BIT.UA at BioASQ 8: Lightweight neural document ranking with zero-shot snippet retrieval

Tiago Almeida[1][0000−0002−4258−3350] and Sérgio Matos[2][0000−0003−1941−3983]

[1] University of Aveiro, IEETA
tiagomeloalmeida@ua.pt
[2] University of Aveiro, DETI/IEETA
aleixomatos@ua.pt

Abstract. This paper presents the participation of the University of Aveiro Biomedical Informatics and Techologies (BIT) group in the eighth edition of the BioASQ challenge for the document and snippet retrieval tasks. Our system follows a two-stage retrieval pipeline, where a group of candidate documents is retrieved based on BM25 and reranked by a lightweight interaction-based model that uses the context of exact matches to refine the ranking. Additionally, we also show a zero-shot setup for snippet retrieval based on the architecture of our interaction based model. Our system achieved competitive results scoring at the top or close to the top for all the batches, with MAP values ranging from 33.98% to 48.42% in the document retrieval task, although being less effective on snippet retrieval.

1 Introduction

Last year (2019), PubMed indexed almost one and a half million articles, which is equivalent to almost three new articles indexed every minute.[3] As a consequence, it is continually more time consuming for a biomedical expert to successfully search this unprecedented amount of available information. So, given the current artificial intelligence (AI) revolution, it is clear that such systems can be exploited to aid with this searching task and ultimately help researchers to rapidly find consistent information about their research topic.

The BioASQ [25] challenge provides annual competitions on document classification, retrieval and question-answering applied to the biomedical domain. These competitions are notable for continuously pushing the development of intelligent systems capable of tackling the previously enunciated problem.

This paper describes the participation of the Biomedical Informatics and Techologies (BIT) group in the eighth edition of the BioASQ challenge, specifically in the document and snippet retrieval tasks of BioASQ 8b Phase A. More

[3] https://www.nlm.nih.gov/bsd/medline_pubmed_production_stats.html

precisely, the objective is to retrieve, from the PubMed/MEDLINE collection, the most relevant articles and documents snippets for a given biomedical question written in English.

Our approach is an evolution of a previous work [1] that develops and applies a two-stage retrieval system to the biomedical searching problem. More concretely, it uses the Elasticsearch engine with BM25 weighing scheme to reduce the search space and then applies a neural ranking model in this smaller space to produce a final ranking order. In this work, we focus on improving the neural ranking model by simplifying the previous architecture and by adopting some modifications based on new assumptions. Furthermore, one of the enhancements enables us to directly extract the importance that the model assigns to each document passage without the need of training the model on this specific task, which makes it a zero-shot learner. In other words, the neural ranking model is only trained to predict the relevance of an entire document for a given question.

The final neural ranking model, presented here, has only 620 trainable parameters, making it an extremely lightweight approach when compared to transformer based models which are the current state-of the-art for NLP related tasks.

Our submissions achieved the top and close to the top positions for every document retrieval batch and also showed interesting results for all of the snippet retrieval batches. These are insightful results, that show the potential of our lightweight neural ranking model and demonstrate a potential zero-shot learning setup that can be easily extended to a snippet retrieval task. The full network configuration is publicly available at `https://github.com/bioinformatics-ua/BioASQ_CLEF`, together with code for replicating the results presented in this paper.

2 Background

In classical IR methods, a ranking function is parameterized by a set of handcrafted features to score the relevance of a query-document pair. Nowadays, recent works on the application of deep learning methods to IR, and question-answering in particular, have shown very good results. In this new perspective, commonly referred to as neural IR, the ranking function is approximated by a neural network that learns its parameters from large data collections. In the literature, neural models are usually subdivided into two categories based on their architecture. In one category, the models learn a semantic representation of the texts and use a similarity measure to score each query-document pair. Examples in this **representation-based** category include the Deep Structured Semantic Model (DSSM) [7] and the Convolutional Latent Semantic Model (CLSM) [23]. On the other hand, in **interaction-based** approaches, query and document matching signals are captured and then fed to a neural network that produces a ranking score based on the extracted matching patterns over these signals. Examples include the Deep Relevance Matching Model (DRMM) [6] and DeepRank [17].

Since 2018, transformer-based architectures, like GPT [20] and BERT [5], have been revolutionizing the NLP field, showing outstanding performance in the majority of tasks. These are large models that explore transfer learning techniques by leveraging the knowledge learned on enormous text collection. Following this trend, some promising works show positive results when applying this type of models for the ad-hoc retrieval task [3, 4, 12]. However, despite the indisputable performance presented by these architectures, it is also undeniable that the dimension of such models is a major drawback, that makes it almost impossible for some institutions to deploy or even use these models given their demanding computational costs.

Endorsed by the annual BioASQ competition, biomedical IR became a challenge with a wide range of different solutions, either based on traditional IR, neural IR, or a combination of both. For example, the system proposed by the USTB_PRIR team [9] uses query enrichment strategies, Sequential Dependence Models (SDM) and pseudo-relevance feedback to obtain a list of relevant documents. This traditional approach scored in the top positions between the third and fifth edition, which highlights early challenges of applying neural models to this task. The system proposed by the AUEB team [2] was the first to show some evidence that deep neural models are capable of outscoring the traditional models by scoring at the top positions in the sixth and seventh editions. Their system uses a variation of DRMM [14] or BERT [5] to rerank the top 100 documents recovered using the BM25 scheme [22]. The importance of the reranking step is evidenced by comparing the results to another work that submitted the top documents directly retrieved based on BM25 [13].

3 Base architecture

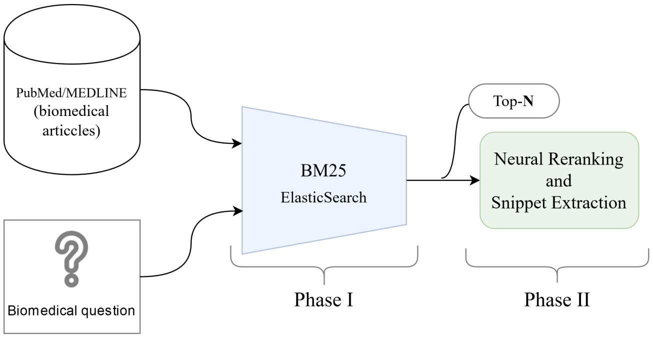
PubMed/MEDLINE (biomedical articcles)
Biomedical question
BM25
ElasticSearch
Top-N
Neural Reranking and Snippet Extraction
Phase I
Phase II

Fig. 1. Overview of our two-stage retrieval system

3.1 Phase-I

The main objective of this phase is to reduce the enormous searching space by only selecting the most **top-N** potential relevant documents for a given question. Given the large dimension of the article collection (approximately 30 million scientific articles), it is important to consider an efficient solution capable of handling this growing collection. With this in mind, we decided to rely on ElasticSearch (ES) with the BM25 weighting scheme described in Equation 1. As mentioned before, only the exact matching signals are considered during this retrieval phase.

$$\begin{aligned} IDF(q_i) &= ln(1 + \frac{C - f(q_i) + 0.5}{f(q_i) + 0.5}), \\ weight(q_i, D) &= IDF(q_i) \times \frac{f(q_i, D) \times (k_1 + 1)}{f(q_i, D) + k_1(1 - b + b \times \frac{|D|}{avg_l(D)})}. \end{aligned} \tag{1}$$

Equation 1 presents the weighting scheme of each query term q_i with respect to a document D, where C corresponds to the total number of documents in the collection, $f(q_i)$ represents the number of documents that contain the term q_i, $f(q_i, D)$ represents the frequency of term q_i in document D, $|D|$ corresponds to the total number of terms in document D, i.e., its length, $avg_t(D)$ represents the average length of the documents in the collection, and k_1, b are hyperparameters that should be finetuned for the collection.

At last, given the weight of each query term with respect to a document, $weight(q_i, D)$, the final query-document score is computed by taking a summation of each query term weight, as shown in Equation 2.

$$score(Q, D) = \sum_{q_i \in Q} weight(q_i, D). \tag{2}$$

3.2 Phase-II

The second phase has the objective of reranking the previously retrieved **top-N** documents by taking into consideration additional matching signals to produce the final ranking order. The rationale here is that the previous step only considers the exact matching signals, i.e., only the words that appear both on the query and the document are taken into account and weighted to produce the phase-I ranking. So a more powerful neural solution may be able to learn how to better explore the context where these exact matches occur.

More precisely, our model is inspired by the DeepRank [17] architecture and represents a direct enhancement of our previous work [1], with the following major differences:

- Passages no longer follow the **query-centric** assumption and now correspond directly to entire document sentences;
- The detection network and the measure network were simplified and now form the **interaction network**;

- The passage position input was dropped;
- The contributions of each passage to the final document score are now assumed to be independent, replacing the self-attention proposed in [1];
- The pooling step now receives more operators, namely average and average over k-max;
- Calculation of the passage relevance score was simplified.

The intuition behind this model is to make a thorough evaluation of the document passages where the exact matches occur, by taking into consideration their context. More precisely, this model explores the interactions presented in the entire passage of each exact match and makes a more refined judgment of the passage relevance based on that.

The updated architecture is depicted in Figure 2 and described here in detail in order to keep this paper self-contained. First, let us define a **query** as a sequence of terms $q = \{u_0, u_1, ..., u_Q\}$, where u_i is the i-th term of the query and Q the size of the query; a **document passage** as $p = \{v_0, v_1, ..., v_T\}$, where v_k is the k-th term of the passage and T the size of the passage; and a document as sequence of passages $D = \{p_0, p_1, ..., p_N\}$.

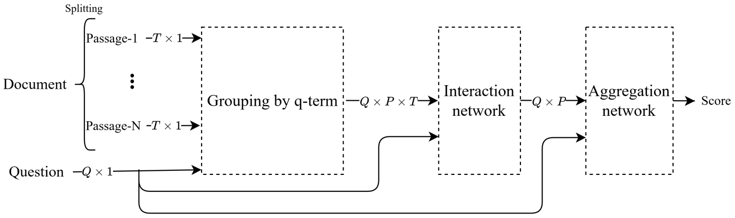

Fig. 2. Overview of the neural ranking model with a tensor representation of the data flow.

From the architecture presented in Figure 2 it is observable that a document is first split into individual sentences, i.e., a sequence of passages. In this step, we rely on the nltk.PunktSentenceTokenizer[4], that implements an unsupervised algorithm for sentence splitting and shows good results on majority of European languages. Then, passages are grouped with each query-term occurring in the passage, and the resulting structure is fed to the interaction network together with the full query to calculate relevance scores for each passage. The final document score is produced in the aggregation network taking into consideration each passage score and the relative importance of each query term.

In more detail, the **Grouping by q-term** block associates each passage with each query term that appears in the passage. Formally, this step produces a set of document passages aggregated by each query term as $D(u_i) = \{p_{i0}, p_{i1}, ..., p_{iP}\}$, where p_{ij} corresponds to the j-th passage with respect to the query term u_i.

[4] https://kite.com/python/docs/nltk.tokenize.punkt.PunktSentenceTokenizer

This aggregated flow facilitates considering the weight of each query term in downstream calculations in a straightforward way, as proposed in DRMM [6].

The **Interaction network** was designed to independently evaluate each query-passage interaction, producing a final relevance score per sentence. In detail, it receives as input the **query** q and the **aggregated set of passages** $D(u_i)$ and creates for each query-passage pair a similarity tensor (interaction matrix) $S \in [-1,1]^{Q \times T}$, where each entry S_{ij} corresponds to the cosine similarity between the embeddings of the i-th query term and j-th passage term, $S_{ij} = \frac{\vec{u_i}^T \cdot \vec{v_j}}{\|\vec{u_i}\| \times \|\vec{v_j}\|}$. Next, an x by y convolution followed by a concatenation of the global max, average and average k-max pooling operation are applied to each similarity tensor, to capture multiple local relevance signals from each feature map, as described in Equation 3,

$$\begin{aligned} h^m_{i,j} &= \sum_{s=0}^{x} \sum_{t=0}^{y} w^m_{s,t} \times S_{i+s,j+t} + b^m \,, \\ h^m_{max} &= \max(h^m),\ m = 1, ..., M \,, \\ h^m_{avg} &= \text{avg}(h^m),\ m = 1, ..., M \,, \\ h^m_{avg-kmax} &= \text{avg}(\text{k-max}(h^m)),\ m = 1, ..., M \,, \\ h &= \{h_{max}; h_{avg}; h_{avg-kmax}\}. \end{aligned} \tag{3}$$

Here, w and b are trainable parameters, the symbol ';' represents the concatenation operator, M corresponds to the total number of filters and the vector $\underset{3M\times 1}{\vec{h}}$ encodes the local relevance between each query-passage, extracted by these pooling operations. At this point, the aggregated set of passages $D(u_i)$ is now represented by their respective vectors $\vec{h}$, i.e., $D(u_i) = \{\vec{h_{p_0}}, \vec{h_{p_1}}, ..., \vec{h_{p_P}}\}$.

The final step of the interaction network is to convert these passage representations $\vec{h}$ to a final relevance score, for which we employed a fully connected layer with sigmoid activation, Equation 4,

$$\underset{P\times 3M}{r_{u_i}} = \sigma(\underset{P\times 3M}{\vec{h_{u_i}}} \cdot \underset{3M\times 1}{\vec{w}} + \underset{1\times 1}{b}). \tag{4}$$

The aim here is to derive a relevance score, relevant (1) or irrelevant (0), directly from the information that was extracted by the pooling operators. So, after this stage the aggregated set of passages $D(u_i)$ is represented by this relevance score, i.e., $D(u_i) = \{r_{p_0}, r_{p_1}, ..., r_{p_P}\} = \vec{r_{u_i}}$.

The **aggregation network**, as already mentioned, takes into consideration the importance of each query term by using a gating mechanism, similar to DRMM [6], over the aggregated set of passages, as described in Equation 5. That is, each passage score is weighted by the importance of its associated query term, following the intuition that in a query different terms carry different importance with respect to the final information goal.

$$\begin{aligned} c_{u_i} &= \underset{1\times E}{\vec{w}} \cdot \underset{E\times 1}{\vec{x_{u_i}}} \,, \\ a_{u_i} &= \frac{e^{c_{u_i}}}{\sum_{u_k \in Q} e^{c_{u_k}}} \,, \\ \underset{P\times 1}{\vec{s_{u_i}}} &= \underset{1\times 1}{a_{u_i}} \times \underset{P\times 1}{\vec{r_{u_i}}} \,, \end{aligned} \tag{5}$$

Here, w is a trainable parameter and $\vec{x_{u_i}}$ corresponds to the embedding vector of the u_i query term. Then the distribution of the query term importance, a, is computed as a softmax and applied to the respective passages scores, $\vec{r_{u_i}}$.

To produce the final document score, a scorable vector $\vec{s}$ is created by performing a summation alongside the query-term dimension of $\vec{s_{u_i}}$. Note that in this step we could have explored other ways to produce this final vector, however, this approach seems to empirically work. Finally, this scorable vector $\vec{s}$ is fed to a Multi-Layer Percepreton (MLP) to produce the final ranking score, as summarized in Equation 6.

$$score = MLP(\sum_{u_i \in Q} \vec{s_{u_i}}) \tag{6}$$

3.3 Snippet Retrieval

As initially stated in [1], this architecture has an interesting property that enables us to directly infer the relevance of each passage according to the model perspective, i.e., the passages scores that most contribute to the final document score. We can therefore derive a final score for each passage as the score given by the interaction network weighted by the query term importance, which was already computed and corresponds to the vector $\vec{s_{u_i}}$.

It is important to note that extracting the most relevant passages per document is not the same as producing a ranked list of passage relevance as intended in the BioASQ competition, which implies comparing passages between different documents. In our case, however, passage scores are not directly comparable since they are obtained with respect to their document, which involves different distributions. So, similarly to [2], we assume that passages from documents with higher document scores are more relevant than passages from documents with a lower score, which seems intuitive. We therefore obtain the list of passages by collecting, from the top ranked documents, all passages with a score above a set threshold. However, a better approach could be explored in the future by producing scores that take into consideration the passage itself and the score of the respective document.

Furthermore, it is noteworthy to reinforce that this strategy works in an unsupervised manner, in the sense that the model does not take into consideration the gold-standard of the passage relevance, but instead produces this relevance based on what is important to increase the final score of a relevant document, according to the document gold-standard. From another perspective, we can argue that the model is pretrained on the document gold-standard and then applied to the snippet retrieval task, making this a zero-shot learning setup since it was never trained on the passage gold-standard.

3.4 Joint Training

Moved by the interesting results, especially in terms of snippet performance, reported in the previous BioASQ challenge [18], we also tried to implement a joint training methodology that explores both document and snippet gold-standards, instead of only training with the document gold-standard. More precisely, we compute the binary cross entropy loss over the passage relevance from Equation 4. Then we added the average cross entropy loss of each passage to the document pairwise loss and trained the model over this combination of the two losses. Note that the architecture for document score and snippet retrieval remained the same, since our main idea at this point was to exploit the snippet gold-standard to, through supervision, enforce the model to distinguish relevant from non-relevant passages. Furthermore, as will be addressed in the following sections and discussed in Section 5, this idea empirically failed to improve the model performance.

4 Submission and Results

In the section, we start by detailing the data collection and some pre-processing steps that are common to our official submissions for the 5 batches. Then we independently show our results for each batch since we continuously refined our base solution by better finetuning the hyperparameters and changing small aspects of the architecture.

4.1 Collection and Pre-processing

In this edition of the BioASQ challenge, the document collection was the 2019 PubMed/MEDLINE annual baseline consisting of almost 30 million articles. However, only roughly 66% of the articles had title and abstract, so following previous observations [2], we decided to discard the remaining 34%, which were rarely relevant according to the gold-standard. At this point, our collection had approximately 20 million documents that were indexed (title and abstract) with Elasticsearch using the **english** text analyzer, which automatically performs tokenization, stemming and stopword filtering.

We adopted a custom tokenizer that uses simple regular expressions to exclude none alphanumeric characters except the hyphen, since many words in the biomedical domain contain a hyphen, like chemical substances. This way we keep these words intact, which enhances the detection of important exact matches. We also trained 200-dimensional word embeddings using the GenSim [21] implementation of word2vec [15], with the 20 million documents (title and abstract) following the described tokenization, which produced a vocabulary of approximately 4 million tokens. We used the default configuration of the word2vec algorithm and fixed the embeddings matrix during the training of the neural ranking model.

4.2 Training Details and Hyperparameters

For training our neural ranking model, we used the gold-standard data from the 1-7 editions of BioASQ, with the exception of one test batch of the seventh edition that we used for validation. Contrarily to our previous work [1], we adopted the pairwise cross-entropy loss, as suggested by Hui et al. [8] and shown in Equation 7.

$$L(q, d^+, d^-) = -log(\frac{e^{(score(q,d^+))}}{e^{(score(q,d^+))} + e^{(score(q,d^-))}}) \tag{7}$$

Since the BioASQ data only provides a list of relevant (positive) documents per query, we sampled the negative documents as the documents that were retrieved by the ES but did not appear in the gold-standard. Another important note is that only the top 10 documents per submission are analyzed by experts in terms of relevance, which may produce an incomplete gold-standard, i.e., positive documents may not be judged since they were not retrieved by participating systems and hence are taken as negative documents during training. To exacerbate the problem, the gold-standard was built as a concatenation of the judged relevance of documents from different years, which implies a different snapshot of the document collection. To alleviate this problem, we restrict the ES search by year so that only the available documents at that time are available to the model training.

We gave a major emphasis to training/validation in order to gain a better intuition of the model behavior and what configuration should be followed in each batch. The neural ranking model was trained using the Adam [10] optimizer, alongside with modern techniques like learning rate finder and cyclical learning rates [24]. The finetuning of this model was a rolling process that took the duration of the 5 batches. More concretely, we searched the kernel size for the convolution, the total number of filters, the pooling operation, the activation functions, and other minor details that ended up not influencing the overall performance. To summarize, Table 1 shows the model configuration that seems to be the strongest producing a model with only **620** trainable parameters.

The model was implemented in TensorFlow[5] and is available at `https://github.com/bioinformatics-ua/BioASQ_CLEF`. The entire training process was conducted with the help of an in-house toolbox that implements pairwise training in TensorFlow. [6].

4.3 BioASQ Evaluation

The BioASQ evaluation is divided in two stages. In the first stage, the submissions are evaluation against a gold-standard annotated by biomedical experts. In

[5] `https://www.tensorflow.org/`

[6] The toolbox is open-sourced here: `https://github.com/T-Almeida/mmnrm`

Table 1. List of the hyperparameters and their respective values. In some cases, the range of tested values is listed, with the best one highlighted in bold.

Hyperparameter	**Value**
BM25 k1 and b	k1∈[0.1, ..., **0.4**, ..., 1.25] and b∈[0.2, ..., **0.4**, ..., 0.7].
ES top-N	**250**, 500, 1000
Number maximum of query tokens, Q	30
Number maximum of passage tokens, T	30
Maximum of passage aggregated to each q-term, P	5
Kernel size	**3 by 3**, multiples (2 by 2, 3 by 3) following [8]
Filters	**16**, 20 , 32
Pooling operations	{max}, {max and avg}, **{max, avg and avg over k-max}**
Activation function	leakyReLU, selu [11], **mish [16]**
Embedding size	200

a second stage, the biomedical experts will manually annotate the relevance of the retrieved documents from each submission. At the time of writing, only the results for the first stage are available, corresponding to the results presented in this paper. In terms of numerical evaluation, the organizers automatically compute five measures (Mean Precision, Recall, F-Measure (F1), MAP and GMAP) over each submission given the current gold-standard. According to the challenge evaluation guidelines [19], the overall system rankings are based on the MAP measure.

Our group submitted five runs for each of the batches, which can be identified by the prefix "bioinfo" on the official results[7]. In the following sections we present a summary table of the results, comparing our five submissions to the top competitor in each batch, i.e., with the top performing system excluding our systems.

4.4 Submission and Results for Batch 1

For the first batch, the BioASQ organizers received a total of 21 submissions from 8 teams[8]. For this run, our main idea was to validate the performance of our phase-I retrieval mechanism and to test if our phase-II reranking model was indeed boosting the original ranking order. As a summary, we submitted one run with the results coming from phase-I, i.e., the BM25 ranking order that was finetuned on the validation set, while the remaining runs were produced by reranking the phase-I results with our neural ranking model:

- bioinfo-0: A finetuned BM25 run produced by the ElastisSearch;
- bioinfo-1 to 4: Neural reranking of the Top-250 documents produced by the finetunned BM25.

At the time of this submission, our ranking model was still in an initial phase of development, which means that it did not completely follow the architecture presented in Section 3.2. More concretely, the model only used the max-pooling operator and a simple linear combination was used for producing the final document score, instead of an MLP.

[7] `http://participants-area.bioasq.org/results/8b/phaseA/`

[8] This number is a speculation based on the names of the submission

Table 2. Summary of the results for the first batch of the BioASQ challenge. Our five submission are presented at the top of the table, starting with the prefix "bioinfo". Additional, we highlight, in bold, the best recorded values per each metric.

System	Rank	Recall	F1	MAP	GMAP	Rank	Recall	F1	MAP	GMAP
	Document Retrieval					Snippet Retrieval				
bioinfo-0	8	44.62	16.27	30.67	0.63	-	-	-	-	-
bioinfo-1	6	44.84	16.92	32.23	1.03	7	17.77	13.86	26.32	0.05
bioinfo-2	2	**48.64**	17.59	33.83	**1.20**	5	17.15	15.00	29.53	0.06
bioinfo-3	1	48.20	17.48	**33.98**	**1.20**	8	18.23	15.91	24.06	0.05
bioinfo-4	4	47.79	17.47	33.59	1.03	6	17.91	15.01	26.53	0.07
Top competitior	3	44.00	16.86	33.59	0.88	1	24.67	**17.52**	**85.75**	0.17

Table 2 reflects the first stage of the BioASQ evaluation for our submissions. In terms of document retrieval, the "bioinfo-3" submission achieved the top score in terms of MAP, which means it was the best performing system in this batch. Additionally, the "bioinfo-2" submission was the second-best performing system and achieved the best result in terms of recall and GMAP. For the snippet retrieval, our best performing system achieved fifth place and also showed interesting results in terms of recall and F-measure when compared to the top-performing system.

4.5 Submission and Results for Batch 2

The second batch received a total of 26 submissions from 9 teams. Our system was built directly on the validation performed for the gold-standard of the previous test batch and the validation set. More precisely, we tested the addition of more pooling operators (average and average over k-max) and the addition of the MLP for scoring, which empirically proved to be beneficial. Additionally, we also decided to pursue a joint training approach described in Section 3.4 and the following list presents a summary of each submitted system:

- bioinfo-0: Neural reranking model with joint training (snippets and documents);
- bioinfo-1,3 and 4: Neural reranking described in Section 3.2;
- bioinfo-2: Neural reranking using max and average pooling operator.

Table 3 shows the performance of the submitted systems, overall only the system that was trained in joint fashion achieved a poor performance. For document retrieval, our top-performing system was the "bioinfo-3", which achieved third place in the overall ranking and also the best score in terms of GMAP. For the snippet retrieval, our best performing system achieved the fourth place on the overall ranking and similarly to the previous batch showed interesting results in terms of recall and F-measures when compared to other systems.

Table 3. Summary of the results for the second batch of the BioASQ challenge. Our five submission are presented at the top of the table, starting with the prefix "bioinfo". Additional, we highlight, in bold, the best recorded values per each metric.

System	Rank	Recall	F1	MAP	GMAP	Rank	Recall	F1	MAP	GMAP
	Document Retrieval					Snippet Retrieval				
bioinfo-0	8	43.41	18.30	29.10	1.17	13	16.17	11.75	18.84	0.09
bioinfo-1	4	47.55	19.94	31.49	1.86	5	21.03	14.61	27.21	0.16
bioinfo-2	7	46.48	19.14	30.84	1.52	6	20.18	14.08	26.37	0.11
bioinfo-3	3	48.80	20.27	31.68	**2.23**	7	20.04	14.08	26.37	0.11
bioinfo-4	5	47.87	20.02	31.20	1.61	4	20.09	14.13	27.67	0.16
Top competitior	1	45.01	**23.00**	**33.04**	1.85	1	25.31	**17.73**	**68.21**	0.15

4.6 Submission and Results for Batch 3, 4 and 5

Contrarily to the previous sections, we now present the results for the third, fourth and fifth batches in the same section, since the submissions for the different batches all follow the same description:

- bioinfo-0: Ensemble of multiple Neural reranking models;
- bioinfo-1 to 4: Neural reranking described in Section 3.2.

The organizers received a total of 28 submissions from 9 teams for the third batch, 26 submissions from 11 teams for the fourth batch, and 25 submissions from 9 teams for the last batch. Given that the proposed joint training seemed to deteriorate the overall performance we decided to keep the focus on the current solution and leave as future work a reformulation of the joint training idea. So, we replaced the joint training submission with a submission that used a naive ensemble of multiple neural reranking models that were trained during validation. Note that for the ensemble run we did not produce a ranked list of snippets since the proposed snippet algorithm does not support multiple relevance values, from different sources, per passage.

Table 4 presents a summary of the results obtained for the last three batches. Focusing now on the document retrieval task, "bioinfo-3" was our best performing system in the third batch achieving a fourth place in the overall ranking and, additionally, "bioinfo-0" was the best system in terms of recall and GMAP. Similarly, "bioinfo-3" was our best performing system in the fourth batch, with a fourth place, and "bioinfo-0" achieved the best result in terms of recall. For the fifth batch, "bioinfo-4" achieved the overall best performance, ranking first place in both MAP and recall. We also achieved the top score in terms of GMAP with the "bioinfo-1" submission. In terms of snippet retrieval, the best ranking was a fifth place on the third and fifth batches.

5 Discussion

In this section we discuss the previously presented results, analyzing first the overall performance on the document retrieval task followed by the results on

Table 4. Summary of the results for the third, fourth and fifth batches of the BioASQ challenge. Our submissions are presented at the top of each batch, starting with the prefix "bioinfo". Additional, we highlight, in bold, the best recorded values per each metric.

System	**Rank**	**Recall**	**F1**	**MAP**	**GMAP**	**Rank**	**Recall**	**F1**	**MAP**	**GMAP**
Batch 3										
	Document Retrieval					Snippet Retrieval				
bioinfo-0	7	**54.15**	18.73	43.50	**2.07**	-	-	-	-	-
bioinfo-1	11	52.43	18.11	42.68	1.55	7	27.01	16.70	39.10	0.37
bioinfo-2	8	53.20	18.08	43.03	1.86	6	28.25	17.29	40.85	0.36
bioinfo-3	4	53.65	18.20	43.69	2.04	5	29.63	17.34	41.37	0.37
bioinfo-4	10	54.08	18.83	42.84	2.02	8	26.79	16.61	37.76	0.32
Top competitior	1	53.77	19.32	**45.10**	1.87	1	**35.58**	**21.40**	**100.39**	0.56
Batch 4										
	Document Retrieval					Snippet Retrieval				
bioinfo-0	7	**55.60**	19.95	39.77	1.92	-	-	-	-	-
bioinfo-1	8	55.28	19.30	39.71	2.01	10	25.29	16.62	34.55	0.13
bioinfo-2	6	55.53	19.77	40.06	2.10	7	25.84	17.23	36.59	0.15
bioinfo-3	4	53.92	19.38	40.24	1.31	9	26.55	17.42	35.00	0.15
bioinfo-4	10	54.44	19.75	38.69	1.54	12	25.29	16.62	34.55	0.13
Top competitior	1	54.46	19.67	**41.63**	2.04	1	**33.03**	**21.51**	**102.44**	0.55
Batch 5										
	Document Retrieval					Snippet Retrieval				
bioinfo-0	4	62.08	19.95	47.47	3.20	-	-	-	-	-
bioinfo-1	3	62.21	19.97	47.80	**3.49**	8	31.94	19.46	42.97	0.24
bioinfo-2	7	59.98	19.09	46.45	2.40	10	31.94	19.47	42.06	0.24
bioinfo-3	5	61.54	19.67	46.65	2.88	9	32.05	19.35	42.70	0.23
bioinfo-4	1	**62.63**	19.78	**48.42**	3.30	5	32.14	19.60	43.79	0.29
Top competitior	2	60.50	39.63	48.25	2.54	1	35.36	24.91	112.67	0.38

snippet retrieval. We complement this discussion with our considerations on what was successful and what has failed.

Addressing the results presented in Tables 2, 3 and 4, we consider that our system had an extremely competitive permanence, being in the top position for the first and fifth batch and close to the top in the remaining batches. Additionally, we note that at least one of our submissions achieved the best performance in at least one metric for all the batches. Furthermore, if we look at the GMAP metric it is observable that our systems achieved the best results in all but the fourth batch.

With respect to the neural reranking performance comparative to the phase-I ranking, we can see in Table 2 that every neural submission was able to improve the original BM25 ranking order, what is in accord with our speculation and validation results. So, these results also seem to be according to our proposed idea of better exploring the context where the exact match occurs to produce a more refined judgment that contributes to the final score.

As previously said, after the first batch and based on some validation tests we decided to change the model by adding more pooling operations and the MLP. However, at a first glance, according to the results on the second, third, and fourth batches, it seems that these changes were not beneficial, since the system was not able to achieve the top performance, similarly to the first batch. However, we argue that this discrepancy can also be a consequence of some improvement of the competitors systems after the first batch. Additionally, we experiment the updated architecture on the first batch and were able to easily achieve a MAP score of over 35%, surpassing the previous best.

Finally, the only metric for which our system does not seem to be able to achieve competitive results is F-measure. However, as noted previously [18], a system that outputs confidence scores instead of ranking scores seems to be able to achieve higher performance in terms of this metric. A possible explanation relies on the BioASQ data and more properly on the questions that have only a few true positive documents[9] in the entire collection. In this case, a system based on confidence scores can easily create a ranked list with fewer than 10 documents (the maximum considered per question), since it selects the relevant documents based on a threshold value over the confidence scores. So, for this type of questions, a system based on confidence is more likely to achieve higher values of Precision and Recall (resulting in a higher F1 measure) when compared to a ranking system that will obtain a higher Recall but lower Precision since it always outputs the top 10 documents.

In terms of snippet retrieval, the submitted system did not present competitive results when compared to the top submissions, being the best performance a fourth place in the second batch. However, given that our method does not use the snippet gold-standard for training and follows a naive ranking approach, we consider these results encouraging, especially in terms of recall and F-measure, and with the potential to be better explored in future work.

Concerning the joint training approach, we considered that it has empirically failed. More precisely, it seems that our intuition to improve the passage relevance with supervision may be more challenging to achieve. One problem is the notion of passage relevance since most of the time a relevant snippet in the gold-standard encompasses multiple sentences, which the model will see and score as independent. So, this supervision may be forcing the model to boost the relevance score of sentences that in isolation carry week matching signals, ending up hindering the overall matching signal extraction. Another problem lies on the naive implementation of the snippet retrieval algorithm. Another idea is to directly use the snippet gold-standard to produce ranking scores, more similar to the winning approach in [18].

6 Conclusion and Future Work

In this paper, we propose a two-stage retrieval pipeline to address the biomedical retrieval problem. Our system first uses BM25 to selected a pool of potential relevant candidates that are then reranked by a neural ranking model. Contrarily to the NLP trend, we focused on building a lightweight interaction based model,

[9] less than 10 documents

which yields a final model with only 620 trainable parameters. The proposed architecture can also be used to produce relevance scores for each document passage according to the model perspective of relevance. This property enables us to perform passage retrieval in a zero-shot learning setup.

The proposed pipeline was evaluated on the eighth edition of BioASQ, were it achieved competitive results for the document retrieval task, being on top and close to the top in all batches. In the snippet retrieval task, it showed interesting results given that they were produced by a naive algorithm in a zero-shot learning setup.

As future work, there are minor questions still open especially on the aggregation network configuration. Additionally, an interesting route is to compare the current architecture with a direct, but parameter-greedy, extension that uses a state-of-the-art transformer based model, such as BERT, which are well suited to our objective of better evaluating the passage context. This may be achieved by replacing the word2vec embeddings by the context-aware embeddings produced by these models or by completely replacing the interaction network.

References

1. Almeida, T., Matos, S.: Calling attention to passages for biomedical question answering. In: Jose, J.M., Yilmaz, E., Magalhães, J., Castells, P., Ferro, N., Silva, M.J., Martins, F. (eds.) Advances in Information Retrieval. pp. 69–77. Springer International Publishing, Cham (2020). https://doi.org/10.1007/978-3-030-45442-5_9
2. Brokos, G.I., Liosis, P., McDonald, R., Pappas, D., Androutsopoulos, I.: AUEB at BioASQ 6: Document and Snippet Retrieval (sep 2018), http://arxiv.org/abs/1809.06366
3. Dai, Z., Callan, J.: Deeper text understanding for ir with contextual neural language modeling. In: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval. p. 985–988. SIGIR'19, Association for Computing Machinery, New York, NY, USA (2019). https://doi.org/10.1145/3331184.3331303
4. Dai, Z., Callan, J.: Context-aware document term weighting for ad-hoc search. In: Proceedings of The Web Conference 2020. p. 1897–1907. WWW '20, Association for Computing Machinery, New York, NY, USA (2020). https://doi.org/10.1145/3366423.3380258
5. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding (2018)
6. Guo, J., Fan, Y., Ai, Q., Croft, W.B.: A deep relevance matching model for ad-hoc retrieval. Proceedings of the 25th ACM International on Conference on Information and Knowledge Management (Oct 2016). https://doi.org/10.1145/2983323.2983769
7. Huang, P.S., He, X., Gao, J., Deng, L., Acero, A., Heck, L.: Learning deep structured semantic models for web search using clickthrough data. In: Proceedings of the 22nd ACM international conference on Conference on information & knowledge management - CIKM '13. pp. 2333–2338. ACM Press, New York, New York, USA (2013). https://doi.org/10.1145/2505515.2505665

8. Hui, K., Yates, A., Berberich, K., de Melo, G.: Co-pacrr: A context-aware neural ir model for ad-hoc retrieval. pp. 279–287 (02 2018). https://doi.org/10.1145/3159652.3159689
9. Jin, Z.X., Zhang, B.W., Fang, F., Zhang, L.L., Yin, X.C.: A multi-strategy query processing approach for biomedical question answering: Ustb_prir at bioasq 2017 task 5b. In: BioNLP (2017)
10. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: Bengio, Y., LeCun, Y. (eds.) 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings (2015), http://arxiv.org/abs/1412.6980
11. Klambauer, G., Unterthiner, T., Mayr, A., Hochreiter, S.: Self-normalizing neural networks (06 2017)
12. MacAvaney, S., Yates, A., Cohan, A., Goharian, N.: Cedr: Contextualized embeddings for document ranking. In: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval. p. 1101–1104. SIGIR'19, Association for Computing Machinery, New York, NY, USA (2019). https://doi.org/10.1145/3331184.3331317
13. Mateus, A., González, F., Montes, M.: Mindlab neural network approach at bioasq 6b (11 2018). https://doi.org/10.18653/v1/W18-5305
14. McDonald, R., Brokos, G.I., Androutsopoulos, I.: Deep Relevance Ranking Using Enhanced Document-Query Interactions (sep 2018), http://arxiv.org/abs/1809.01682
15. Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2. p. 3111–3119. NIPS'13, Curran Associates Inc., Red Hook, NY, USA (2013)
16. Misra, D.: Mish: A self regularized non-monotonic neural activation function (2019)
17. Pang, L., Lan, Y., Guo, J., Xu, J., Xu, J., Cheng, X.: Deeprank. Proceedings of the 2017 ACM on Conference on Information and Knowledge Management (Nov 2017). https://doi.org/10.1145/3132847.3132914
18. Pappas, D., McDonald, R., Brokos, G.I., Androutsopoulos, I.: Aueb at bioasq 7: Document and snippet retrieval. In: Cellier, P., Driessens, K. (eds.) Machine Learning and Knowledge Discovery in Databases. pp. 607–623. Springer International Publishing, Cham (2020)
19. Prodromos Malakasiotis, Ioannis Pavlopoulos, I.A.d., Nentidis, A.: Evaluation measures for task b, http://participants-area.bioasq.org/Tasks/b/eval_meas_2020/
20. Radford, A., Narasimhan, K., Salimans, T., Sutskever, I.: Improving language understanding by generative pre-training (2018)
21. Řehůřek, R., Sojka, P.: Software Framework for Topic Modelling with Large Corpora. In: Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks. pp. 45–50. ELRA, Valletta, Malta (May 2010), http://is.muni.cz/publication/884893/en
22. Robertson, S., Zaragoza, H.: The probabilistic relevance framework: Bm25 and beyond. Found. Trends Inf. Retr. **3**(4), 333–389 (Apr 2009). https://doi.org/10.1561/1500000019
23. Shen, Y., He, X., Gao, J., Deng, L., Mesnil, G.: A Latent Semantic Model with Convolutional-Pooling Structure for Information Retrieval. In: Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management - CIKM '14. pp. 101–110. ACM Press, New York, New York, USA (2014). https://doi.org/10.1145/2661829.2661935

24. Smith, L.N.: Cyclical learning rates for training neural networks (2015)
25. Tsatsaronis, G., Balikas, G., Malakasiotis, P., Partalas, I., Zschunke, M., Alvers, M., Weißenborn, D., Krithara, A., Petridis, S., Polychronopoulos, D., Almirantis, Y., Pavlopoulos, J., Baskiotis, N., Gallinari, P., Artieres, T., Ngonga Ngomo, A.C., Heino, N., Gaussier, E., Barrio-Alvers, L., Paliouras, G.: An overview of the bioasq large-scale biomedical semantic indexing and question answering competition. BMC Bioinformatics **16**, 138 (04 2015). https://doi.org/10.1186/s12859-015-0564-6