Question Answering Chatbots for Biomedical Research using Transformers

Evdokia Xygi
Computer Engineering and Informatics Dpt.
University of Patras
Patras, Greece
up1047172@upnet.gr

Andreas D. Andriopoulos
Computer Engineering and Informatics Dpt.
University of Patras
Patras, Greece
a.andriopoulos@upatras.gr

Dimitrios A. Koutsomitropoulos
Computer Engineering and Informatics Dpt.
University of Patras
Patras, Greece
koutsomi@ceid.upatras.gr

Abstract — **Professionals as well as the general public need effective help to access, understand and consume complex biomedical concepts. The existence of an interaction environment capable of automatically processing such information - thus replacing human intervention - such as chatbots, is however challenging. In this paper we propose a method of utilizing chatbots in the domain of biomedicine. In the implementation we choose to incorporate the BERT algorithm, so as to adopt a modern technique for natural language processing tasks. We use several pre-trained models (RoBERTa, XLM-R, BERT Large, and BioBert) in order to evaluate their ability to back the chatbot infrastructure. The data is retrieved from the PubMed repository, with the final set being formed into full sentences or potential chatbot responses, thus preserving their conceptual meaning. Response selection is performed using similarity metrics and F-score. The results create a ranking of the models placing related ones closely, recognizing the ability to always answer each question and highlighting the importance of the training previously applied to them. These are compared to the Count Vectorizer technique, which appears to perform better, but with several weaknesses, as many questions could not be answered.**

Keywords—: chatbot, natural language processing, word embeddings, BERT, Count Vectorizer

I. INTRODUCTION

The acquisition of biomedical knowledge has been a major task in Information Retrieval (IR) and knowledge management in recent years [1]. There has been an increase in the interest of the general public in their personal health research and medical data via the Internet. Until recently, IR systems such as PubMed [2] have been developed and used to meet such needs. Nowadays, chatbot technologies see rapid growth, especially in the medical field, and many medical chatbot systems have been proposed over the years with the goal of being the best quality system that can be created for the medical community and beyond [3]. Medical chatbots provide guidelines for proper nutrition, lifestyle changes and further pathology tests based on patients' symptoms provided upon entry and use AI. They can also prescribe appropriate medication, which can be further approved by the patient's physician [4].

In our approach we investigate the relationship that could be developed between chatbots and biomedical data. By choosing to create a Question Answering (QA) chatbot and inserting biomedical data - and not only - into it, we aim through various tests to evaluate possible performance and to make observations about the reaction of the chatbot in each case. For the implementation we use word embeddings and word counting algorithms, thus integrating state-of-the-art Machine Learning (ML) techniques for Natural Language Processing (NLP) and Natural Language Understanding (NLU) tasks.

We use four BERT models (RoBERTa, XLM-R, BERT Large, and BioBERT) as well as a count-based model for input recognition and we search for each probable answer with the use of the cosine similarity metric. We set the suggestions of the system, which we evaluate using F-score, to consist of full sentences. We, thus, utilize the biomedical information by preserving the semantic background and avoiding any corruption to accurate and, above all, valid data accumulated after years of research. The results, although small-scale, combined with the implementation methodology and sources provide the guidelines for future research. Our source code is openly available at: https://github.com/Evdokia97/MediBot.

To the best of our knowledge, biomedical chatbots based on pure transformed-based language models, such as BERT, have not been extensively investigated before. Most existing related studies appear to utilize classifiers such as K-nearest neighbor (KNN), Support Vector Machines (SVM), Decision Trees, etc. for the output. In addition, they are limited to short length answers, that is, without incorporating essential knowledge for obtaining a medical opinion.

In the following, we first review related work in the field of chatbots in Biomedicine. Next, in Section III, we present our methodology and architectural details for word embeddings (BERT), the count vectorizer and how these are utilized by our chatbot. Section IV describes the design of our experiments, datasets used for evaluation and the evaluation metrics. Section V discusses results by testing our approach in terms of similarity scores, precision, recall and F-score. Our conclusions and future work are summarized in the last section VI.

II. RELATED WORK

In recent years, several attempts have been made to create various kinds of chatbots in the domain of biomedicine. Selected chatbots which use NLP to develop conversational systems for this field are presented below.

Text Messaging-Based Chatbot [5] is a personalized diagnosis system utilizing self-input from users, developed for the Covenant University Doctor (CUDoctor) telehealth system, to effectively diagnose diseases. It focuses on assessing the symptoms of tropical diseases in Nigeria. It is based on fuzzy logic rules and fuzzy inference. The knowledge base consisting of diseases and symptoms is acquired from medical ontologies. The inputs, Short Messaging Service (SMS), are recognized by NLP, important keywords are extracted and forwarded to the CUDoctor for decision support. Finally, the usability of the developed system is positively evaluated with 80.4 score using the System Usability Scale (SUS).

Chatbot for Disease Prediction [6] is built to be a conversational agent that discusses with users about their health issues and based on their symptoms, which it is able to identify, returns the diagnosis. Using these extracted symptoms, the chatbot predicts the disease and recommends a proper treatment. To achieve this, the system combines NLP for simple symptom analysis and KNN for classification of symptoms, predicts the disease, and finally recommends the suitable treatment.

Health Chatbot [7] is a proposed chatbot using NLP to provide some information about health. It is capable to understand and answer the questions asked by the user. The cosine similarity is used to find the similarities between the query words and the documents and then return the answers of the document with the highest similarity. Also, to enhance chatbot performance, ID3 Decision tree is used. This medical chatbot has successfully diagnosed the user's illness with approximately 87 percent accuracy.

Medbot [8] is a multilingual conversational AI-based application which attempts to reduce COVID-19 transmission among patients and clinicians. This is achieved by permitting patients to receive supportive care without physically visiting a hospital in rural India. Also, it detects various common diseases and suggests home remedies. It has additional features including local food diets, age and gender-specific health check-up advice, and emergency helpline numbers with a real-time messaging. This application is based upon a serverless architecture on Google Cloud Platform (GCP), embedded NLP and NLU to understand the user's query and return respective responses.

A Smart Chatbot Architecture [9] is proposed to build an intelligent chatbot for health care assistance. A fundamental component is an NLP Engine which is used to identify the intent of the user. It also includes an intent classifier and an entity extractor to interpret the meaning and extract the critical information from a user's query respectively. Furthermore, an agent for Dialogue Management handling the real context of the user's query, a QA system with manual or automated trainings, an application programming interface (API), a server which handles the user's request, other plugins-components, and finally, a user-friendly Front-End system are included in the system.

III. Design and Methodology

A. BERT Algorithm

Word embeddings use techniques to convert words into vectors so that if they are semantically close, they have a similar vector representation. In recent years many approaches have been made with Transformers [10], since their introduction in 2017, offering improved parallelization and better modeling of long-range dependencies. Pre-trained language models such as Bidirectional Encoder Representations from Transformers (BERT) [11], have proven effective for NLP tasks.

BERT is a model designed for a priori training, with deep, bidirectional, label-free text representations, with shared context conditioning at all layers. It makes use of an attention mechanism that learns contextual relations between words (or sub-words) in a text corpus. Such a model can be used to estimate semantic similarity between texts [12], or offer its improvement (get fine-tuned), with just one additional layer of output, thus offering state-of-the-art models which can be applied in various domains, such as question answering, classification and language inference, without substantial architecture modifications.

Next, four BERT models have been selected with criteria such as improved results, complex architecture, multilingual representations, biomedical domain corpus and are presented in detail.

B. BERT Models

- RoBERTa model is the Robustly Optimized BERT Pre-training Approach (RoBERTa) [13]. This particular improved version of BERT is based on published BERT results, in GLUE, RACE and SQuAD. It is an extension of the BERT model, which is done with longer training of the model on additional data (such as CC-NEWS), longer batches and sequences, removing the next sentence prediction objective, thus giving the possibility for performance to be substantially improved, compared to the way of dealing with the data of the simple BERT. Another difference between these two models is the usage of the dynamic masking pattern of RoBERTa, which replicates the training data and performs a series of masking strategies, different for each run, as opposed to BERT, which uses a static strategy masking, which is performed during data preprocessing.

- Bert-Large model is a type of BERT model [11]. This model performs better at the results of GLUE benchmarks than BERT-base. Important differences between the two models are found in the complexity they adopt in their architecture. Specifically, BERT Large model has 24 encoder layers stacked on top of each other, whereas BERT Base has the half, 12 layers. BERT Large has a total of 16 attention heads and 340 million parameters, while BERT Base has 12 attention heads with 110 million parameters. BERT Large has 1024 hidden layers, whereas BERT Base has 768 hidden layers. Finally, BERT-large can achieve a good performance on NLP tasks; however, it is a relatively expensive model which requires a lot of computational power and memory availability.

- XLM-R is a model which further improved the results of XLM, where 'R' stands for RoBERTa, without relying on the supervised Translation Language Modeling (TLM) [14]. It uses 2.5TB training data extracted from Common Crawl and a larger vocabulary size of 250K, covering 100 different languages. This particular multilingual Masked Language Model (MLM) has the potential to study unsupervised cross-linguistic representations of general-purpose text at a very large scale, exposing the effectiveness of multilingual models over monolingual models. In addition, in Cross-Lingual Understanding (CLU) task, a model is used with other languages without additional training data, trained only in one language. It also provides state-of-the-art performance on cross-lingual classification, sequence labeling, and question-answering (QA), tasks which provide strong gains over previous multilingual models like mBERT and XLM.

- BioBERT model is a pre-trained language representation model for the biomedical domain with its weights initialized based on the BERT weights [15]. While BERT is pre-trained on general domain text (English Wikipedia and Books Corpus), BioBERT is trained on biomedical domain texts (PubMed abstracts and PMC full-text articles). While BERT has demonstrated the effectiveness of contextual word representations, it cannot achieve high performance on biomedical texts as it is only pre-trained on general domain texts. On the contrary, BioBERT outperforms BERT models on biomedical text mining tasks such as Named-Entity Recognition (NER), Relation Extraction (RE), and Question-Answering (QA).

C. *Count-based method*

CountVectorizer is a tool for extracting features from a text corpus [16]. It is used to transform a given text into a vector, equal to the size of our vocabulary, based on the frequency of each word that appears in the given text. This is helpful in the case of multiple such texts where we wish to convert each word in each text into vectors for use in further text analysis. It creates a matrix in which each unique word is represented by a column of the matrix and each document is a row in the matrix. The value of each cell is the count of occurrences of the word in that particular text. Therefore, it is a flexible feature representation module for text with less need for computational power.

D. *Implementation*

Chatbots manage a conversation with humans, by understanding their language and providing a response. Algorithms such as those presented at the beginning of this chapter help with this task. The vectorization they achieve on existing data leads firstly to understanding and then to the retrieval or generation of an answer. Each answer case requires a large amount of training data, with the generation being unable to avoid high-probability generic responses.

In terms of our chatbot type (Fig. 1), we chose to use a QA chatbot. These are information retrieval systems which can handle complex NLP queries in order to return answers related to input data. Retrieval-based chatbots provide the best possible response from a dataset of predefined responses, and do not produce new output. In our chatbot implementation, by using existing biomedical information and word embeddings techniques such as Domain Specific BERT models and CountVectorizer method, we determine the most appropriate response.

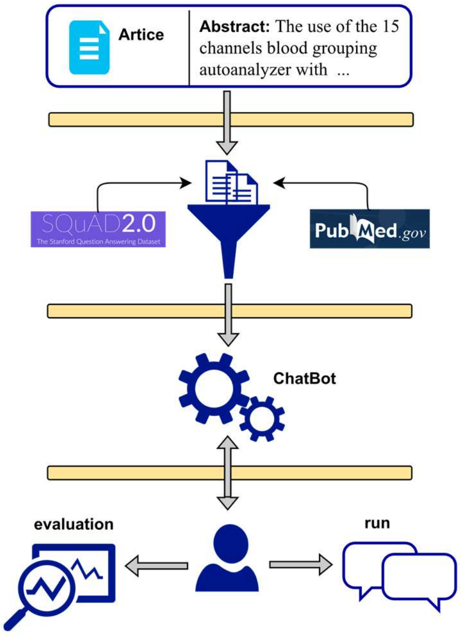
Artice
Abstract: The use of the 15 channels blood grouping autoanalyzer with ...
SQuAD2.0
The Stanford Question Answering Dataset
PubMed.gov
ChatBot
evaluation
run

Fig. 1. A general structure of the system.

Another reason why we chose the QA chatbot is the type of data we drive to our input. Our data concerns in-depth medical texts with precise definitions and numerical data. Therefore, our goal was the most possible accurate response from the chatbot and to avoid generating suggestions with a possible risk of mixing data or producing incorrect medical suggestions.

We have developed two main chatbot variants: one version containing BERT algorithms and a second one containing the CountVectorizer. After our data is first imported and processed so that it can be treated separately as sentences and not as paragraphs, it is loaded into the respective model and stored with it in a local environment for future use. Then, the chatbot asks for the user's name and starts the conversation with them to learn their intention (run or evaluate), and to act accordingly. The set of processes that the chatbot can perform is captured in the following algorithm (Table I).

TABLE I. CHATBOT OPERATION ALGORITHM

D: dataset, d: sentence, L: set of sentences, V: vectors, G: greetings, GT: ground truth, t, k: user response, h: threshold, S_d: similarity score

```
$t \leftarrow input()$
$if\ t \in G$:
    $continue$    //with predefined answers from chatbot
$if\ run\_mode$:
    $V_t \leftarrow infer(t)$
    $\forall d \epsilon D$:
        $S_d \leftarrow compute\_similarity(V_t, V_d)$
        $L \leftarrow L \cup \{ d \mid S_d > h \}$
    $return\ L$
$if\ eval\_mode$:
    $k \leftarrow input(GT)$
    $V_t \leftarrow infer(t)$
    $\forall d \epsilon D$:
        $S_d \leftarrow compute\_similarity(V_t, V_d)$
        $L \leftarrow L \cup \{ d \mid S_d > h \}$
    $return\ f \leftarrow compute\_fscore(k, L)$
```

With the *run* option, the chatbot asks the user to enter a sentence, entering their data and asking for a response. The chatbot's response depends on whether the user's data is about a welcome word, their intent to end the conversation, or their intent to ask a question. In the first case, the response to the user will be given by some predefined sentences. In the case of a query, the user's input data will be first fed into the model to compare it with the existing data. A response is provided to the user only when data exceeding a certain similarity threshold are found.

With the *eval* option, operation remains the same, with the difference that as soon as the question is given by the user, the chatbot asks the user to enter the possibly correct answer that should be given by it. The chatbot then compares its answer with the user's and prints its answer along with the success or failure rates. For various similarity thresholds we compare inputs, i.e., the answer of the chatbot (prediction) and the answer we consider correct (ground truth), and we compute precision, recall and F-score.

IV. Experimental Setup

A. Comparison Metrics

Cosine similarity is the metric we use to determine the similarity of two vectors on an inner product space. Regarding the operation, it accepts two vectors as arguments, which are inferred from sentences with word embeddings, and returns the score in the closed interval between zero and one.

One of the most common ways to evaluate the performance of a model, usually in text classification problems, is the F-score. The F-score is the harmonic mean of precision and recall of a prediction model. To calculate precision and recall, we enumerate common words between ground truth and prediction.

B. Datasets

We use 1,100 abstracts from medical papers, as well as 250 SQuAD v2.0 data. These texts are separated into sentences after entering the program, offering 8,737 sentences for comparison.

Regarding the PubMed data, after parsing it into text format, only the abstract part is kept. In the preprocessing, the corpus is split into sentences. The topics covered by the archive are varied. They concern various diseases, genes, microorganisms and more general techniques for dealing with diseases.

As for the SQuAD v2.0 data [17], this new version retains the possibility that there is no short answer in the given paragraph, making the problem more realistic. More specifically, the data set of this publication consists of 536 sample top Wikipedia articles, which cover a wide range of topics. The data set is constructed from questions and answers of the specific texts and can be expanded according to the answer.

V. Results and Discussion

The experiments are performed on a chatbot with four different BERT models and with CountVectorizer. 100 questions are selected based on the data we enter, which include general, composite or highly specific ones. However, there are cases where no response is given that meets the similarity threshold percentages we have set, named as *Not Answered*. There are also responses which do not have any success rate, due to zero common words with the ground truth; these are classified as *Not Related*. Therefore, overall metrics are calculated exclusively for data which yield scores greater than zero, i.e., only *Related*. The following tables II-IV show the results for all the models in detail.

TABLE II. EXPERIMENT RESULTS, SIMILARITY>0.5

Models	**Cosine Similarity 0.5**					
	Not Ans-wered	**Not Related**	**Related**	**Precision**	**Recall**	**F score**
RoBERTa	18%	2%	80%	0.256	0.392	0.293
XLM-R	3%	1%	96%	0.177	0.289	0.203
BERT Large	0%	2%	98%	0.224	0.385	0.263
BioBERT	0%	2%	98%	0.211	0.349	0.248
Count-Vectorizer	64%	0%	36%	0.804	0.902	0.827

TABLE III. EXPERIMENT RESULTS, SIMILARITY>0.4

Models	**Cosine Similarity 0.4**					
	Not Ans-wered	**Not Related**	**Related**	**Precision**	**Recall**	**F score**
RoBERTa	4%	0%	96%	0.215	0.353	0.252
XLM-R	1%	1%	98%	0.175	0.285	0.200
BERT Large	0%	2%	98%	0.224	0.384	0.259
BioBERT	0%	1%	99%	0.211	0.346	0.247
Count-Vectorizer	40%	1%	59%	0.659	0.810	0.689

TABLE IV. EXPERIMENT RESULTS, SIMILARITY>0.3

Models	**Cosine Similarity 0.3**					
	Not Ans-wered	**Not Related**	**Related**	**Precision**	**Recall**	**F score**
RoBERTa	4%	0%	96%	0.205	0.342	0.242
XLM-R	1%	1%	98%	0.175	0.287	0.202
BERT Large	0%	2%	98%	0.224	0.385	0.263
BioBERT	0%	1%	99%	0.211	0.347	0.248
Count-Vectorizer	19%	3%	78%	0.460	0.728	0.531

The results show that, of the various BERT models, RoBERTa is more sensitive to the similarity threshold, because, for threshold 0.5, 18% of the questions are not answered, while, when the threshold is reduced to 0.3, only 4% remain as unanswered questions. Additionally, F-score also decreases; therefore, we observe that this model responds selectively for greater percentages of similarity and this strict selection turns in favor of the F-score. The other BERT models instead allow more sentences to pass the threshold, which helps them with lower threshold values. For threshold 0.3, BERT Large with its complex architecture comes first, while BioBERT, which uses a different tokenizer and RoBERTa, with dynamic masking pattern, follow. Finally, the multilingual XLM-R ranks last.

Regarding the CountVectorizer, the threshold functions decisively in the value of the F-score (Fig. 2). For a threshold of 0.5, 64% of the questions are not answered, while, as it decreases to 0.3, the percentage drops to 19%. As for the rest of the metrics, regardless of the threshold, recall is higher than precision in the total of BERT as well as CountVectorizer models, indicating that the correct answer is sometimes found verbatim within the set of answers given by the model.

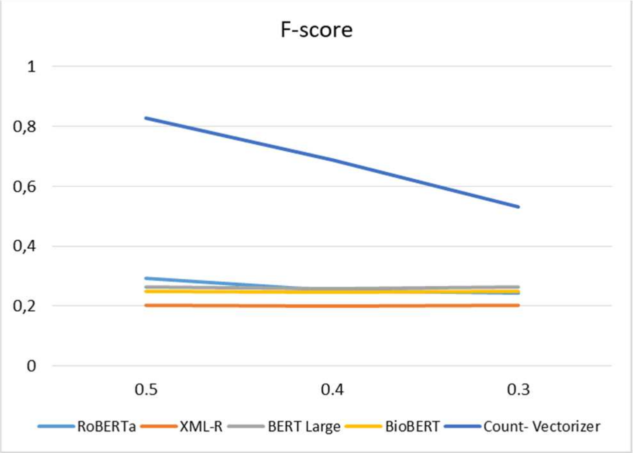

Figure 2. F-score comparison between chatbot variants.

In general, BERT can generate different word embeddings for a word by capturing its context and infer vectors after training. Instead, the CountVectorizer is used to convert a collection of text documents into a vector of term counts. It is observed that although a pre-trained model is expected to give better results, the chatbot which contains the CountVectorizer performs better, but with only a few response attempts. This results from the fact that CountVectorizer returns an encoded vector with the length of the vocabulary of our data set, containing the number of word occurrences, emphasizing each word individually and collecting the fraction of words within a sentence. On the other hand, BERT parses sentences holistically and makes predictions with the help of the initial vocabulary and outputs a weighted sum of the value vectors, paying close attention either to fixed offset positions or sentence as a whole. All in all, the results from CountVectorizer are context specific and cannot be generalized, unlike the BERT algorithm.

VI. CONCLUSIONS AND FUTURE WORK

The exponential growth of chatbot interaction into everyday life has shown that their evolution is ongoing, and their full potential is yet to come. We have shown a methodology and limitations of building a domain-specific QA chatbot based on state-of-the-art transformer-based language models. Bag-of-words approaches, such as CountVectorizer appear unsurprisingly to work better for 1-1 word matching; Nevertheless, the true merit of language models lies within the fact that not all questions can be replied verbatim and would at some point require generative text answering and/or open-ended conversation.

As a next step, we intend to observe the behavior of the model in data about specific medical issues and its performance in such cases. Regarding the target audience, the use of more data or data related to a specific subfield could form a chatbot to the aid of experts and professionals. Finally, an interactive platform could be developed, both for visual enhancement and for the convenience of potential users and the greater familiarity which could be developed by them.

REFERENCES

[1] L. Goeuriot, G. J. F. Jones, L. Kelly, H. Müller and J. Zobel, (2016). ‘‘Medical Information Retrieval: Introduction to the Special Issue,’’ Information Retrieval, January 2016, DOI: 10.1007/s10791-015-9 277-8.

[2] U.S. National Library of Medicine. PubMed.gov [Online]. https://www.nlm.nih.gov/databases/download/pubmed_medline.html

[3] Q. Jin, Z. Yuan, G. Xiong, Q. Yu, H. Ying, C. Tan, M. Chen, S. Huang, X. Liu and S Yu, “Biomedical Question Answering: A Survey of Approaches and Challenges,” In submission to ACM Computing Surveys, February 2021.

[4] J. Chaudhary, V. Joshi, A Khare, R. Gawali and A. Manna, “A Comparative Study of Medical Chatbots,” International Research Journal of Engineering and Technology (IRJET), Vol 08, Issue 02, February 2021.

[5] N.A Omoregbe, I.O. Ndaman, S. Misra, O.O. Abayomi-Alli and R. Damaševičius, ‘‘Text Messaging-Based Medical Diagnosis Using Natural Language Processing and Fuzzy Logic’’, in Journal of Healthcare Engineering, 2020, pp 1-14.

[6] R.B. Mathew, S.Varghese, S.E. Joy and S.S. Alex, ‘‘Chatbot for Disease Prediction and Treatment Recommendation using Machine Learning’’, in 3rd International Conference on Trends in Electronics and Informatics (ICOEI), 2019, pp. 851-856.

[7] P. I. Prayitno, R. P. Pujo Leksono, F. Chai, R. Aldy and W. Budiharto, ‘‘Health Chatbot Using Natural Language Processing for Disease Prediction and Treatment’’, in 1st International Conference on Computer Science and Artificial Intelligence (ICCSAI), 2021, pp. 62-67, doi: 10.1109/ICCSAI53272.2021.9609784.

[8] U. Bharti, D. Bajaj, H. Batra, S. Lalit, S. Lalit and A. Gangwani, ‘‘Medbot: Conversational Artificial Intelligence Powered Chatbot for Delivering Tele-Health after COVID-19,’’ in 5th International Conference on Communication and Electronics Systems (ICCES), 2020, pp. 870-875, doi: 10.1109/ICCES48766.2020.9137944.

[9] S. Ayanouz, B.A. Abdelhakim and M. Benahmed, ‘‘A Smart Chatbot Architecture based NLP and Machine Learning for Health Care Assistance’’, in Proceedings of the 3rd International Conference on Networking, Information Systems & Securit, 2020, DOI:10.1145/3386723.3387897

[10] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, ‘‘Attention is all you need’’, In Advances in Neural Information Processing Systems, pp. 6000–6010, 2017

[11] J. Devlin, M. W. Chang, K. Lee and K. Toutanova, ‘‘BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding’’, arXiv:04805v2[cs.CL], May 2019.

[12] D. A. Koutsomitropoulos and A. Andriopoulos, ‘‘Thesaurus-based Word Embeddings for Automated Biomedical Literature Classification,’’ Neural Computing and Applications. DOI: 10.1007/s00521-021-06053-z. Springer, 2021.

[13] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, et al. ‘‘Roberta: A robustly optimized bert pretraining approach,’’ ICLR 2020, arXiv preprint, arXiv:1907.11692.

[14] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzman, E. Grave, M. Ott, L. Zettlemoyer and V. Stoyanov, ‘‘Unsupervised Cross-lingual Representation Learning at Scale,’’ Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 8440-8451. 10.18653/v1/2020.acl-main.747.

[15] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. So and J. Kang, (2019). ‘‘BioBERT: a pre-trained biomedical language representation model for biomedical text mining. Bioinformatics’’ (Oxford, England). Bioinformatics, vol 36, Issue 4, February 2020, pp 1234–1240, 10.1093/bioinformatics/btz682.

[16] Scikit-learn, Machine Learning in Python. scikit-learn.org [Online]. https://scikit-learn.org/0.18/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html

[17] P. Rajpurkar, R. Jia and P. Liang, ‘‘Know What You Don’t Know: Unanswerable Questions for SQuAD,’’ 2018, pp. 784-789. DOI:10.18653/v1/P18-2124.