Farming Open LLMs for Biomedical Question Answering

Dimitra Panou[1,2], Alexandros C. Dimopoulos[1,3] and Martin Reczko[1,*]

[1]Institute for Fundamental Biomedical Science, Biomedical Sciences Research Center "Alexander Fleming", 34 Fleming Street, 16672 Vari, Greece

[2]Department of Informatics and Telecommunications, National and Kapodistrian University of Athens, 15784 Athens, Greece

[3]Hellenic Naval Academy, 18539 Piraeus, Greece

Abstract

A large number of performant, open Large Language Models (LLMs) are continuously appearing. Here we deploy a selection of these for embedding and retrieval of documents and snippets as well as retrieval-augmented generators to answer biomedical questions within the BioASQ competition. Dense retrieval based on distances between dense representations obtained by LLM embeddings of the corpus and the question and hybrid sparse/dense methods result in higher mean average precisions compared to traditional sparse retrieval methods. In the exact answer category, which is processed using open LLMs in a zero-shot approach, our submission shares one first place in the last batch of the BioASQ 12b competition.

Keywords

Biomedical Question Answering, BioASQ, Large Language Models, Retrieval-augmented generation

1. Introduction

The emergence of open-source Large Language Models (LLMs) marks a notable trend in the tech landscape. These models are increasingly tailored to address diverse tasks such as powering chatbots, providing tech support, aiding in healthcare, and facilitating multilingual capabilities [1]. The significance of open-source LLMs merits deeper exploration, especially with the availability of supportive tools and platforms like Ollama [2] and GPT4All [3]. These resources not only promote the use of open-source LLMs but also simplify testing and implementation processes. While industry-standard models like ChatGPT have long been utilized, open-source LLMs [4] offer distinct advantages, notably in terms of transparency, reproducibility and cost. These attributes, often lacking in commercial models, foster a level of trust and accountability that resonates with developers and users alike.

2. Methodology

2.1. Phase A: Document Retrieval

For Phase A, the BioASQ team releases biomedical questions posed by their experts [5]. Participants have 24 hours to respond with 10 relevant article abstracts per question, extracted from PubMed, along with the most relevant snippets from these abstracts. In figure 1, our processing during Phase A is summarized.

For batch 1 and 2 of BioASQ12, we utilized our previously developed GANBERT model [6] with optimized parameters for document selection. In brief, GANBERT extends the fine-tuning of a BERT architecture with unlabeled data using a Generative Adversarial Network (GAN) framework, where a generator is trained to produce samples of the internal BERT representation resembling the distribution over the unlabeled data, and a discriminator that is trained to distinguish samples of the generator from the real instances. This semi-supervised method can improve generalization. We expanded the training data for GANBERT by augmenting the unlabeled dataset with random segments from Pubmed

CLEF 2024: Conference and Labs of the Evaluation Forum, September 09–12, 2024, Grenoble, France

*Corresponding author.

panou@fleming.gr (D. Panou); dimopoulos@fleming.gr (A. C. Dimopoulos); reczko@fleming.gr (M. Reczko)

0000-0002-9824-4489 (D. Panou); 0000-0002-4602-2040 (A. C. Dimopoulos); 0000-0002-0005-8718 (M. Reczko)

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073

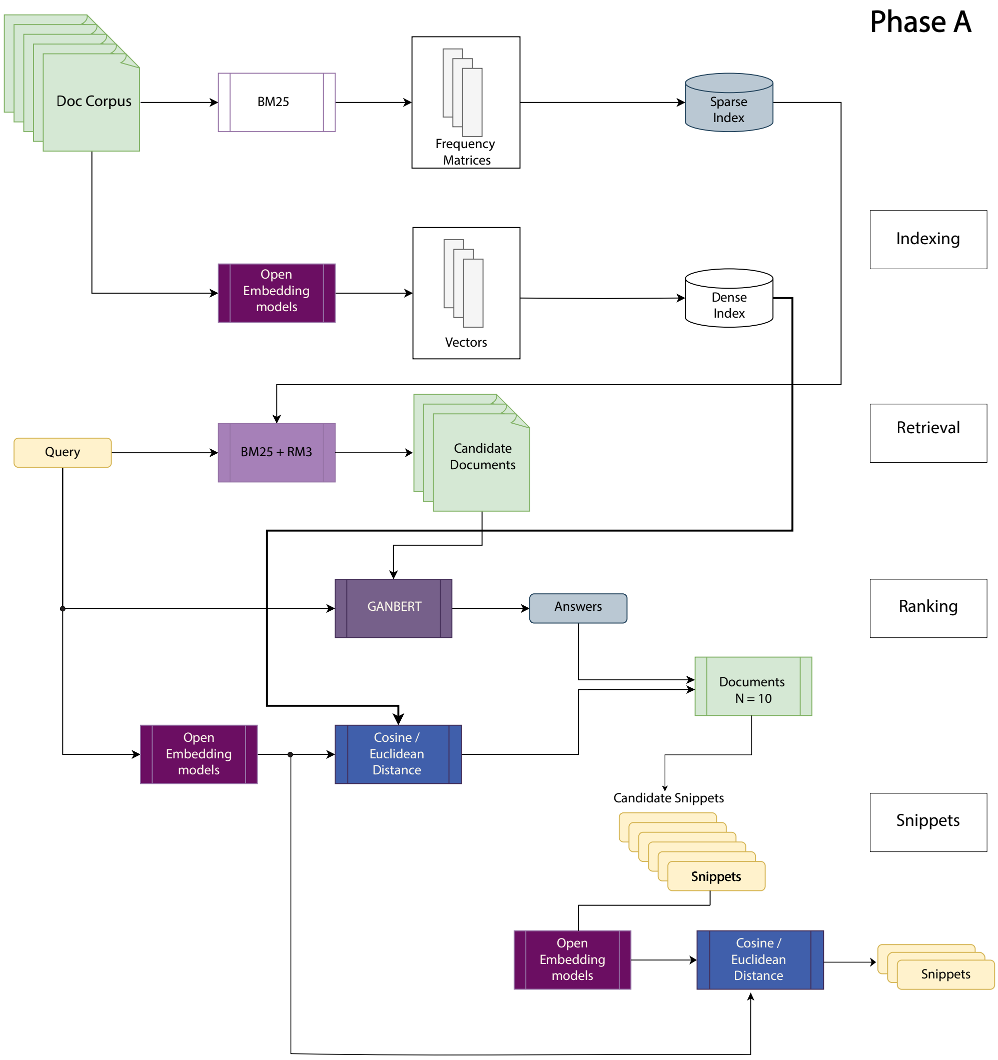

Figure 1: Processing for Phase A.
The document corpus undergoes two types of indexing: Sparse Indexing using BM25 to generate frequency matrices stored in a sparse index, and dense indexing using open embedding models to generate vectors stored in a dense index. During sparse retrieval, a query is processed with BM25 combined with RM3 to produce candidate documents, which are then ranked using GANBERT. In dense retrieval, the query is processed by the same embedding model used to generate the dense index and the resulting embedding is compared against the index using cosine or Euclidean distance and the closest documents are returned. During the Snippet Generation phase, candidate snippets are extracted from the top documents, passed through Open Embedding models and the cosine/Euclidean distance is measured against the embedding of the query, returning the snippet with the closest distance for each document.

abstracts. For the systems Fleming-2 and Fleming-3 in batch 2 we adopted a methodology where the top 20 retrieved documents were processed with a prompt asking to assess the relevance of each document for the given question using several quantized open LLMs and finally MIXTRAL[1] via GPT4All and Ollama, a software designed to streamline access to open models locally, eliminating the need for manual downloading and scripting. This additional selection step did not improve retrieval performance.

For batch 3 and 4, we evaluated dense retrieval methods that compare the embeddings of the corpus with the embedding of the question obtained with different open LLMs that have embedding dimensions $<=$ 1024, selected from the Massive Text Embedding Benchmark (MTEB) Leaderboard at https://huggingface.co/spaces/mteb/leaderboard. We utilized both Euclidean distance and cosine similarity metrics to evaluate the proximity between the documents and the query. This facilitated the identification of the ten most closely related documents, as determined by smaller distances.

[1] https://mistral.ai/news/mixtral-of-experts

Subsequently, to combine the advantage of sparse retrieval methods finding documents with less frequent words with the higher sensitivity of dense methods for semantic similarities, we use a threshold for the distances returned from the dense search to replace documents exceeding this threshold with the top documents returned from the sparse search. The details and performances of the sparse, dense and hybrid retrieval methods are shown in table 1. Hybrid sparse and dense retrieval methods have been suggested e.g. in [7]. All indices generated for the comparison in the table process Pubmed abstracts published later than November 2001. As older questions in the BioASQ12 training set also require older documents as correct answers, the performance on the complete training set is lower than on the current four batches of BioASQ12. Tested on these batches, the dense retrieval using the embedder `jamesgpt1/sf_model_e5` has the best performance for batches 1 to 3, while on batch 4 and on the training set, the hybrid combination of the dense embedder `BAAI/bge-small-en-v1.5` with our sparse BM25-rm3 retrieval version outperforms the other tested methods.

Table 1
Mean Average Precision (MAP) performance of sparse, dense and hybrid retrieval using different embeddings for the BioASQ12 training set and batches 1 to 4.

method	embed. dimension	tra. set	batch				Avg.
			1	2	3	4	
jamesgpt1/sf_model_e5	1024	0.0875	**0.1655**	**0.1483**	**0.1640**	0.2558	**0.1642**
BAAI/bge-small-en-v1.5	384	0.0879	0.1520	0.1539	0.1725	0.2292	0.1591
Hyb. BM25+bge-small dist-thresh=0.125	384	**0.1259**	0.0895	0.0924	0.1448	0.1541	0.1213
Hyb. BM25+jamesgpt dist-thresh=0.602	1024	0.1252	0.1089	0.0898	0.1633	0.2168	0.1408
Hyb. BM25+jamesgpt dist-thresh=0.8	1024	0.0978	0.1578	0.1332	0.1682	**0.2580**	0.1630
BM25-rm3	-	0.1116	0.0678	0.0633	0.0692	0.1072	0.0838

2.2. Phase A: Snippet identification

A standard approach [8] is used to identify snippets. The query and each candidate snippet are embedded by various open LLMs and the cosine similarity between the embeddings is measured. Various window sizes were explored to effectively isolate snippets from document abstracts and assess their correlation with the provided question. Our primary aim was to extract a single snippet for each document-question pair. Initially, we tested window sizes of 30 and 50, and subsequently implemented a two-step embedding process. Initially, we evaluated the question-snippet pairs' scores using an embedding model and cosine similarity for a window size of 30. Then, we selected the snippet with the highest score for the window size of 30. We further experimented with adjusting the starting and ending positions of the window within the ranges [-10, 10] and [half window size, end of abstract], respectively. Additionally, we explored segmenting the text into sentences and either preserving entire sentences or utilizing a window size of [0, 4] sentences. This approach yielded superior results in terms of precision, recall, and F-measure. The BioASQ questions are tagged with either "yes/no," "factoid," "summary," or "list" to indicate the required format for the exact answers to be created by these systems. In table 2, the candidate snippets are split according to the type of the question and the recall and F measure are reported for the tested LLMs. Neither the question type nor the used LLM have a severe effect on the measured metrics. For batch 1, 2 and 3, the model `intfloat/multilingual-e5-large-instruct` and for batch 4 the model `hkunlp/instructor-xl` was used to identify snippets. It should be noted that the model `jamesgpt/sf_large_all` had the best overall F measure, but the performance difference to the other models is very small.

Table 2
Performance metrics (in %) of various embedding models for snippet extraction on the BioASQ12 training set.

Models	Yes		Lists		Factoid		Summary		Average
	Recall	F meas.	Recall	F meas.	Recall	F meas.	Recall	F meas.	F meas.
jamesgpt1/sf_large_all[9]	**29.177**	**30.813**	31.936	**34.039**	**34.632**	**36.607**	36.077	38.843	**35.075**
BAAI/bge-large-en-v1.5[10]	29.101	30.801	**32.088**	34.018	34.624	36.567	**36.132**	**38.857**	35.060
WhereIsAI/UAE-Large-V1[11]	29.103	30.799	32.068	34.017	34.614	36.570	36.118	38.821	35.051
llmrails/ember-v1[12]	29.092	30.789	32.068	33.993	34.618	36.566	36.122	38.845	35.048
hkunlp/instructor-large[13]	29.087	30.794	32.045	34.014	34.583	36.532	36.059	38.814	35.038
hkunlp/instructor-xl[14]	29.095	30.802	32.055	34.016	34.611	36.554	36.092	38.811	35.045
avsolatorio/GIST-large-Embedding-v0[15]	29.094	30.800	32.000	33.977	34.558	36.516	36.098	38.836	35.037
thenlper/gte-large[16]	29.090	30.791	32.013	34.007	34.547	36.508	36.118	38.821	35.031
mixedbread-ai/mxbai-embed-2d-large-v1[17]	29.092	30.808	32.010	33.993	34.544	36.507	36.058	38.814	35.030
intfloat/multilingual-e5-large-instruct[18]	29.040	30.770	31.934	33.933	34.527	36.499	36.063	38.807	35.002

2.3. Phase A+ / Phase B

In Phase A+, participants will submit exact and/or ideal answers before the expert selected (gold) documents and snippets (released in Phase B) are known. Thus, each participant has to use their predictions for documents and snippets for further processing. Participants will have 24 hours to provide exact answers for various question types ("yes/no," "factoid," "list") and ideal answers in the form of paragraph-sized summaries. In figure 2 processing for both Phase A+ and Phase B is illustrated.

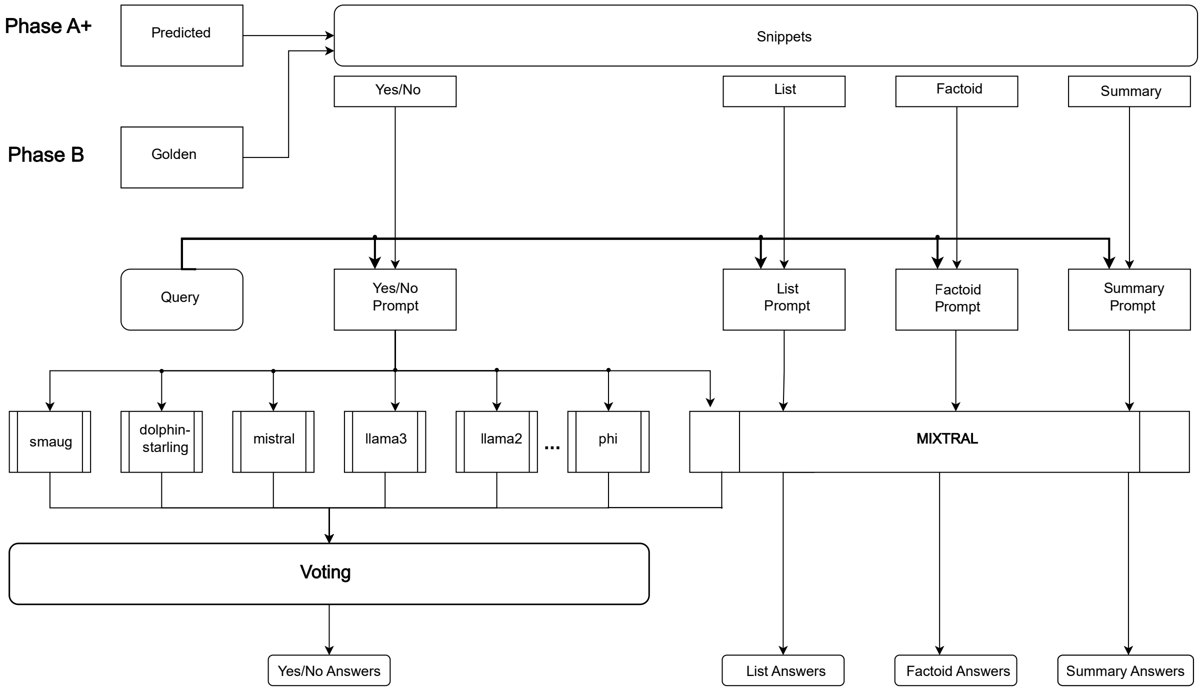

Figure 2: The image illustrates the pipeline of the two phases, the Phase A+ & Phase B, for generating exact answers using various models. The snippets used in Phase A+ are the predicted ones from Phase A, while Phase B starts with a golden standard provided by the BioASQ competition. Queries are directed into different prompt types (Yes/No, List, Factoid, Summary). For List, Factoid, and Summary type questions, the prompts are processed by MIXTRAL. For Yes/No type questions, the prompts are processed by a farm of various models (smaug-72B, phi2-2.7B, dolphin-llava-7B, dolphin-starling-7B, llama2-13b, llama3-8B, llama3-70b, mixtral-8*7B, vicuna-7B) and then a majority Voting System aggregates the responses from the models to produce final answers. Ideal answers with text summarizing the most relevant information for each question type are generated the same way Summary answers are generated.

We experimented with different prompts using various models to generate the answers using the information given in the snippets in a zero-shot approach. Ultimately, we chose the MIXTRAL model for List, Factoid, and Summary type of questions. The prompts used for each question type are as follows:

Yes/No Prompt

Given only the following **INFORMATION** and **QUESTION**, answer the **QUESTION** only with ’Yes’ or ’No’ **INFORMATION**: %s **QUESTION**: %s

List Prompt

Answer the **QUESTION** using only the **TEXT** by only returning a list of entity names, numbers, or similar short expressions that are an answer to the question and are separated by commas. Only the list should be returned. If you do not know any answer return the word EMPTY. **TEXT**: %s **QUESTION**: %s

Factoid Prompt

Answer the **QUESTION** using only the **TEXT** by only returning a list of entity names, numbers, or similar short expressions that are an answer to the question and are separated by commas,ordered by decreasing confidence. Only the list should be returned. If you do not know any answer return the word EMPTY. **TEXT**: %s **QUESTION**: %s

Summary Prompt

##**ABSTRACT**: %s ##**QUESTION**: %s ##**TASK**: Answer the **QUESTION** by returning a single paragraph sized text ideally summarizing only the most relevant information in the **ABSTRACT**.

In all these prompts, the %s after QUESTION is replaced by the actual question, and the %s after INFORMATION, TEXT or ABSTRACT is replaced with the collection of the related snippets, concatenated and separated by a single blank. The answers by the LLMs are processed by custom awk scripts that eliminate doublettes in the case of list and factoid questions and extract the difference of ’Yes’ and ’No’ for Yes/No type questions.

The performances obtained for the Yes/No questions of the training set using different open LLMs are listed in table 3. As also observed in other applications, a larger number of parameters typically also leads to a higher prediction accuracy. Llama3-70B is a noteworthy exception in our tests, performing slightly worse than Llama3-8B.

Table 3

Performance of different LLMs for Yes/No questions in a chronologic 67%/15%/18% - tra/val/tes split of the 1357 questions of this type in the BioASQ12 training set. $\%predicted$ indicates the percentage where the model answers with either 'Yes' or 'No'. MC is the Matthews correlation. Entries are sorted according to average $\%$ correct predictions (Q).

$model(-size)$		$\%predicted$	Q	$sensitivity$	$specificity$	MC
smaug-72B https://github.com/abacusai/smaug	tra	100.00	95.91	98.60	96.32	0.8730
	val	100.00	96.50	97.56	96.77	0.9260
	tes	100.00	**94.86**	96.95	95.21	**0.8868**
	average	100.00	**95.76**	97.70	96.10	**0.8953**
aya-35B https://ollama.com/library/aya	tra	100.00	95.91	98.74	96.19	0.8729
	val	100.00	96.50	**100.00**	94.62	0.9274
	tes	100.00	94.47	96.95	94.64	0.8779
	average	100.00	95.63	98.56	95.15	0.8927
yi-34B https://huggingface.co/TheBloke/Yi-34B-Chat-GGUF	tra	99.34	95.99	96.92	98.02	0.8788
	val	100.00	97.00	99.19	96.06	0.9369
	tes	100.00	93.68	92.68	**97.44**	0.8661
	average	99.78	95.56	96.26	**97.17**	0.8939
dolphin-starling-7B https://huggingface.co/bunnycore/Starling-dolphin-E26-7B	tra	99.00	95.42	97.19	97.05	0.8589
	val	99.00	**97.47**	98.36	**97.56**	**0.9465**
	tes	98.42	93.57	93.17	96.77	0.8627
	average	98.81	95.49	96.24	97.13	0.8894
mistral-7B https://huggingface.co/mistralai/Mistral-7B-v0.1	tra	98.45	94.38	97.03	95.95	0.8248
	val	99.50	96.98	100.00	95.31	0.9375
	tes	97.23	94.72	96.84	95.03	0.8844
	average	98.39	95.36	97.96	95.43	0.8822
llama3-8B https://huggingface.co/nvidia/Llama3-ChatQA-1.5-8B [19]	tra	100.00	95.35	97.07	97.07	0.8590
	val	100.00	96.00	97.56	96.00	0.9153
	tes	100.00	94.47	95.12	96.30	0.8794
	average	100.00	95.27	96.58	96.46	0.8846
llama3-70B https://huggingface.co/aaditya/Llama3-OpenBioLLM-70B	tra	100.00	**96.35**	96.93	**98.44**	**0.8921**
	val	100.00	96.00	98.37	95.28	0.9155
	tes	100.00	93.28	92.07	97.42	0.8585
	average	100.00	95.21	95.79	97.05	0.8887
openorca-13B https://huggingface.co/Open-Orca/OpenOrca-Preview1-13B	tra	99.45	95.22	**98.88**	95.29	0.8480
	val	100.00	96.50	99.19	95.31	0.9265
	tes	100.00	92.89	**98.78**	91.01	0.8449
	average	99.82	94.87	**98.95**	93.87	0.8731
mixtral-8x7B https://huggingface.co/mistralai/Mixtral-8x7B-v0.1	tra	99.00	95.42	96.91	97.32	0.8601
	val	99.50	95.48	98.36	94.49	0.9048
	tes	98.42	92.37	90.68	97.33	0.8414
	average	98.97	94.42	95.32	96.38	0.8688
dolphin-llava-7B https://huggingface.co/liuhaotian/llava-v1.5-7b	tra	98.56	92.93	95.91	95.24	0.7804
	val	100.00	94.00	99.19	91.73	0.8753
	tes	100.00	90.51	95.73	90.23	0.7896
	average	99.52	92.48	96.94	92.40	0.8151
phi3-medium-14B https://huggingface.co/ bartowski/Phi-3-medium-4k-instruct-GGUF	tra	98.23	93.58	96.46	95.52	0.7996
	val	99.50	93.47	98.36	91.60	0.8634
	tes	97.63	88.66	93.08	89.70	0.7501
	average	98.45	91.90	95.97	92.27	0.8044
llama2-13B https://huggingface.co/meta-llama/Llama-2-13b-hf	tra	94.91	88.00	91.76	93.25	0.6253
	val	99.00	84.34	96.72	81.38	0.6722
	tes	98.42	87.55	95.03	86.93	0.7235
	average	97.44	86.63	94.50	87.19	0.6737
phi2-2.7B https://huggingface.co/ microsoft/phi-2	tra	95.91	86.85	90.63	92.77	0.6031
	val	91.50	81.97	84.96	85.71	0.6193
	tes	91.70	81.03	83.67	86.01	0.5959
	average	93.04	83.28	86.42	88.16	0.6061
vicuna-7B https://huggingface.co/lmsys/vicuna-7b-v1.5	tra	93.58	82.27	92.11	86.78	0.3704
	val	94.50	73.54	94.02	71.90	0.4241
	tes	94.47	73.22	90.20	73.80	0.3864
	average	94.18	76.34	92.11	77.49	0.3936

3. Results

3.1. Document retrieval

In table 4 the performances of our document retrieval submissions for the BioASQ12 competition are listed.

Table 4
BioASQ12 document relevance prediction performance measured as mean average precision (MAP). As the document order in our submissions was by mistake scrambled for batch 1 to 3, we also include the performance with the correct order in the column MAP (Corrected). The column 'details' specifies the hyperparameters of the pipeline in figure 1.

batch	MAP	MAP (Corrected)	system	per team rank /corrected rank	details
1	**0.2067**		bioinfo-4	1	
	0.1195	**0.1886**	Fleming-2	4/2	BM25, $documents = 50$ + Ganbert
	0.1143	0.1793	Fleming-1	4/2	BM25, $documents = 20$ + Ganbert
	0.1101	0.1677	Fleming-3	4/2	BM25, $documents = 10$
2	0.2293		dmiip2024_4	1	
	0.1585	0.1580	Fleming-1	5	BM25, $documents = 50$ + Ganbert
	0.1381	0.1452	Fleming-2	8/7	BM25, $documents = 20$ + Mixtral
	0.1076	0.1076	Fleming-3	10	BM25, $documents = 50$ + Mixtral
3	**0.2549**		dmiip2024_4	1	
	0.1183	**0.2228**	Fleming-3	8/3	Dense Jamesgpt/BM25 hybrid
	0.1063	0.2007	Fleming-1	9/3	Dense bge-small-v1.5
	0.0993	**0.2123**	Fleming-5	12/3	BM25, $documents = 10$
4	**0.3930**		dmiip2024_3	1	
	0.2615		Fleming-5	6	BM25, $documents = 50$ + Ganbert
	0.2558		Fleming-1	6	Dense Jamesgpt/BM25 hybrid

3.2. Snippet prediction

In table 5 the performances of our snippets predictions for the BioASQ12 competition are listed.

Table 5
BioASQ12 snippets relevance prediction performance measured as F-Measure.

batch	F-Measure	system	per team rank	embedding model
1	**0.0638**	dmiip2024_2	1	
	0.0530	Fleming-3	2	intfloat/multilingual-e5-large-instruct
2	**0.0746**	dmiip2024	1	
	0.0282	Fleming-3	5	intfloat/multilingual-e5-large-instruct
3	**0.0940**	dmiip2024_4	1	
	0.0267	Fleming-5	3	intfloat/multilingual-e5-large-instruct
4	**0.1191**	dmiip2024_4	1	
	0.0615	Fleming-1	3	hkunlp/instructor-large

3.3. Exact answer prediction

In tables 6 and 7 the performances of our submissions for Phase A+ and Phase B of the BioASQ12 competition are listed. In batch 4 of Phase B, our submissions share the first place in average rank with the system labeled 'IISR 4th submit', which is the criterium to evaluate the overall performance for exact answers.

Table 6
Phase A+: Exact answers performance.

batch	System	Yes/No		Factoid		List		avg. rank
		acc.	rank	MRR	rank	F-Meas.	rank	
1	**UR-IW-3**	0.92	1	0.0952	10	0.4089	3	**4.6**
	Fleming-1	0.8	7	-	17	0.2079	15	13
2	**dmiip2024**	0.9615	1	0.6842	2	0.5047	2	**1.6**
	Fleming-3	0.8077	10	0.307	6	0.1708	10	8.6
3	**dmiip2024_1**	0.875	4	0.3269	2	0.3571	2	**2.6**
	Fleming-3	0.75	16	0.125	18	0.1643	18	17.3
4	**dmiip2024_1**	0.8889	1	0.3947	1	0.3219	1	**1**
	Fleming-1	0.8148	5	0.1158	14	0.1494	9	9.3

Table 7
Phase B: Exact answers performance.

batch	System	Yes/No		Factoid		List		avg. rank
		acc.	rank	MRR	rank	F-Meas.	rank	
1	**UR-IW-5**	0.96	2	0.254	12	0.579	3	**5.6**
	Fleming-1	0.8	21	0.0714	31	0.4717	15	22.3
2	**UR-IW-1**	0.9615	2	0.6842	2	0.5047	8	**4**
	Fleming-3	0.9615	4	0.4342	12	0.5243	6	7.3
3	**IISR 4th submit**	1	4	0.4231	5	0.5247	6	**5**
	Fleming-3	1	3	0.2404	28	0.5413	5	12
4	**Fleming-2**	0.963	2	0.5526	9	0.6401	3	**4.6**
	IISR 4th submit	0.9259	9	0.5965	3	0.646	2	**4.6**

4. Conclusion and Future Work

At the time of writing, manual scores to assess the free text in the ideal answers were not ready and we cannot evaluate our submissions in this category. The higher performance of our hybrid sparse and dense retrieval system are promising and might be further improved by an adaptive combination of the two results and by using an optimized subset of the embedding for distance measurement. The open LLM 'farming' approach employing a collection of (complementary) LLMs and used for the Yes/No questions can be transferred to the other question categories. With the observed rapid progress in the development of open LLMs, novel systems are easily incorporated into our pipelines.

Acknowledgments

We thank the anonymous reviewers for their comments. GPU computations for GANBERT were offered by HYPATIA, the Cloud infrastructure of the Greek ELIXIR node. Other GPU computations were performed on two servers obtained within the project ID 16624, "Creation - Expansion - Upgrading of the Infrastructures of research centers supervised by the General Secretariat for Research and Innovation (GSRI)" with code MIS 5161770. This project was funded under the "National Recovery and Resilience Plan Greece 2.0".

References

[1] X. Zhu, J. Li, Y. Liu, C. Ma, W. Wang, A survey on model compression for large language models, 2023. arXiv:2308.07633

[2] Ollama, github, 2024. URL: https://github.com/ollama/ollama

[3] Y. Anand, Z. Nussbaum, B. Duderstadt, B. Schmidt, A. Mulyar, Gpt4all: Training an assistant-style chatbot with large scale data distillation from gpt-3.5-turbo, https://github.com/nomic-ai/gpt4all, 2023

[4] A. Spirling, Why open-source generative ai models are an ethical way forward for science, Nature 616 (2023) 413–413

[5] A. Krithara, A. Nentidis, K. Bougiatiotis, G. Paliouras, BioASQ-QA: A manually curated corpus for Biomedical Question Answering, Scientific Data 10 (2023) 170

[6] D. N. Panou, M. Reczko, Semi-supervised training for biomedical question answering, in: Conference and Labs of the Evaluation Forum, 2023. URL: https://doi.org/10.5281/zenodo.8410284

[7] P. Mandikal, R. Mooney, Sparse meets dense: A hybrid approach to enhance scientific document retrieval, in: The 4th CEUR Workshop on Scientific Document Understanding, AAAI, 2024

[8] M. Feng, B. Xiang, M. R. Glass, L. Wang, B. Zhou, Applying deep learning to answer selection: A study and an open task, CoRR abs/1508.01585 (2015). URL: http://arxiv.org/abs/1508.01585. arXiv:1508.01585

[9] jamesgpt1/sf_large_all, Huggingface, 2022. URL: https://huggingface.co/jamesgpt1/sf_large_all

[10] BAAI/bge-large-en-v1.5, Huggingface, 2023. URL: https://huggingface.co/BAAI/bge-large-en-v1.5

[11] WhereIsAI/UAE-Large-V1, Huggingface, 2023. URL: https://huggingface.co/WhereIsAI/UAE-Large-V1

[12] ember-v1 (Revision 8119998) , Huggingface, 2023. URL: https://huggingface.co/llmrails/ember-v1. doi:10.57967/hf/1241

[13] instructor-large, Huggingface, 2023. URL: https://huggingface.co/hkunlp/instructor-large

[14] instructor-xl, Huggingface, 2023. URL: https://huggingface.co/hkunlp/instructor-xl

[15] GIST-large-Embedding-v0, Huggingface, 2024. URL: https://huggingface.co/avsolatorio/GIST-large-Embedding-v0

[16] gte-large, Huggingface, 2023. URL: https://huggingface.co/thenlper/gte-large

[17] mxbai-embed, Huggingface, 2024. URL: https://huggingface.co/mixedbread-ai/mxbai-embed-2d-large-v1

[18] multilingual-e5, Huggingface, 2024. URL: https://huggingface.co/intfloat/multilingual-e5-large-instruct

[19] AI@Meta, Llama 3 model card (2024). URL: https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md