Deep Learning solutions based on fixed contextualized embeddings from PubMedBERT on BioASQ 10b and traditional IR in Synergy

Tiago Almeida[1], André Pinho[1], Rodrigo Pereira[1] and Sérgio Matos[1]

[1] *University of Aveiro, IEETA*

Abstract

This paper presents the participation of the University of Aveiro Biomedical Informatics and Technologies group (BIT) in the tenth edition of the BioASQ challenge for document retrieval (task B phase A and Synergy) and 'yes or no' answering (task B phase B). For the Synergy task, we adopted a relevance feedback approach that leveraged the traditional BM25 retrieval model combined with query expansion based on the positive documents. Regarding task B phase A, we adopted a two-stage retrieval pipeline that consists of the traditional BM25 retrieval with pseudo-relevance feedback followed by a neural retrieval model. For the neural models, we experimented with the publicly available Transformer-UPWM already trained on last year's BioASQ data and with a local implementation of the PARADE-CNN model. Lastly, for the 'yes or no' questions, we trained a binary classifier over the pretrained PubMedBERT model, and also studied the impact of data augmentation and dataset balancing techniques. In terms of results, our Synergy and task B phase B systems underperformed, scoring below the average. Our best results came from the task B phase A systems that achieved above-average results, being in the top three in terms of teams. Furthermore, after the challenge, we also conducted additional experiments to evaluate the impact of the Transformer-UPWM when trained on 10b data. This trial produced an improvement of 1 to 5 MAP percentage points in all official evaluation batches. Code to reproduce our submissions are available on https://github.com/bioinformatics-ua/BioASQ-10.git.

Keywords

Neural ranking, Document Retrieval, Binary classification, BioASQ 10B, Relevance feedback, Synergy

1. Introduction

The rapid growth of biomedical literature, through open or peer reviewed publications, creates a situation of information overload that affects researchers, doctors or any other health care practitioner. According to Klerings et al. [1], the main problem is not the abundance of information, but rather the lack of more capable information retrieval systems that can effectively search this continually growing literature.

The BioASQ challenge [2] is an annual competition that promotes the development of such intelligent retrieval systems that can aid with the current problem of information overload.

CLEF 2022: Conference and Labs of the Evaluation Forum, September 5–8, 2022, Bologna, Italy

tiagomeloalmeida@ua.pt (T. Almeida); andre.s.pinho@ua.pt (A. Pinho); rodrigo.pereira@ua.pt (R. Pereira); aleixomatos@ua.pt (S. Matos)
https://t-almeida.github.io/online-cv (T. Almeida)
0000-0002-4258-3350 (T. Almeida); 0000-0003-1919-5058 (A. Pinho); 0000-0002-5178-7056 (R. Pereira); 0000-0003-1941-3983 (S. Matos)

Currently, the challenge is in its tenth edition and is divided into tasks **A** and **B** plus the recent **Synergy** task. More concretely, task **A** is concerned with biomedical semantic indexing, while task **B** and **Synergy** focus on information retrieval and question answering. Furthermore, task **B** is additionally subdivided into phase A and phase B, where the first is concerned with finding the relevant documents/snippets that answer a given biomedical question (IR) and the later is concerned with answer extraction and generation (QA). The aim of tasks **A** and **B** is the development of state-of-the-art systems that can find evidence or answer an open biomedical question, while the primary objective of the Synergy task is in using IR and QA systems for collectively finding answers to open questions about Covid-19.

This paper describes the participation of the Biomedical Informatics and Technologies (BIT) group of the Aveiro University in BioASQ task B and BioASQ Synergy. For task B, our approaches rely on the fixed contextualized embeddings produced by the PubMedBERT model. More precisely, for phase A we adopted the PARADE-CNN [3] and Transformer-UPWM [4] models. The first of these two models was trained on this year's training data, while for the latter we re-used the model trained on last year's data. This produced an interesting duality of new training data against one year old training data that we later explored in a post-challenge experience (Section 4). For phase B we trained a binary classifier to answer 'yes or no' type questions, while experimenting with data augmentation through a paraphrasing technique and dataset balancing. Regarding the Synergy task, we followed a simple, yet successful, relevance feedback approach [5], where we used positively annotated documents to expand the current search query that was fed to the traditional BM25 retrieval model.

The remainder of the paper is organized as follows. In Section 2 we present the datasets and corpora used, followed by a detailed description of each method. The corresponding submissions and results are presented and discussed in Section 3. Then, in Section 4 we analyze the impact of training the Transformer-UPWM in BioASQ 9b data against 10b data. We finish the paper with a conclusion in Section 5.

2. Materials and Methods

This section starts by presenting corpora used in each task (**B** and Synergy). Then it presents, for each task that we participated, the dataset preparation, the method and the official submissions.

2.1. Corpora

As previously mentioned, the Synergy task aims to develop systems capable of answering Covid-19 related questions, while task **B** focuses on the development of more broad systems that can answer any biomedical type of question. Therefore, the Synergy questions were envisioned to be answered using the CORD-19 [6] dataset, while task **B** uses articles from PubMed/MEDLINE dataset as collection to answer the given biomedical questions.

2.1.1. CORD-19

CORD-19 dataset is a joint effort by several research groups in response to the Covid-19 pandemic [6]. It aggregates several academic papers related to the topic of Covid-19 and general coronavirus research from peer reviewed sources such as PubMed, the World Health Organization, and open source archives such as bioRxiv, medRxiv and arXiv, in a structured manner that facilitate the creation of text mining and information retrieval systems, which can help the scientific community better understand this virus. Nowadays, the CORD-19 dataset is updated on a weekly basis, which means that each round of the Synergy task used a different snapshot of the CORD-19 dataset.

Each entry of this dataset represents a publication, described by multiple fields. For our task, we only focus on the title, abstract, PubMed ID, PubMedCentral ID and arXiv ID.

2.1.2. MEDLINE/PubMed

MEDLINE [7] is a database that aggregates abstracts and metadata from biomedical literature from several sources, while PubMed [8] is a free searching engine over the MEDLINE data. The PubMed index is updated on daily basis and a baseline snapshot is produced at the end of each year. In this year's challenge task **B** used the 2022 PubMed baseline as the source of documents.

Similar to CORD-19, each datapoint describes a publication composed of multiple fields. The subset of these fields useful to us are the article title, the abstract and the PubMed ID.

2.2. Synergy

The Synergy task had four discrete moments of evaluation, also called rounds, where the organizers provided a set of unanswered questions related to Covid-19, and teams had three days to submit their systems' answers. Table 1 shows a summary of the number of questions made available in each round and the total number of feedback documents from the previous round. These are documents that either were consider to be positive or negative. Finally, in this task we only submitted results regarding the document retrieval task.

Table 1
Statistics of the testset provided by the organizers of the Synergy task. It presents the number of questions that were made available in each round and in parenthesis the number of new queries relatively to the previous set of questions. It also shows the total number of feedback documents that were available to each question

	Round 1	**Round 2**	**Round 3**	**Round 4**
Nr of Questions	72 (+72)	70 (-2)	70 (+0)	64 (-6)
Nr of Feedback Documents	-	17820	20601	24268

2.2.1. Dataset Preparation

Upon inspecting the CORD-19 dataset, we concluded that it contains some datapoints that need further treatment. Some of the issues present are unstructured abstracts, abstracts in foreign languages and even papers without an abstract. Considering these problems, we created a module to preprocess and structure the input dataset.

The first step of this process is to remove all articles that don't contain a title, abstract or at least one of the three identifiers (PubMed ID, PubMed Central ID, arXiv ID). Afterwards, considering that very short abstracts contained irrelevant information, we filter out all the articles whose abstracts contain less than six words. We also observed that multiple articles were written in other languages (such as Chinese). Unfortunately our models are not designed to deal with multi-languages, even less with a different character set. To filter these, we leveraged FastText [9] using the model `lid.176.bin`.

Having done basic filtering, the module advances to a more fine filtering. Each of the remaining abstracts are split into a list of sentences. To perform this task we used the NLP module of the Spacy Python library [10]. In the case that some non-English characters passed through the initial removal of foreign languages (for example abstracts where the main language was in fact English, but that did contain characters from another language), they are deleted.

We also noticed that multiple articles contained URLs scattered in the abstract. Once again, our model is not designed to deal with this, so we filter them out using a regular expression. At the end, all of these steps contributed to a total reduction of roughly 58% of the total number of documents in the collection. Note that this number slightly changed from round to round since the CORD-19 was updated continually.

As an additional step, besides indexing the documents as a whole, we also indexed directly the individual sentences of the documents to have a more fine grain control of the search. However, when looking at the training data, it is observable that most of the relevant snippets are composed of a sequence of sentences. Therefore, we implemented a sentence merging mechanism in order to concatenate some of the sentences. More specifically, adjacent sentences are merged until the sum of the sizes does not exceed 80 characters. This also drastically increases the efficiency of the neural rerankers.

In terms of the training data, these correspond to last year's questions to the BioASQ 9b Synergy v1 and v2 challenge and their respective feedback data. Regarding data preparation, some duplicate questions were identified and merged, and their answers were combined. Afterwards, we also identified that for some questions there were documents with inconsistent labeling. In other words, for a question, the same document could be labeled as both relevant and irrelevant. In those cases we considered these documents to be irrelevant (as opposed to unknown or relevant).

Furthermore, considering that articles can be revoked from CORD-19 on each update, it was necessary to guarantee that each of the relevant documents for a specific question was still present in the recent version of CORD-19. If this was not the case, those document references were removed from the feedback data. Finally, questions that did not have any associated relevant documents were also removed. The final clean training set contained 175 questions, each with 28.8 positive documents and 149.5 negative documents on average.

2.2.2. Method

Our main method adopted for the Synergy task relies on a relevance feedback approach, since in [5] it was shown to be a simple but quite effective technique. Therefore, our method directly follows the ideas presented in [5], however, with a different implementation. More concretely, we adopted the Pyserini [11] (Python wrapper of Anserini [12]) information retrieval toolkit, instead of the Elastic Search (ES) engine, due to its increasing popularity within the information retrieval community and, from our experience, better out-of-the box standard retrieval models.

Similarly to [5], we used the BM25 probabilistic ranking model with query expansion to find documents that answer biomedical questions. Here, the query expansion acts as the relevance feedback mechanism, where the intuition is to introduce the most important terms from the set of positive documents into the original query. Internally, the Anserini uses the rm3 [13] relevance model to perform the query expansion. First, this model tries to estimate, from a set of positive feedback data, $\mathcal{R}$, the importance of each document term. Then all of the terms are normalized and sorted, where the top M are selected to compose the new query. Additionally, in order to maintain the original query information, the new query weights are linearly interpolated with the original query weights, which ensures that the original query terms are present in the new query. After this step, a new BM25 search is performed, where the previously computed query weights are kept. For a more complete description framed within the probabilistic framework, refer to Section 3.1.4 in [13]. Finally, it is important to mention that the rm3 relevance model can be used for relevance or for pseudo-relevance feedback, where in the first case the set $\mathcal{R}$ is provided by the data, as explicit feedback, while in the later case $\mathcal{R}$ corresponds to the top-N retrieved by the BM25 ranking model over the original query. Recalling Table 1, some of the test queries did not have feedback data and in those cases we performed pseudo-relevance feedback.

Regarding the positive feedback data, $\mathcal{R}$, the obvious choice is to use the documents that are classified as relevant for each query. But as suggested in [5], we also used the snippet sentences as the positive feedback data, since, when comparing to a whole document, these are more focused on the answer. In order to find the best set of parameters for $\mathcal{R}$, M, $k1$ and b we used Optuna [14] to search a good combination over the training data.

Besides this method we also tried to train the PARADE-CNN [3] ranking model (refer to Section 2.3.2 for more details) to rerank the top-100 documents retrieved by the BM25. However, we did not achieved satisfactory results and hence we did not submit any run with this system. Nevertheless, for sake of completeness we present on Table 2 our preliminary benchmark, in terms of MAP@10, between the BM25 and the PARADE-CNN calculated on the same validation subset. We believe that the scarcity of the data and the difficulty of the questions are the main reason for the under-performance of the PARADE-CNN reranking model.

Table 2
Comparison of the PARADE-CNN model trained on the Synergy training dataset against the finetuned BM25 (pyserini)

Method	Map@10
BM25 (pyserini)	0.1924
PARADE-CNN [3]	0.1494

2.3. Task 10b Phase A

Task 10B Phase A had five moments of evaluation, also called batches, with an additional non-official batch at the end of the challenge. In each batch, the organizers provided a set of biomedical questions for which the teams had 24 hours to find the top-10 most relevant documents (document retrieval task) and/or to also retrieve the top-10 most relevant snippets extracted from relevant documents (snippet retrieval task). Each batch contained 90 new questions, except the sixth batch which only contained 37.

2.3.1. Dataset Preparation

The training dataset for Task 10B consists of all of the previous year's questions and their respective relevant information that was annotated by experts. However, since in each year BioASQ uses different PubMed baselines as document collections this raises an issue of time inconsistency. For example, questions that belonged to Task 3B had the PubMed 2015 as a baseline, which means that only documents that where indexed in the 2015 baseline were consider for retrieval. So, using those questions in the 2022 baseline may produce several inconsistencies. For example, the article identifier may have changed, the article may have been removed, the article content may have changed and new articles that answer that question could have been added. We believe that all of these inconsistencies must be addressed in order to produce a clean set of training data. To solve this issue, our solution was to ensure that each question would only use their corresponding baseline. To accomplish that, we firstly downloaded all the baselines since 2013 and, after that, we iterated through these baselines and generated a data structure which contained the most recent version of the abstract and title of each article and kept record of all the baselines in which that article appeared on. Then, when searching, we filter out the documents that did not appear in that respective question baseline. This approach ensures that each training question only sees the documents that were available at the time they were written. Figure 1 shows, for each year's question set, a distribution of the removed articles from the top-150 BM25 retrieval. As observable, the older questions (1B, 2B) have a larger amount of documents that needed to be removed, since these are documents that were not available at the time that the gold-standard were produced, hence its relevance value is unknown.

Regarding the preprocessing of the documents, it uses the sentence splitting (using the same approach as in Synergy) and merging strategy that was also applied to the Cord-19 dataset, however for this particular case, the foreign charset and URL cleaning was not necessary.

2.3.2. Method

For this task we decided to use a two-stage retrieval system, where we adopted the traditional BM25 with rm3 (in pseudo-relevance feedback mode) as the first stage ranker and then the PARADE-CNN [3] model as a top-100 reranking model. Additionally, we also tried the BioASQ 9B Transformer-UPWM [4] model to rerank the same top-100 documents as a way of comparison.

Transformer-based architectures have emerged to the top, as the most adopted models for almost every NLP task, inclusively information retrieval. However, when considering the subtask of document retrieval it still remains a challenge to use these type of models, due to the limited number of tokens that they can process, which are bound by the quadratic complexity order of the self-attention mechanism. To alleviate this problem, it is practical to divide the input document into a set of sentences and then use a transformer model to compute a query-sentence relevance score, that are later aggregated to compute the final query-document relevance score.

Similarly to this paradigm, Li et al. propose a set of PARADE models that use a BERT based

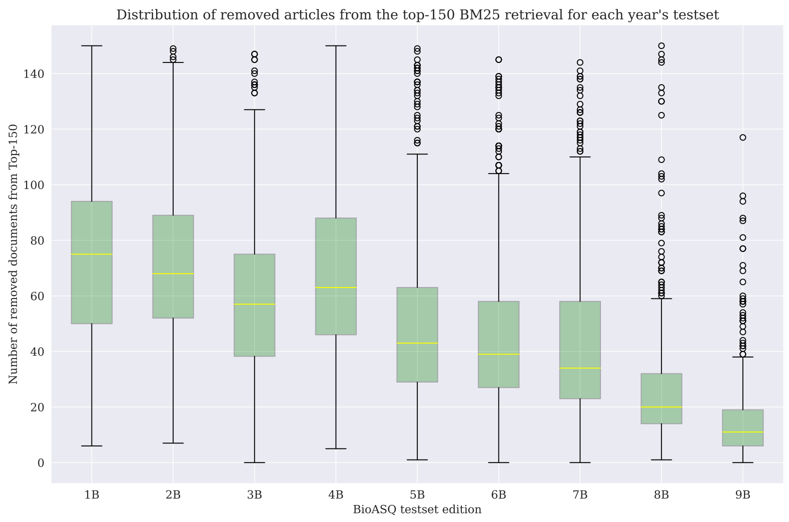

Figure 1: Distribution of the removed articles from the top-150 BM25 retrieval for each year's testset. More precisely, for each year's edition testset we retrieve the top-150 most relevant articles using the BM25 from the 2022 PubMed baseline. Then from those 150, we removed the documents that did not exist at that time.

model to encode query-sentence relevance in a highly dimensional space (reflected in the CLS token). Then the authors tested several aggregation strategies, like a MultiLayer Perceptron (MLP), 2-Layer Transformer, and CNN to produce the final document score from the set of query-sentences CLSs. We decided to adopt the PARADE-CNN model, presented in Figure 2, based on the reported results and on the fact that the CNN kernel ($d \times 2$ size, where d corresponds to the CLS dimension, corresponding to a 1D CNN) can easily encode the relative positions of sentences. The downside of this was that the PARADE-CNN was the only model that did not have an open-source implementation at that time. Therefore, we made a local implementation of the model in Tensorflow. Furthermore, in contrast to the original model, we used PubMedBERT as the transformer query-sentence encoder and during train we did not update its weights. We believed that this could harm the overall performance, but we were also interested in seeing if the out-of-the-box projections were good enough for this task. Additionally, we did not possess the hardware to do such training.

Taking into consideration that we made a local implementation of the PARADE-CNN, we also decided to adopt the Transformer-UPWM [4] model that was already trained in the BioSAQ 9b data. This way we could benchmark against the PARADE-CNN and have an idea if the PARADE-CNN implementation was working as expected, since we believe that the results would be similar.

The Transformer-UPWM (Figure 3) is, as the name suggests, a transformer-based model that follows the same ideology of dividing a document into sentences, processing each individual query-sentence pair using a transformer and then aggregating them. However, differently to the

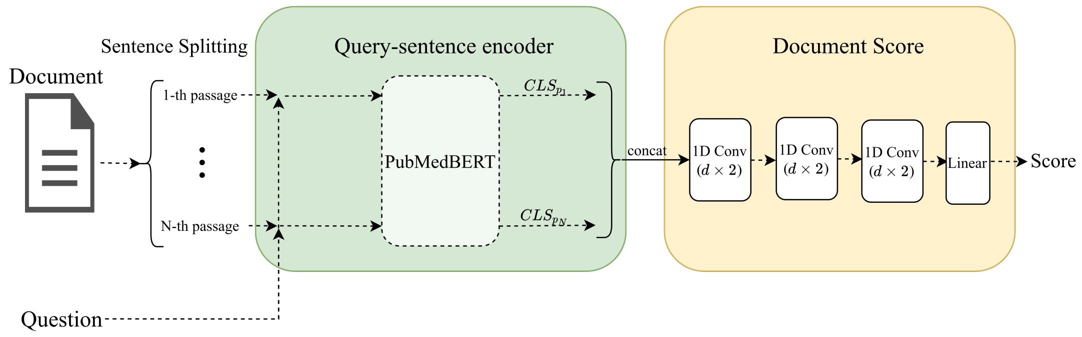

Figure 2: A general overview of the PARADE-CNN architecture.

PARADE-CNN, this model does not encode the query-sentences pair to a highly dimensional space, but instead tries to predict a query-relevance score, bounded between 0 and 1. This model is inspired by the way humans search for information [15, 16]. More precisely, a person usually scans a document for regions of interest, usually denoted by relevant keywords, then reads it and finally produces a mental score of relevance. Therefore, to compute the query-sentence score the model considers the query-sentence relevance score produced by a classifier layer over the PubMedBERT model and the query-sentence relevance score produced by the "*a priori*" layer. In other words, the "*a priori*" layer tries to find the regions of interest by query-sentence embedding matching and relative query term importance. For more details about the architecture, consider our previous work [4].

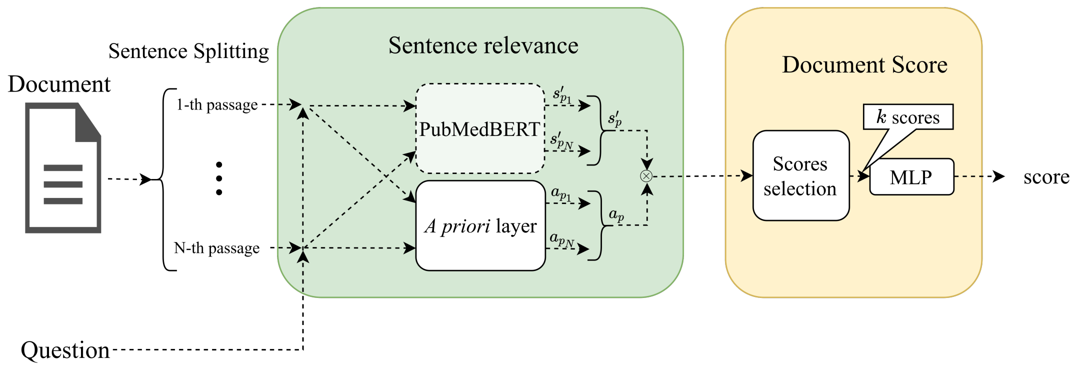

Figure 3: A general overview of the Transformer-UPWM architecture.

To train the PARADE-CNN model we used the standard AdamW optimizer with the pairwise cross-entropy loss. Regarding the negative sampling, we consider as pool of negatives every document that does not belong to the set of positive documents and as hard negative every negative document that belongs to the top-100 BM25 ranking order. We tried some negative sampling strategies, but found that training only with hard negatives seems to be the most stable approach. Note that the Transformer-UPWM was already pre-trained on BioASQ 9b data so it was not trained at this phase. The reasons for not training the Transformer-UPWM model were two fold. The first was that we wanted to focus our efforts on making sure that the

PARADE-CNN model was working correctly, while the other reason was that we also wanted to see how a one year old model, trained on a one year old dataset handled the new questions provided by the organizers.

2.4. Task 10b Phase B

Similarly to Phase A, Phase B also had six moments of evaluation, where the sixth was a non-official batch. Here, the teams also had 24 hours to answer questions of four types, namely 'yes or no', 'list', 'factoid' and 'summary'. Due to time constrains we only participated in the 'yes or no' subtask.

2.4.1. Dataset Preparation

Regarding the training data, we used a combination of the testsets from the previous editions of BioASQ task B. The data consists of a list of questions and the corresponding collections of snippets, obtained from multiple documents, capable of answering each question. From that collection of questions we first extracted the ones of type "yesno". Associated with each 'yes or no' question, a boolean label is also given to represent the correct answer (Yes/No) of that question. To process this dataset, we pair every question (and classification label) with each snippet. These are then processed, expanded and filtered in several steps as explained next.

In an attempt to increase the size of the dataset, we explored the use of paraphrasing tools. Initially, we performed tests with T5 Sentence Paraphrasing Model [17]. However, after a preliminary analysis we opted for the use of QuillBot [18], which offered better results. Also based on a deep learning architecture, this tool was capable of analyzing text segments and generating legitimate alternative wordings while preserving the original internal meaning. This method was applied to increase the number of questions, and corresponding answering snippets. In order to fairly distribute the generated snippets and questions, their pairings are shuffled. This means that for a question (may it be a paraphrased or original one), the list of snippets corresponds to the mixture of original and paraphrased versions.

Previous works in this task describe the use of down sampling strategies based on the unbalance between "yes" and "no" questions, with the original dataset being heavily skewed towards "yes" questions [19]. This down sampling however, presents a problem in itself, namely the fact that not only is the dataset skewed towards "yes" question, but also that "yes" questions have a higher average of "corroborative snippets", meaning that the number of input pairs Question + Snippet will be unbalanced. As a preventive measure we decided to be more aggressive in the down sampling of "yes" questions making it so that the unbalanced ratio between "yes" and "no" questions offsets the difference in number of average "corroborative snippets".

The balanced data obtained was then split into training and validation sets. Different ratios where experimented, from very small validations sets, that mimicked the size of the test sets provided by BioASQ, and at the same time provided a larger training set to the model, to more conventional split ratios (80%/20%) that gave us a better understanding of whether the model was generalizing.

2.4.2. Method

Our approach to this particular task was to perform a more traditional classification method, centered on the PubMedBERT deep-learning model (Figure 4). The datapoint input composition of our model corresponds to the pairing between one question and one corresponding snippet. This datapoint is fed to the PubMedBERT model (using the two phrase input option), and its CLS output is retrieved. This CLS is then processed by a configurable amount of dense layers. The last dense layer of the model has only one output that serves as the pair score to be used for classification. If this value is less than 0.5 then we consider the answer to be "no", and "yes" otherwise. Earlier experiments indicated that enabling the training of the internal BERT layers closest to the output could be beneficial. As such, we also considered this option as an hyperparameter. For the training setup of the model we used a binary cross entropy (BCE) loss function paired with the AdamW optimization strategy.

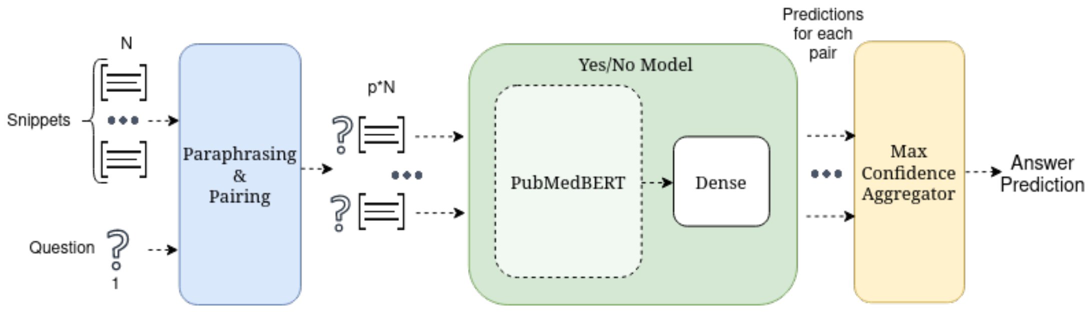

Figure 4: A general overview of the Task 10b phase B Yes/No architecture.

2.4.3. Model Prediction and Optimization

Given our initial approach of using Question + Snippet pairings as the model input, it was necessary for us to create mechanisms that allowed to obtain the classification of a single question regarding all their snippet pairings. To this end two different approaches were implemented: Majority voting and Max confidence. The majority voting strategy reaches a consensus based on the most frequent label (yes or no) for that question. On the other hand the Max Confidence strategy follows a less democratic approach, where only the *(question, snippet)* pair with highest confidence in its classification (value most close to 1 for yes or 0 for no) is used as the final classification for the correspondent question.

The performance of the model is dependent on a number of variables, from the splitting approach, to the use of paraphrasing, to the model architecture and its hyperparameters. As such we tested various combinations for these variables, while tracking their validation performance. To that end we made use of one of many machine learning tracker tools (Weights and Biases [20]) in order to log and more easily compare them.

3. Results and Discussion

In this section, for each BioASQ task, we present a description of the systems that we submitted, their corresponding results followed by a discussion.

3.1. Synergy

Regarding the submissions, due to time constrains we only enrolled in the third and fourth rounds of the Synergy challenge. Furthermore, after the third round submission we noticed that our generated outputs erroneously contained the known positive documents, which were fed as feedback data to our method. This strongly affected our results, as these documents do not count for the ranking metrics. Table 3 shows a summary with the systems description for each of the rounds, where the main difference was the source of the positive feedback data, $\mathcal{R}$, used. System 0 directly used the whole documents as the feedback data, while System 2 uses snippets. System 1 and 3 are ensemble runs produced by the rank reciprocal fusion (RRF) [21] method, from slightly variations of hyperparameters on individual runs from the System 0 and System 2, respectively. Finally, System 4 is an RRF run that combines System 1 and System 3, which are runs produced with different sources of the positive feedback data (documents and snippets).

Table 3
Summary containing the description of the submitted systems to the BioASQ 10 Synergy challenge

System	Round 3	Round 4
System 0	$\mathcal{R}$ = Documents	$\mathcal{R}$ = Documents
System 1	RRF -> System 0	RRF -> System 0
System 2	$\mathcal{R}$ = Snippets	$\mathcal{R}$ = Snippets
System 3	RRF -> System 2	RRF -> System 2
System 4	-	RRF -> System 1 and System 3

The official results for each system are presented in Table 4, with the scores achieved by the first place team presented at the bottom. As previously mentioned, the results of round 3 are under representatives. Nevertheless, it is interesting to see that the systems that used the snippets as source of feedback data achieved better results when compared to using the documents as source of feedback data. This result is indicative that when using the snippets as feedback data, the retrieval system did not score the corresponding positive documents as higher, suggesting that the query expansion is less biased towards the positive documents. Regarding the fourth round results, our best submission scored close to the best participation teams, being only 0.0251 MAP percentage points away of the top score. Another interesting result was the fact that here the systems that used the documents as source of feedback data achieved slightly higher results than those using the snippets. This results contradicts the results achieved by [5] who achieved the best results when using snippets as feedback data. We believe the main reason behind this is due to the difference in the relevance feedback method used. More precisely, we used the rm3 relevance model to produce a new weighted query, while [5] uses ES that implements a more simplistic query expansion mechanism. Additionally, it was also clear that using RRF resulted in minimal improvements when similar runs are combined and a much larger improvement when we combined runs that used different source of positive feedback data (System 4).

Table 4
Batch performance (Rank, Mean Average Precision) of submitted models for the Synergy challenge. Note, that we had an issue with the third round submission, where the known positive feedback documents were not removed from the final ranking order

System	Round 3		Round 4	
	#	MAP@10	#	MAP@10
System 0	17	0.0745	11	0.1244
System 1	16	0.0764	9	0.1288
System 2	11	0.1122	18	0.1152
System 3	10	0.1158	16	0.1192
System 4	-	-	7	0.1446
Top Competitor	1	0.2622	1	0.1697

3.2. Task 10B Phase A

For the phase A challenge we enrolled in all batches, although in the first batch we could only submit runs using the Transformer-UPWM model trained on last year's data and not with the PARADE-CNN model. Table 5 shows a summary containing the characteristics of each submitted system.

Table 5
Summary containing the description of the submitted systems to the BioASQ 10 B phase A challenge

System	Batch 1	Batch 2	Batch 3, 4, 5 and 6
System 0	RRF -> T-UPWM	RRF -> T-UPWM	RRF -> T-UPWM
System 1	RRF -> T-UPWM	RRF -> T-UPWM	RRF -> T-UPWM
System 2	RRF -> T-UPWM	RRF -> T-UPWM + PARADE-CNN	RRF -> T-UPWM + PARADE-CNN
System 3	RRF -> T-UPWM	RRF -> T-UPWM + PARADE-CNN	RRF -> T-UPWM + PARADE-CNN
System 4	-	PARADE-CNN	RRF -> T-UPWM

As observable in Table 5, we only submitted one individual run that used the PARADE-CNN (System 4 on batch 2). That decision was based on pre-challenge experiments that compared the PARADE-CNN performance trained on exactly the same data under comparable conditions against the Transformer-UPWM. The results, presented on Table 6, show that the PARADE-CNN was slightly under-performing when compared to the Transformer-UPWM. Therefore, we made one single submission using the PARADE-CNN model to confirm if these results hold in the final testset. We believe that the main reason for this difference in results is related to the fact that we did not train the encoder transformer layers and hence the pre-trained PubMedBERT model may be less sensitive to finding matching signals between the query and document. This becomes more evident when we consider that the Transformer-UPWM utilizes the same frozen transformer encoder, but has an additional layer (*a priori layer*) that focuses on finding matching evidence, hence reinforcing our belief.

Table 6
Comparison of the PARADE-CNN model against the Transfomer-UPWM model. Both models were trained in the exact same data and under similar conditions

Method	Map@10
Transformer-UPWM	0.4623
PARADE-CNN [3]	0.3787

Regarding the results, Table 7 details the achieved performance for each system submitted, compared to the top competitor at the bottom of the table. Overall, we consider our results to be above average, with some runs in the top 3 in terms of teams. Our best submissions were those that used the Transformer-UPWM model, which was trained with last year's data, which is a surprising result since we expected this would achieve a lower performance due to a slightly larger amount of training data. Given this unexpected results, we conducted an additional experiment following the challenge, where we trained the Transformer-UPWM model using this year's training data (see Section 4). Our weakest submission was based on the PARADE-CNN model, confirming our initial suspicions. However, despite the lower individual performance, it achieved the best results in batch 2 and 3 when combined with the Transformer-UPWM, which also suggests that the two models are focusing on different types of matching signals.

Table 7
Batch performance (Rank, Mean Average Precision) of submitted models for the Phase A Document retrieval challenge

Model	Batch 1		Batch 2		Batch 3		Batch 4		Batch 5		Batch 6	
	#	MAP	#	MAP	#	MAP	#	MAP	#	MAP	#	MAP
System 0	10	0.4098	14	0.3186	6	0.4352	15	0.3404	8	0.3734	17	0.0862
System 1	8	0.4130	12	0.3223	8	0.4290	14	0.3502	10	0.3706	18	0.0854
System 2	11	0.4059	11	0.3339	9	0.4290	12	0.3552	11	0.3704	19	0.0854
System 3	12	0.4042	13	0.3199	13	0.4174	10	0.3613	9	0.3718	21	0.0806
System 4	-	-	19	0.2813	10	0.4279	13	0.3519	6	0.3760	14	0.0928
Top Competitor	1	0.4805	1	0.3977	1	0.5063	1	0.4058	1	0.4154	1	0.1704

3.3. Task 10B Phase B Yes/No

During the challenge, we submitted five different models, four of those following the architecture described in Section 2.4.2, each with slightly different hyperparameters (see Table 8 for an overview on the differences). The last model is an ensemble model, that aggregates the results of several variations of our model using a majority voting strategy (the output label is given based on the most common predicted label across the ensembled models).

Although we submitted systems for all six batches, in this manuscript we only focus on the last four. This is due to the first two batches our model not including paraphrasing, which we determined to be beneficial according to the experiment shown in Table 9.

The paraphrasing strategy adopted was effective, resulting in an increase of 3 percentage points in F1 score when comparing the best performing model without paraphrasing data

Table 8
Hyperparameter configurations of the submitted models for the Yes/No challenge

Model	Learning Rate	Batch Size	Nr epochs	Nr trainable BERT layers	Nr dense layers	Size dense layers	Dropout
System 1	0.001	64	15	0	1	[768]	0.2
System 2	0.0001	64	10	5	1	[768]	0.4
System 3	0.0005	64	15	0	1	[192]	0.4
System 4	0.0001	64	10	7	1	[768]	0.4

Table 9
Experiment on the use of paraphrasing data augmentation on the Yes/No challenge

Model	Validation F1 w/ Majority Voting
No Paraphrasing	0.8476
Paraphrasing	0.8814

augmentation against the best performing model that used paraphrasing, with a majority voting approach. Considering this, we opted to use paraphrasing in the subsequent batches. The performance reached in these is shown on Table 10.

Table 10
Batch performance (Rank, Macro F1) of submitted models for the Yes/No challenge

Model	Batch 3		Batch 4		Batch 5		Batch 6	
	#	M.F1	#	M.F1	#	M.F1	#	M.F1
System 0	26	0.6528	21	0.7723	23	0.7846	21	0.4857
System 1	25	0.7024	29	0.6581	28	0.7497	22	0.4857
System 2	24	0.7029	20	0.8634	21	0.8212	2	1.0000
System 3	28	0.6362	23	0.7474	22	0.8212	28	0.3333
System 4	27	0.6528	22	0.7723	24	0.7846	3	1.0000
Top Competitor	1	1.0000	1	1.0000	1	0.93	1	1.0000

Overall, the results obtained from our submitted systems are placed within or slightly bellow the average on all batches. The exception to this pattern is the sixth batch, where two of our systems placed within the first three spots. These last results were surprisingly high, since we did not perform any modifications to the system between the batches. Being this last round a non official batch, we believe that this unexpected results can be attributed to a different data distribution when compared to the official batches.

4. Post-Challenge results

As previously mentioned, we present on Table 11 a comparison of the Transformer-UPWM model that was trained on the BioASQ 9B data, against a Transformer-UPWM that was trained on this year's data. For both models we trained 6 instances that were used to report the best, average and ensemble scores in terms of MAP@10 measured on the BioASQ 10b batch 5 testset.

Furthermore, both models used the same training hyperparameters, which were finetuned for the BioASQ 9b dataset and as such may present some bias towards that data.

Table 11
Direct comparison of last year's Transformer-UPWM model (BioASQ 9b), with the same model trained with this year's training data (BioASQ 10b). We computed the MAP@10 scores, over the BioASQ 10B Batch 5 testset, of 6 models trained with the 9b and 10b datasets, and report the best individual score, the average and standard deviation and an ensemble performance with RRF

Data	Best individual	Average	Ensemble (RRF)
BioASQ 10b	0.3823	0.3792 (0.004)	0.3834
BioASQ 9b	0.3776	0.3670 (0.008)	0.3706

As previously suspected, training the model with the BioASQ 10b gave a slight boost in terms of the results. Even more interestingly, this experience enable us to conclude that the addition of roughly more 500 biomedical questions (difference between BioASQ 9b and BioASQ 10b) only gave us a boost of 1 percentage point in terms of MAP@10. This makes us question if the BioASQ phase A has entered the phase of marginal gains, where exponential more annotated data would be needed to consistently improving the state-of-the-art.

After this experiment, for the sake of completeness, we also present, in Table 12, the results of the Transformer-UPWM model trained on the BioASQ 10b data, which we defined as system 0, against the same model trained on the BioASQ 9b data, defined as system 1, for all of the evaluation batches. As observable, system 0 consistently outperforms system 1,

Table 12
Performance results (Rank, MAP@10) on the five official batches of RRF runs produced from 6 T-UPWM models trained on BioASQ 10b and 9b data. Note that the run "RRF-> T-UPWM (BioASQ 9b)" corresponds to the System-2 from Table 7

System	Batch 1		Batch 2		Batch 3		Batch 4		Batch 5	
	#	MAP	#	MAP	#	MAP	#	MAP	#	MAP
0) RRF -> T-UPWM (BioASQ 10b)	5	0.4381	12	0.3291	5	0.4795	12	0.3605	6	0.3834
1) RRF -> T-UPWM (BioASQ 9b)	8	0.4130	12	0.3223	8	0.4290	14	0.3502	10	0.3706

reinforcing the benefit of training with the additional 500 questions. Overall, we can consider the improvements to be marginal except on batch 3, where system 0 outscored system 1 by 5 MAP points. Furthermore, we also reported the corresponding BioASQ system ranking position if we had made submissions with system 0.

5. Conclusion and Future Work

In this paper we detailed our participation on tasks B phase A, B phase B and Synergy of the tenth edition of the BioASQ challenge. For each task, we described a different method that was the core of our submissions. Regarding Synergy we tested a relevance feedback based approach, that despite its simplicity achieved results that were close to the overall average. In task B phase A, we explored the PARADE-CNN and Transformer-UPWM models, where the later had been previously trained with last year's data. Here, we achieved quite interesting results, namely, the fact that an one year old model remained competitive, raising the question how it would performed if it was trained with this year's training data. After additional post-challenge experiments, we found that the additional training data contributed for an 1 point increase in terms of MAP@10. Finally on task B phase B, we made use of the PubMedBERT model for the classification task. Given the relatively small size of the dataset when compared to the complexity of the model used, we explored and made use of data augmentation techniques based on paraphrasing. While the inclusion of this data augmentation did indeed show an increase of 3 points when compared with the basic setup, it was not enough to bring the model past the average performance of the submissions by other teams.

In terms of future work, we aim to jointly address phase A (document retrieval and snippet retrieval) and phase B (yes or no), leveraging a multi-objective training paradigm. Furthermore, for the task b phase B "yes no" question answering, we would like to also explore the integration of sentiment-word scores into our classification score function, given the results reported on [22].

Acknowledgments

This work was partially supported by national funds through the Foundation for Science and Technology (FCT) in the context of the project UIDB/00127/2020 and from the EU/EFPIA Innovative Medicines Initiative 2 Joint Undertaking under grant agreement No 806968. Tiago Almeida is funded by FCT under the grant 2020.05784.BD.

References

[1] I. Klerings, A. S. Weinhandl, K. J. Thaler, Information overload in healthcare: too much of a good thing?, Z. Evid. Fortbild. Qual. Gesundhwes. 109 (2015) 285–290.

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

[3] C. Li, A. Yates, S. MacAvaney, B. He, Y. Sun, Parade: Passage representation aggregation for document reranking, arXiv preprint arXiv:2008.09093 (2020).

[4] T. Almeida, S. Matos, Universal passage weighting mecanism (UPWM) in bioasq 9b, in: G. Faggioli, N. Ferro, A. Joly, M. Maistro, F. Piroi (Eds.), Proceedings of the Working Notes of CLEF 2021 - Conference and Labs of the Evaluation Forum, Bucharest, Romania, September 21st - to - 24th, 2021, volume 2936 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2021, pp. 196–212. URL: http://ceur-ws.org/Vol-2936/paper-13.pdf.

[5] T. Almeida, S. Matos, Bioasq synergy: A strong and simple baseline rooted in relevance feedback, in: G. Faggioli, N. Ferro, A. Joly, M. Maistro, F. Piroi (Eds.), Proceedings of the Working Notes of CLEF 2021 - Conference and Labs of the Evaluation Forum, Bucharest,

Romania, September 21st - to - 24th, 2021, volume 2936 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2021, pp. 188–195. URL: http://ceur-ws.org/Vol-2936/paper-12.pdf.

[6] L. L. Wang, K. Lo, Y. Chandrasekhar, R. Reas, J. Yang, D. Burdick, D. Eide, K. Funk, Y. Katsis, R. M. Kinney, Y. Li, Z. Liu, W. Merrill, P. Mooney, D. A. Murdick, D. Rishi, J. Sheehan, Z. Shen, B. Stilson, A. D. Wade, K. Wang, N. X. R. Wang, C. Wilhelm, B. Xie, D. M. Raymond, D. S. Weld, O. Etzioni, S. Kohlmeier, CORD-19: The COVID-19 open research dataset, in: Proceedings of the 1st Workshop on NLP for COVID-19 at ACL 2020, Association for Computational Linguistics, Online, 2020. URL: https://www.aclweb.org/anthology/2020.nlpcovid19-acl.1.

[7] MEDLINE - database, 2002. URL: https://lhncbc.nlm.nih.gov/ii/information/MBR.html.

[8] PubMed - National Library of Medicine, 1996. URL: https://www.ncbi.nlm.nih.gov/pubmed/.

[9] A. Joulin, E. Grave, P. Bojanowski, T. Mikolov, Bag of tricks for efficient text classification, in: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers, Association for Computational Linguistics, Valencia, Spain, 2017, pp. 427–431. URL: https://aclanthology.org/E17-2068.

[10] M. Honnibal, I. Montani, S. Van Landeghem, A. Boyd, spaCy: Industrial-strength Natural Language Processing in Python (2020). doi:10.5281/zenodo.1212303.

[11] J. Lin, X. Ma, S.-C. Lin, J.-H. Yang, R. Pradeep, R. Nogueira, Pyserini: A Python toolkit for reproducible information retrieval research with sparse and dense representations, in: Proceedings of the 44th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2021), 2021, pp. 2356–2362.

[12] P. Yang, H. Fang, J. Lin, Anserini: Enabling the use of lucene for information retrieval research, in: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '17, Association for Computing Machinery, New York, NY, USA, 2017, p. 1253–1256. URL: https://doi.org/10.1145/3077136.3080721. doi:10.1145/3077136.3080721.

[13] N. Abdul-jaleel, J. Allan, W. B. Croft, O. Diaz, L. Larkey, X. Li, M. D. Smucker, C. Wade, Umass at trec 2004: Novelty and hard, in: In Proceedings of TREC-13, 2004.

[14] T. Akiba, S. Sano, T. Yanase, T. Ohta, M. Koyama, Optuna: A next-generation hyperparameter optimization framework, in: Proceedings of the 25rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2019.

[15] H. Wu, R. W. Luk, K. Wong, K. Kwok, A retrospective study of a hybrid document-context based retrieval model, Information Processing & Management 43 (2007) 1308–1331. URL: https://www.sciencedirect.com/science/article/pii/S0306457306001865. doi:https://doi.org/10.1016/j.ipm.2006.10.009, patent Processing.

[16] L. Pang, Y. Lan, J. Guo, J. Xu, J. Xu, X. Cheng, Deeprank: A new deep architecture for relevance ranking in information retrieval, in: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, CIKM '17, Association for Computing Machinery, New York, NY, USA, 2017, p. 257–266. URL: https://doi.org/10.1145/3132847.3132914. doi:10.1145/3132847.3132914.

[17] Huggingface - ramsrigouthamg/t5_sentence_paraphraser, https://huggingface.co/ramsrigouthamg/t5_sentence_paraphraser, 2022. Accessed: 2022-06-20.

[18] Quillbot, https://quillbot.com/, 2022. Accessed: 2022-06-20.

[19] W. Yoon, J. W. Yoo, S. Seo, M. Sung, M. Jeong, G. Kim, J. Kang, Ku-dmis at bioasq 9: Data-centric and model-centric approaches for biomedical question answering, in: CLEF, 2021.

[20] L. Biewald, Experiment tracking with weights and biases, 2020. URL: https://www.wandb.com/, software available from wandb.com.

[21] 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, SIGIR '09, Association for Computing Machinery, New York, NY, USA, 2009, p. 758–759. URL: https://doi.org/10.1145/1571941.1572114. doi:10.1145/1571941.1572114.

[22] M. Sarrouti, S. Ouatik El Alaoui, A yes/no answer generator based on sentiment-word scores in biomedical question answering, International Journal Of Healthcare Information Systems And Informatics 12 (2017) 62–74. doi:10.4018/IJHISI.2017070104.