Exploring Biomedical Question Answering with BioM-Transformers At BioASQ10B challenge: Findings and Techniques

Sultan Alrowili[1], K. Vijay-Shanker[1]

[1]Department of Computer and Information Science, University of Delaware, Newark, Delaware, USA

Abstract

This paper details the methods and techniques we used at the BioASQ10B challenge with our BioM-Transformers models. As of last year, we continue to use our BioM-Transformers: an adaptation of both ELECTRA and ALBERT models to the biomedical domain. However, this year, we extend the investigation of our biomedical Questing Answering models with BioM-Transformers models by extending our grid search for hyper-parameters and addressing the limited size of the BioASQ10B-Factoid training set by merging it with the List training set. Additionally, in Transformers-based models (e.g., ELECTRA, ALBERT), task-specific (e.g., question answering) layers are randomly initialized for every new run causing the performance to fluctuate on downstream tasks (e,g SQuAD, BioASQ). We study the range of this randomness at the BioASQ10B challenge by running two identical models with the same hyperparameters. Our results show that tuning our hyper-parameters led to significant performance gain (e.g., 20% lead in list questions and 100% accuracy on several Yes/No batches). Moreover, our approach to merge both BioASQ10B (Factoid / List) training set show better performance than our model, which was fine-tuned only on the Factoid training set. Finally, our results also show that the randomness caused by task-specific wights initializations causes a significant performance variance, especially in small datasets such as BioASQ.

Keywords

BERT, ELECTRA, ALBERT, BioASQ, BioM-Transformers

1. Introduction

The introduction of the BioBERT model [1], introduces the idea of domain adaptation of the BERT model [2] to the biomedical domain. This adaptation of BERT shows significant performance gains on downstream tasks such as Question Answering, Text Classification, and Named Entity Recognition (NER). BioBERT model has been widely used by the majority of the teams on both BioASQ7B [3], and BioASQ8B [4] challenges. However at BioASQ9B challenge [5], we have witnessed using variety of State-of-The-Art models including RoBERTa [6], ELECTRA [7], XLNET [8], and ALBERT [9]. We also had, at the BioASQ9B challenge, introduced our large biomedical question answering models [10] , which we built based on both BioM-ELECTRA and BioM-ALBERT models [11]. However, our focus last year was more on evaluating the reproducibility of our BioM-Transformers model by using the exact hyperparameters settings that we chose in our early work [11], that we published prior to BioASQ9B challenge.

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

alrowili@udel.edu (S. Alrowili); vijay@udel.edu (K. Vijay-Shanker)

This year at the BioASQ10B challenge, however, we extended our investigation scope by increasing our hyperparameters grid search to explore the potential of our BioM-Transformers model. In addition, we address the issue of the limited size of the BioASQ10B-Factoid dataset by combining both Factoid and List training datasets. Finally, several studies [12] indicate that the random initialization of task-specific layers' weights inside Transformer models causes a fluctuation in performance between each run. In this paper, we study the range of this randomness with our Models on BioASQ10 challenges.

We can summarize the main findings of our investigations in the following points:

- We show that improving the hyperparameters choices of our BioM-Transformers models led to a significant performance improvement, especially on both list and yes/no questions.
- We introduce a new approach that merges both Factoid and List training datasets, which resulted in us taking the lead on two batches of the BioASQ10B Factoid task.
- Adapting Text Classification task for BioASQ Yes/No question led us to take the lead in batches 2, 3, and 4 and score second in both the first and last batch.

2. Relate Work

The adaption of BERT to the biomedical domain with BioBERT [1] has demonstrated significant success in addressing the performance of BERT on biomedical Question Answering tasks. Consequently, new state-of-the-art Transformers model have followed similar approach to BioBERT model including PubMedBERT$_{\text{base}}$ [12], PubMedBERT$_{\text{large}}$ [13], BioRoBERTa [14], BioMegaTron [15], and recently BioLinkBERT [16].

PubMedBERT is a new model introduced by Microsoft which pretrained BERT on both PubMed abstracts and PMC full articles. However, it differs from BioBERT that it uses a large batch size (8,192). Using a large batch size has shown effectiveness in improving the Language Model's perplexity, and performance on downstream tasks [6] [17]. Moreover, the PubMedBERT team recently introduced a large-scale variant of PubMedBERT [13], which improves the result on downstream biomedical tasks, including BioASQ7B Yes/No task. They have also introduced in the same paper both PubMedELECTRA$_{\text{base}}$ and PubMedELECTRA$_{\text{large}}$, which follow similar design factors of PubMedBERT but replacing the Masked Language Model MLM (BERT) objective with ELECTRA objective.

Additionally, more large-scale biomedical language models have been introduced in the last two years, including BioMegaTron, BioRoBERTA, and recently BioLinkBERT. Both BioMegaTron and BioRoBERTA follow a similar approach by studying the impact of design factors (e.g., corpora and vocabulary domain, batch size, training steps) on improving the performance on downstream biomedical tasks, including the BioASQ challenge. On the other hand, BioLinkBERT is a new biomedical language model that adds an additional pre-training objective to the Masked Language Model (MLM) objective of BERT. This new training objective is called Document Relation Prediction (DPR), and it aims to capture dependencies from citation links inside PubMed corpora, which they added to the pre-train corpora. By adopting this approach, BioLinkBERT shows better results on downstream biomedical tasks than the PubMedBERT model, including BioASQ7B Yes/No task.

3. Methods

3.1. BioM-Transformers

Similar to what we did last year at the BioASQ9B challenge, we continue to use BioM-Transformers [11] models at the BioASQ10B challenge. BioM-Transformers are large biomedical language models, which we introduced last year, pre-trained on biomedical corpora (PubMed Abstracts) and use specific domain vocabulary (PubMed Abstracts). We use both BioM-ELECTRA and BioM-ALBERT.

BioM-ELECTRA is a large biomedical language model that uses ELECTRA loss function [7] instead of Masking Language Model MLM. We pre-train BioM-ELECTRA for 434K steps with a batch size of 4096 on TPU3-512 units. BioM-ELECTRA model uses the same vocabulary as the PubMedBERT [12]. In contrast to the BERT loss function, which uses Masked Language Model MLM, ELECTRA uses generative and discriminative loss functions. Figure 1 illustrates the idea of the ELECTRA function. The generator inside ELECTRA is a small Masked Language Model MLM aiming to generate fake tokens that could fit the context around [MASK] token. On the other hand, the discriminator is a model that aims to judge whether the generated tokens are original (real) or replaced. Both the generator and the discriminator are pre-trained jointly inside ELECTRA, and both are improving simultaneously in a way described in the Game Theory field as a "Cat and Mouse" game.

On the other hand, our BioM-ALBERT model is a large model based on ALBERTxxlarge architecture [9], which has a hidden size of 4096 compared to BioM-ELECTRA, which has only 1024 hidden size. Although ALBERT still uses the traditional Masked Language Model MLM, it incorporates several techniques that decrease the pre-training cost and support the model's scalability. Those techniques include parameter-sharing, large batch size optimizer technique LAMB [17], and factorization of vocabulary embedding matrix.

The parameter-sharing technique allows the ALBERT model to address the parameters redundancy issue inside the Transformers. LAMB optimizer allows ALBERT to be pre-trained with large batch size, 8192, compared to 256 in the case of the BERT model [2]. The factorization of the vocabulary embedding matrix technique allows ALBERT-xxlarge to control the size of the parameters (235M), despite having a larger hidden size (4096). We pre-train our BioM-ALBERT model for 264K steps and a batch size of 8192 on TPUv3-512 units. Like BioM-ELECTRA, BioM-ALBERT pre-trained on collecting 27GB of PubMed Abstracts and uses a specific domain vocabulary that has a size of 30K tokens. We build our BioM-ALBERT's vocabulary by training the SentencePeice model on PubMed Abstracts.

3.2. Yes/No Question as a Text Classification Problem

We continue to adopt a binary classification approach to address Yes/No questions, which is the same approach we did last year at the BioASQ9B challenge. Thus, we use a snippet (context) as "sentence 1", questions as "sentence 2" and the answer (yes/no) as our "label." We use a pre-processing script by the PubMedBERT team [12] to generate the BioASQ classification dataset. We optimized our best hyperparameters for the Yes/No task using the training and testing set of BioASQ9B with Huggingface Transformers [18] implementation of text classification.

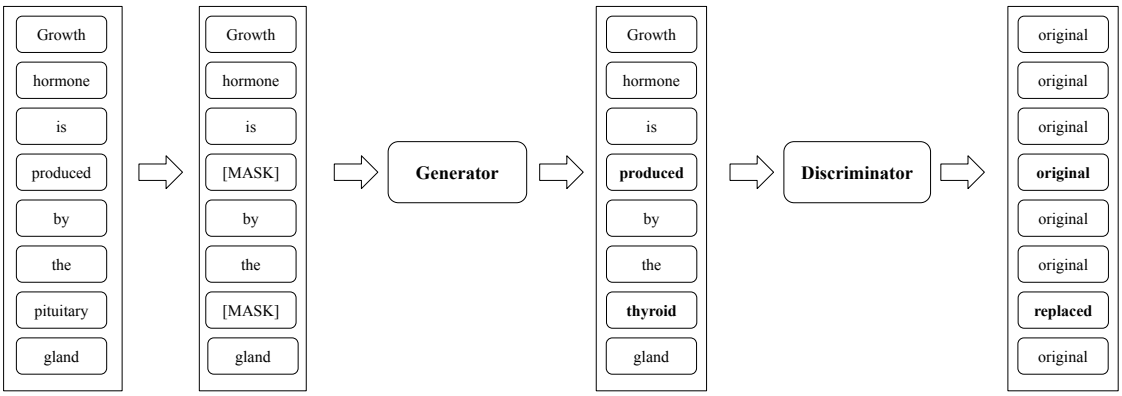

Figure 1: Overview of ELECTRA Function. Figure adapted from ELECTRA paper [7]

3.3. Combining Factoid and List Question

Most of the participants' teams at the previous BioASQ challenges BioASQ9B [5] adapt the SQuAD1.1 [19] format with BioASQ-Phase-B list questions. Tasks in the format of SQuAD1.1 treat the question answering task as a reading comprehension task where the model will scan through the context (snippet), looking for the start and end indexes of the answer span. Thus, to treat BioASQ list questions as reading comprehension tasks, we need to convert the training set of BioASQ-B list questions into Factoid-SQuAD-style questions. The BioBERT team initially proposed this idea at the BioASQ7B challenge [20], and Figure 2 illustrates this process.

Since those list-type questions are in a similar format to BioASQ10B-Factoid questions, we hypothesize that combining both the list and factoid training sets could improve the performance of our models on Factoid questions. To test this hypothesis at the BioASQ10B challenge, we combine the training set of the BioASQ10B Factoid/list tasks as one training set. This resulted in a training set that has 18,587 triplets of a question, context, and answer compared to 4,691 Factoid-only questions. We report our findings and analysis of this approach in the results section for all five batches except the second batch. Due to technical and time constraint issues, we did not use this approach in the second batch of the BioASQ10B-Factoid task.

3.4. Hyperparamters Fine-Tuning

Reproducibility is one of the major concerns in the research community, especially with Transformers-based models where hyperparameter settings (e.g., learning rate, training steps) play a significant role in improving the results on downstream tasks. Last year at the BioASQ9B challenge, we decided to use the same hyperparameters setting that we reported in our early work [11], which we published prior to the BioASQ9B challenge. Last year, we decided to test how our hyperparameters settings would perform on the new BioASQ test set (BioASQ9B). However, this year at BioASQ10B, we increased our grid search for hyperparameter settings and used BioASQ9B training and test dataset to find the best hyperparameters settings instead

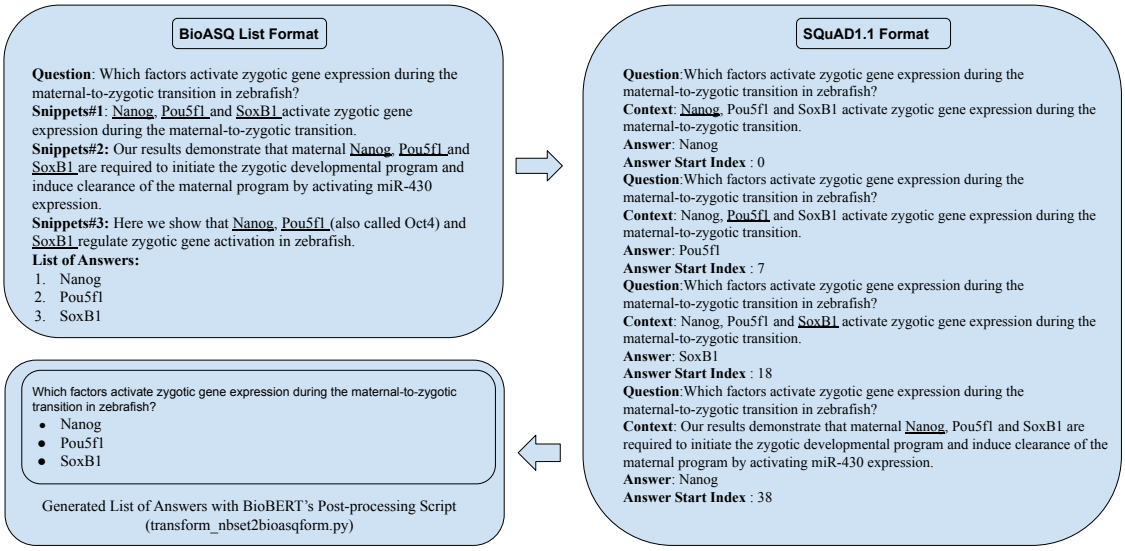

Figure 2: Overview of the idea of converting BioASQBList questions to Factoid-style question as proposed by BioBERT Team at BioASQ7B [3].

of using BioASQ7B as we did last year. Our grid search for hyperparameters this year includes: batch size of [16,24,32,40,48,64,128], learning rate of [1e-5,2e-5,3e-5,5e-5] and epochs number of [2,3,4,5]. To help reproduce our results, we share our best hyper-parameters setting for both BioM-ELECTRA and BioM-ALBERT in the appendix section A.

Additionally, we investigate studying the configuration of the post-processing script (transform_nbset2bioasqform.py) that we used in list-type questions. As we previously illustrated in Figure 2, we convert the list question into SQuAD format, and then we fine-tune our BioM-Transformers models using this SQuAD-style list question. However, following the fine-tuning process, we need to convert the list of SQuAD-like prediction to a list of answers and its common practice at this stage to use a post-processing script developed by BioBERT team [1] to obtain the final predictions for list-type questions. This post-processing script has a threshold parameter that sets the limit at which specific probability an answer can be part of a correct list of answers. We study the impact of this "threshold" value on the BioASQ9B list question's performance by using different threshold values ranging from [0.1-0.4]. We use the training and test dataset of BioASQ9B-list questions to optimize this value, and we use the best threshold we find to predict our answers on the BioASQ10B challenge.

3.5. Randomness of Transformers-based Models

In Transformers-based models (e.g., BERT, ELECTRA, ALBERT), the initial weights of task-specific layers (e.g., question answering, text classification layers) are randomly assigned before

[1]This post-processing script can be obtained at https://github.com/dmis-lab/biobert/blob/master/biocodes/transform_nbset2bioasqform.py

the fine-tuning. This randomness will cause the performance of Transformers-based models to fluctuate on downstream tasks, especially in smaller datasets such as BioASQ [12]. We examine the range of this randomness at the BioASQ10B challenge by using two identical BioM-ALBERT models (UDEL-LAB1, UDEL-LAB2), where both use the same hyper-parameters and fine-tuning dataset. In the result section, we report our observation and the effect of weights randomness on both Factoid and List questions.

3.6. Task-To-Task Transferability

The transferability between SQuAD and The Multi-Genre Natural Language Inference MNLI [21] dataset at the BioASQ challenge was first observed by the BioBERT team in their early work [22]. MNLI is a text classification task that is part of the GLUE benchmark. This year, we followed a similar approach by testing the transferability effect on the performance with our BioM-ELECTRA models. At the BioASQ10B challenge, we use BioM-ELECTRA-SQuAD (UDEL-LAB3) and BioM-ELECTRA-SQuAD-MNLI (UDEL-LAB4) models to test this transferability effect. For the BioM-ELECTRA-SQuAD model, we first fine-tune our BioM-ELECTRA model on the SQuAD2.0 dataset and then on the training set of the BioASQ10B. For our BioM-ELECTRA-SQuAD-MNLI model, we fine-tune our BioM-ELECTRA model on the MNLI dataset first, then on the SQuAD2.0 set, and finally on the training set of the BioASQ10B dataset. We report our findings of this approach for both BioASQ10B-Factoid and List questions.

4. Results and Discussion

In this section, we reported the results of our models at the BioASQ10B challenge. We split our results and discussion based on Yes/No, Factoid, and List questions. Results are taken from the official leaderboard of the BioASQ10B challenge [2]. For both Factoid and List questions, we analyze the effect of Transferability (MNLI-to-SQuAD) with both our models BioM-ELECTRA-SQuAD (UDEL-LAB3) and BioM-ELECTRA-SQuAD-MNLI (UDEL-LAB4). For both Factoid and List tasks, we reported our findings of the randomness of the results by running two BioM-ALBERT models (UDEL-LAB1, UDEL-LAB2) where both use the same hyper-parameters and fine-tuning setting. Results are sorted using Accuracy, mean reciprocal rank (MRR), and mean F-Measure as main ranks for Yes/No, Factoid and List questions, respectively.

4.1. Yes/No Questions

Table 1 shows the result of our models against other participants' teams on Yes/No questions. Results show that adapting the text-classification approach with both BioM-ELECTRA and BioM-ALBERT shows effectiveness in addressing the performance on Yes/No questions. This approach leads us to take the lead in Batch 2,3, and 4. In batch 3, the result shows that we took the lead against other systems by scoring 100% accuracy with both BioM-ELECTRA and BioM-ALBERT. Also, we can observe the consistency in results between BioM-ELECTRA and

[2] Our results on this paper are from preliminary results reported on the BioASQ10B leaderboard, which can be accessed at http://participants-area.bioasq.org/results/10b/phaseB/. We only reported the name of other team systems without a description of each since the BioASQ team has not released further details.

BioM-ALBERT in all batches except batch 4. This consistency also highlights that randomness is less likely to occur with the binary dataset (Yes/No).

Table 1
Results of our Models on BioASQ10B-YesNo task.

Batch	Model	**Accuracy**	**Macro F1**
Batch1	bio-answerfinder	1.0000	1.0000
	bio-answerfinder-2	1.0000	1.0000
	LaRSA	0.9565	0.9464
	BioM-ELECTRA	0.9565	0.9403
	BioM-ALBERT	0.9565	0.9403
Batch2	orpheus_kg	1.0000	1.0000
	BioM-ALBERT	1.0000	1.0000
	BioM-ELECTRA	1.0000	1.0000
	Ir_sys1	1.0000	1.0000
	bio-answerfinder	0.9444	0.9345
Batch3	BioM-ALBERT	1.0000	1.0000
	BioM-ELECTRA	1.0000	1.0000
	KU-AAA637-system2	0.9600	0.9524
	KU-AAA637-system3	0.9600	0.9524
	Ir_sys1	0.9600	0.9524
Batch4	BioM-ELECTRA	1.0000	1.0000
	Ir_sys1	1.0000	1.0000
	Ir_sys2	1.0000	1.0000
	lalala	1.0000	1.0000
	bio-answerfinder	0.9583	0.9473
Batch5	KU-AAA637-system2	0.9286	0.9271
	Ir_sys1	0.9286	0.9282
	Ir_sys2	0.9286	0.9282
	lalala	0.9286	0.9282
	BioM-ALBERT	0.8929	0.8893

4.2. Factoid Questions

Table 2 shows the results of our models on BioASQ10B-Factoid questions ranked by the mean reciprocal rank (MRR). Results illustrate that combining both Factoid and List questions led us to take the lead in both batch 1 and batch 3. Taking the lead in two batches, not only one, with this method indicates that randomness was not the reason for this performance but our technique to combine both Factoid and List questions. Moreover, we can observe significant randomness (3%) in MRR performance with both BioM-ALBERT-SQuAD-Run2 and BioM-ALBERT-SQuAD-Run1 in the first and third batches.

Similar to last year's BioASQ9B result, Task-to-Task transferability did not always lead to better performance than the BioM-ELECTRA-SQuAD model. Furthermore, We can observe that our BioM-ELECTRA models outperform our BioM-ALBERT on all four batches, suggesting that it is better to use BioM-ELECTRA models for BioASQ factoid questions. Indeed BioM-ELECTRA

Table 2
Results of our Models at BioASQ10B-Factoid task. Results show the top-5 best performing models from the BioASQ10B leaderboard. We extend the results in both Batch1 and Batch3 to show the effect of randomness by showing the performance of different runs of BioM-ALBERT.

Batch	Model	Strict Acc.	Lenient Acc.	**MRR**
Batch1	BioM-ELECTRA-Factoid+List [UDEL-LAB5]	0.3824	0.5588	0.4608
	lr_sys1	0.4118	0.5000	0.4559
	BioM-ALBERT-SQuAD-Run2 [UDEL-LAB2]	0.3824	0.5588	0.4534
	lr_sys3	0.4118	0.5000	0.4485
	lalala	0.3824	0.5000	0.4363
	BioM-ALBERT-SQuAD-Run1 [UDEL-LAB1]	0.3529	0.5588	0.4299
Batch2	BioM-ELECTRA-SQuAD [UDEL-LAB3]	0.5588	0.6765	0.6000
	BioM-ELECTRA-SQuAD-MNLI [UDEL-LAB4]	0.5588	0.6765	0.5912
	BioM-ALBERT-SQuAD-Run2 [UDEL-LAB2]	0.5588	0.6176	0.5882
	BioM-ALBERT-SQuAD-Run1 [UDEL-LAB1]	0.5588	0.6176	0.5809
	lr_sys3	0.5588	0.6176	0.5809
Batch3	BioM-ELECTRA-Factoid+List [UDEL-LAB5]	0.5313	0.6563	0.5792
	KU-AAA637-system4	0.5000	0.6875	0.5755
	BioM-ALBERT-SQuAD-Run1 [UDEL-LAB1]	0.5313	0.6250	0.5729
	LaRSA	0.5000	0.6563	0.5677
	KU-AAA637-system2	0.5000	0.6875	0.5661
	KU-AAA637-system3	0.5000	0.6875	0.5651
	KU-AAA637-system1	0.4688	0.6875	0.5505
	BioM-ALBERT-SQuAD-Run2 [UDEL-LAB2]	0.4688	0.6250	0.5417
Batch4	lalala	0.5806	0.6452	0.5995
	lr_sys3	0.5161	0.6774	0.5806
	KU-AAA637-system4	0.5161	0.6452	0.5656
	BioASQ-2022_UNCC	0.5161	0.6129	0.5645
	BioM-ELECTRA-SQuAD-MNLI [UDEL-LAB4]	0.5484	0.6129	0.5613
Batch5	lr_sys3	0.4828	0.5862	0.5098
	NCU-IISR-AS-GIS-4	0.4483	0.5862	0.4983
	NCU-IISR-AS-GIS-5	0.4483	0.5862	0.4983
	BioM-ELECTRA-SQuAD [UDEL-LAB3]	0.4138	0.5862	0.4828
	BioASQ-2022_UNCC1	0.4138	0.6207	0.4764

models also have less hidden layer size (1024) than BioM-ALBERT models (4096), and this leads to better inference and fine-tuning time (0.33x), as we show in our early work [11].

4.3. List Questions

The threshold value, part of the "nbset2bioasqform" script, plays a significant role in choosing which answers can be considered candidate answers for list-type questions. We optimized the threshold value using the training and test dataset of the BioASQ9B challenge. Figure 3 shows how our BioM-ALBERT performs in terms of F-Measure score on last year's BioASQ9B-List task with different threshold values. Results in figure 3 show that we gain %4 (0.68-0.64) improvement in the F-measure score at the 0.18 threshold compared to the default value of the threshold set

by the BioBERT team (0.42). Based on these results, we chose our threshold value for this year's challenge to be 0.18.

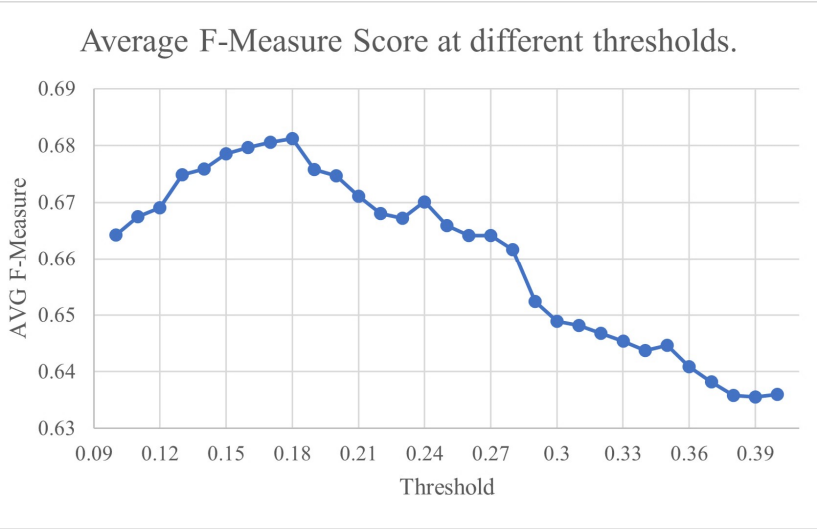

Figure 3: Performance of our BioM-ALBERT on BioASQ9B-List questions with different threshold values. The default threshold value assigned by BioBERT team is 0.42. We take the average F-Measure score of five different batches of BioASQ9B-List task. We use last year BioASQ9B-List since we have access to both the training and golden dataset.

We believe that this choice alone explains the significant margin we have in terms of performance against other models at the BioASQ10B-list task this year. Table 3 shows the performance of our models on the BioASQ10B-list task. Our models score among the top best-performing models in all five batches of the BioASQ10B-list task. In contrast to the Factoid task, BioM-ALBERT performs better than BioM-ELECTRA on list questions. In addition, the performance of two runs of BioM-ALBERT models shows a larger range of randomness (1-7%) in performance than the Factoid task (0-%3).

5. Conclusion

This paper illustrates and explains the methods and models we use at the BioASQ10B challenge. We show that by adapting the Text Classification approach for Yes/No questions and tuning our hyper-parameters, we reach 100% accuracy in batches 2,3, and 4 and score second with 95.6% accuracy on the First and last Batch. In addition, our new technique to combine both Factoid and List questions contributes to the lead we achieve in two batches of BioASQ10B-factoid questions. Finally, we show how tuning threshold probability from 0.42 to 0.18 led to a significant performance gain on the BioASQ10B-list task, leading us to rank first in all five batches of the BioASQ10B-list task. For our future work, we will focus on addressing the limited size of the BioASQ dataset through data augmentation and investigate building an ensemble model with both our models BioM-ELECTRA and BioM-ALBERT.

Table 3
Results of our Models at BioASQ10B-List task. We only show top-5 best performing systems on F-Measure score.

Batch	Model	Mean Prec.	Recall	**F-Measure**
Batch1	BioM-ALBERT-Run2	0.7201	0.8405	0.7469
	BioM-ALBERT-Run1	0.6974	0.8226	0.7346
	BioM-ELECTRA-SQuAD-MNLI	0.6762	0.8464	0.7229
	BioM-ELECTRA-SQuAD	0.6893	0.7429	0.6731
	lalala	0.6046	0.7286	0.6459
Batch2	BioM-ELECTRA-SQuAD	0.7042	0.7400	0.7051
	BioM-ELECTRA-SQuAD-MNLI	0.6859	0.7530	0.7011
	BioM-ALBERT-Run2	0.6914	0.7193	0.6787
	BioM-ALBERT-Run1	0.6707	0.6530	0.6393
	lalala	0.4955	0.6067	0.5177
Batch3	BioM-ALBERT-Run2	0.5442	0.6742	0.5655
	BioM-ALBERT-Run1	0.5174	0.6591	0.5558
	BioM-ELECTRA-SQuAD-MNLI	0.5293	0.6439	0.5255
	BioM-ALBERT-Run1	0.5263	0.5985	0.5188
	bio-answerfinder	0.6273	0.4472	0.4843
Batch4	BioM-ALBERT-Run2	0.5834	0.5844	0.5386
	BioM-ALBERT-Run1	0.5799	0.5017	0.4950
	BioM-ELECTRA-SQuAD-MNLI	0.6162	0.4753	0.4752
	BioM-ELECTRA-SQuAD	0.5584	0.4438	0.4501
	lalala	0.4089	0.4507	0.3835
Batch5	BioM-ELECTRA-SQuAD-MNLI	0.6009	0.6313	0.6016
	BioM-ALBERT-Run2	0.5587	0.6297	0.5793
	BioM-ALBERT-Run1	0.6003	0.5795	0.5707
	BioM-ELECTRA-SQuAD	0.5651	0.5929	0.5669
	NCU-IISR-AS-GIS-4	0.6222	0.4975	0.5332

Acknowledgments

The authors would like to Acknowledge the ultimate support from Google Research Cloud TRC for providing access to Tensor Processing Unit TPU which we use to pre-train our BioM-Transformers and fine-tune our model for Both BioASQ9B and BioASQ10 challenges. The authors also would like to thank BioBERT team for their continuous effort to make their codes (e.g transform_nbset2bioasqform.py) available to the public community and share their codes and resources on their GitHub repository.

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: https://doi.org/10.1093/bioinformatics/btz682. doi:10.1093/bioinformatics/btz682.

[2] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional

transformers for language understanding, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 4171–4186. URL: https://www.aclweb.org/anthology/N19-1423. doi:10.18653/v1/N19-1423.

[3] A. Nentidis, K. Bougiatiotis, A. Krithara, G. Paliouras, Results of the seventh edition of the bioasq challenge, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, Springer, 2019. URL: https://arxiv.org/pdf/2006.09174.pdf.

[4] 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.

[5] A. Nentidis, G. Katsimpras, E. Vandorou, A. Krithara, L. Gasco, M. Krallinger, G. Paliouras, Overview of BioASQ 2021: The ninth BioASQ challenge on large-scale biomedical semantic indexing and question answering, in: Lecture Notes in Computer Science, Springer International Publishing, 2021, pp. 239–263. URL: https://doi.org/10.1007%2F978-3-030-85251-1_18. doi:10.1007/978-3-030-85251-1_18.

[6] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, Roberta: A robustly optimized bert pretraining approach, 2019. arXiv:1907.11692.

[7] K. Clark, M.-T. Luong, Q. V. Le, C. D. Manning, Electra: Pre-training text encoders as discriminators rather than generators, 2020. arXiv:2003.10555.

[8] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, Q. V. Le, Xlnet: Generalized autoregressive pretraining for language understanding, in: H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, R. Garnett (Eds.), Advances in Neural Information Processing Systems, volume 32, Curran Associates, Inc., 2019. URL: https://proceedings.neurips.cc/paper/2019/file/dc6a7e655d7e5840e66733e9ee67cc69-Paper.pdf.

[9] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, R. Soricut, Albert: A lite bert for self-supervised learning of language representations, 2020. arXiv:1909.11942.

[10] S. Alrowili, V. K. Shanker, Large biomedical question answering models with albert and electra, in: CLEF, 2021. URL: http://ceur-ws.org/Vol-2936//paper-14.pdf.

[11] S. Alrowili, V. Shanker, BioM-transformers: Building large biomedical language models with BERT, ALBERT and ELECTRA, in: Proceedings of the 20th Workshop on Biomedical Language Processing, Association for Computational Linguistics, Online, 2021, pp. 221–227. URL: https://www.aclweb.org/anthology/2021.bionlp-1.24. doi:10.18653/v1/2021.bionlp-1.24.

[12] Y. Gu, R. Tinn, H. Cheng, M. Lucas, N. Usuyama, X. Liu, T. Naumann, J. Gao, H. Poon, Domain-specific language model pretraining for biomedical natural language processing, 2021. arXiv:2007.15779.

[13] R. Tinn, H. Cheng, Y. Gu, N. Usuyama, X. Liu, T. Naumann, J. Gao, H. Poon, Fine-tuning large neural language models for biomedical natural language processing, 2021. URL: https://arxiv.org/abs/2112.07869. doi:10.48550/ARXIV.2112.07869.

[14] P. Lewis, M. Ott, J. Du, V. Stoyanov, Pretrained language models for biomedical and

clinical tasks: Understanding and extending the state-of-the-art, in: Proceedings of the 3rd Clinical Natural Language Processing Workshop, Association for Computational Linguistics, Online, 2020, pp. 146–157. URL: https://www.aclweb.org/anthology/2020.clinicalnlp-1.17. doi:10.18653/v1/2020.clinicalnlp-1.17.

[15] H.-C. Shin, Y. Zhang, E. Bakhturina, R. Puri, M. Patwary, M. Shoeybi, R. Mani, BioMegatron: Larger biomedical domain language model, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Online, 2020, pp. 4700–4706. URL: https://www.aclweb.org/anthology/2020.emnlp-main.379. doi:10.18653/v1/2020.emnlp-main.379.

[16] M. Yasunaga, J. Leskovec, P. Liang, LinkBERT: Pretraining language models with document links, in: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Dublin, Ireland, 2022, pp. 8003–8016. URL: https://aclanthology.org/2022.acl-long.551. doi:10.18653/v1/2022.acl-long.551.

[17] Y. You, J. Li, S. Reddi, J. Hseu, S. Kumar, S. Bhojanapalli, X. Song, J. Demmel, K. Keutzer, C.-J. Hsieh, Large batch optimization for deep learning: Training bert in 76 minutes, 2019. URL: https://arxiv.org/abs/1904.00962. doi:10.48550/ARXIV.1904.00962.

[18] 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. Le Scao, S. Gugger, M. Drame, Q. Lhoest, A. 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://aclanthology.org/2020.emnlp-demos.6. doi:10.18653/v1/2020.emnlp-demos.6.

[19] P. Rajpurkar, R. Jia, P. Liang, Know what you don't know: Unanswerable questions for SQuAD, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Association for Computational Linguistics, Melbourne, Australia, 2018, pp. 784–789. URL: https://www.aclweb.org/anthology/P18-2124. doi:10.18653/v1/P18-2124.

[20] W. Yoon, J. Lee, D. Kim, M. Jeong, J. Kang, Pre-trained language model for biomedical question answering, 2019. URL: https://arxiv.org/abs/1909.08229. doi:10.48550/ARXIV.1909.08229.

[21] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, S. Bowman, GLUE: A multi-task benchmark and analysis platform for natural language understanding, in: Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP, Association for Computational Linguistics, Brussels, Belgium, 2018, pp. 353–355. URL: https://www.aclweb.org/anthology/W18-5446. doi:10.18653/v1/W18-5446.

[22] 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.

A. Appendix

Table 4 shows the details of our hyper-parameters that we use for both BioM-ELECTRA and BioM-ALBERT for Yes/No, Factoid, and List questions.

Table 4
Details of fine-tuning hyperparameters that we use for both BioM-ALBERT and BioM-ELECTRA. (MSL=Max Seq. Length)

Task	Model	Learning Rate	Warmup Ratio	Batch	Epochs	MSL
Yes/No	BioM-ELECTRA	3e-5	0.0	16	5	512
Yes/No	BioM-ALBERT	3e-5	0.0	8	5	256
Factoid	BioM-ELECTRA	2e-5	0.0	24	4	512
Factoid	BioM-ALBERT	1e-5	0.1	128	5	384
List	BioM-ELECTRA	2e-5	0.0	24	4	512
List	BioM-ALBERT	1e-5	0.1	128	5	384