HU-WBI at BioASQ12B Phase A: Exploring Rank Fusion of Dense Retrievers and Re-rankers

Oğuz Şerbetçi[1,†], Xing David Wang[1,†] and Ulf Leser[1,*]

[1]Humboldt-Universität zu Berlin, Rudower Chaussee 25, 12489 Berlin, Germany

Abstract

This paper describes the participation of the HU-WBI team at the BioASQ 12B Phase A shared task for retrieving the top 10 most relevant PubMed abstracts for a given natural language query. Our approach focuses on using BM25 for sparse first-stage retrieval and combining a variety of dense, neural models via rank fusion for second-stage re-ranking. We train our neural models on question-answer pairs from previous editions of the task. The neural re-rankers consist of (1) two distinct cross-encoders, one trained on a pairwise loss and one trained on both token-level and document-level features, as well as (2) a large language model, generating synthetic queries from documents returned by the first-stage retrieval and comparing their similarities to the original test query. Our systems achieved competitive results placing third in the second official evaluation batch and eleventh both in the third and forth batch which corresponds to the second best team in the second batch and third best team in the other ones.

Keywords

Information Retrieval, BM25, Dense Re-Rankers, Rank Fusion, BioASQ

1. Introduction

BioASQ [1, 2] is a long-running challenge dealing with indexing, classification and question answering over a corpus of biomedical academic publications. This year is the twelfth edition of the challenge [3] and we here present the contribution of the HU-WBI group to the task 12B of BioASQ [4]. Task 12B is about returning a ranked list of PubMed abstracts answering a given biomedical question formulated in natural language. Further parts of the task are then concerned with identifying relevant text spans in the retrieved documents and crafting natural language answers to the given questions, we here focus on the information retrieval and document ranking problem described above.

Information retrieval (IR) is a key field to systematically handle the evergrowing amount of information and literature available each day. One of the main task in IR is to return a ranked list of documents over a given document corpus sorted by their relevance according to a given user query. Classic strong approaches for solving the task are keyword matching and vector-based methods like BM25 [5]. Recently, IR models based on neural networks have gained popularity with the advent of Transformers-based architectures [6]. Neural or dense retrievers like DPR [7] encode both query and documents into d-dimensional vectors where $d << |V|$, V being the vocabulary of the underlying corpus. Similarity measures of the query and vector embeddings are computed and the most similar documents are ranked on top.

In the previous iterations of the task, there has been widespread experimentation with dense retriever architectures on different kinds of training data: [8] train dense retrievers on PubMed search click logs whereas approaches like BitUA [9] tinker with BM25 negatives. All these approaches do make use of the manually annotated query-article relevance pairs from previous editions of the BioASQ challenge.

Our approach focuses on using BM25 for sparse first-stage retrieval and combining a variety of dense, neural models via rank fusion for second-stage re-ranking. We explored a plethora of dense retrievers, both pre-trained models and models fine-tuned by us and experimented which combination

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

[*]Corresponding author.

[†]These authors contributed equally.

oguz.serbetci@hu-berlin.de (O. Şerbetçi); xing.wang@informatik.hu-berlin.de (X. D. Wang); leser@informatik.hu-berlin.de (U. Leser)

of models performed the best in aggregation using reciprocal rank fusion [10]. The neural re-rankers we developed consist of (1) two distinct cross-encoders, one trained on a pairwise loss and one trained on both token-level and document-level features, as well as (2) a large language model, generating synthetic queries from documents returned by the first-stage retrieval and comparing their similarities to the original test query.

Our systems achieved competitive results placing third in the second official test batch reaching a MAP score of 22.23 percent compared to the 22.93 percent MAP score of the best model and eleventh both in the third and final forth batch (6 percentage points (pp) and 4 pp difference to the best competitor). These results place us overall as the second best team in the second batch and third best team in the other ones as each team was allowed to submit up to five systems per batch.

The rest of the paper is structured as follows: In Section 2, we go over the models developed and used for our experiments. In Section 3, we present the results our models achieved on the official test batches and in Section 4, we present ablation studies on the models we have considered using. Finally, we sum up the findings of our experiments in Section 5.

2. Methodology

We here present the pre-trained models we used for initial document retrieval as well as subsequent re-ranking and the models we fine-tuned ourselves. For the final model predictions, we implemented an ensemble approach by applying reciprocal rank fusion [11] to consolidate the outputs to a single result.

2.1. Retriever

We tested both sparse and dense approaches for the initial candidate retrieval: BM25 [5], MedCPT [12] and E5 [13] . For BM25, we set the hyper-parameters $b = 0.75$ and $k = 1.2$ as found to work best for BioASQ training data by [9] and store the indexed documents in ElasticSearch[1]. For MedCPT and E5, we store and retrieve the dense document embeddings via FAISS [14]. Document retrieval is performed over the PubMed 2024 baseline and returns the top $k \in \{100, 1000\}$ documents as results.

2.2. Re-ranker

After retrieving the top k documents from the retriever, the document re-rankers assign new scores to the documents via vector similarity or cross-encoder attention. The new updated ranking based on the re-ranker scores is then returned as the output. We now present the pre-trained and fine-tuned approaches we have considered for our experiments.

Pre-trained Dense Rerankers Regarding dense-retrieval, we experimented with the MedCPT retriever (MedCPT), the MedCPT cross-encoder (MedCPT CE) and the E5 cross-encoder (E5 CE). We also fine-tuned a version of the MedCPT cross-encoder (MedCPT CE v2) on the BioASQ training data continuing from their publicly available checkpoint.

Cross-encoder with sequence-level loss and token-level loss (SeqTok) For our first fine-tuned model, we train a cross-encoder on both sequence-level and token-level features from the BioASQ training data. Our aim is to make use of the annotation of the exact text spans relevant in a retrieved document which are of importance for the subsequent snippet identification task of Phase A. We calculate two separate losses for each document: The sequence-level loss is a cross entropy loss based on the binary relevance of a given query-document pair, i.e. the document was marked as relevant to the question. The token-level loss is a token level cross entropy loss for whether the token answers the query, i.e. token is part of the answer span as provided in the training data. We use IOB-notation to denote the labels for each tokens.

[1] https://www.elastic.co/

During model training, we combine both losses via a weighted sum

$$\text{Loss} = \lambda_{Seq} \cdot \text{SeqLoss} + \lambda_{Tok} \cdot \text{TokLoss},$$

where we set $\lambda_{Seq} = \lambda_{Tok} = 1$. As additional training signal, we make use of hard negatives provided by BM25 [9].

When we calculate re-ranking scores we calculate two independent document relevance scores, one for sequence level and one for token level features. The token score is the average logit score of all the tokens that have been predicted as part of the answer span (Seq). The sequence score is just the logit score of the whole document given by the BERT $[CLS]$ token (Tok). We include both scores as separate models to be considered for the rank fusion.

Cross-encoder with pairwise loss (Pair) For our second fine-tuned dense retriever, we are training a neural network using pairs of documents given an input query. Instead of assigning a binary relevance label relevant/not relevant to the input query, we are ranking the relevance of an input pair (doc_1, doc_2) relative to each other, i.e., either doc_1 is more relevant than doc_2 or vice versa.

Let doc_1 be a relevant document and doc_2 be an irrelevant document for a given query q. We create training pairs (doc_1, doc_2) in two variants as follows:

- Pair v1: For each relevant doc_1, we randomly sample one BM25 negatives from the same query.
- Pair v2: For each relevant doc_1, we randomly sample two BM25 negatives from the same query and one relevant document from another query as our negative documents doc_2s. Then for each query q, we sample negative documents doc_2s up to the number $max_{neg} = 20$ and pair them with at least one relevant document doc_1 for the query q. This ensures that we see a minimum amount of max_{neg} training pairs for each query instead of potentially only having one example as in Pair v1.

To implement the pairwise loss, we then use a sigmoid function to force all document logits to be in the range of $[0, 1]$. Afterwards, we use the `torch.nn.MarginRankingLoss`[2] loss function with a margin of 0.1 to ensure that the relevant document doc_1 receives a higher score than the irrelevant doc_2.

Model parameters of all our custom re-rankers are initialized based on the BioLink-BERT base checkpoint [15].

GPT Re-ranker with query generation (GPT and GPT-CoT) For each document retrieved, we use GPT-3.5 [16] to generate queries that the title and the abstract answers. We use one model with chain-of-thought prompting [17] (GPT-CoT) and one without (GPT), asking the model to answer the query and to finally discriminate whether the query is answered by the document. We show an example with the prompt in Figure 1. After generating the questions for each document, we re-rank the documents using the average of the top two similarities between generated queries and the query from the task. To calculate the similarities we use the MedCPT-Query-Encoder [12].

2.3. Reciprocal Rank Fusion

In order to aggregate the predictions of our various models we use reciprocal rank fusion (RRF) [10]. The final ranking score of a given document d in the whole corpus D is then calculated over its individuals ranks $r(d)$ in the set of all rankings R we take into account by

$$RRF(d \in D) = \sum_{r \in R} \frac{1}{k + r(d)} \ ,$$

where the constant k helps reducing the influence of outlier rankings significantly differing from the other one and is determined empirically on our training data.

In our preliminary studies, we have also considered other rank fusion methods like Min-Max Norm, combsum and combmnz with z/zmuv normalization [18, 19] that did not perform as well as RRF however.

[2] https://pytorch.org/docs/stable/generated/torch.nn.MarginRankingLoss.html

Please write at least 10 different search queries that a clinician might need answers to that are answered in the abstract of the peer-reviewed academic paper provided. For each search query, provide a short and concise answer using only the abstract provided. After you have answered the search query, make a final correction as to whether the abstract provided actually answers the query using the following labels: "ANSWERS", "DOES NOT ANSWER". Be careful not to refer directly to the abstract or study provided in the search queries. We are only interested in general search queries that a clinician would look for answers to before seeing the abstract and the details provided. The search queries are answered by the abstract coincidentally.
Use following structure:
QUESTION: [QUESTION OR QUERY]
ANSWER: [SHORT BUT SOUND ANSWER]
CORRECTION: [ANSWERS|DOES NOT ANSWER]
For example, consider following abstracts and corresponding question and answer pairs:
ABSTRACT
# In vivo topical gene therapy for recessive dystrophic epidermolysis bullosa: a phase 1 and 2 trial
Recessive dystrophic epidermolysis bullosa (RDEB) is a lifelong genodermatosis associated with blistering, wounding, and scarring caused by mutations in COL7A1, the gene encoding the anchoring fibril component, collagen VII (C7). Here, we evaluated beremagene geperpavec (B-VEC), an engineered, non-replicating COL7A1 containing herpes simplex virus type 1 (HSV-1) vector, to treat RDEB skin. B-VEC restored C7 expression in RDEB keratinocytes, fibroblasts, RDEB mice and human RDEB xenografts. Subsequently, a randomized, placebo-controlled, phase 1 and 2 clinical trial (NCT03536143) evaluated matched wounds from nine RDEB patients receiving topical B-VEC or placebo repeatedly over 12 weeks. No grade 2 or above B-VEC-related adverse events or vector shedding or tissue-bound skin immunoreactants were noted. HSV-1 and C7 antibodies sometimes presented at baseline or increased after B-VEC treatment without an apparent impact on safety or efficacy. Primary and secondary objectives of C7 expression, anchoring fibril assembly, wound surface area reduction, duration of wound closure, and time to wound closure following B-VEC treatment were met. A patient-reported pain-severity secondary outcome was not assessed given the small proportion of wounds treated. A global assessment secondary endpoint was not pursued due to redundancy with regard to other endpoints. These studies show that B-VEC is an easily administered, safely tolerated, topical molecular corrective therapy promoting wound healing in patients with RDEB.
QUESTION: Beremagene Geperpavec is tested for which disease?
ANSWER: Beremagene Geperpavec was tested for recessive dystrophic epidermolysis bullosa.
DECISION: ANSWERS
ABSTRACT
{. . . EXAMPLE #2 . . . }
Here is the provided abstract for an academic paper and the question:
ABSTRACT

Figure 1: Prompt for the GPT re-ranker. Few-shot example is shown in color (blue).

3. Results

In this section, we present the final systems that we submitted for each batch and report the performance they achieved. An overview of the results can be found in Table 1. We experimented with the output of single models only but found out that an aggregation of model rankings with RRF always worked best. All the systems submitted using RRF include the following model rankings: BM25, MedCPT CE and SeqTok. Additionally each system used additional rankings from different models in their RRF. We present all systems used in our test submission in Table 1 whereas the model abbreviatons correspond to the model names introduced above.

Each team could submit up to five systems for each test batch. The results for all our systems and the best overall competitor are shown in Table 1. In the first batch, we only submitted our base RRF. In the individual system ranking, it reached the 20th place with a MAP score of 10.94 around 10 percentage points (pp) behind the top competitor system. However, for this batch we had not updated the document corpus to the PubMed 2024 baseline so the model operated on the 2023 baseline. Two third of the relevant documents in first batch originated from the 2024 PubMed baseline, making our model underperform. We fixed this for all model rankings in the second batch, except for the GPT re-ranker, which negatively affected the results for the respective systems 2-5 of that batch.

System using RRF	Batch 1			Batch 2			Batch 3			Batch 4		
	F1	MAP	Rank	F1	MAP	Rank	F1	MAP	Rank	F1	MAP	Rank
BM25, MedCPT CE, SeqTok	07.12	**10.94**	**20**	**13.97**	**22.23**	**3**	11.80	16.55	17	13.86	33.88	13
+ GPT		–		12.00	15.53	18		–			–	
+ GPT, MedCPT		–		10.89	13.57	23		–			–	
+ GPT, Pair v1		–		13.61	18.78	12		–			–	
+ GPT-CoT, MedCPT, Pair v1		–		12.05	16.99	14		–			–	
+ Pair v2		–			–		12.85	18.63	13	**14.06**	32.74	15
+ GPT-CoT, Pair v2		–			–		**12.87**	**19.28**	**11**	13.91	34.87	12
+ GPT-CoT, Pair v2, MedCPT, E5		–			–		11.16	14.18	19		–	
+ GPT-CoT, Pair v2, MedCPT, E5, E5 RR		–			–		11.22	15.26	18		–	
+ Pair v2, MedCPT CE v2		–			–			–		13.89	33.39	14
+ Pair v2, MedCPT CE v2, GPT-CoT		–			–			–		13.88	**35.06**	**11**
Top competitor	14.85	20.67	1	14.28	22.93	1	13.09	25.49	1	16.09	39.30	1
Median	9.51	10.73	20.5	10.29	11.51	26	9.09	12.50	29.5	11.02	17.69	24

Table 1
Overview of retriever and re-ranker models used for the reciprocal rank fusion (RRF) and their corresponding results for BioASQ-task 12B on the official evaluation batches for phase A. Each team was allowed to submit up to five systems for each batch.

In the second batch, our base RRF model reached the third place from all submitted systems with a MAP of 22.23, around 0.7 pp behind the top competitor and second place considering the team rankings showing a strong result for the proposed Seq-Tok training objective.

In the third and forth batches, our best models ranked the eleventh out of all submitted systems and third in the team rankings. Here, the pairwise cross-encoder, the fine-tuning of the MedCPT cross-encoder and the GPT re-ranker proved to mostly improve the base system, where as including E5 dense embeddings for initial retrieval (E5) and re-ranking of the BM25 results (E5 RR) performed significantly worse.

4. Discussion

4.1. Individual model performance

To determine which models to include in the RRF, we examine the results of our individual models without RRF on our development set, test batch 2 of the BioASQ edition 11B, in Table 2. We observe that dense retrievers like MedCPT Ret and E5 Ret perform much worse for initial retrieval than BM25. That

System Individual	Batch 1	
	F1	MAP
BM25	16.73	25.36
SeqTok (k=100)	18.75	22.96
SeqTok (k=1000)	14.34	20.02
MedCPT Ret	7.22	10.17
MedCPT CE v2 (k=100)	20.25	29.12
MedCPT CE v2 (k=1000)	20.29	28.75
GPT	18.75	22.96
Pair v2 (k=100)	**21.98**	**34.53**
Pair v2 (k=1000)	19.77	32.98
E5 CE (k=100)	19.25	19.93
E5 CE (k=1000)	16.93	17.10
E5 Ret	13.61	13.21

Table 2
Evaluation results on the test batch 2 of BioASQ-task 11B for our explored models. k=100 denotes that the top-100 BM25 documents were fed into the re-ranker and k=1000 denotes the top-1000 documents respectively.

is why all our subsequent re-rankers use BM25 for the initial retrieval step instead of the output of the dense retrievers. Regarding the top-k documents to be extracted by BM25, we see that only re-ranking the top 100 documents instead of 1000 performs consistently better on our development set, so we only re-rank the top 100 BM25 documents in our submitted systems.

Regarding individual re-ranker performance, we find that most of our re-rankers do not surpass BM25 performance alone. However, using RRF, their inclusion improves the performance of the overall ensemble. Only the fine-tuned MedCPT CE and the Pairwise v2 model consistently outperformed BM25. The Pairwise v2 performed better than the first pairwise model in our preliminary experiments therefore we omitted the first pairwise model from the batches three and four. However, it showed that the addition of other models like GPT-CoT still benefitted the overall RRF results.

4.2. Model selection

Model selection of the rankings to be included in each RRF was conducted semi-automatically. For each combination of initial retriever models and subsequent re-rankers, we calculated their RRF score on our development sets and then decided which combination to submit.

In our experiments, we found that including the BM25 results in the RRF along with the re-rankings separately is essential for good ensemble performance. In addition, both sequence and token scores seemed to be important, so we always use both as separate models in the fusion. They added two pp in our evaluations with a previous test batch. The MedCPT cross-encoder also consistently improved the rank fusion results, so we included it in our base RRF system. However, including a version of the MedCPT cross-encoder fine-tuned to the BioASQ training data only showed a 0.2 pp improvement.

4.3. Using LLM-generated queries

Th query matching approach with GPT represents an interpretable, reviewable approach which is expandable by humans and needs to be only computed once. One interesting expansion on this idea for future work is to actually store the queries that lead to a paper as metadata, which can be utilized as we just did. Since we use it just to re-rank, the computation cost is minimal if the associated queries are already available as metadata on the retrieved documents.

5. Conclusion

In this paper, we presented our contribution to the twelfth edition of the BioASQ challenge Task B Phase A for document retrieval. We showed how to combine simple baselines like BM25, already pre-trained neural models and own fine-tuned models via reciprocal rank fusion to achieve competitive results.

Acknowledgments

Xing David Wang is supported by the DFG as part of the Research Training Group CompCancer (No. RTG2424). Oğuz Şerbetçi is supported by the project IntSimOnko funded by the Gemeinsame Bundesausschuss (G-BA, 01VSF22041).

References

[1] G. Tsatsaronis, G. Balikas, P. Malakasiotis, I. Partalas, M. Zschunke, M. R. Alvers, D. Weissenborn, A. Krithara, S. Petridis, D. Polychronopoulos, Y. Almirantis, J. Pavlopoulos, N. Baskiotis, P. Gallinari, T. Artiéres, A.-C. N. Ngomo, N. Heino, E. Gaussier, L. Barrio-Alvers, M. Schroeder, I. Androutsopoulos, G. Paliouras, An overview of the BIOASQ large-scale biomedical semantic indexing and question answering competition, BMC Bioinformatics 16 (2015) 138. URL: https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-015-0564-6. doi:10.1186/s12859-015-0564-6.

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

[3] A. Nentidis, G. Katsimpras, A. Krithara, S. Lima-López, E. Farré-Maduell, M. Krallinger, N. Loukachevitch, V. Davydova, E. Tutubalina, G. Paliouras, Overview of BioASQ 2024: The twelfth BioASQ challenge on Large-Scale Biomedical Semantic Indexing and Question Answering, in: L. Goeuriot, P. Mulhem, G. Quénot, D. Schwab, L. Soulier, G. Maria Di Nunzio, P. Galuščáková, A. García Seco de Herrera, G. Faggioli, N. Ferro (Eds.), Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Fifteenth International Conference of the CLEF Association (CLEF 2024), 2024.

[4] A. Nentidis, G. Katsimpras, A. Krithara, G. Paliouras, Overview of BioASQ Tasks 12b and Synergy12 in CLEF2024, in: G. Faggioli, N. Ferro, P. Galuščáková, A. García Seco de Herrera (Eds.), Working Notes of CLEF 2024 - Conference and Labs of the Evaluation Forum, 2024.

[5] S. Robertson, H. Zaragoza, The Probabilistic Relevance Framework: BM25 and Beyond, Foundations and Trends® in Information Retrieval 3 (2009) 333–389. URL: https://www.nowpublishers.com/article/Details/INR-019. doi:10.1561/1500000019, publisher: Now Publishers, Inc.

[6] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Advances in neural information processing systems 30 (2017).

[7] V. Karpukhin, B. Oğuz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, W.-t. Yih, Dense Passage Retrieval for Open-Domain Question Answering, 2020. URL: http://arxiv.org/abs/2004.04906, arXiv:2004.04906 [cs].

[8] A. Shin, Q. Jin, Z. Lu, Multi-stage Literature Retrieval System Trained by PubMed Search Logs for Biomedical Question Answering, Thessaloniki, Greece, 2023.

[9] T. Almeida, R. A. A. Jonker, R. Poudel, J. M. Silva, S. Matos, BIT.UA at BioASQ 11B: Two-Stage IR with Synthetic Training and Zero-Shot Answer Generation, Thessaloniki, Greece, 2023.

[10] G. V. Cormack, C. L. A. Clarke, S. Buettcher, Reciprocal rank fusion outperforms condorcet and individual rank learning methods, in: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval, ACM, Boston MA USA, 2009, pp. 758–759. URL: https://dl.acm.org/doi/10.1145/1571941.1572114. doi:10.1145/1571941.1572114.

[11] G. V. Cormack, C. L. A. Clarke, S. Büttcher, Reciprocal rank fusion outperforms condorcet and individual rank learning methods, in: SIGIR, ACM, 2009, pp. 758–759.

[12] Q. Jin, W. Kim, Q. Chen, D. C. Comeau, L. Yeganova, W. J. Wilbur, Z. Lu, MedCPT: Contrastive Pre-trained Transformers with large-scale PubMed search logs for zero-shot biomedical information retrieval, Bioinformatics 39 (2023) btad651. URL: https://academic.oup.com/bioinformatics/article/doi/10.1093/bioinformatics/btad651/7335842. doi:10.1093/bioinformatics/btad651.

[13] L. Wang, N. Yang, X. Huang, B. Jiao, L. Yang, D. Jiang, R. Majumder, F. Wei, Text Embeddings by Weakly-Supervised Contrastive Pre-training, 2024. arXiv:2212.03533.

[14] J. Johnson, M. Douze, H. Jégou, Billion-scale similarity search with GPUs, 2017. URL: http://arxiv.org/abs/1702.08734, arXiv:1702.08734 [cs].

[15] M. Yasunaga, J. Leskovec, P. Liang, Linkbert: Pretraining language models with document links, arXiv preprint arXiv:2203.15827 (2022).

[16] OpenAI, Gpt-4, 2023. URL: https://openai.com, accessed: 2023-05-27.

[17] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. Chi, Q. Le, D. Zhou, Chain-of-Thought Prompting Elicits Reasoning in Large Language Models, 2023. arXiv:2201.11903.

[18] E. A. Fox, J. A. Shaw, Combination of multiple searches, in: TREC, volume 500-215 of *NIST Special Publication*, National Institute of Standards and Technology (NIST), 1993, pp. 243–252.

[19] J. A. Aslam, M. H. Montague, Models for metasearch, in: W. B. Croft, D. J. Harper, D. H. Kraft, J. Zobel (Eds.), SIGIR 2001: Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, September 9-13, 2001, New Orleans, Louisiana, USA, ACM, 2001, pp. 275–284. URL: https://doi.org/10.1145/383952.384007. doi:10.1145/383952.384007.