AUEB-NLP at BioASQ 8: Biomedical Document and Snippet Retrieval

Dimitris Pappas[1,2], Petros Stavropoulos[1,2], and Ion Androutsopoulos[1]

[1] Department of Informatics, Athens University of Economics and Business, Greece
pappasd@aueb.gr,pstav1993@aueb.gr,ion@aueb.gr
[2] Institute for Language and Speech Processing, Research Center 'Athena', Greece
dpappas@ilsp.gr,petros.stavropoulos@athenarc.gr

Abstract. We present the submissions of AUEB's NLP group to the BIOASQ 8 document and snippet retrieval tasks. We relied mostly on JPDRMM, our top performing model of BIOASQ 7, but we also tested feeding JPDRMM with word embeddings obtained by applying a graph node embedding method to a biomedical co-occurrence graph; the latter approach was competitive to using biomedical WORD2VEC embeddings in JPDRMM. We also experimented with neural methods to encode, index, and directly retrieve snippets (sentences) and indirectly documents containing the retrieved snippets, instead of relying on conventional information retrieval to pre-fetch possibly relevant documents and invoking JPDRMM to re-rank the pre-fetched documents and their snippets; conventional BM25-based pre-fetching, however, was far better. Our JPDRMM-based document and snippet retrieval methods scored at the top or near the top for all test batches of BIOASQ 8.

Keywords: Biomedical Document Retrieval · Biomedical Snippet Retrieval · BioASQ · Deep Learning.

1 Introduction

BIOASQ [34] is an annual competition for biomedical document classification (Task A), as well as document, snippet, structured data retrieval, question answering, and summarization (Task B).[3] This work pertains to Task B, which consists of two 'phases'. In Phase A, systems are provided with English biomedical questions and are required to retrieve relevant documents and document snippets from a collection of MEDLINE/PUBMED articles.[4] In Phase B, systems are

[3] In some years, BIOASQ includes additional tasks. Consult http://bioasq.org/ and http://bioasq.org/participate/challenges.

[4] See http://www.ncbi.nlm.nih.gov/pubmed/. The 'documents' of BIOASQ are actually article titles concatenated with their abstracts.

provided with English biomedical questions, along with gold relevant documents and gold document snippets per question; they are required to respond with 'exact answers' (e.g., named entities) and 'ideal' answers, i.e.g, paragraph-sized summaries. Here we provide an overview of the submissions of AUEB's NLP group to the document and snippet retrieval tasks (parts of Task 8b, Phase A). We also participated in exact answer extraction (part of Task 8b, Phase B) this year, but we describe this aspect of our work very briefly, because it was only a quick attempt to reuse work from cloze-style biomedical machine reading comprehension (MRC) [29], which led to poor results. By contrast our best document and snippet retrieval systems scored at the top or near the top for all test batches of BIOASQ 8, as in BIOASQ 6 and 7 [4, 28].

For document and snippet retrieval, we employed our JPDRMM model [28], which had achieved top performance in BIOASQ 7. This year, we tested JPDRMM with biomedical WORD2VEC embeddings [23], as in our previous work, but also with word embeddings obtained by applying a graph node embedding method [14] to a biomedical entity co-occurrence graph; both approaches were equally good. We also experimented with neural methods to encode, index, and directly retrieve relevant snippets (sentences) and indirectly retrieve documents containing the retrieved snippets, instead of relying on conventional information retrieval to pre-fetch possibly relevant documents and invoking JPDRMM to re-rank the pre-fetched documents and their snippets; conventional BM25-based pre-fetching, however, followed by JPDRMM reranking was far better.[5]

As already noted, we also participated (for the first time) in exact answer extraction this year. We experimented with SCIBERT-MAX-READER, a SCIBERT-based model [2] that we recently introduced for cloze-style biomedical MRC [29]. Although SCIBERT-MAX-READER has been found to reach or even exceed human expert performance in biomedical cloze-style MRC [29], it performed poorly in BIOASQ 8, indicating that BIOASQ's exact answer extraction task is not as similar as we had hoped to the MRC task and dataset SCIBERT-MAX-READER was developed for. Hence, we do not provide here any further information about this aspect of our work, though we hope to examine in future work if it can be used to pre-train exact answer extraction components, which will be subsequently fine-tuned on BIOASQ exact answer extraction training instances.

2 JPDRMM-based Models for Document and Snippet Retrieval

Following last year's successful approach [28], we used JPDRMM as our main model for document and snippet retrieval. JPDRMM is actually a (neural) re-ranking model; it is fed with the top N documents retrieved by a conventional (and computationally more efficient) information retrieval engine, and

[5] As in recent BIOASQ editions, gold snippets are almost always sentences in BIOASQ 8. Hence, we take snippets to be sentences in our experiments. When gold snippets contain multiple sentences, we break them into multiple single-sentence snippets.

it is trained to jointly re-rank the top N documents and their snippets. We do not discuss JPDRMM further here, since it is presented in detail in our previous work [28]. We note, however, that JPDRMM computes one loss for document re-ranking (L_{doc}) and another one for snippet re-ranking (L_{snip}). In our previous work, we simply added the two losses, but this year we used a linear combination of the two losses, $L = \lambda_{doc} L_{doc} + \lambda_{snip} L_{snip}$, and we tuned the two loss weights (λ_{doc}, λ_{snip}) by performing a 10-fold cross-validation on training data.[6]

Furthermore, in our previous work JPDRMM had been used with biomedical WORD2VEC embeddings [23]. Here we also experimented with word embeddings obtained by applying a graph node embedding method [14] to a biomedical entity co-occurrence graph. The node embedding method we used is an extension of NODE2VEC [11] that considers both the topology of the graph it is applied to and text associated with each node of the graph. In our case, nodes are biomedical entities and the text of each node is the (often multi-word) English name of the corresponding entity. Roughly speaking, the graph node embedding method uses an RNN to obtain a node embedding from the word embeddings of the text (name) of the node, and then applies graph convolutions to make sure that the embeddings of nodes with common neighbors are close to each other. In effect, the embeddings of two nodes (entities) end up being close to each other if the two nodes have similar names (e.g., 'acute cardiomyopathy', 'cardiomyopathy') or similar neighbors. To construct the entity co-occurrence graph, we used PUBTATOR [38] to identify the biomedical entities in a randomly selected set of approx. 5 million PUBMED abstracts. Whenever a biomedical entity was found in the same abstract with another one, a link between the two entities was added to the graph. We then pruned links corresponding to co-occurrences with frequencies lower than 10. Although the graph embedding method is primarily intended to generate node embeddings, it also generates word embeddings, which we used as an alternative to WORD2VEC embeddings. The intuition was that nodes (entities) with similar neighborhoods in the co-occurrence graph are probably related, the graph node embedding method places their embeddings close to each other, and this might also help place close to each other the embeddings of the words of the names of the two related nodes, since node embeddings are based on the word embeddings of the node names. We call GRAPH-JPDRMM the JPDRMM version that uses word embeddings obtained via the graph embedding method, and W2V-JPDRMM the original JPDRMM version with biomedical WORD2VEC embeddings.

3 SEMantic Indexing for SEntence Retrieval (SEMISER)

Our JPDRMM-based models of the previous section rely on conventional information retrieval to obtain a set of N possibly relevant documents from the document collection, and then invoke JPDRMM to re-rank the retrieved N documents and their snippets. Instead, in this section we use a neural encoder to

[6] We tuned for λ_{snip} and λ_{doc} in $\{0, 0.001, 0.01, 0.1, 0.2, 1.0, 5.0, 10.0, 100.0\}$, which led to $\lambda_{snip} = 0.01$ and $\lambda_{doc} = 1.0$.

map each sentence of the document collection to a sentence embedding, and we index the sentences of the document collection by their sentence embeddings. We use a similar encoder to map each query to a query embedding, and approximate *k*-NN retrieval algorithms [3, 20] to retrieve the sentences of the document collection whose embeddings are closest to the query embedding; the retrieved sentences are ranked by increasing distance to the query embedding. When required to retrieve documents too, we simply report the documents that contained the retrieved sentences; the relevance score of each document is the minimum query-sentence distance over all the sentences of the document.[7] The encoder of the sentences and the encoder of the queries are jointly trained in a 'self-supervised' manner, detailed below, which does not require manually labeled gold relevant documents and snippets per training query. The resulting method, called SEMISER (SEMantic Indexing for SEntence Retrieval), is a new deep learning model for semantic indexing of sentences [40].

SEMISER takes a sentence and a query as input (Fig. 1). Each word of the sentence and query is mapped to the corresponding word embedding. In BIOASQ 8, we used the same biomedical WORD2VEC embeddings as in the JPDRMM methods.[8] Two stacked trigram convolutional layers (with tanh activations) are used to obtain a context-aware embedding for each word, and a self-attention layer (different for sentences and queries) then computes the sentence and query embeddings. The self-attention layer actually produces two sentence embeddings (vectors) and two query embeddings. The intuition is that the two vectors will capture different views of the sentence and query, respectively, similar in spirit to the multiple representations obtained when using multiple attention heads in Transformer-based models [35]. To force the two sentence (or query) embeddings to learn different views of the sentence (or query), we compute a cosine similarity loss between the two sentence (or query) embeddings during training. We also compute the maximum cosine similarity over all four pairs of sentence and query embeddings, and require it to be 1 (or 0) when SEMISER is given a query and a relevant (or irrelevant) sentence, using binary cross-entropy loss.[9] We simply added the three losses, but we plan to tune the loss weights in future work. Although SEMISER can be trained in a supervised manner, by using pairs consisting of queries and relevant (or irrelevant) sentences as positive (or negative) training instances, BIOASQ provides relatively few training instances by today's standards (approx. 2.6k training queries, with approx. 1.24 relevant snippets per query on average). Instead, we opted for a 'self-supervised' approach, using an auxiliary training task for which very large numbers of training instances can be obtained without manual annotation.

For the auxiliary training task, we used sentences from 50k randomly selected PUBMED documents. The positive training instances were pairs consisting of one of the sentences and a (possibly multi-word) keyterm extracted from

[7] We also maintain an index that maps sentences to their documents.

[8] The word embeddings are not updated during training, in any of the methods we experimented with.

[9] We replace all negative cosine similarity values by zeros, using a RELU activation.

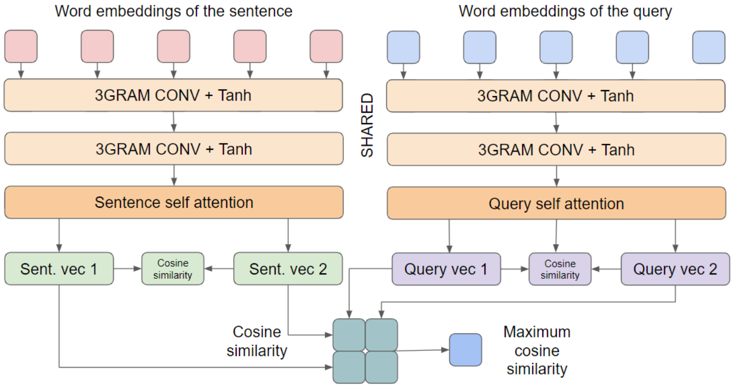

Fig. 1. Illustration of the SEMISER model.

the sentence using SGRANK [8], an unsupervised keyterm extraction method. The negative training instances were pairs consisting of one of the sentences and a keyterm extracted from another randomly selected sentence. This process led to approx. 2.3 million training instances; we generated an equal number of positive and negative instances. In effect, the auxiliary task requires SEMISER to be able to generate sentence and query embeddings containing enough information to decide if a sentence contains a keyterm (treated as a query) or not. The intuition is that in most cases relevant sentences contain keyterms of the queries, hence being able to predict if a keyterm included in a sentence is important. By forcing keyterms (more generally queries) to be represented by low-dimensional embeddings, we also hope that similar queries will end up being close in the vector space, and that similar sentences will also end up being close in a similar manner.

Having trained SEMISER, we use its left part (Fig. 1) to obtain and index (offline) sentence embeddings, and the right part to convert queries (on the fly) to query embeddings. To retrieve sentences (and the documents that contain them), we query the index of sentence embeddings (using approximate k-NN matching) to obtain the sentences with the most similar sentence embeddings. For each retrieved sentence, we compute again the maximum similarity score over all four pairs of sentence-query embeddings.

4 Overall System Architecture

In the JPDRMM-based methods (Section 2), we use ElasticSearch[10] to index the approx. 30 million PUBMED 'documents' (concatenated titles and abstracts) of BIOASQ 8. Figure 2 illustrates the overall architecture of our JPDRMM-based methods. Given a question, we submit it as a query to ElasticSearch to retrieve

[10] https://www.elastic.co/elasticsearch/

the N documents with the best BM25 scores; in our experiments, $N = 100$. Then JPDRMM (W2V-JPDRMM or GRAPH-JPDRMM) jointly re-ranks the N documents and their snippets, assigning relevance scores to each one. We return the n_d documents with the highest relevance scores, and the n_s snippets with the highest relevance scores among the snippets of the n_d documents. We set $n_d = n_s = 10$, as required by BIOASQ 8.

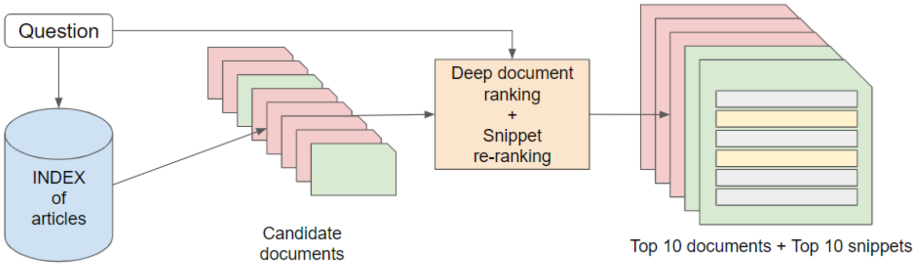
Question
INDEX of articles
Candidate documents
Deep document ranking + Snippet re-ranking
Top 10 documents + Top 10 snippets

Fig. 2. Architecture of our JPDRMM-based systems. The IR engine retrieves candidate relevant documents. JPDRMM (W2V-JPDRMM or GRAPH-JPDRMM) re-ranks the retrieved documents and their snippets. We return the 10 documents with the highest scores, and the 10 snippets of those documents with the highest scores.

When using SEMISER (Section 3), we sentence-split[11] all the PUBMED 'documents' of BIOASQ 8, we map each sentence to its two sentence embeddings (left part of Fig. 1, already trained, see also Fig. 3), and we index (offline) all the sentence embeddings in a single index. Given a BIOASQ question, we map it on the fly to its two query embeddings (right part of Fig. 1, already trained, see also Fig. 3), and we query the index of sentence embeddings (using approximate k-NN matching) to retrieve the n_s sentences with the highest similarity scores.[12] The similarity (relevance) score of each sentence is the maximum cosine similarity over all four pairs of sentence-query embeddings (as in the lower part of Fig. 1). To retrieve documents, we assign to each document the score of its best (most relevant) snippet, and return the n_d documents with the best scores. Again, we set $n_d = n_s = 10$, as required by BIOASQ 8. Since multiple snippets may come from the same document, we actually initially retrieve more than 10 snippets, to always be able to return 10 documents.

Unfortunately, when used on its own to directly retrieve sentences and documents (Fig. 3), SEMISER performed poorly. Hence, we also experimented with combinations of SEMISER with other methods (e.g., applying SEMISER only to documents retrieved by ElasticSearch), which are discussed in the next section.

[11] We use NLTK's English sentence splitter; see `https://www.nltk.org/api/nltk.tokenize.html`.

[12] We use HNSWLIB [20] for approximate k-NN matching.

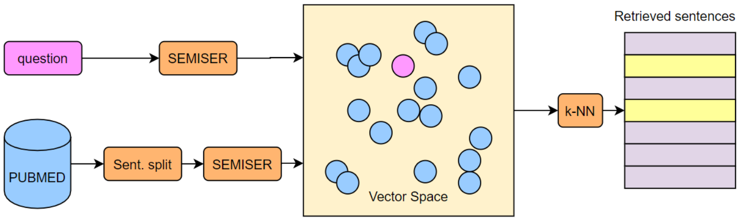

Fig. 3. Using SEMISER for snippet (sentence) retrieval. Documents inherit the scores of their best snippets.

5 Data and Experiments

The BIOASQ 8 document collection consists of approx. 31M 'documents' (concatenated titles and abstracts) from the openly available 'MEDLINE/PubMed Baseline 2020' collection.[13] We discarded approx. 11M articles that contained only titles. The average 'document' length is 197.19 words, the minimum length is 11 words, and the maximum length is 1,500 words. There are 2,647 training questions, from which we held out 100 for development. The average training question length is 9.02 words, and the maximum length is 30 words. Each one of the five (not publicly available) test batches contains 100 questions.

5.1 AUEB-NLP Submissions

We submitted the following five systems to BIOASQ 8 (Task 8b, Phase A). In all cases, we used BM25 when scoring documents with ElasticSearch.

AUEB-NLP-1: W2V-JPDRMM for document and snippet retrieval, with BM25 for initial document retrieval.

AUEB-NLP-2 (batches 2–5 only): Same as AUEB-NLP-1, but with GRAPH-JPDRMM instead of W2V-JPDRMM.

AUEB-NLP-3: Same as AUEB-NLP-1, but we use SEMISER to re-score the sentences of the n_d documents that W2V-JPDRMM retrieves. Each one of the n_d documents is then re-ranked by the score of its best snippet.

AUEB-NLP-4: SEMISER for document and snippet retrieval (Fig. 1) in batches 1–2. An ensemble of AUEB-NLP-1 and AUEB-NLP-2 in batches 3–5. In batches 3–4, the ensemble summed the scores of the two models (both when scoring documents and snippets); in batch 5, it used the maximum score of the two models.

AUEB-NLP-5: BM25 for document retrieval, then SEMISER for snippet retrieval.

The last three systems were intended to test the performance of SEMISER, when used on its own for both document and snippet retrieval (AUEB-NLP-4, batches 1–2), when pipelined after BM25 (AUEB-NLP-5), or when used as an additional rescoring mechanism after W2V-JPDRMM (AUEB-NLP-3). Since SEMISER

[13] Available from `ftp://ftp.ncbi.nlm.nih.gov/pubmed/baseline/`.

performed very poorly when used on its own (AUEB-NLP-3, batches 1–2), in the last three batches we used the slot of AUEB-NLP-3 to experiment with ensembles of our two best systems (AUEB-NLP-1 and AUEB-NLP-2).

5.2 Results

The official BIOASQ evaluation measure for document and snippet retrieval is Mean Average Precision (MAP). Table 1 reports the official MAP scores of our systems for batches 1–5, along with the best score achieved by other participants in each batch. We also report system rankings, again based on MAP.

A first observation is that AUEB-NLP-1 and AUEB-NLP-2, which use W2V-JPDRMM and GRAPH-JPDRMM respectively, performed particularly well in snippet retrieval. In batches 1–4, they were the top two systems in snippet retrieval, largely outperforming all other systems in MAP, and they ranked 2nd and 3rd in batch 5, where their MAP was close to that of the best system.[14] The two systems also performed well in document retrieval, where they were ranked in the top 8 positions in all batches among more than 20 participants. These document and snippet retrieval results also indicate that JPDRMM works equally well with the original biomedical WORD2VEC embeddings (W2V-JPDRMM) and the word embeddings we obtained from the entity co-occurrence graph via the graph node embedding method (GRAPH-JPDRMM).

Another key observation is that SEMISER, which uses self-supervised neural encoders to index and retrieve sentences and indirectly documents (AUEB-NLP-4, batches 1–2 only), performed poorly, both in document and snippet retrieval. When SEMISER was used only to score the sentences of documents retrieved by BM25 (AUEB-NLP-5), its snippet MAP improved substantially (see batches 1–2), but remained well below the snippet MAP of the best systems. For document retrieval, AUEB-NLP-5 uses BM25, hence the corresponding document MAP results show the performance of conventional information retrieval. When SEMISER was used to re-score sentences and documents retrieved by BM25 and W2V-JPDRMM (AUEB-NLP-3), both document MAP and snippet MAP were lower than those of AUEB-NLP-5. Overall, we were unable to obtain benefits by including SEMISER in any of our systems. We note, however, that SEMISER is still in early development stages. We plan to investigate its failures further and try to improve it. The simplistic ensembles (summing or taking the maximum score) of AUEB-NLP-1 and AUEB-NLP-2 that we experimented with (AUEB-NLP-4, batches 3–5) did not improve document MAP and, more surprisingly, led to much worse snippet MAP compared to the scores of the systems we combined. We also need to investigate these results further.

[14] We are surprised by the fact that the official MAP scores occasionally exceed 100%, which may be due to using a wrong normalization.

DOCUMENT RETRIEVAL										
System	**Rank**	**MAP**	**Rank**	**MAP**	**Rank**	**MAP**	**Rank**	**MAP**	**Rank**	**MAP**
Batch	Batch 1		Batch 2		Batch 3		Batch 4		Batch 5	
AUEB-NLP-1	3	33.59	2	31.81	3	44.41	5	40.09	8	45.97
AUEB-NLP-2	n/a	n/a	6	31.03	**1**	**45.1**	**1**	**41.63**	6	46.57
AUEB-NLP-3	14	24.37	14	27.02	24	32.56	25	28.29	23	33.89
AUEB-NLP-4	21	4.93	26	7.12	2	45.02	2	41.47	9	45.96
AUEB-NLP-5	13	28.62	9	28.43	15	38.31	19	34.77	18	40.93
Top Comp.	**1**	**33.98**	**1**	**33.04**	4	43.69	3	41.21	**1**	**48.42**
SNIPPET RETRIEVAL										
System	**Rank**	**MAP**	**Rank**	**MAP**	**Rank**	**MAP**	**Rank**	**MAP**	**Rank**	**MAP**
Batch	Batch 1		Batch 2		Batch 3		Batch 4		Batch 5	
AUEB-NLP-1	**1**	**85.75**	**1**	**68.21**	2	96.32	**1**	**102.44**	2	108.31
AUEB-NLP-2	n/a	n/a	2	65.49	**1**	**100.39**	2	99.2	3	106.39
AUEB-NLP-3	9	21.71	15	15.56	13	23.85	13	17.54	13	26.37
AUEB-NLP-4	15	2.73	17	3.28	9	35.94	11	34.24	12	33.56
AUEB-NLP-5	4	36.36	10	22.17	10	32.17	8	35.31	11	37.34
Top Comp.	2	54.49	3	33.74	3	65.58	3	71.63	**1**	**112.67**

Table 1. Performance on BIOASQ Task 8b, Phase A for document and snippet retrieval. Top Comp. is the top scoring submission of other teams. AUEB-NLP-2 (W2V-JPDRMM) did not participate in batch 1. AUEB-NLP-4 was different in batches 1–2 (SEMISER on its own) and batches 3–5 (W2V-JPDRMM and GRAPH-JPDRMM ensembles).

6 Related Work

6.1 Document Retrieval

Neural models for re-ranking have shown promising results in multiple domains [1,39]. The introduction of BERT sparked the creation of more deep learning approaches for document retrieval and re-ranking [12,22,25,27,41]. Recently several deep learning models have also been introduced for neural document retrieval using document vector representations [6,7,12,19,24,36,44].

In the document retrieval task, several approaches that benefit from data structured as graphs have been proposed [9,21,32,44,45]. A model that benefits from structured data in biomedical IR is GRAPHENE [46]. It uses graph-augmented document representation learning, query expansion, and representation learning to rank biomedical articles. Its creators concatenated the titles and abstracts of biomedical articles from the TREC Precision Medicine track [31] to create a pool of documents. They then used the MESH terms of the articles as queries, aiming to retrieve articles labeled with the MESH terms of each query. GRAPHENE managed to surpass other deep learning models [17,24] in this task, which resembles the auxiliary task of SEMISER.

6.2 Snippet Retrieval

Several deep learning approaches have been proposed for biomedical snippet retrieval [10, 26, 43]. TANDA [10] is a BERT based deep learning model for answer sentence selection. The authors fine-tuned a pre-trained BERT model, using non-biomedical data obtained from the Natural Questions dataset [15]. Their tuning led to a BERT model trained for answer sentence selection on Wikipedia articles. A second fine-tuning step adapts the obtained model to the specific target domain of each dataset they use for testing. Their model outperformed the previous state of the art model in the TREC-QA dataset [31], which partly consists of biomedical documents. Yoon et al. [43] proposed a clustering-based method for sentence selection. Their clustering significantly improved performance leading to state of the art performance on WIKIQA [42] and TREC-QA [31] when it was published, but was later surpassed by TANDA.

The COVID-19 pandemic and the need to keep up with rapidly increasing related biomedical literature led to new document collections and retrieval challenges for COVID-19 [30, 37]. These datasets, however, provide only gold documents, not snippets, and methods proposed for them focus only on document retrieval [5, 16, 18, 41], some also on summarization [13, 33].

7 Conclusions and Future Work

In this short system description paper, we presented our submissions to the document and snippet retrieval tasks of BIOASQ 8. Our joint JPDRMM model scored at the top or near the top in both tasks across all batches, as in BIOASQ 7. Its performance was equally good when its biomedical WORD2VEC embeddings were replaced by word embeddings obtained from a biomedical entity co-occurrence graph via a graph embedding method. In future work, we also plan to experiment with JPDRMM fed with random word embeddings, to investigate to what extent JPDRMM is affected by the word embeddings used. We also experimented with self-supervised neural encoders to index and directly retrieve snippets and indirectly documents, instead of initially retrieving documents using conventional information retrieval and then re-ranking the documents and their snippets with JPDRMM. This approach performed poorly, but is still in early stages and we hope to improve it further.

References

1. Ahmad, W.U., Chang, K.W., Wang, H.: Context attentive document ranking and query suggestion. Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (2019)
2. Beltagy, I., Lo, K., Cohan, A.: Scibert: Pretrained language model for scientific text. In: EMNLP (2019)
3. Boytsov, L., Novak, D., Malkov, Y., Nyberg, E.: Off the beaten path: Let's replace term-based retrieval with k-nn search. Computing Research Repository (CoRR) **abs/1610.10001** (2016)
4. Brokos, G., Liosis, P., McDonald, R., Pappas, D., Androutsopoulos, I.: AUEB at BioASQ 6: Document and Snippet Retrieval. In: Proceedings of the 6th BioASQ Workshop. pp. 30–39. Brussels, Belgium (2018)
5. Chen, Q., Allot, A., Lu, Z.: Keep up with the latest coronavirus research. Nature **579**(7798), 193 (2020)
6. Dai, Z., Callan, J.P.: Context-aware sentence/passage term importance estimation for first stage retrieval. ArXiv **abs/1910.10687** (2019)
7. Dai, Z., Callan, J.P.: Context-aware document term weighting for ad-hoc search. Proceedings of The Web Conference 2020 (2020)
8. Danesh, S., Sumner, T., Martin, J.H.: SGRank: Combining statistical and graphical methods to improve the state of the art in unsupervised keyphrase extraction. In: Proceedings of the Fourth Joint Conference on Lexical and Computational Semantics (2015)
9. Farhi, S.H., Boughaci, D.: Graph based model for information retrieval using a stochastic local search. Pattern Recognition Letters **105**, 234 – 239 (2018)
10. Garg, S., Vu, T., Moschitti, A.: Tanda: Transfer and adapt pre-trained transformer models for answer sentence selection. In: The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020. pp. 7780–7788 (2020)
11. Grover, A., Leskovec, J.: node2vec: Scalable feature learning for networks. In: KDD. pp. 855–864. ACM (2016)
12. Khattab, O., Zaharia, M.: Colbert: Efficient and effective passage search via contextualized late interaction over bert. ArXiv **abs/2004.12832** (2020)
13. Kieuvongngam, V., Tan, B., Niu, Y.: Automatic text summarization of covid-19 medical research articles using bert and gpt-2. ArXiv **abs/2006.01997** (2020)
14. Kotitsas, S., Pappas, D., Androutsopoulos, I., McDonald, R., Apidianaki, M.: Embedding biomedical ontologies by jointly encoding network structure and textual node descriptors. In: Proceedings of the 18th BioNLP Workshop and Shared Task. pp. 298–308. Association for Computational Linguistics, Florence, Italy (Aug 2019)
15. Kwiatkowski, T., Palomaki, J., Redfield, O., Collins, M., Parikh, A., Alberti, C., Epstein, D., Polosukhin, I., Kelcey, M., Devlin, J., Lee, K., Toutanova, K.N., Jones, L., Chang, M.W., Dai, A., Uszkoreit, J., Le, Q., Petrov, S.: Natural questions: a benchmark for question answering research. Transactions of the Association of Computational Linguistics (2019)
16. Lee, J., Jeong, M., Sung, M., Yoon, W., Sung, M., Choi, Y., Ko, M., Lee, S.W., Kang, J.: Answering domain-specific questions in real-time for covid-19 research. arxiv (2020)
17. Lu, Z., Li, H.: A deep architecture for matching short texts. In: Advances in Neural Information Processing Systems 26, pp. 1367–1375. Curran Associates, Inc. (2013)

18. MacAvaney, S., Cohan, A., Goharian, N.: Sledge: A simple yet effective baseline for coronavirus scientific knowledge search. ArXiv **abs/2005.02365** (2020)
19. MacAvaney, S., Nardini, F.M., Perego, R., Tonellotto, N., Goharian, N., Frieder, O.: Efficient document re-ranking for transformers by precomputing term representations. ArXiv **abs/2004.14255** (2020)
20. Malkov, Y.A., Yashunin, D.A.: Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. Computing Research Repository (CoRR) **abs/1603.09320** (2016)
21. Malliaros, F.D., Vazirgiannis, M.: Graph-based text representations: Boosting text mining, NLP and information retrieval with graphs. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing: Tutorial Abstracts. Association for Computational Linguistics, Copenhagen, Denmark (Sep 2017)
22. Mass, Y., Carmeli, B., Roitman, H., Konopnicki, D.: Unsupervised FAQ retrieval with question generation and BERT. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 807–812. Association for Computational Linguistics, Online (2020)
23. McDonald, R., Brokos, G.I., Androutsopoulos, I.: Deep Relevance Ranking Using Enhanced Document-Query Interactions. In: Proc. of the Conf. on Empirical Methods in Natural Language Processing. Brussels, Belgium (2018)
24. Mohan, S., Fiorini, N., Kim, S., Lu, Z.: A fast deep learning model for textual relevance in biomedical information retrieval. Computing Research Repository (CoRR) **abs/1802.10078** (2018)
25. Nogueira, R., Cho, K.: Passage re-ranking with BERT. Computing Research Repository (CoRR) **abs/1901.04085** (2019)
26. Ozyurt, I.B., Bandrowski, A., Grethe, J.S.: Bio-answerfinder: a system to find answers to questions from biomedical texts. Database : the journal of biological databases and curation **2020** (January 2020)
27. Padigela, H., Zamani, H., Croft, W.B.: Investigating the successes and failures of BERT for passage re-ranking. Computing Research Repository (CoRR) **abs/1905.01758** (2019)
28. Pappas, D., Brokos, G., McDonald, R., Androustopoulos, I.: Aueb at bioasq 7: Document and snippet retrieval. In: BioASQ (2019)
29. Pappas, D., Stavropoulos, P., Androutsopoulos, I., McDonald, R.: BioMRC: A dataset for biomedical machine reading comprehension. In: Proceedings of the 19th SIGBioMed Workshop on Biomedical Language Processing (2020)
30. Roberts, K., Alam, T., Bedrick, S., Demner-Fushman, D., Lo, K., Soboroff, I., Voorhees, E.M., Wang, L.L., Hersh, W.R.: Trec-covid: Rationale and structure of an information retrieval shared task for covid-19. Journal of the American Medical Informatics Association : JAMIA (2020)
31. Roberts, K., Demner-Fushman, D., Voorhees, E.M., Hersh, W.R., Bedrick, S., Lazar, A.J., Pant, S., Meric-Bernstam, F.: Overview of the trec 2017 precision medicine track. In: TREC (2017)
32. Rospocher, M., Corcoglioniti, F., Dragoni, M.: Boosting document retrieval with knowledge extraction and linked data. Semantic Web **10**, 753–778 (2019)
33. Su, D., Xu, Y., Yu, T., Siddique, F.B., Barezi, E.J., Fung, P.: Caire-covid: A question answering and multi-document summarization system for covid-19 research. ArXiv **abs/2005.03975** (2020)
34. 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, A., Heino, N., Gaussier, E., Barrio-Alvers, L., Schroeder, M., Androutsopoulos, I., Paliouras, G.: An overview of the BioASQ Large-Scale Biomedical Semantic Indexing and Question Answering Competition. BMC Bioinformatics **16**(138) (2015)

35. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L.u., Polosukhin, I.: Attention is all you need. In: Advances in Neural Information Processing Systems 30, pp. 5998–6008. Curran Associates, Inc. (2017)
36. Wang, L., Luo, Z., Li, C., He, B., Sun, L., Yu, H., Sun, Y.: An end-to-end pseudo relevance feedback framework for neural document retrieval. Information Processing and Management **57**(2) (2020)
37. Wang, L.L., Lo, K., Chandrasekhar, Y., Reas, R., Yang, J., Eide, D., Funk, K., Kinney, R., Liu, Z., Merrill, W., Mooney, P., Murdick, D., Rishi, D., Sheehan, J., Shen, Z., Stilson, B., Wade, A.D., Wang, K., Wilhelm, C., Xie, B., Raymond, D., Weld, D.S., Etzioni, O., Kohlmeier, S.: Cord-19: The covid-19 open research dataset (2020)
38. Wei, C.H., Allot, A., Leaman, R., Lu, Z.: PubTator central: automated concept annotation for biomedical full text articles. Nucleic Acids Research **47**(W1), W587–W593 (2019)
39. Xu, P., Ma, X., Nallapati, R., Xiang, B.: Passage ranking with weak supervsion. ArXiv **abs/1905.05910** (2019)
40. Yan, Y., Yin, X.C., Zhang, B.W., Yang, C., wei Hao, H.: Semantic indexing with deep learning: a case study. Big Data Analytics **1**, 1–13 (2016)
41. Yang, W., Zhang, H., Lin, J.: Simple applications of bert for ad hoc document retrieval. Computing Research Repository (CoRR) **abs/1903.10972** (2019)
42. Yang, Y., Yih, W.t., Meek, C.: WikiQA: A challenge dataset for open-domain question answering. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. pp. 2013–2018. Association for Computational Linguistics, Lisbon, Portugal (2015)
43. Yoon, S., Dernoncourt, F., Kim, D.S., Bui, T., Jung, K.: A compare-aggregate model with latent clustering for answer selection. In: Proceedings of the 28th ACM International Conference on Information and Knowledge Management. p. 2093–2096 (2019)
44. Zamani, H., Dehghani, M., Croft, W.B., Learned-Miller, E., Kamps, J.: From neural re-ranking to neural ranking: Learning a sparse representation for inverted indexing. In: Proceedings of the 27th ACM International Conference on Information and Knowledge Management (2018)
45. Zhang, Z., Wang, L., Xie, X., Pan, H.: A graph based document retrieval method. In: 2018 IEEE 22nd International Conference on Computer Supported Cooperative Work in Design (CSCWD). pp. 426–432 (2018)
46. Zhao, S., Su, C., Sboner, A., Wang, F.: Graphene: A precise biomedical literature retrieval engine with graph augmented deep learning and external knowledge empowerment. Proceedings of the 28th ACM International Conference on Information and Knowledge Management (2019)