Post-processing BioBERT And Using Voting Methods for Biomedical Question Answering

Margarida M. Campos[1], Francisco M. Couto[1]

[1]LASIGE, Departamento de Informática, Faculdade de Ciências, Universidade de Lisboa, 1749–016 Lisboa, Portugal

Abstract

There have been remarkable advances in the field of Biomedical Question Answering (QA) through the application of Transfer Learning to overcome the scarcity of domain-specific corpora. The fine-tuning of BioBERT on general purpose larger datasets prior to fine-tuning on a specific biomedical task has proven to significantly improve performance. There are, however, a lot of post-processing techniques to the outputs of fine-tuned models to be explored.

In this paper we present our QA system, developed for the BioASQ 9th challenge - Task B, Phase B, developed by our team - LASIGE_ULISBOA. Using the outputs from the fine-tuning of BioBERT on both the Multi-Genre Natural Language Inference (MNLI) and the Stanford Question Answering Dataset (SQuAD) datasets. We compare different post processing strategies for prediction retrieval for Yes/No, Factoid, and List type questions.

We show that using Softmax in the proper location of the pipeline of answer retrieval leads to better performance and also increases the explainability of a prediction's confidence level in QA. We also present a method for applying voting system algorithms to choose candidates for List type answers, how they can increase MacroF1 score and how one can use them to optimize for either Precision or Recall. The obtained results, averaged over batches, were 0.798 MacroF1 for Yes/No, 0.478 MRR for Factoid, and 0.466 F1 for List.

The used software is available in an open access repository.

1. Introduction

BioASQ is an annual challenge that comprises different biomedical semantic indexing and question answering (QA) tasks. The presented system is a solution for Task B - Phase B, which consists of providing exact and ideal answers to questions, given related snippets. Biomedical QA is particularly challenging due to the highly domain-specific vocabulary and limited availability of curated datasets. In order to minimize these limitations we used **BioBERT** [1] as our base model and Transfer Learning - by fine tuning the base model on non-medical larger datasets, prior to training on the task's training data. There are three types of questions that require exact answers in Task B:

- **Yes/No** - binary answer
- **Factoid** - answer is a string
- **List** - answer is a list of strings, each identifying a different entity

CLEF 2021 – Conference and Labs of the Evaluation Forum, September 21–24, 2021, Bucharest, Romania
margarida.moreira.campos@gmail.com (M. M. Campos); fjcouto@edu.ulisboa.pt (F. M. Couto)

Our approach is concerned only with the retrieval of exact answers, and therefore it it was not designed to retrieve answers to *Summary* type questions or ideal answers (paragraph-sized summaries).

For factoid and list questions, predictions are always substrings of the provided passages (snippets), making the success of the previous task of snippet retrieval paramount to obtain good results.

Although the most significant advances in the area have been made by fine-tuning on different and bigger datasets or the development of new and complex transformers architectures [2], we aim to show the importance of post-processing and the use of proper final layers for each task.

Considering that a tractable and meaningful measure of the level of confidence of a prediction is as important as the prediction itself, we present as well a proposal of said confidence level for Yes/No and Factoid questions.

All the software used can be found in https://github.com/lasigeBioTM/BioASQ9B.

2. Related Work

Our baseline approach was inspired in the work done by DMIS Laboratory (Korea University) for the previous edition of BioASQ challenge[3].

2.1. BioBERT

The base model for our system is BioBERT, a BERT[4] which was pre-trained using PubMed abstracts and PubMed Central (PMC) articles. BioBERT has obtained state-of-the-art results in several biomedical NLP tasks, including QA [1].

2.2. Sequential Transfer Learning

Substantial advances have been made in Natural Language Processing (NLP), specially in domain-specific tasks with the use of Transfer Learning - using the learnt model from a task for a subsequent task [5]. The use of extra corpora to train is particularly important given the reduced size of the BioASQ dataset. Research has found that fine-tuning on the SQuAD dataset [6] improves the performance of QA systems where the correct answer is a segment of a provided passage. Another dataset that has proven important is the Multi-Genre Natural Language Inference (MNLI)[7], which is widely used to improve questions of type Yes/No, but has also proven to be useful for factoid and list question types, as was shown by DMIS Laboratory (DMIS)[3].

3. Methods

3.1. Data & Pre-Processing

MNLI Training data consists of pairs of sentences, each classified with a label from $\{Entailment, Contradiction, Neutral\}$. The cardinality of each label set can be found in Table 1. Intuitively there is a mapping (MNLI $\leftrightarrow$ $BioASQ$): $Entailment \leftrightarrow Yes$ and $Contradiction \leftrightarrow No$.

Table 1
Statistics MNLI training data.
Number of paired sentences per class

Relation	# Pairs
Entailment	130,899
Contradiction	130,903
Neutral	130,900

Table 2
Statistics of BioASQ 8b training data, used in training.
Number of unique questions(Q) and median
number of related snippets (S) per question

Question Type	$\#Q$	$Median \#S/Q$
Yes/No	881	10
Factoid	941	9
List	644	11

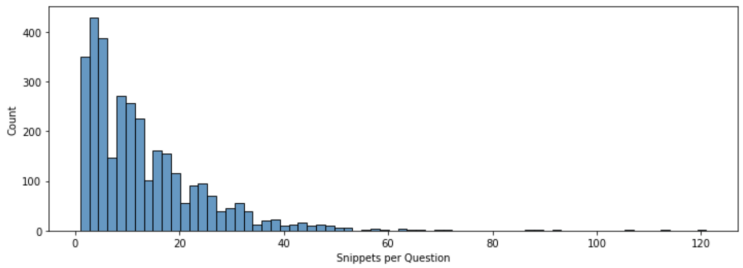


Figure 1: Distribution of number of associated snippets per question, in training data from task 8B

This could suggest that training without the $Neutral$ pairs could improve performance, however our experiments showed that our system's performance did not benefit from this strategy, hence the entire dataset was used.

SQuAD Training data consists of pairs $\{Question, Snippet\}$ and the correct answer as well as its starting position. For training the QA model, the end position was identified and added as input.

BioASQ Training of the systems was done using BioASQ 8B training data, and evaluation was done on BioASQ 8B test batches. In Table 2 we can see the number of questions in the BioASQ training data, and in Figure 1 we can see the distribution of the number of snippets associated to a question. Examples of questions can be found in Table 3, and the number of train and test questions for each type of question can be found in Table 4. It is important to mention that only 177 (20%) of the Yes/No questions have the label *No*, making the classification extremely imbalanced. To handle this, undersampling[8] of the *Yes* class was performed, resulting in an even smaller set of 354 unique questions. Oversampling the *No* class proved to be ineffective.

For list questions, each entity in the golden label was considered a correct answer for the given $\{Question, Snippet\}$ pair, *i.e.* a pair whose golden list contains m entities will appear as m distinct input observations, each labeled with a different correct answer. A summary of different type of inputs can be seen in Table 5.

Both factoid and list inputs were converted to the mentioned SQuAD format - containing the answer's start and end positions. Observations whose snippets did not contain the correct answer were discarded.

As with all BERT inputs, $\{Question, Snippet\}$ pairs are added a [CLS] token in the beginning -for classification - and a separation token ([SEP]) is added in between the two input texts, as

Table 3
Example of training questions, snippets and answers from BioASQ training data

	Question	Snippet	Gold Answer
Yes/No	Is Baloxavir effective for influenza?	"Baloxavir marboxil is a selective inhibitor of influenza cap-dependent endonuclease. It has shown therapeutic activity in preclinical models of influenza A and B virus infections,including strains resistant to current antiviral agents."	yes
Factoid	Cemiplimab is used for treatment of which cancer?	"Cemiplimab is a PD-1 inhibitor that is approved for treatment of metastatic or locally advanced cutaneous squamous cell carcinoma."	cutaneous squamous cell carcinoma
List	Which organs are mostly affected in Systemic Lupus Erythematosus (SLE)?	"In systemic lupus erythematosus (SLE), brain and kidney are the most frequently affected organs. The heart is one of the most frequently affected organs in SLE. Any part of the heart can be affected, including the pericardium, myocardium, coronary arteries, valves, and the conduction system"	kidney, brain, heart,skin

Table 4
Number of questions and snippets in training and test sets used for obtaining the reported experimental results.

Type of Question	Train		Test	
	Questions	Snippets	Questions	Snippets
Yes/No	881	11,976	152	1,262
Factoid	941	11,633	151	1,249
List	644	88,36	75	662

well as in the end of the input.

Additional biomedical datasets could have been curated to be used for fine tuning the system, however this was not done due to time constraints.

3.2. Fine Tuning

For the fine-tuning of BioBERT the best performing sequences of training reported in [3] were used. For Yes/No questions the sequence is ***BioBERT-MNLI-BioASQ***, as for factoid and list

Table 5
Input form of each type of dataset used

	Input
MNLI	{Sentence A, Sentence B, Label}
Yes/No	{Question, Snippet, Label}
SQuAD Factoid List	{Question, Context, Answer start, Answer end}

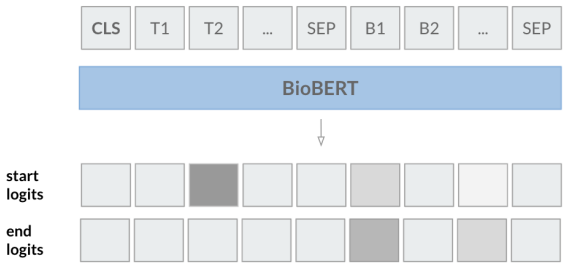


Figure 2: Simplified representation of BERT for QA

type questions ***BioBERT-MNLI-SQuAD-BioASQ*** was used.

For fine-tuning in the MNLI dataset we used a slightly altered version of *BertForSequenceClassification* model from the *Transformers* [9] library, which consists of adding a linear layer which receives as input the hidden vector of the $[CLS]$ token [4]. To train in the binary classification of BioASQ *Yes/No* questions, following *BertForSequenceClassification* last 3 neuron layer tests were made with addition of:

- one extra binary layer (*[CLS]-3-2*)
- fully-connected 512 neuron layer followed by a binary one (*[CLS]-3–256-2*)
- fully-connected 256 neuron layer followed by a binary one (*[CLS]-3-512-2*)
- replacing previous MNLI 3 neuron layer with a binary one (*[CLS]-2*)

For training on the *SQuAD* corpus, the final classification layers are removed and the architecture of *BertForQuestionAnswering* model from the *Transformers* library is used. A simplified overview of Input/Output from *BertForQuestionAnswering* can be seen in Figure 2. In QA the input provided contains the start and end positions of the tokens representing the span of the correct answer within the passage. Training is done by creating two new vectors - *start logits* and *end logits* of shape $(input\_length, 1)$ that represent the likelihood of each token being the start and end of the answer, respectively.

3.3. Post-Processing and Output Aggregation

Given that the same question can have multiple snippets associated to it, leading to different $\{Question, Snippet\}$ pairs as input, a strategy is needed to combine the different outputs

Tokens	effects	of	lb	-	100	a	novel	inhibitor	of	pp	##2	##a	against
Start Logits	-9.01	-10.41	-8.49	-10.40	-9.98	-9.11	-9.55	-8.67	-10.81	3.11	-7.07	-4.72	-9.23
End Logits	-5.78	-9.66	-10.48	-9.96	-7.87	-10.31	-9.27	-9.78	-9.30	-4.96	-5.23	5.41	-9.54

Figure 3: Example of the output Start and End logits vectors for a snippet. Highlighted cells represent the allowed maximum score pairs, identically highlighted in Figure 4

into single predictions. Each type of question demands a different approach, hence they are presented separately.

3.3.1. Yes/No

Let p_{ij} and $\overline{p}_{ij}$ represent the model's output probability of question i given the snippet j having answer *Yes* and *No*, respectively. The predicted answer will be the one with a highest mean probability over the J snippets associated to question i. P_i represents the level of confidence that the provided answer is correct.

$$p_i = \frac{1}{J}\sum_{j=1}^{J} p_{ij}$$
$$\overline{p}_i = \frac{1}{J}\sum_{j=1}^{J} \overline{p}_{ij}$$

$$P_i = \begin{cases} P(Yes) = p_i, & \text{if } p_i \geq \overline{p}_i \\ P(No) = \overline{p}_i, & \text{otherwise} \end{cases}$$

3.3.2. Factoid

The relevant outputs from the fine-tuned QA model are the start and end logits vectors. In Figure 3 an output example from the BioASQ golden set from Batch 1 of task 8B can be seen.

Let s_{ij}^{l} and e_{ij}^{l} be the start and end logits value corresponding to token T_{ij}^{l}, the l^{th} of the j^{th} snippet associated to question i, and $Pred_{ij}$ and $Prob_{ij}$ be the lists of predictions and associated confidence levels for the same input.

In order to choose the best prediction for each input, one should find the span (a, b) that maximizes some combination of s_{ij}^{a} and e_{ij}^{b}. Given the logits are not normalized, to use merely the sum of start and end logits would result in an unfounded comparison between confidence levels for predictions of different snippets.

To minimize this discrepancy, our approach for each input was implemented as follows:

1. Create upper triangular matrix M where, $M_{p,q} = s_{ij}^{p} + e_{ij}^{q}$ (See Figure 4), for $q \geq p$, guaranteeing end does not precede the start
2. Choose positions a and b that maximize $M_{p,q}$
3. If the expression resulting from the span from T_{ij}^{a} to T_{ij}^{b} satisfies admission rules, append expression to $Pred_{ij}$ and $M_{a,b}$ to $Prob_{ij}$
4. Remove entry $M_{a,b}$ from M
5. Repeat steps 2 to 4, until lists have length k, where k is an hyperparameter chosen by the user
6. Apply the softmax function to vector $Prob_{ij}$ of length k

	0	1	2	3	4	5	6	7	8	9	10	11	12
0	-14.79	-18.67	-19.49	-18.97	-16.88	-19.32	-18.28	-18.79	-18.31	-13.97	-14.24	-3.60	-18.55
1		-20.07	-20.89	-20.37	-18.28	-20.72	-19.68	-20.19	-19.71	-15.37	-15.64	-5.00	-19.95
2			-18.97	-18.45	-16.36	-18.80	-17.76	-18.27	-17.79	-13.45	-13.72	-3.08	-18.03
3				-20.36	-18.27	-20.71	-19.67	-20.18	-19.70	-15.36	-15.63	-4.99	-19.94
4					-17.85	-20.29	-19.25	-19.76	-19.28	-14.94	-15.21	-4.57	-19.52
5						-19.42	-18.38	-18.89	-18.41	-14.07	-14.34	-3.70	-18.65
6							-18.82	-19.33	-18.85	-14.51	-14.78	-4.14	-19.09
7								-18.45	-17.97	-13.63	-13.90	-3.26	-18.21
8									-20.11	-15.77	-16.04	-5.40	-20.35
9										-1.85	-2.12	8.52	-6.43
10											-12.30	-1.66	-16.61
11												0.69	-14.26
12													-18.77

Figure 4: Matrix M where each entry represents the sum of i^{th} position start logits vector, and j^{th} position of end logits vector. Highlighted in red are scores that are not eligible - end position must be equal or greater to start position, and end position token must not contain a split word, identified with the characters $\#\#$ (see Fig 3)

To select the top 5 predictions for question i, we simply select the 5 expressions from the concatenation of the J vectors $Pred_{ij}$ with the 5 highest corresponding values in the concatenated $Prob_{ij}$.

3.3.3. List

Potential answers for list questions are retrieved using the same method as for factoid questions. The process however requires some extra processing steps, given that for list questions different entities need to be discriminated.

To select the best list of candidates, we used voting systems treating each distinct obtained answer as a candidate and the frequency in the answers as votes. The systems of Single Transferable Vote (STV) and Preferential Block Voting (PBV) were tested, with STV having the best performance. Elections are performed in rounds, in each round candidates are categorized in states: *Elected* - if the candidate has already won, *Rejected* - if the candidate is already unable to win, *Hopeful* - if the candidate has neither won nor has yet been discarded.

Candidates for answers are obtained by splitting the predictions by all usual separator characters and words(*e.g.* ',', 'and', ';', 'or'). We tested the approach of doing the splitting after the voting - treating full answers as candidates for the STV (*STV + PostProcess*), and doing the splitting before the voting - separate distinct entities are treated as a vote, with the score for the ranked ballot being the average score of all the answers that contain that entity. An example of ranked candidates before and after being processed can be seen in Tables 6 and 7. *E.g.* the score of candidate "dizziness" will be the average of scores where the candidate is contained: 0.21, 0.20 and 0.18 (1^{st},2^{nd} and 5^{th} entries of Table 6). Each snippet will contribute to the voting with a ballot of ranked candidates, which then enter the voting algorithm.

A potential handicap of using voting system algorithms for answer selection is the need to predefine the number of elected entities, since in an election the number of winners is

Table 6
Example of ranked list candidate answers from one snippet and respective scores

Predictions	Scores
dizziness	0.21
dizziness, orthostatic hypotension	0.20
orthostatic hypotension	0.20
hallucination	0.19
hallucination, dizziness	0.18

Table 7
Example of ranked list candidate answers after being split by seperators, considering the average score of all answers that contain each entity

Predictions	Scores
orthostatic hypotension	0.20
dizziness	0.197
hallucination	0.185

Table 8
Summary of hyperparameters used for the fine-tuning that led to the reported results.

Epochs	3
Batch Size	18
Optimizer	Adam
Learning Rate	5×10^{-5}

established beforehand. This is not ideal since the correct number of answers for a given list question is not defined. Two characteristics of the implemented algorithms that allow us to minimize this problem are:

- If the number of non rejected candidates is inferior to the number of selected winners, these are elected
- If there are ties in the election, all tied candidates are elected - even if this means electing a superior number of candidates

Although these factors allow for some flexibility in the number of predictions, a more flexible approach can be used. Since elections are performed in rounds, one can define that the selected answers are the ones that are not rejected in the pre-final round, *i.e.*, all candidates with states in {*Hopeful,Elected*}. When referencing this approach we call the number of candidates *Hopeful*.

3.4. Software

Our team tried to replicate the results of 4 state of the art systems and found some reproducibility issues. Some of the causes were: outdated versions of packages, compatibility issues due to the use of conflicting code libraries like the use of both *Tensorflow* and *PyTorch* for different stages of the pipeline.

To avoid the aforementioned issues our implementation was done in a modularized fashion, built in *Python 3.6*[10], using the *Pytorch*[11] versions of model implementations from the *Transformers*[9] library as main structure. In spite of its fully *Pytorch* architecture, the system accepts as input *Tensorflow*[12] checkpoints (model's saved parameters).

Fine-tuning was performed using parallelization on 6 GPUs (Tesla M10) with 8GB of memory each. Total batch size is 18 (3 samples per GPU). Summary of the training details of the reported results can be found in Table 8

Table 9
Experimental results of *Yes/No* models. Trained on training data for task 8B, evaluated on the task's 5 test batches. [CLS]-3-2 architecture has the addition of a binary layer on top of the MNLI classification model, [CLS]-256-2 and [CLS]-512-2 has the addition of a fully connected layer of 256 and 512, respectively, before the extra binary layer. DMIS represents the average scores of DMIS Lab systems.

Architecture	Acc	F1no	F1yes	MacroF1
[CLS]-3-2	0.7039	0.4828	0.7926	0.6377
[CLS]-3-256-2	0.7434	0.6286	0.8040	**0.7163**
[CLS]-3-512-2	0.7368	0.5745	0.8095	0.6920
DMIS	0.8513	0.8071	0.8733	0.8402

3.5. Metrics

For evaluation and comparison of different models, the official BioASQ measures of performance were used [13].

For *Yes/No* questions the official metric is the MacroF1 - mean of F1 score of both *Yes* and *No* classes. Accuracy is also calculated for completeness. Factoid questions are evaluated using Mean Reciprocal Rank (Metric). Strict Accuracy (SAcc) and Lenient Accuracy (LAcc) are also calculated. List questions are evaluated by the average F1 score of all questions, with the mean precision and recall also reported.

4. Results

4.1. Experimental Results

In this section we present the experimental results of the referred approaches. Training was done in task 8B training set and evaluation was done in the aggregation of all 8B Phase B batches. Results are compared with the average (weighted by number of questions) of all DMIS systems results in the 5 batches.

4.1.1. Yes/No

In Table 9 we can see the results of the different classification architectures for the Yes/No question type. Results are significantly better with the extra fully connected layer, before the final binary one. Experiments showed that performance differs slightly with the number of neurons of the middle layer if it lays between 128 and 512. Higher MacroF1 was obtained with 256 neurons (**[CLS]-256-2**).

4.1.2. Factoid

For factoid questions performance increased substantially with the use of the *k-candidates* approach. The results can be seen in Table 10. Best results were obtained with $k = 2$ number of candidate answers per snippet. It is interesting to point out that for $k > 4$ the results almost do

not differ. This is due to the fact that candidates of order higher than 4 typically have extremely low scores and end up with probabilities close to 0, therefore are discarded when the top 5 predictions are extracted.

Table 10
Experimental results of *Factoid* models. Trained on training data for task 8B, evaluated on the task's 5 test batches. *Start+End* represents the classic approach of applying Softmax to both Start and End Logits prior to finding the scores' maximizing answers. *Top k* represents the approach of applying Softmax to the k selected candidates. Different values of k are presented (2,5,10). DMIS represents the average scores of DMIS Lab systems.

Strategy	k	SAcc	LAcc	MRR
Start + End	-	0.1060	0.2119	0.1485
Top k	2	0.3179	0.5232	0.3991
	5	0.2195	0.5121	0.3390
	10	0.2195	0.5121	0.3390
DMIS	-	0.3603	0.5656	0.44

4.1.3. List

In Table 11 we can see the results of experiments with the list questions. We can observe the impact of requesting different number of winners from the algorithm. Unsurprisingly, a larger number of winners leads to an increase in Recall and a decrease in Precision. Maximum performance (MacroF1) is obtained with the *Hopeful* strategy, for both processing strategies.

Results show that splitting candidates prior to the voting leads to better results.

4.2. BioASQ Official Results

A summary of the official results from BioASQ Task9B - Phase B can be seen in Table 12, where we present the results of the top teams along with ours (LASIGE), considering the BioASQ ordering. The place in each batch is considered to be the place of the best scoring system for each team, considering all systems of each team as one.

4.3. Unanswerability

An important aspect to look at when evaluating performance, is the unanswerability of some questions in the dataset. Several questions in the test sets have an answer which can not be extracted from the provided snippets. Ideally, to measure the actual performance of answer extracting systems, these would be removed from the test set. Examples of such questions can be seen in Table 13.

For the test set of task 8B (resulting of the aggregation the 5 test batches), $\mathbf{22,5}\%$ of factoid questions do not contain the golden answer in any provided snippet, and $\mathbf{25.3}\%$ of list

Table 11
Experimental results of *List* models. Trained on training data for task 8B, evaluated on the task's 5 test batches. STV-PostProcess and STV-PreProcess refer to the strategies of using the Single Transferable Vote algorithm for answer candidates, considering answers splitted by separators after and before voting,respectevily. *Elected* represents the number of winners required for the voting, with *Hopeful* representing the strategy of selecting the non-rejected candidates as winners.DMIS represents the average scores of DMIS Lab systems.

Method	Elected	Precision	Recall	MacroF1
STV-PostProcess	2	0.5111	0.3437	0.3921
	5	0.3906	0.4766	0.4115
	Hopeful	0.4944	0.3289	0.3751
STV-PreProcess	2	0.4527	0.3306	0.3706
	5	0.4333	0.4167	0.4107
	Hopeful	0.5	0.4167	0.4524
DMIS	-	0.4761	0.4206	0.3940

questions have at least one entity that is not contained in the snippets. For Yes/No questions unanswerability would have to be manually done.

5. Discussion

5.1. Analysis of Results

All reported results were obtained using BioBERT Base (12 stacked encoding layers). Although we tested with BioBERT Large (24 stacked encoding layers), which usually obtains better results, the results were very poor. This is probably due to memory restrictions. Since BioBERT Large has over three times more trainable parameters, reductions in input size and batch size had to be made, which are probably the cause for the low performance.

5.1.1. Yes/No

The addition of a fully connected layer between the MNLI classification layer (3 neurons) and the BioASQ binary classification layer improved performance on the test set. This indicates that the relation between the knowledge obtained on the NLI data and the one needed for the BioASQ questions is not as obvious as one might expect. This is not uncommon when we are dealing with corpora from different domains (general purpose vs biomedical), and might also be related to the existence of unanswerable questions in the dataset that difficult model's learning of what represents agreement between question and snippet, since the inputs with no relation are inducing noise for the binary task.

In Figure 5 we can see the distribution of the confidence levels for Yes ($P(Yes)$) and No ($P(No)$) predictions, compared between the actual value of the correct answer. Note that

Table 12
Summary of the official results from BioASQ Task9B - Phase B. The place in each batch is considered to be the place of the best scoring system for each team, and considering all systems of each team as one. Reported scores are taken from the overall best ranked system for each team.

Batch	System	Place	Yes/No	Factoid	List
			MacroF1	MRR	F1
1	DMIS	1	0.9258	0.3856	0.4143
	Ir_Sys	2	0.8183	0.4149	0.2800
	LASIGE	3	**0.7699**	**0.3506**	**0.4860**
2	LASIGE	1	**0.9454**	**0.5539**	**0.4818**
	bio-answerfinder	2	0.8952	0.5000	0.4571
	DMIS	3	0.8854	0.5294	0.4554
3	lalala	1	0.9532	0.5347	0.5887
	bio-answerfinder	2	0.9023	0.5811	0.4209
	LASIGE	7	**0.7292**	**0.4919**	**0.4918**
4	DMIS	1	0.9480	0.5310	0.7061
	Ir_sys	2	0.9480	0.6929	0.6312
	LASIGE	5	**0.7807**	**0.5577**	**0.5872**
5	DMIS	1	0.8246	0.4722	0.3561
	MDS_UNCC	2	0.7841	0.5204	0.2678
	LASIGE	4	**0.7564**	**0.4546**	**0.2798**

Table 13
Examples of unanswerable questions in 8B - Phase B test set. Answers represent the golden label provided by BioASQ.

Question	Snippet	Answer
Which biological process takes place in nuclear speckles?	here we demonstrate that mrnas containing alrex-promoting elements are trafficked through nuclear speckles	mrna processing
Can LB-100 downregulate miR-33?	PP2A inhibition from LB100 therapy enhances daunorubicin cytotoxicity in secondary acute myeloid leukemia via miR-181b-1 upregulation.	No

model's discriminatory power (distance between $P(Yes)$ and $P(No)$) is much greater for answers with Yes label. This can also be seen by looking at the differences between the F1 scores of both classes, noting that $F1_{yes}$ is much higher than $F1_{no}$ across experiments. This

is not surprising in NLI, as it is easier to identify entailment than it is to distinguish between contradiction and neutral relations. Entailment is usually distinctly expressed in the passage, whilst contradiction sometimes needs to be inferred from more complicated relations between sentences.

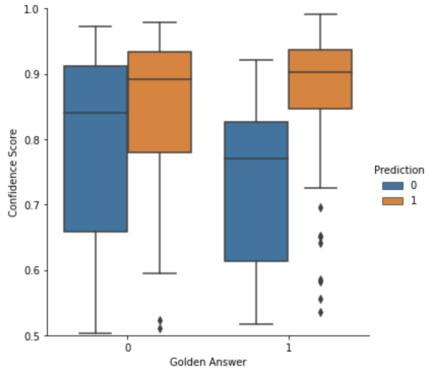


Figure 5: Confidence scores for Yes/No predictions split by correct golden label

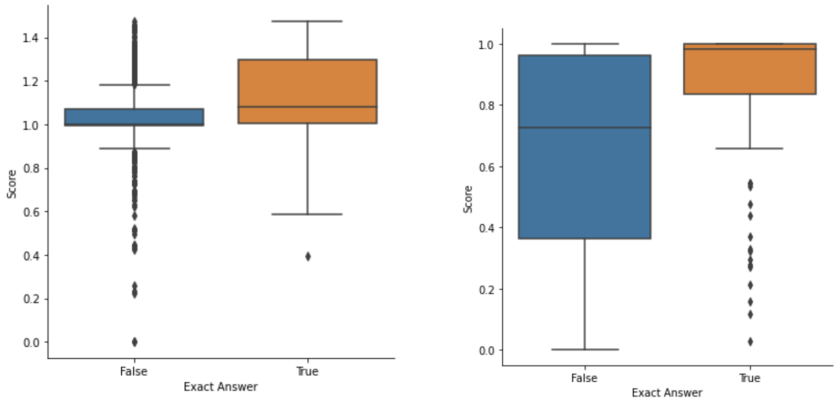
5.1.2. Factoid

Looking at experimental results (Table10) we can see that sorting predictions using scores obtained by applying Softmax to the k predictions for each snippet strongly improved all metrics. Moreover, we can look at the fitness of the scores by analysing Figure 6 where we compare the distribution of confidence levels for predictions when the answer was in fact correct or not. We can see that for the classic approach there is an almost 100% overlap of incorrect scores with correct ones, which implies the scoring is not strong. Although there is still some expected overlap in the *k-candidates* approach, one can distinctly see a higher level of confidence for correct answers, indicating the validity of the proposed score as a confidence level metric.

5.1.3. List

Using voting systems for the choice of list questions proved to be effective, and we can see in Table 12 that the proposed system obtained overall strong results for List type questions, with the exception of Batch 5.

By using the *Hopeful* approach, one has flexibility in the number of entities that are selected, and in fact this approach has the best MacroF1 scores across experiments. With the application of the voting systems, opposed to using a predefined threshold for answer selection, we make use not only of the confidence level of each answers but also of the occurrence of the answer and its relative certainty amongst other answers from the same input.



(a) Scores from Softmax(Start Logits) plus Softmax(End Logits)

(b) Softmax(k top predictions)

Figure 6: Distribution of prediction confidence scores for Factoid questions of the Task 8B - Phase B test set.

6. Conclusion

In this paper we used transfer learning to fine-tune BioBERT on general purpose datasets (MNLI and SQuAD) prior to fine-tuning on the BioASQ dataset. We showed how the post-processing of the model outputs greatly impacts performance, revealing that applying Softmax on the output scores from only the k selected candidates, for obtaining predictions' confidence level improves overall performance and makes scores more meaningful. We also showed that using the Single Transferable vote system for electing list questions candidates for answers obtains promising results, outperforming the previous approach of selecting candidates merely based on a defined threshold.

To increase the current model's performance in the future, one can: enrich transfer learning sequences with additional biomedical domain corpora, train current system using BioBERT Large in larger memory GPUs, with same learning parameters (input size, learning rate and batch size). Another possibility is to adapt BERT architecture to allow for training of start and end logits combined, *i.e.*, train QA for finding the exact span of the answer within the text - conditioning end of answer to its start - instead of training them separately and doing the conditioning in the post-processing phase.

Acknowledgments

This work was supported by FCT through project DeST: Deep Semantic Tagger project, ref. PTDC/CCI-BIO/28685/2017, and the LASIGE Research Unit, ref. UIDB/00408/2020 and ref. UIDP/00408/2020.

We would like to thank Doctor Maria Fernandes from the University of Luxembourg, who provided us access to larger GPUs for running experiments, for all her help and support.

References

[1] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, J. Kang, Biobert: a pre-trained biomedical language representation model for biomedical text mining, Bioinformatics (2019). URL: http://dx.doi.org/10.1093/bioinformatics/btz682. doi:10.1093/bioinformatics/btz682.

[2] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, 2017. arXiv:1706.03762.

[3] M. Jeong, M. Sung, G. Kim, D. Kim, W. Yoon, J. Yoo, J. Kang, Transferability of natural language inference to biomedical question answering, 2021. arXiv:2007.00217.

[4] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, 2019. arXiv:1810.04805.

[5] J. Yosinski, J. Clune, Y. Bengio, H. Lipson, How transferable are features in deep neural networks?, 2014. arXiv:1411.1792.

[6] P. Rajpurkar, J. Zhang, K. Lopyrev, P. Liang, Squad: 100,000+ questions for machine comprehension of text, 2016. arXiv:1606.05250.

[7] A. Williams, N. Nangia, S. R. Bowman, A broad-coverage challenge corpus for sentence understanding through inference, 2018. arXiv:1704.05426.

[8] S. Dendamrongvit, M. Kubat, Undersampling approach for imbalanced training sets and induction from multi-label text-categorization domains, 2009, pp. 40–52. doi:10.1007/978-3-642-14640-4_4.

[9] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, A. M. Rush, Transformers: State-of-the-art natural language processing, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Association for Computational Linguistics, Online, 2020, pp. 38–45. URL: https://www.aclweb.org/anthology/2020.emnlp-demos.6.

[10] Python Core Team, Python: A dynamic, open source programming language, Python Software Foundation, Vienna, Austria, 2016. URL: https://www.python.org/.

[11] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, S. Chintala, Pytorch: An imperative style, high-performance deep learning library, in: H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, R. Garnett (Eds.), Advances in Neural Information Processing Systems 32, Curran Associates, Inc., 2019, pp. 8024–8035. URL: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[12] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, X. Zheng, TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL: https://www.tensorflow.org/, software available from tensorflow.org.

[13] A. Nentidis, A. Krithara, K. Bougiatiotis, M. Krallinger, C. Rodriguez-Penagos, M. Villegas, G. Paliouras, Overview of bioasq 2020: The eighth bioasq challenge on large-scale

biomedical semantic indexing and question answering, in: International Conference of the Cross-Language Evaluation Forum for European Languages, Springer, Springer, 2020. URL: https://link.springer.com/chapter/10.1007/978-3-030-58219-7_16.