.6386	0.4432	0.0579	0.0154
3	LLM as entity embedder for linking	PREV	LLM	0.6701	0.4330	0.1335	0.1020
4	best extractor, baseline linking	PREV	BASE	0.6701	**0.5691**	0.1335	0.1601
5	retrain with all optimizations	FIN	FIN	0.6523	0.5084	0.1115	0.1478
6	extractor trained on dev, baseline linking	DEV	BASE	0.6857	0.5525	0.1554	0.1451
7	extractor overfit	OFIT	DEV	0.7347	0.4751	0.2459	0.1653
8	extractor overfit, baseline linking	OFIT	BASE	0.7347	0.6040	0.2459	0.2442
25	ensemble of other predictions	-	-	**0.6943**	0.5477	**0.1546**	**0.1952**

Learning Rates The results in Table 4 suggest that the entity branch quickly converges, requiring a lower initial learning rate than the URI branch.

Quality Weight Our findings for this parameter are noisy. We vary the bronze weight between 0.2 and 1, and gold between 1 and 2. We find no consistent pattern in the results, but choose the configuration which yields the highest validation F1: bronze-0.5, gold-1.

Embedding Dimensions We run a gridsearch of dimensions between 64 and 256 for entity-, URI and shared embedding dimension. The results are again noisy. A larger entity bottleneck is positively correlated (Pearson's r: 0.313) with higher F1, while the opposite is true for the uri bottleneck (r: -0.226). We pick the best performing (shared, uri, entity) combination of (256, 64, 128).

Context Window Size We experiment with a context window size between 0 and 1000 characters around each mention and find context size to be negatively correlated with better F1 (r: -0.648) with the most effective model scoring at 0.57 F1 with no context, around 0.02 higher than models with 100 or more characters of context. We attribute this to the fact that the Small Language Model (SLM) used to encode the context does not have the expressive power to extract the relevant information from context in a way that is specific enough to the entity mention to aid in classification. To remedy this, we experiment with a specifically prompted Large Language Model (LLM) but find no significant improvement.

Table 6
Configurations of individual components.

Extractor	LoRA r	Sample Weighting	Quality Weights [dev, gold, silver, bronze]	Learning Rate	Epochs
PREV	8	false	[0, 1.5, 1, .5]	$5e^{-4}$	11
FULL	full	false	[0, 1, 1, 1, 1]	$1e^{-4}$	25
FIN	8	true	[0, 1.25, 1, .5]	$1e^{-4}$	7
DEV	8	true	[1.25, 1.25, 1, .5]	$1e^{-4}$	6
OFIT	8	true	[1.25, 1.25, 1, .5]	$1e^{-4}$	50

Linker	Embed. Dimension [shared, entity, uri]	Learning Rates [τ, entity, uri]	Quality Weights [dev, gold, silver, bronze]	Text Embedder	Epochs
PREV	[128, 128, 128]	[$5e^{-3}$, $1e^{-4}$,$1e^{-2}$]	[0, 1.25, 1, .75]	pubmedbert	3
LLM	[128, 128, 64]	[$5e^{-3}$, $1e^{-4}$,$1e^{-2}$]	[0, 1.25, 1, .5]	Qwen3-4B	24
FIN	[256, 128, 64]	[$5e^{-3}$, $1e^{-4}$,$5e^{-3}$]	[0, 1.25, 1, .5]	pubmedbert	12
DEV	[128, 128, 64]	[$5e^{-3}$, $1e^{-4}$,$5e^{-3}$]	[1.25, 1.25, 1, .5]	pubmedbert	11

3.5. Results

We propose a modular pipeline for entity extraction, linking and relationship classification that incorporates information from mention context and relational structure between entities to make predictions, thereby addressing several conceptual shortcomings of the competition baseline. Table 5 shows the configurations we submit, alongside their evaluation on the dev set. The table references identifiers for specific runs. The hyperparameter choices corresponding to these identifiers are listed separately in Table 6.

In the final evaluation our extraction component lags behind the baseline with a micro F1 of 0.725 vs. 0.799 and 0.186 vs. 0.388 in the NER and RE subtasks, respectively [3]. Despite the worse extractions, the graphlinker architecture is more effective than the baseline on the NERD subtask with a micro F1 of 0.478 vs. 0.439 and is almost as effective as the linked RE with 0.125 vs. 0.134. The latter reduces the relative effectiveness decline from 52% to less than 9%, demonstrating a much better raw linking effectiveness.

4. Conclusion

In these working notes, we present the results of the submissions from team Health-NLP on two tasks at the BioASQ 2026 lab: BioASQ Task B on Biomedical Semantic QA and BioASQ Task GutBrainIE. Through these tasks, we investigate how well recent state-of-the-art approaches generalise to the biomedical domain. Our investigation highlights several limitations of these approaches and the domain-specific adaptations used to make them more effective, such as implementing scaled PRAG for weight merging and MMR for diverse document selection in Task B on Biomedical Semantic QA, and, hyperparameter tuning for GLiNER2 architecture for the BioASQ Task GutBrainIE. We highlight several promising directions for the future work that can further improve these approaches.

Declaration on Generative AI

The authors used generative AI tools for minor result analysis and spelling and grammar checking. Claude Sonnet 4.6 was used to help rephrase paragraphs to improve clarity, conciseness, and style. Claude Sonnet 4.6 was used as support when writing code. This was merely support and not the automatic generation of larger parts of the program. All AI-generated content was critically reviewed and verified by the authors before inclusion in this work. The authors reviewed and edited all content

and take full responsibility for the publication's content.

References

[1] 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.

[2] 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 2026 Working Notes, 2026.

[3] M. Martinelli, V. Bonato, G. M. Di Nunzio, G. Faggioli, N. Ferro, O. Irrera, B. Kantz, S. Marchesin, L. Menotti, S. Merlo, R. Michelotto, G. Tussardi, F. Vezzani, P. Waldert, G. Silvello, Overview of GutBrainIE@CLEF 2026: Gut-Brain Interplay Information Extraction, in: E. Sánchez Salido, A. Barrón-Cedeño, A. García Seco de Herrera, S. MacAvaney, J. M. Struß (Eds.), CLEF 2026 Working Notes, 2026.

[4] S. Wang, H. Scells, B. Koopman, G. Zuccon, Autobool: Reinforcement-learned llm for effective automatic systematic reviews boolean query generation., in: EACL, 2026, pp. 1468–1493. URL: https://aclanthology.org/2026.eacl-long.68/.

[5] E. Sayers, The e-utilities in-depth: Parameters, syntax and more, 2009. URL: https://www.ncbi.nlm.nih.gov/books/NBK25499/.

[6] 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. (2023). doi:10.1093/BIOINFORMATICS/BTAD651.

[7] W. Su, Y. Tang, Q. Ai, J. Yan, C. Wang, H. Wang, Z. Ye, Y. Zhou, Y. Liu, Parametric retrieval augmented generation., in: SIGIR, 2025, pp. 1240–1250. doi:10.1145/3726302.3729957.

[8] U. Zaratiana, G. Pasternak, O. Boyd, G. Hurn-Maloney, A. Lewis, Gliner2: An efficient multi-task information extraction system with schema-driven interface, 2025. URL: http://arxiv.org/abs/2507.18546v1.

[9] J. Qi, Z. Xu, Q. Wang, L. Huang, Ar-rag: Autoregressive retrieval augmentation for image generation, 2025. URL: http://arxiv.org/abs/2506.06962v3.

[10] P. Yadav, D. Tam, L. Choshen, C. A. Raffel, M. Bansal, Ties-merging: Resolving interference when merging models., in: NeurIPS, 2023. URL: http://papers.nips.cc/paper_files/paper/2023/hash/1644c9af28ab7916874f6fd6228a9bcf-Abstract-Conference.html.

[11] O. Ostapenko, Z. Su, E. M. Ponti, L. Charlin, N. L. Roux, L. Caccia, A. Sordoni, Towards modular llms by building and reusing a library of loras., in: ICML, 2024, pp. 38885–38904. URL: https://proceedings.mlr.press/v235/ostapenko24a.html.

[12] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding (2018). arXiv:10.48550/arXiv.1810.04805.

[13] U. Zaratiana, N. Tomeh, P. Holat, T. Charnois, Gliner: Generalist model for named entity recognition using bidirectional transformer. (2024) 5364–5376. doi:10.18653/V1/2024.NAACL-LONG.300.

[14] 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. (2020) 1234–1240. doi:10.1093/BIOINFORMATICS/BTZ682.

[15] S. Chakraborty, O. E. Bisong, S. Bhatt, T. Wagner, R. Elliott, F. Mosconi, Biomedbert: A pre-trained

biomedical language model for qa and ir., in: COLING, 2020, pp. 669–679. doi:10.18653/V1/2020.COLING-MAIN.59.

[16] P. Shi, J. Lin, Simple bert models for relation extraction and semantic role labeling, 2019. URL: https://arxiv.org/abs/1904.05255. arXiv:1904.05255.

[17] W. Zhou, K. Huang, T. Ma, J. Huang, Document-level relation extraction with adaptive thresholding and localized context pooling., in: AAAI, 2021, pp. 14612–14620. doi:10.1609/AAAI.V35I16.17717.

[18] G. Faggioli, N. Ferro, P. Rosso, D. Spina (Eds.), Working Notes of the Conference and Labs of the Evaluation Forum, CLEF 2025, Madrid, Spain, 9-12 September 2025, volume 4038 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2025. URL: https://ceur-ws.org/Vol-4038.

[19] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, Lora: Low-rank adaptation of large language models. (2022). URL: https://openreview.net/forum?id=nZeVKeeFYf9.

[20] C. Chevalier, B. B. Tournier, M. Marizzoni, R. Park, A. Paquis, K. Ceyzériat, A. M. Badina, A. Lathuiliere, S. Saleri, F. D. Cillis, A. Cattaneo, P. Millet, G. B. Frisoni, Fecal microbiota transplantation from a human at low risk for alzheimer's disease improves short-term recognition memory and increases neuroinflammation in a mouse model, Genes, Brain and Behavior 24 (2025). doi:10.1111/gbb.70012.

[21] Z. Yan, F. Yang, J. Cao, W. Ding, S. Yan, W. Shi, S. Wen, L. Yao, Alterations of gut microbiota and metabolome with parkinson's disease, Microbial Pathogenesis (2021). doi:10.1016/j.micpath.2021.105187.

[22] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Liò, Y. Bengio, Graph attention networks. (2018). URL: https://openreview.net/forum?id=rJXMpikCZ.

[23] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, I. Sutskever, Learning transferable visual models from natural language supervision., in: ICML, 2021, pp. 8748–8763. URL: http://proceedings.mlr.press/v139/radford21a.html.

[24] A. van den Oord, Y. Li, O. Vinyals, Representation learning with contrastive predictive coding (2018). URL: http://arxiv.org/abs/1807.03748v2.

[25] L. R. Andersen, M. H. Dolmer, M. I. Gardshodn, J. M. Rodriguez, D. Dell'Aglio, Trusting gut instincts: Transformer-based extraction of structured data from gut-brain axis publications., in: CLEF, 2025, pp. 99–119. URL: https://ceur-ws.org/Vol-4038/paper_6.pdf.