I.J. Intelligent Systems and Applications, **2023, 4, 53-69**
Published Online on August 8, 2023 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijisa.2023.04.05

MediBERT: A Medical Chatbot Built Using KeyBERT, BioBERT and GPT-2

Sabbir Hossain
Department of Computer Science, Faculty of Science and Technology, American International University-Bangladesh, Dhaka, Bangladesh
E-mail: 18-39103-3@student.aiub.edu, shasabbir234@gmail.com
ORCID iD: https://orcid.org/0009-0006-8067-564X

Rahman Sharar*
Department of Computer Science, Faculty of Science and Technology, American International University-Bangladesh, Dhaka, Bangladesh
E-mail: 18-38695-3@student.aiub.edu, rahmansharar@gmail.com
ORCID iD: https://orcid.org/0009-0000-6071-0344
*Correspond Author

Md. Ibrahim Bahadur
Department of Computer Science, Faculty of Science and Technology, American International University-Bangladesh, Dhaka, Bangladesh
E-mail: 18-38671-3@student.aiub.edu, ibrahimbahadur574@gmail.com
ORCID iD: https://orcid.org/0009-0000-3297-3825

Abu Sufian
Department of Computer Science, Faculty of Science and Technology, American International University-Bangladesh, Dhaka, Bangladesh
E-mail: 18-38659-3@student.aiub.edu, sufian6543@gmail.com
ORCID iD: https://orcid.org/0009-0000-1198-4533

Rashidul Hasan Nabil
Department of Computer Science, Faculty of Science and Technology, American International University-Bangladesh, Dhaka, Bangladesh
E-mail: rashidul@aiub.edu
ORCID iD: https://orcid.org/0000-0002-8414-6423

Received: 01 March 2023; Revised: 23 April 2023; Accepted: 29 May 2023; Published: 08 August 2023

Abstract: The emergence of chatbots over the last 50 years has been the primary consequence of the need of a virtual aid. Unlike their biological anthropomorphic counterpart in the form of fellow homo sapiens, chatbots have the ability to instantaneously present themselves at the user's need and convenience. Be it for something as benign as feeling the need of a friend to talk to, to a more dire case such as medical assistance, chatbots are unequivocally ubiquitous in their utility. This paper aims to develop one such chatbot that is capable of not only analyzing human text (and speech in the near future), but also refining the ability to assist them medically through the process of accumulating data from relevant datasets. Although Recurrent Neural Networks (RNNs) are often used to develop chatbots, the constant presence of the vanishing gradient issue brought about by backpropagation, coupled with the cumbersome process of sequentially parsing each word individually has led to the increased usage of Transformer Neural Networks (TNNs) instead, which parses entire sentences at once while simultaneously giving context to it via embeddings, leading to increased parallelization. Two variants of the TNN Bidirectional Encoder Representations from Transformers (BERT), namely KeyBERT and BioBERT, are used for tagging the keywords in each sentence and for contextual vectorization into Q/A pairs for matrix multiplication, respectively. A final layer of GPT-2 (Generative Pre-trained Transformer) is applied to fine-tune the results from the BioBERT into a form that is human readable. The outcome of such an attempt could potentially lessen the need for trips to the nearest physician, and the temporal delay and financial resources required to do so.

Index Terms: Medical chatbot, RNN, LSTM, GRU, TNN, KeyBERT, BioBERT, GPT-2.

1. Introduction

In computing terms, a chatbot is a virtual entity that engages in communication with humans primarily via text. As the latter would almost always make use of their natural language to do so, it is the task of the chatbot to convert this into a form that it can interpret, that is, in machine language, and then give a fitting reply to the message in question [1]. This whole procedure is generally known as Natural Language Processing (NLP).

In the dawn of the 21st century, where automation and artificial intelligence has reduced time, effort and monetary wastage to a great degree, it was only a matter of time before chatbots that are capable of assessing people's symptoms and providing medical counsel came to the scene. Now, of course, they would arguably never healthcare professionals, especially not when it comes to physically treating ailments, however, their use is prominent in certain key areas. People may easily and conveniently acquire healthcare information and guidance via medical chatbots. Due to its round-the-clock accessibility, consumers may get help right away without having to make an appointment or wait in line. Chatbots can also assist with preliminary diagnosis and offer suggestions depending on an individual's symptoms. They can provide preliminary evaluations, advise self-care measures, or indicate when professional medical assistance is necessary by posing relevant and essential questions and examining the answers.

Numerous resources of medical information and research are available to medical chatbots. They can inform users on many aspects of health, describe ailments, drugs, and therapies, and encourage pro-active self-care habits. Chatbots may aid with mental health issues by offering emotional support, coping techniques, and recommendations for the right resources or expert assistance when necessary. They provide people with a private, secure setting where they may vent their emotions and look for advice. By assisting with the identification of urgent situations, recommending appropriate levels of treatment, and guiding patients to the best healthcare professionals, they can optimize the use of resources and boost overall effectiveness.

Certain medical chatbots can work with wearable technology or health applications to measure users' vital signs, keep tabs on their chronic diseases, or send them reminders to take their medications as prescribed. This encourages early diagnosis of possible problems and enables people to actively manage their health. In general, chatbots close the information discrepancy among patients and healthcare resources by facilitating rapid access to trustworthy information, assistance, and direction. They improve accessibility and effectiveness in the delivery of healthcare and provide people the power to decide what is best for their health.

However, some key concepts need to be explained before proceeding any further.

1.1. Natural Language Processing

In other words, Natural Language Processing usually deals with analyzing human language to convert it into a form that can be deciphered by an artificial intelligence; in this case, a chatbot. Methods include segmentation of words [2], text mining [3], pre-processing of data (tokenization [4], eliminating non-essential terms [5], stemming), etc. NLP has its uses in fields beyond the realm of chatbots, from data mining to search engine optimization techniques and handling spam messages [6].

1.2. Deep Learning

Deep Learning, a term coined by G.E. Hinton in 2006 [7], is a segment of the much broader domain called Machine Learning that involves the training of neural networks, which are designed and developed into an architecture that mimics the intricately sophisticated webbing of neurons known as the human brain [8]. As the name suggests, it is capable of teaching itself to work as intended, with an emphasis of fine-tuning its accuracy of generating results through rigorous and consecutive cycles of training with the help of existing datasets. The internal architecture consists of multiple layers of neurons working synergistically towards the same ultimate objective, as depicted by Figure 1-1. Common neural networks include CNN (Convolutional Neural Network), RNN (Recurrent Neural Network), TNN (Transformer Neural Network), and DBN (Deep Belief Network); all of these are remarkably adroit at both supervised (labelled data with a designated attribute) and unsupervised (unlabeled data) learning.

Deep Learning exceeds mainstream Machine Learning-based chatbots in several ways.

First off, deep learning algorithms can automatically learn multi-level features, doing away with laborious manual extracting features in machine learning, and outcomes are frequently more accurate than those of conventional techniques due to deep neural networks' potent training and expression abilities. However, because of its robust expression capability, a lot of pointless features will be developed at runtime, necessitating a substantial amount of data pieces for training stage. As can be observed, traditional approaches are much more reliable for analyzing miniscule amounts of data whereas our method is better suited for analyzing big volumes of data.

The focus of conventional machine learning techniques and dictionary creation approaches, on the other hand, is on how to develop a computational formula and what characteristics to extract. Deep learning techniques, on the other hand, concentrate on how to create a network structure that is more effective, as well as how train a network with more precise parameters.

Finally, deep neural networks can autonomously change the weights of model parameters to as nearly reach the intended impact as feasible because of the robust autonomous learning function. Various issues can be addressed utilizing the same model and training approach, but different problems require different network structures and

parameter weights. The overall structure is similar to a function, with input and output correlating one to one. Deep learning may therefore be used in many other domains and has shown positive outcomes.

Aside from its use in building chatbots, deep learning has applications in image processing, pattern recognition, sensors in self-driven vehicles, calamity prediction, etc.

A neural network generally has three key areas:

- The input layer takes input from the source and transfers it into the hidden layer.
- The hidden layer performs the "process" part of the neural network, that is, it takes data from the input layer, processes it however it is supposed to in order to achieve the end result, and passes it to the output layer. The hidden layer is, of course, hidden in the sense that the nodes are kept concealed.
- As the last layer of the network, the output layer merely receives information from the hidden layer and releases it as the result.

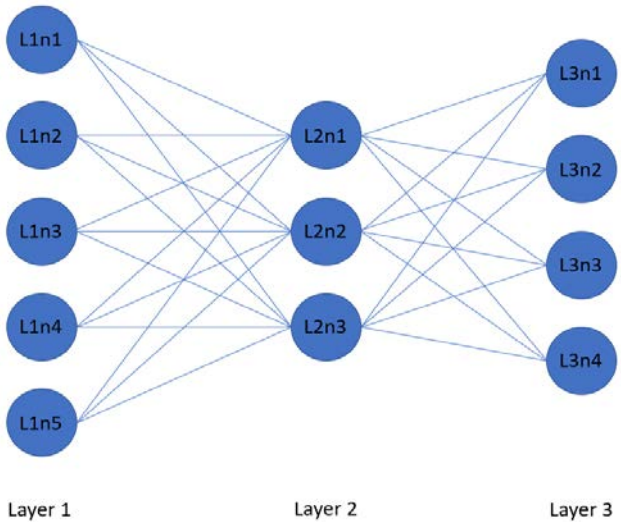

Fig.1. Hidden layer in a neural network

1.3. Miscellaneous Concepts

It is of essence to comprehend the meanings of three distinct keywords before advancing any further:

- Backpropagation - the weights of the network connections are regularly modified in an effort to bring the expected output and real output as close as possible.
- Feedforward Propagation - data moves in the forward direction. Median functions are computed in the hidden layer using the input layer, which in turn is used to determine the result.
- Activation Function - The purpose of an activation function is to add non-linearity to the neural network. Although adding an Activation Function to an already convoluted system may seem cumbersome, it has a high reward: effort ratio. If we were to work with a neural network that does not have an Activation Function, every neuron would be implementing a linear transformation on the fed data with the help of weights and biases. Introducing multiple hidden layers would not change the outcome as they will work in a similar fashion as a whole. The only task this type of network would be able to perform is linear regressions.

This paper is divided into 6 sections. The first section introduces the paper and delivers a background for it, while also providing a headfirst dive into several concepts, the comprehension of which is of paramount importance before proceeding any further. The second section follows it up with a summary of pertinent studies that have been conducted since the 1900s till the present day. Section 3 elucidates the methodologies used by this paper in great detail, that is, the procurement of the datasets and the TNN models used. Section 4 discusses the accuracy of the results obtained by comparing it to those of three other similar neural network models that were used for the same purpose in a 2020 paper, while the fifth section draws conclusions from the entirety of the project, summarizing it and mentioning the limitations experienced and the scope for future research. The sixth and last section merely lists the references used.

2. Related Works

The domain of NLP is primarily driven with the frameworks offered by Deep Neural Networks due to the latter's high computational abilities and abundance of open-source models, which are primarily of two categories- Transformer Neural Networks (TNN) and Recurrent Neural Networks (RNN) [9].

2.1. Recurrent Neural Network (RNN)

RNNs are progressive topologies with the ability to model entities successively. In an RNN, the results of one phase are fed into the next phase's computations. Classic neural networks have inputs and outputs that are autonomous of one another, yet there is a requirement to recall the prior statements in situations where it is necessary to anticipate the following word in a phrase. As a result, RNN was developed, which utilized a hidden unit to resolve this problem. The hidden unit, which retains certain data about a series, is the primary and perhaps the most significant characteristic of RNNs. RNNs are better suited for sequential datasets, such as text.

RNNs employ two distinct activation functions quite frequently, which are Tanh and Sigmoid.

The numbers moving across the network are controlled with the aid of the tanh function. Numbers are all compressed by the tanh function to fall within -1 and 1.

Sigmoid functions can be found in gates. The tanh function is comparable to a sigmoid function. It compresses data across 0 and 1 rather than from -1 and 1. Since every integer multiplied by 0 is 0, numbers vanish or are "forgotten," making it easy to modify or forget data. Any integer multiplied by one has the same result, hence that integer is "preserved" or remains equal. The system learns which information is crucial to maintain and which should be deleted based on importance.

However, RNNs have a major drawback as a result of its short-term memory, which is the vanishing gradient issue, that is, it has problems remembering data from earlier stages as it undergoes additional phases. Given the characteristics of back-propagation, a method used to prepare and tune neural networks, short-term memory and the vanishing gradient arise [10].

Three main processes are involved in RNN. It executes a forward pass first, then predicts. Second, a loss function is used to contrast the forecast to the actual data. The deviation produced by the loss function serves as an estimation of how inadequately the system is functioning. Finally, it does back propagation, which determines the gradients for every network node by using that magnitude of deviation.

The gradient is the quantity utilized to modify the intrinsic weights of the system, thus enabling learning. The modifications increase in size as the gradient does, and conversely. The issue is at this point. Each unit in a layer quantifies its gradient with regard to the consequences of the gradients in the layer preceding it when back propagation is being used. Revisions to the present layer will thus be negligible if those to the ones preceding it were already modest. Since it back propagates downward, gradients rapidly diminish as a result. Given that the intrinsic weights are scarcely being altered by the very modest gradients, the previous layers are unable to learn anything. And that is the issue with the disappearing gradient. In order to solve this issue, RNNs have been modified to its main two derivatives, namely LSTM and GRU [11].

LSTM, or Long Short-Term Memory, is a specialized RNN developed in 1997 owing to the vanishing gradient problem (differential values converge to zero around the origin of the graph) of previous RNNs [12], including Backpropagation Through Time (BPTT) [13, 14] and Real-Time Recurrent Learning (RTRL) [15]. These erroneous backflow issues can be solved via LSTM. Even with distorted, incompressible input sequences, it can learn to bridge temporal spans beyond 1000 steps without losing its short-time lag skills. This is accomplished by an effective gradient-based algorithm for an architecture that enforces constant error flow through the internal states of special units (thus, neither exploding nor vanishing error flow, providing the gradient processing is compressed only at structure points; however, this does not affect long-term error flow) [16]. LSTM usually consists of three gates- forget, input, output, [17].

The forget gate determines what data must be deleted or retained. The sigmoid function processes data from the prior hidden unit as well as data from the present input. There are numbers within 0 and 1. To forget implies inching nearer to 0, while to retain implies getting nearer to 1.

The input gate modifies the cell unit, which is the storage of the system. Initially, a sigmoid function is used to process the present data and the prior hidden unit. The numbers are changed to range from 0 to 1, and that determines whichever data would be altered. 0 denotes unimportant, whereas 1 denotes significant. Additionally, we supply the tanh function with the hidden unit and present data to compress results amongst -1 and 1, which aids in network regulation. The outcome of the sigmoid is then multiplied by the result of the tanh. The data that should be retained from the tanh result will be determined by the sigmoid result.

The subsequent hidden unit is decided by the output gate. Keep in mind that the hidden unit holds data about prior entries. Forecasts are also made using the hidden unit. Initially, a sigmoid function is used to process the present data and the prior hidden unit. The freshly altered cell unit is then relayed to the tanh function. To determine what data the hidden unit will contain, we multiply the tanh value by the sigmoid value. The hidden unit is the product. The new hidden and cell units are then transferred over to the following sampling interval. [18].

While LSTM takes into account the sequence reliance between lexical items to detect components in both long- and short-range forms, Bi-LSTM can carry out both directional scans, and it is possible to navigate both aspects simultaneously in both forward and backward orientations, hence aptly termed Bidirectional LSTM. This specific feature makes it an upgraded version of LSTM [19]. Bi-LSTM is fundamentally an amalgamation of the assets LSTM and Bi-RNN [20], bearing the capability of stockpiling both past and future information, ultimately leading to its superiority over LSTM [21]. Due to this, Bi-LSTM can predict the context of a given word and enable the chatbot to reply appropriately.

Both LSTM and Bi-LSTM are adept at handling relatively substantial volumes of data, evident by their modal accuracies being over 80% in most cases involving chatbots and NLP [22].

Gated Recurrent Unit (GRU) was developed in 2014 [23], making it another version of RNN akin to LSTM but is easier to process and deploy. Furthermore, it constitutes of two gates rather than three, which are reset and update. The reset gate works in a similar fashion to that of LSTM's forget and input gates, while the update gate determines to what extent a neuron would update its existing data [24]. Like LSTM, GRU also tackles the vanishing gradient problem with an identical mechanism.

Bidirectional-GRU is to GRU as Bidirectional-LSTM is to LSTM, in the sense that both the bidirectional neural networks are enhanced versions of their respective base models, capable of traversing either direction and gathering both past and future data for finer context prediction [25].

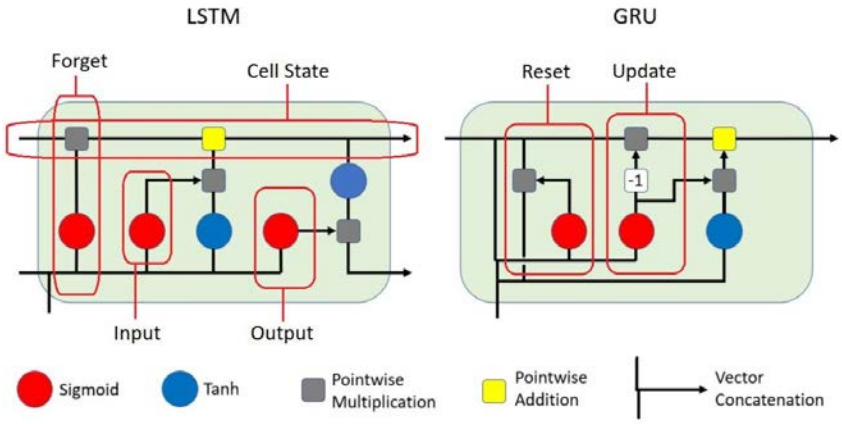

Fig.2. LSTM and GRU architecture

According to conventional research, both LSTM and GRU may maintain significant characteristics via a variety of gates, preventing the loss of significant distinct characteristics over protracted propagation. GRU's design is less complex and can save a significant amount of time without affecting quality since it has one fewer gate than LSTM, which decreases matrix multiplication. On the contrary, practical evidence suggests that this GRU edge only applies when dealing with lengthy texts and limited datasets. In some cases, the performance reduction of GRU is more severe than that of LSTM. Since computational complexity is no more a limiting factor, LSTM is appropriate in such situations [26].

2.2. Transformer Neural Network (TNN)

TNN, modeled by a Google Brain group in 2017 [27], is a DNN that uses the self-attention process and weights the importance of each component of the input variables using their derivatives. It is largely utilized in the disciplines of NLP, and computer vision and pattern recognition (CVPR) [28].

The TNN architecture has two distinct parts- an encoder and a decoder [29, 30, 31]. The encoder deals with the input, while the more nuancedly designed decoder handles the output.

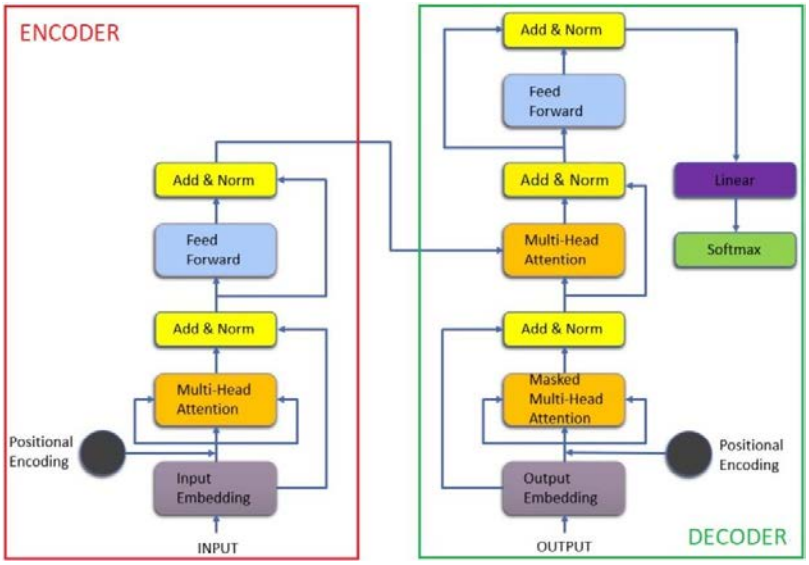

Fig.3. TNN architecture

In the encoder, the primary task of the Input Embedding section is NLP, that is, converting natural language into vectors for the system to comprehend; whereas the Positional Encoding part appreciates the context of the entire sentence by judging the location of the words.

Next comes the concept of Self-Attention. It emphasizes the notion of a word's significance in relation to certain other words used throughout the sentence. It appears as a vector of attention. An attention vector that reflects the contextual relationship among words in a phrase may be constructed for each word. The main issue it has is that it gives each word a far more weight than it deserves in the phrase, when we are more interested in how those words interact with one another. Therefore, to determine the ultimate attention vector for each word, we calculate multiple attention vectors for each word and then take an arithmetic mean. It is known as the Multi-Head Attention component since we are employing various attention vectors.

All attention vectors are subjected to a rather straightforward Feed-Forward network, and its primary function is to change the attention vectors to a format that the subsequent encoder or decoder phase will tolerate. Attention vectors are collected by the Feed Forward network individually. The finest part is that, in contrast to RNN, all of the attention vectors are autonomous. Therefore, parallelization is possible here, and that is what really changes everything. We can now input every one of the words concurrently into the encoder unit to obtain the collection of encoded vectors with each word.

The Decoder, as previously mentioned, is responsible for the output data. Akin to the Encoder, it has (Output) Embedding, Positional Encoding and Multi-Head Attention chambers that work with similar mechanisms; however, the aforementioned third unit has the word ‘Masked’ attached as a prefix. The innate nature of its learning ability resides in the fact that it first converts the vectors for use by the Decoder with the help of previous data, then it compares this result to the dataset that had been fed to it originally. The matrix result will be updated after contrasting the two. Over numerous repetitions, it would learn in this manner. Consequently, when conducting the matrix operation in tandem, we ensure that the matrix would conceal words that arrive later by turning them into 0s, preventing the attention network from using these; hence the ‘Mask’ prefix.

The next Multi-Head Attention compartment is then given the output attention vectors originating from the preceding stage and the vectors of the Encoder (at this point, the Encoder's findings likewise become apparent. The output from the Encoder is plainly evident in the graphic as well, arriving here.) It is termed an encoder-decoder attention unit for this reason. Attention vectors for each word in sentences are the outcome of this block. Each vector illustrates how a word in either set relates to others.

Now that each attention vector has been sent through a feed-forward unit, the output vectors will be transformed into a form that any decoder or linear layer—another feed-forward layer—can readily accept. It goes via a Softmax Layer, which converts the input data into a probability distribution that can be understood by humans. A Softmax function serves as an amalgamation of several Sigmoid Functions by evaluating empirical probability of every stage, and is used for the outermost layer in a network that works with multinomial classification.

For each word, the Transformer first creates embeddings, or initial representations. Then, by leveraging self-attention, it compiles data from every other word, creating a new representation for each word that is influenced by the full context. Then, for each word, this phase is performed several times simultaneously to produce new representations one after the other. Similar to the encoder, the decoder creates words individually, spanning left to right. It pays attention to both the final representations produced by the encoder and the other words that were created earlier.

As stated earlier, RNN suffers from the vanishing gradient issue, which is sorely lacking in TNN. TNN also parses entire sentences at once while simultaneously giving context to it via embeddings, unlike RNN and its derivatives (LSTM/GRU), which sequentially process one word at a time. This allows for parallelization among TNN models [32-34].

One of the most commonly used TNN derivatives is Bidirectional Encoder Representations from Transformers (BERT), which was introduced in 2018 by Google [31], and is primarily used for NLP pre-processing. Dissimilar to other models that perform related tasks [11, 35], the purpose of BERT is to prepare bidirectional deep representations from unlabeled data in every layer by combining the context of both directions (left and right). The outcome is the opportunity to adjust the pre-trained BERT model by adding just one output layer to develop cutting-edge models for a variety of activities, including answering questions without any meaningful task-specific language architectural alterations. These are the result of the bidirectional self-attentions heads (12 in $BERT_{BASE}$ and 16 in $BERT_{LARGE}$) which were both fed unlabeled data for the pre-training stage from BooksCorpus [36] and the English version of Wikipedia, having 800 million and 2500 million words, respectively. BERT has consistently scored between 80-90% accuracy with versatile NLU operations, such as GLUE, SQuAD, SWAG, and Sentiment Analysis [37]. Two of BERT’s variants are discussed below.

BioBERT is used for medical text analysis [38] in order to overcome the more generalized approach of text mining offered by BERT, the latter of which is difficult to assess and may lead to ambiguities. The pre-training dataset is extracted from PubMed and PMC articles, which are two major medical blogs, and finetuned using three more medical datasets, namely NER, RE, and QA.

The sole purpose of KeyBERT is to extract keywords from phrases along with taking their semantics into account [39]. To start, texts are first transformed into BERT embeddings via KeyBERT. Then, word n-gram embeddings with predetermined lengths using BERT are produced. The keyphrases that best characterize the full document are then extracted using cosine commonalities among text and keyphrase embeddings.

The first version of Generative Pre-trained Transformer (GPT) was developed in 2018 [40] It demonstrated how pre-training on a heterogeneous dataset with extensive stretches of continuous text allows a linguistic generative model to gain global information and comprehend long-range relationships.

GPT-2 is an enhanced edition of the original GPT, introduced in 2019 [41], was not made fully accessible by the general population due to worries about potential abuse, including tools for creating fake news. The training dataset used was WebText, which consists of little more than 8 million items totaling 40 Gigabytes of text from hyperlinks posted in Reddit contributions that have received at least three upvotes. By adopting byte pair encoding, it eliminates several problems that might arise when encoding vocabulary using word tokens. By encoding separate characters and multi-character tokens, this enables the representation of just about any string of characters [42].

GPT-3 is an extended version of GPT-2, released in 2020 [43]. The whole iteration of GPT-3 has 175 billion parameters, which is hundred times more than the complete edition of GPT-2's 1.5 billion parameters; the accuracy and efficiency have thus been enhanced several folds. As of 2020, GPT3 has been acquired by Microsoft and is no longer available to the general public, other than the trial version offered for two months.

2.3. Existing Chatbots

A. Domain-based

There are two types of chatbots when it comes to knowledge are: open domain and closed domain, and the data which the chatbot is addressing is referred to as the domain. Open domain includes both fundamental human understanding and broad information, including contemporary events. The chatbot software would be categorized as a closed domain chatbot if its expertise is specialized in a certain field, such as healthcare assistance, football, etc. [9]. Studies have showed that closed domain chatbots are simpler to create and are now yielding positive outcomes [44, 45, 46], while a lot of misleading findings were produced by open domain chatbots, which are currently difficult to construct [47, 48].

Examples of closed domain chatbots include

- DeepProbe, which employs the criterion of "bad, fair, good, excellent," while "fair, good, excellent" is seen as favorable [49]
- AliME, which utilizes business analysts as its human judges, using a scale of 0, 1, and 2, with 0 being inappropriate, 1 being a remark that is only appropriate in specific situations, and 2 being a reasonably appropriate answer [50]
- SuperAgent, which has no metric for sentient assessment in its review process [51]. Whereas open domain chatbots include
- MILLABOT, which uses a scale of 1 to 5, where 1 is unsuitable or illogical, 3 is a reasonable answer, and 5 is very suitable [52]
- RubyStar, whose performance metrics are similar to that of MILLABOT [53]

B. Objective-based

Objective-based chatbots are categorized according to the main objective they seek to accomplish. They are created with a specific objective in mind and are arranged for quick exchanges of data with the users [54].

Some examples of objective-based chatbots:

- Casper (Insomnobot-3000) is the lone chatbot in the world that is open for a conversation from 12 pm to 4 am, right when a user may have trouble falling asleep. Using the chatbot, they can talk about practically any topic because it was designed to mimic interpersonal interactions [55].
- One Remission is a bot created by a New York-based startup with the intention of providing the data required by individuals active in the cancer battle. In order to reduce their dependence on physicians, this supportive bot gives customers a detailed range of post-cancer routines, meals, and regular workouts. For instance, users may look up the advantages and disadvantages of a particular dietary item in relation to cancer, with the option of contacting a professional oncologist at any time. The bot functions as a psychological and physical helper, giving individuals the freedom to express any good or skewed opinions. Users can speak with the bot orally or through texts, while they will receive correct answers to their queries in response. One Remission is available to offer the best guidance available whether they require assistance with their meals, workout routines, or circadian rhythms [56].
- Babylon Health, developed in 2013, provides healthcare background and standard health knowledge-based consultations, in addition to the option for a user to speak with a medical professional through video conference if necessary. In the initial scenario, people tell the bot about their sickness' indicators, and the bot compares them to a library of ailments before understanding what they're saying and suggesting the best course of action. The personal engagement with a qualified practitioner, who attentively observes and analyzes, to evaluate the user and then writes an adequate treatment or refers them to a consultant, if necessary, significantly exceeds the regulatory norms of a bot in the later scenario [56].

C. Older Examples

Objective ELIZA is among the first popular bots in history. It was created in 1966 at the MIT Lab [57] with the goal of demonstrating human language communication between people and machines in order to deliver psychiatric treatment, focusing on encouraging the client to communicate more. ELIZA's replies take the form of open-ended inquiries that are supposed to pique the client's interest and encourage more discussion. It matches keywords from a library of frameworks with semantics to provide answers to the client's inquiries using regulatory approaches and a program. The algorithm picks the relevant replies after identifying the proper framework. In the event that there are several frameworks, one is chosen at random, which the algorithm subjects to a series of perspectives to further effectively prepare the message for a reply. ELIZA is able to persuade certain individuals to support the client's therapy. However, Eliza is unable to offer something like to counseling with a human counselor. ELIZA's flaw is its inability to continue conversations. Additionally, ELIZA is unable to pick up new behavioral traits, find background for voice or phrases via conversation and logic-based deductive skills [58].

ALICE (Artificial Linguistic Internet Computer Entity), released in 1995, used ELIZA as its base [59]. ALICE remains solely reliant on pattern recognition and a depth-first approach to human input, however. It is an XML variant in which the criteria for queries and responses are encoded. The replies are generated using a collection of artificial intelligence markup language (AIML) elements based on the conversation record and human speech [43]. The human phrase is initially inputted into AIML and placed in a class. Each class consisted of a reply pattern and a set of circumstances that provide the background, or content, to the pattern. After that, the system preprocesses it and compares it to decision tree nodes. The bot will respond or take measures when text entry matches. The recursion procedures used by the AIML patterns to duplicate the user's supplied statement result in sometimes meaningless answers. In order to assess if the answer generates an accurate or relevant result, string-based restrictions are necessary. The disadvantage of ALICE is the character modeling used to describe the bot's behavior, including its features, perspectives, temperament, and bodily conditions [60]. The AIML ought to have character components, based on the human controller. Nevertheless, it is far from a simple process. Additionally, ALICE lacks the ability to rationalize or produce replies that are sentient and therefore cannot provide suitable replies. A sophisticated chatbot needs a lot of classifications, which might make the program impractical, challenging to manage, or cumbersome. In order to speak coherently, ALICE lacks cognitive elements like NLU, sentiment analysis, and grammatical analysis. Additionally, ALICE frequently regurgitates the same responses if the exact data is entered repeatedly.

D. Newer Examples

Google created Dialogflow, also called as Api.ai, which is a component of Google Cloud Platform [61]. It enables programmers to offer their consumers voice and text conversations with UI that are supported by ML and NLP. This enables them to concentrate on other crucial aspects of developing the program instead of outlining intricate language patterns. Dialogflow understands the background and purpose of human input. After that, employ objects to retrieve pertinent information from them via matching input from the user to certain purposes. Lastly, it permits replies to be provided by the chat agent. Dialogflow's shortcomings include a lack of a mobile app, a static UI, and inadequate paperwork.

For creating speech and text-based interactive UI for apps, Amazon built Lex, an AWS tool [62]. In order to create very appealing UX with genuine-sounding communication, it offers deep learning capabilities and versatility of natural language understanding (NLU) and automated speech recognition (ASR). Because of the integration between Amazon Lex and AWS Lambda, users may quickly launch operations to carry out back-end business rules for data modification and extraction. The limitation of Amazon Lex is that it only supports English at the moment. Lex must adhere to a strict procedure for web application. Additionally, it is difficult to prepare the dataset, and connecting the statements to the objects is fairly crucial.

3. Methods

3.1. Workflow

The methodology is divided into three parts: Dataset Collection and Preprocessing, Finetuning the BioBERT Model, and Finetuning the GPT-2 Model.

Q/A based datasets are collected and separated into questions and answers, which are fed into the KeyBERT the extract keywords and attach tags before storing them.

Q/A pairs are vectorized and passed into the BioBERT model, which again separates and further vectorizes them into numbers, before preforming dot product similarity matching between them.

GPT-2 converts vectors into text as the final output.

New questions are concatenated with existing similar Q/A pairs in the BioBERT, and the entire aforementioned process of GPT-2 repeats.

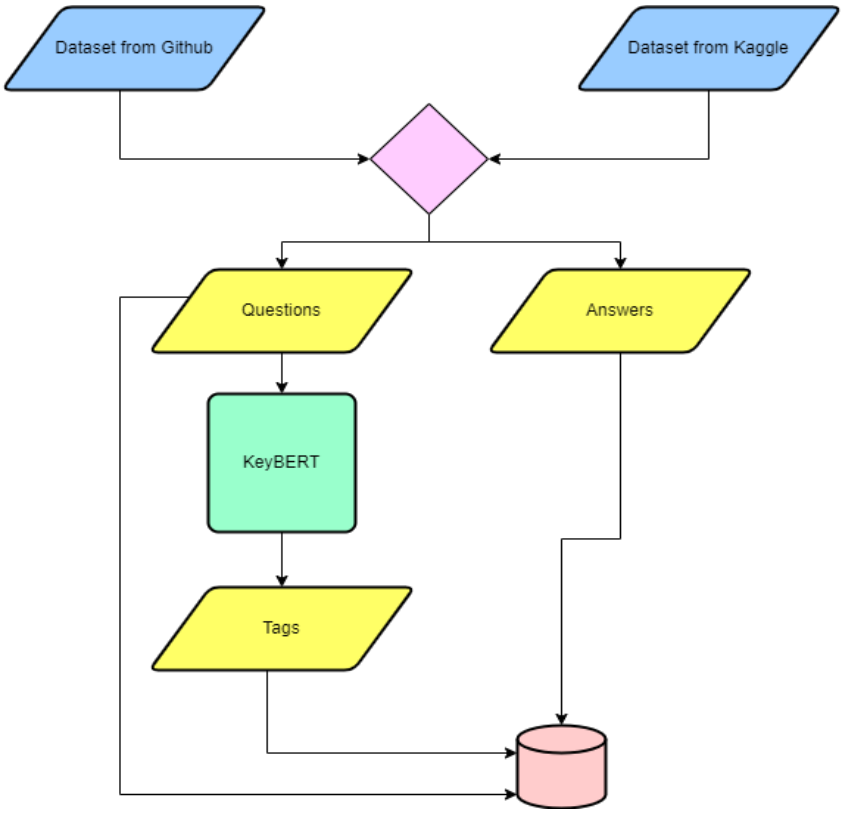

Fig.4. Part 1 – Dataset Preprocessing

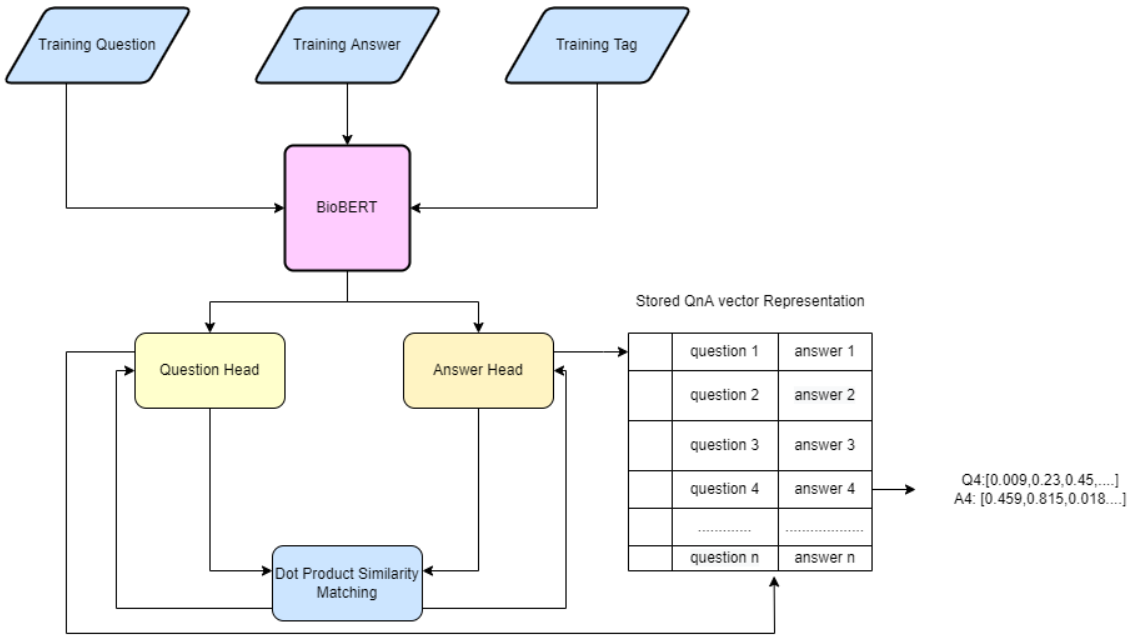

Fig.5. Part 2 – Finetuning BioBERT

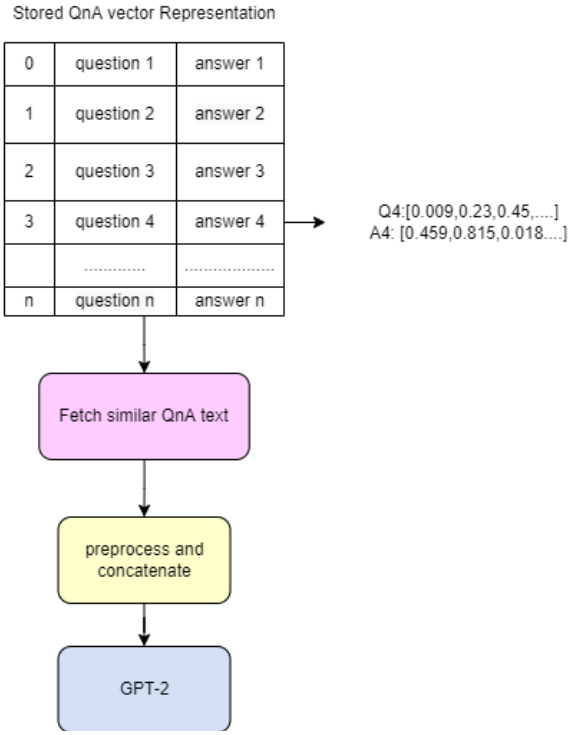

Fig.6. Part 3 – Finetuning GPT-2

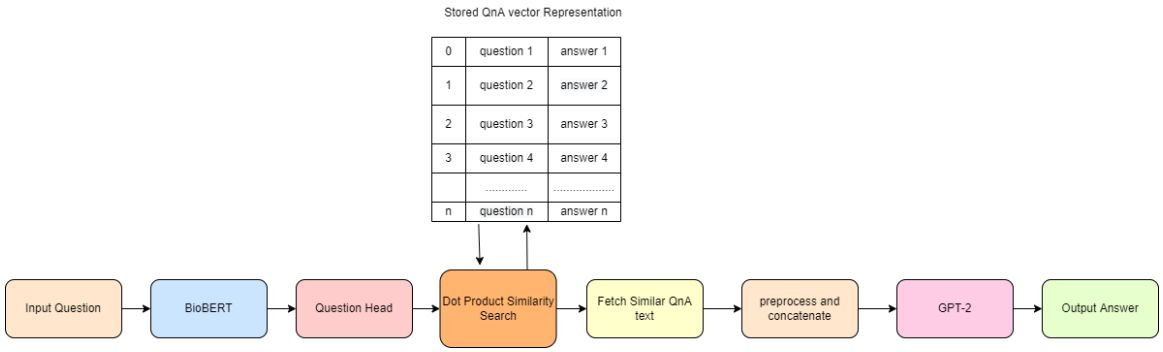

Fig.7. Deployment

3.2. Dataset Preparation

Due to the versatile nature of a chatbot's functionalities, the dataset(s) used for this project originate from four different domains:

- Mental Health (depression, anxiety, panic attacks, suicidal tendencies)
- Virtual Friend (chatbot) that is available for conversation 24/7
- Emergency Services (911) - police, fire service, ambulance
- Customer Service systems that can aid in alleviating the user's issues (e.g., Sheba)

The four main sources of these datasets were Kaggle, GitHub, UCI Machine Learning Repository, and Google Forms with approximately 500 responders. All these datasets exist as either readymade Q/A pairs (which are relatively easy to deal with), or as a complex structure which had to be extracted via additional programming and converted into Q/A pairs. In both cases, the file format is either CSV or JSON, which was handled by the pandas library of the Python programming language. The dataset had 50000 instances, of which 10000 were used to train and 40000 were used to test the model.

An important step is to remove stop words before training the BioBERT model. Stop words are non-essential (in the context of our system) and language-specific words which bear no information. Examples include conjunctions, articles, pronouns, prepositions, interjections, and words that have been abbreviated in a informal manner (such as "they will" can be abbreviated into "they'll"). The English language has about 500 stop words.

3.3. Neural Network Models Used

As stated earlier, TNN models were used due to their lack of the vanishing gradient issue in comparison with traditional RNN models. Furthermore, enhanced versions of RNN (LSTM and GRU), although do not face this hurdle either, they can only process individual words sequentially; whereas TNN models process entire sentences at once while simultaneously giving context to it via embedding. This leads to increased efficiency, accuracy and the scope for parallelization.

Specifically, three TNN models were used- KeyBERT, BioBERT, and GPT-2. Both BERT models contain bidirectional heads that can traverse left and right to extract past data and future context, respectively. KeyBERT was used for keyword extraction, BioBERT serves as the central and main component of the model, and GPT-2 is the final layer which finetunes the entire chatbot.

3.4. Methodology

A. KeyBERT

The keywords in each sentence of the dataset are given a specific tag by the KeyBERT using a loop. Each keyword is made up of two words, and each question or answer has two to three keywords on average. The questions and answers are concatenated to decipher their context before the KeyBERT can be extracted from them. These Q/A pairs, along with their keywords, are saved in a storage system.

The KeyBERT model did not undergo pre-training; it was used out-of-the-box.

B. BioBERT

The Q/A pairs that were saved earlier are converted from text into vectors using an open-source algorithm called Word2Vec [43], and then fed into the BioBERT. These vectors undergo an embedding procedure in the BioBERT that is similar to the previous Word2Vec algorithm, but the vectorization method here is contextual, so homonyms (words that have the same spelling and/or pronunciation but different meanings and origins) have different vector values, which are represented as numbers. The vectors stemming from questions are directed into a unit called the Question Head, and the ones stemming from answer are directed into another unit called the Answer Head; essentially ending up as matrices on either side. These are subjected to matrix multiplication in the form of dot product calculation, and the result is yet another Q/A combination.

	short_question	short_answer	tags	label
0	can an antibiotic through an iv give you a ras...	yes it can even after you have finished the pr...	['rash' 'antibiotic']	1.0
1	can you test positive from having the hep b va...	test positive for what if you had a hep b vacc...	['hepatitis b']	1.0
2	what are the dietary restrictions for celiac d...	omitting gluten from the diet is the key to co...	['celiac disease']	1.0
3	can i transmit genital warts seventeen years a...	your symptoms refers to either a kidney infect...	['wart']	-1.0
4	is all vitamin d the same	when you check in tell them that your doctor t...	['vitamin d']	-1.0

Fig.8. Training of BioBERT

The BioBERT model was trained using 10000 instances of data, and in 5 epochs.

C. GPT-2

The Q/A vector representation from the previous stage are fed into the pre-trained GPT-2 to decode the dataset into a form that can be interpreted by humans. In short, BioBERT acts as the encoder and GPT-2 acts as the decoder of the entire system.

	question	answer	Q_FFNN_embeds	A_FFNN_embeds
0	can an antibiotic through an iv give you a ras...	yes it can even after you have finished the pr...	[-0.0251857045366356, 0.04886164566957911, 0.0...	[-0.018387219490760776, 0.06435742783851145, 0...
1	can you test positive from having the hep b va...	test positive for what if you had a hep b vacc...	[-0.03262645238537212, -0.02002499675267078, 0...	[-0.027307724928990213, 0.018306292806955527, 0...
2	what are the dietary restrictions for celiac d...	omitting gluten from the diet is the key to co...	[0.05482535632046998, -0.027704004034438897, -...	[0.05141538089539002, 0.0023206423138006794, 0...

Fig.9. Training of GPT-2

The GPT-2 model was trained using 10000 instances of data, and in two stages of 10 epochs each.

D. Deployment

The pre-trained model is put to use by first inputting questions into the system after removing the stop words in them. But since Machine Learning models cannot interpret natural language directly, so these questions need to be vectorized into a numeric representation, which is done using BioBERT's tokenizers. These are compared against the

existing Question Head to find similar questions and discern their individual context. A dot product calculation is performed between the new questions (together with the old ones) and answers that were paired against similar questions in the training dataset; these are then concatenated into a new set of Q/A pair embedding. These are finally passed to the GPT-2 model that converts them back into a textual form.

The entire system has been deployed on an internal IP using FastAPI, which is a Python-based web framework, and ngrok, which is a cross-platform application that connects local servers to the internet. The UI is developed using React, which is a JavaScript library.

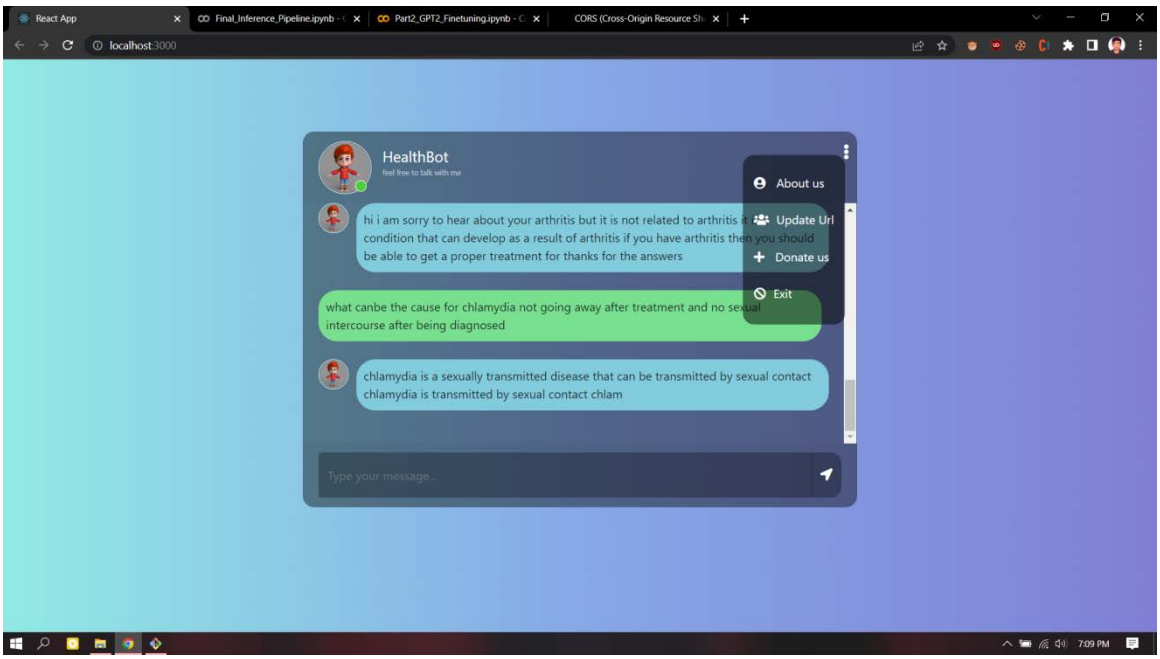

Fig.10. System UI

As the screenshot above depicts, users can read up on the system in detail and the development team in the “About us” section, may make charitable contributions using the “Donate us” option, and exit the system using the “Exit” button. The development team may change the URL using the “Update URL” settings, which cannot be accessed by mainstream users for obvious reasons.

4. Results and Discussion

The deployed BioBERT model achieved an accuracy of 89.64% after 5 epochs.

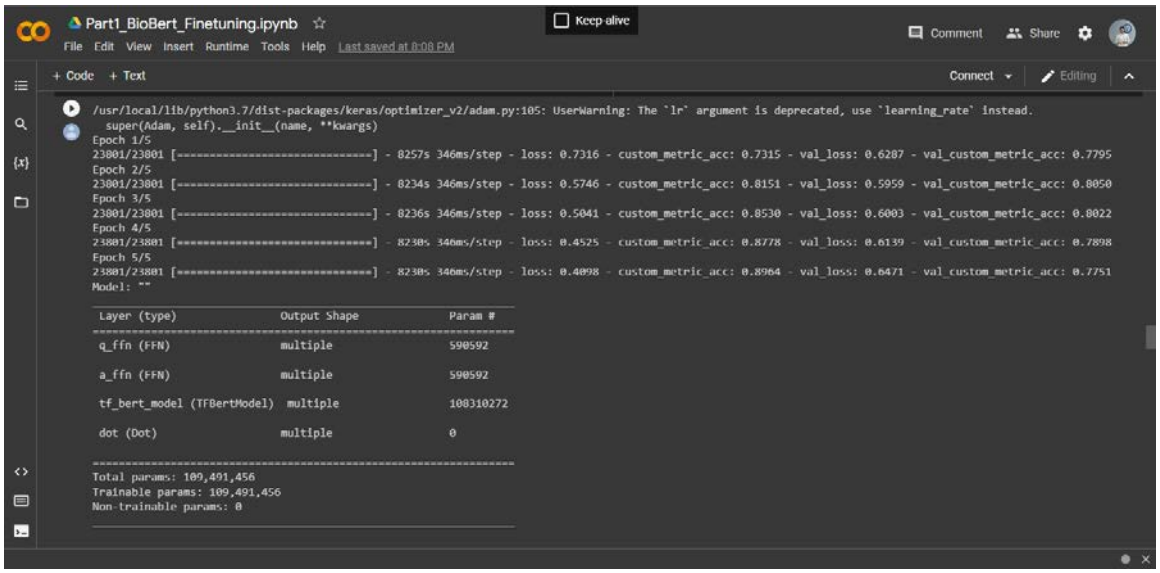

Fig.11. Achieved accuracies of the BioBERT model

A paper published in 2020 makes use of a hybrid model-based chatbot to compare the accuracy of three neural networks, namely Manhattan LSTM (MaLSTM), Hierarchical Bi-LSTM Attention Model (HBAM), and the basic BERT model [48]. Medical datasets from Quora, WebMD, eHealthForum, and QuestionDoctors were used, consisting of 20000 train and 50000 test instances in the form of Q/A pairs.

Since this paper has a domain, methodology, dataset size and a neural network very similar to that of BioBERT, the two can be subjected to a comparative analysis of their accuracies.

Table 1. Neural Network Accuracy Comparison

Neural Network Model Used	Average Evaluation Accuracy
BioBERT	89.6%
BERT	78.2%
MaLSTM	78.4%
HBAM	81.2%

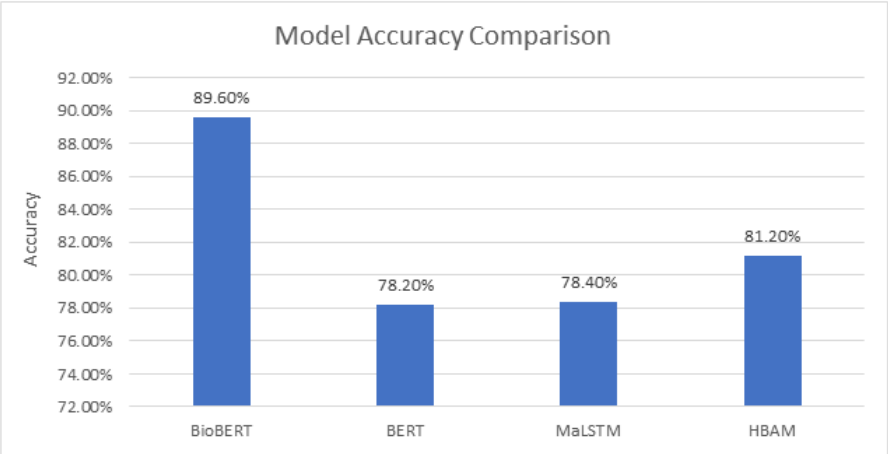

Fig.12. Bar Graph showing Neural Network Accuracy Comparison

As can be seen above, BioBERT outperforms the rest by a significant margin, boasting an accuracy of 89.6%, while the rest experienced accuracies of below 80% with the exception of HBAM, which scored 81.2%. Both MaLSTM and HBAM were RNNs, and due to their lack of parallelization, where they would parse each word sequentially, it was expected that they would score lower than any modified TNN, especially when compared against the BioBERT model, which not only has undergone a keyword tagging phase previously in the KeyBERT layer, but has had its results fine-tuned by the GPT-2 layer as well. Intuitively speaking, it may come off as a shock that the standalone BERT model would score the lowest among all four, even though it is a TNN model that was "competing" against two RNNs, namely MALSTM and HBAM. However, it should be noted that the HBAM model was equipped with an attention layer for keyword identification. As for the MaLSTM model, it is plausible that the standalone BERT model's generalized embedding and 12 layers of TNNs caused a certain degree of overestimation.

5. Conclusions

With artificial intelligence shaping every aspect of our lives, utilizing chatbots for medical purposes is a no-brainer. This paper aimed to develop one such chatbot that is capable of not only analyzing human text (and speech in the near future), but also refining the ability to assist them medically through the process of accumulating data from relevant datasets, using Transformer Neural Networks (TNNs), which parses entire sentences at once while simultaneously giving context to it via embedding, leading to increased parallelization. Two variants of the TNN Bidirectional Encoder Representations from Transformers (BERT), namely KeyBERT and BioBERT, are used for tagging the keywords in each sentence and for contextual vectorization into Q/A pairs for matrix multiplication, respectively. A final layer of GPT-2 (Generative Pre-trained Transformer) is applied to fine-tune the results from the BioBERT into a form that is human readable. When compared against three other contemporary neural network models that used similar datasets and were built for the same purpose, BioBERT outperformed the rest by a significant margin, boasting an accuracy of 89.6%, while the one that came closest scored 81.2%. Although this paper achieved what it had set out to do, it was bound by a few limitations, and certain areas could be improved upon in the future.

5.1. Limitations

The dataset used was relatively small, with only 50000 instances. A larger dataset with a greater number of epochs could potentially boost the accuracy to over 90%.

Furthermore, the hardware used had only 8 GB HDD RAM, Intel Core i5 8th Gen processors and Nvidia GeForce MX150 2 GB GPU. Since accuracy is dependent on processing power, better configurations could have affected the results in a positive manner.

5.2. Future Work

$BioBERT_{BASE}$ was used as the main model, which has only 12 bidirectional self-attentions heads. Using $BioBERT_{LARGE}$, consisting of 24 bidirectional self-attentions heads, would likely be more efficient and produce better results. The MediBERT can also be made available to the general public by deploying it on a domain that is accessible worldwide.

The bot can only process text as of now; with the proper resources and a longer development time, it may be possible to integrate voice command and text-to-speech (and vice versa) capabilities. Raspberry Pi may be used as the computer, along with a speaker and a microphone, to emulate the contemporary virtual assistants present in the form of Amazon's Alexa and Apple's Siri; except this would be solely used for medical aid, and due to the specialization, it could potentially outperform both virtual assistants in this domain.

References

[1] Ayanouz, S., Abdelhakim, B. and Benhmed, M., 2020. A Smart Chatbot Architecture based NLP and Machine Learning for Health Care Assistance. *Proceedings of the 3rd International Conference on Networking, Information Systems & Security*, doi: https://doi.org/10.1145/3386723.3387897

[2] A. Mullen, L., Benoit, K., Keyes, O., Selivanov, D., & Arnold, J. (2018). Fast, Consistent Tokenization of Natural Language Text. *Journal of Open Source Software*, *3*(23), 655. https://doi.org/10.21105/joss.00655

[3] Kumar, L., & Bhatia, P. K. (2013). TEXT MINING: CONCEPTS, PROCESS AND APPLICATIONS. *Journal of Global Research in Computer Sciences*, *4*(3), 36–39. https://www.rroij.com/open-access/text-mining-concepts-process-and-applications.php?aid=38178

[4] Ferilli, S., Esposito, F., & Grieco, D. (2014). Automatic Learning of Linguistic Resources for Stopword Removal and Stemming from Text. *Procedia Computer Science*, 38, 116–123. https://doi.org/10.1016/j.procs.2014.10.019

[5] du Buf, J., Kardan, M., & Spann, M. (1990). Texture feature performance for image segmentation. *Pattern Recognition*, *23*(3–4), 291–309. https://doi.org/10.1016/0031-3203 (90)90017-f

[6] Fattahi, J., & Mejri, M. (2021). SpaML: a Bimodal Ensemble Learning Spam Detector based on NLP Techniques. *2021 IEEE 5th International Conference on Cryptography, Security and Privacy (CSP)*. https://doi.org/10.1109/csp51677.2021.9357595

[7] Hinton, G. E., Osindero, S., & Teh, Y. W. (2006). A Fast Learning Algorithm for Deep Belief Nets. *Neural Computation*, *18*(7), 1527–1554. https://doi.org/10.1162/neco.2006.18.7.1527

[8] Williams, R. J., & Zipser, D. (1995). Gradient-based learning algorithms for recurrent networks and their computational complexity. *L. Erlbaum Associates Inc. EBooks*, 433–486. https://dl.acm.org/citation.cfm?id=201801

[9] Kim, Y., Denton, C., Hoang, L., & Rush, A. M. (2017). Structured Attention Networks. *ArXiv: Computation and Language*. https://arxiv.org/pdf/1702.00887.pdf

[10] Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep Contextualized Word Representations. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. https://doi.org/10.18653/v1/n18-1202

[11] Hu, Y., Huber, A. E. G., Anumula, J., & Liu, S. (2018). Overcoming the vanishing gradient problem in plain recurrent networks. *Cornell University - ArXiv*. https://doi.org/10.48550/arxiv.1801.06105

[12] Werbos, P. J. (1988). Generalization of backpropagation with application to a recurrent gas market model. *Neural Networks*, *1*(4), 339–356. https://doi.org/10.1016/0893-6080(88)90007-x

[13] Robinson, A. J. & Fallside, F. (1987). *The Utility Driven Dynamic Error Propagation Network* (CUED/F-INFENG/TR.1). Engineering Department, Cambridge University.

[14] Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, *9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

[15] Yu, Y., Si, X., Hu, C., & Zhang, J. (2019). A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures. *Neural Computation*, *31*(7), 1235–1270. https://doi.org/10.1162/neco_a_01199

[16] Gers, F. A., Schmidhuber, J., & Cummins, F. (2000). Learning to Forget: Continual Prediction with LSTM. *Neural Computation*, *12*(10), 2451–2471. https://doi.org/10.1162/089976600300015015

[17] Li, W., Qi, F., Tang, M., & Yu, Z. (2020). Bidirectional LSTM with self-attention mechanism and multi-channel features for sentiment classification. *Neurocomputing*, *387*, 63–77. https://doi.org/10.1016/j.neucom.2020.01.006

[18] Shao, D., Zheng, N., Yang, Z., Chen, Z., Xiang, Y., Xian, Y., & Yu, Z. (2019). Domain-Specific Chinese Word Segmentation Based on Bi-Directional Long-Short Term Memory Model. *IEEE Access*, *7*, 12993–13002. https://doi.org/10.1109/access.2019.2892836

[19] Attri, I., & Dutta, D. M. (2019). Bi-Lingual (English, Punjabi) Sarcastic Sentiment Analysis by using Classification Methods. *International Journal of Innovative Technology and Exploring Engineering*, *8*(9), 1383–1388. https://doi.org/10.35940/ijitee.i8053.078919

[20] Ouerhani, N., Maalel, A., Ghézala, H. B., & Chouri, S. (2020). Smart Ubiquitous Chatbot for COVID-19 Assistance with Deep learning Sentiment Analysis Model during and after quarantine. *Research Square*. https://doi.org/10.21203/rs.3.rs-33343/v1

[21] Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014b). Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. https://doi.org/10.3115/v1/d14-1179

[22] Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *ArXiv: Neural and Evolutionary Computing.* https://arxiv.org/pdf/1412.3555
[23] Zhao, R., Wang, D., Yan, R., Mao, K., Shen, F., & Wang, J. (2018). Machine HealthMonitoring Using Local Feature-Based Gated Recurrent Unit Networks. *IEEE Transactionson Industrial Electronics*, *65*(2), 1539–1548. https://doi.org/10.1109/tie.2017.2733438
[24] Yang, S., Yu, X., & Zhou, Y. (2020). LSTM and GRU Neural Network Performance Comparison Study: Taking Yelp Review Dataset as an Example. *2020 International Workshop on Electronic Communication and Artificial Intelligence (IWECAI)*. https://doi.org/10.1109/iwecai50956.2020.00027
[25] Arai, K., Bhatia, R., & Kapoor, S. (Eds.). (2019). Proceedings of the Future Technologies Conference (FTC) 2018. *Advances in Intelligent Systems and Computing*. https://doi.org/10.1007/978-3-030-02686-8
[26] Cui, L., Huang, S., Wei, F., Tan, C., Duan, C., & Zhou, M. (2017). SuperAgent: A Customer Service Chatbot for E-commerce Websites. *Proceedings of ACL 2017, System Demonstrations*. https://doi.org/10.18653/v1/p17-4017
[27] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is All you Need. *Neural Information Processing Systems*, *30*, 5998–6008. https://arxiv.org/pdf/1706.03762v5
[28] Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., & Zagoruyko, S. (2020). End-to-End Object Detection with Transformers. *Computer Vision – ECCV 2020*, 213–229. https://doi.org/10.1007/978-3-030-58452-8_13
[29] Zeyer, A., Bahar, P., Irie, K., Schluter, R., & Ney, H. (2019). A Comparison of Transformer and LSTM Encoder Decoder Models for ASR. *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. https://doi.org/10.1109/asru46091.2019.9004025
[30] Bahdanau, D., Cho, K., & Bengio, Y. (2015). Neural Machine Translation by Jointly Learning to Align and Translate. *International Conference on Learning Representations*. https://arxiv.org/pdf/1409.0473
[31] Sutskever, I., Vinyals, O., & V. Le, Q. (2014). Sequence to Sequence Learning with Neural Networks. *Cornell University - ArXiv*. https://doi.org/10.48550/arXiv.1409.3215
[32] Kaiser, U., & Sutskever, I. (2016). Neural GPUs Learn Algorithms. *International Conference on Learning Representations*. https://arxiv.org/pdf/1511.08228
[33] Kalchbrenner, N., Espeholt, L., Simonyan, K., Van Den Oord, A., Graves, A., & Kavukcuoglu, K. (2016). Neural Machine Translation in Linear Time. *ArXiv: Computation and Language*. https://arxiv.org/pdf/1610.10099
[34] Devlin, J., Chang, M., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *ArXiv: Computation and Language*. https://arxiv.org/pdf/1810.04805v2
[35] Zhu, Y., Kiros, R., Zemel, R., Salakhutdinov, R., Urtasun, R., Torralba, A., & Fidler, S. (2015). Aligning Books and Movies: Towards Story-Like Visual Explanations by Watching Movies and Reading Books. *2015 IEEE International Conference on Computer Vision (ICCV)*. https://doi.org/10.1109/iccv.2015.11
[36] Acheampong, F. A., Nunoo-Mensah, H., & Chen, W. (2021). Transformer models for text-based emotion detection: a review of BERT-based approaches. *Artificial Intelligence Review*, *54*(8), 5789–5829. https://doi.org/10.1007/s10462-021-09958-2
[37] Qudar, M. M. A., & Mago, V. (2020). TweetBERT: A Pretrained Language Representation Model for Twitter Text Analysis. *ArXiv: Computation and Language*. https://arxiv.org/pdf/2010.11091.pdf
[38] Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C. H., & Kang, J. (2019b). BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*. https://doi.org/10.1093/bioinformatics/btz682
[39] Mathur, A., & Suchithra, M. (2022). Application of Abstractive Summarization in Multiple Choice Question Generation. *2022 International Conference on Computational Intelligence and Sustainable Engineering Solutions (CISES)*. https://doi.org/10.1109/CISES54857.2022.9844396
[40] Hegde, C. V., & Patil, S. (2020). Unsupervised Paraphrase Generation using Pre-trained Language Models. *ArXiv: Computation and Language*. https://arxiv.org/abs/2006.05477
[41] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (n.d.). *Language Models are Unsupervised Multitask Learners*. https://d4mucfpksywv.cloudfront.net/better-language models/language_models_are_unsupervised_multitask_learners.pdf
[42] Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., . . . Amodei, D. (2020). Language Models are Few-Shot Learners. *ArXiv: Computation and Language*. https://arxiv.org/pdf/2005.14165.pdf
[43] Mikolov, T., Chen, K., Corrado, G. S., & Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. *ArXiv: Computation and Language*. http://export.arxiv.org/pdf/1301.3781
[44] Papineni, K., Roukos, S., Ward, T., & Zhu, W. J. (2001). BLEU. *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics - ACL '02*. https://doi.org/10.3115/1073083.1073135
[45] Liu, H., Lin, T., Sun, H., Lin, W., Chang, C., Zhong, T., & Rudnicky, A. I. (2017). RubyStar: A Non-Task-Oriented Mixture Model Dialog System. *ArXiv: Computation and Language*. http://export.arxiv.org/pdf/1711.02781
[46] Serban, I. V., Sankar, C., Germain, M., Zhang, S., Lin, Z., Subramanian, S., Kim, T., Pieper, M., Chandar, S., Ke, N. R., Rajeshwar, S., De Brébisson, A., Sotelo, J., Suhubdy, D., Michalski, V., Nguyen, A., Pineau, J., & Bengio, Y. (2017). A Deep Reinforcement Learning Chatbot. *ArXiv: Computation and Language*. http://export.arxiv.org/pdf/1709.02349
[47] Adamopoulou, E., & Moussiades, L. (2020). An Overview of Chatbot Technology. *IFIP Advances in Information and Communication Technology*, 373–383. https://doi.org/10.1007/978-3-030-49186-4_31
[48] Tyen, G., Brenchley, M., Caines, A., & Buttery, P. (n.d.). Towards An Open-Domain Chatbot For Language Practice. *Association for Computational Linguistics*. https://doi.org/10.18653/v1/2022.bea-1.28
[49] Yin, Z., Chang, K., & Zhang, R. (n.d.). DeepProbe: Information Directed Sequence Understanding and Chatbot Design via Recurrent Neural Networks. *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2017*. https://doi.org/10.48550/arXiv.1707.05470
[50] Qiu, M., Li, F. L., Wang, S., Gao, X., Chen, Y., Zhao, W., Chen, H., Huang, J., & Chu, W. (2017). AliMe Chat: A Sequence to Sequence and Rerank based Chatbot Engine. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. https://doi.org/10.18653/v1/p17-2079

[51] Cui, L., Huang, S., Wei, F., Tan, C., Duan, C., & Zhou, M. (2017b). SuperAgent: A Customer Service Chatbot for E-commerce Websites. *Proceedings of ACL 2017, System Demonstrations*. https://doi.org/10.18653/v1/p17-4017
[52] Koehler, B. J. (2017, December 1). AhriBot: A Python Bot Written for Discord Tasks. https://keep.lib.asu.edu/items/134113
[53] Liu, H., Lin, T., Sun, H., Lin, W., Chang, C., Zhong, T., & Rudnicky, A. I. (2017b). RubyStar: A Non-Task-Oriented Mixture Model Dialog System. *ArXiv: Computation and Language*. http://export.arxiv.org/pdf/1711.02781
[54] Tiong, R. L., & Alum, J. (1997). Evaluation of proposals for BOT projects. *International Journal of Project Management*, 15(2), 67–72. https://doi.org/10.1016/s0263-7863(96)00003-8
[55] Rick, S. R., Goldberg, A. P., & Weibel, N. (2019). SleepBot. *Proceedings of the 24th International Conference on Intelligent User Interfaces: Companion*. https://doi.org/10.1145/3308557.3308712
[56] Epstein, J., & Klinkenberg, W. (2001). From Eliza to Internet: a brief history of computerized assessment. *Computers in Human Behavior*, *17*(3), 295–314. https://doi.org/10.1016/s0747-5632(01)00004-8
[57] Weizenbaum, J. (1966). ELIZA—a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, *9*(1), 36–45. https://doi.org/10.1145/365153.365168
[58] Jurafsky, D., & Martin, J. (2000). Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. *Prentice Hall EBooks*. https://nats-www.informatik.uni-hamburg.de/pub/CDG/JurafskyMartin00Comments/JurafskyMartin00-Review.pdf
[59] Wallace, R. S. (2007). The Anatomy of A.L.I.C.E. Parsing the Turing Test, 181–210. https://doi.org/10.1007/978-1-4020-6710-5_13
[60] Bao, Q., Ni, L., & Liu, J. (2020). HHH: An Online Medical Chatbot System based on Knowledge Graph and Hierarchical Bi-Directional Attention. *Proceedings of the Australasian Computer Science Week Multiconference*. https://doi.org/10.1145/3373017.3373049
[61] Sabharwal, N., & Agrawal, A. (2020). Introduction to Google Dialogflow. *Cognitive Virtual Assistants Using Google Dialogflow*, 13–54. https://doi.org/10.1007/978-1-4842-5741-8_2
[62] Samuel, I., Ogunkeye, F. A., Olajube, A., & Awelewa, A. (2020, November). Development of a voice chatbot for payment using amazon lex service with eyowo as the payment platform. In *2020 International Conference on Decision Aid Sciences and Application (DASA)* (pp. 104-108). IEEE.

Authors' Profiles

Sabbir Hossain was born in Dhaka, Bangladesh. He received a BSc degree in Computer Science & Engineering from the American International University-Bangladesh, with a major in software engineering, in 2022.

He is a software developer and researcher at Dr. Anwarul Abedin Institute of Innovation, currently working on a TRP management project in collaboration with Bangabandhu Satellite, all channels, streaming services, and ad agencies in Bangladesh. His research focuses on natural language processing, deep learning, and data science.

Rahman Sharar (correspond author) was born in Dhaka, Bangladesh in 1997. He received a BSc degree in Computer Science & Engineering from the American International University-Bangladesh (AIUB), majoring in software engineering, in 2022.

He served as an intern in the J2SE and C++ laboratory classes at AIUB from May 2022 to August 2022. His research areas comprise of data science and natural language processing, with a keen interest in machine learning.

Md. Ibrahim Bahadur was born in Dhaka, Bangladesh in 2000. He received a BSc degree in Computer Science & Engineering from the American International University-Bangladesh (AIUB), with a major in information systems, in 2022.

He is currently serving as intern at AIUB. His fields of interest lie in Data Science and Web Development.

Abu Sufian was born in Dinajpur, Bangladesh in 2000. He received a BSc degree in Computer Science & Engineering from the American International University-Bangladesh (AIUB), majoring in computer engineering, in 2022.

He is currently working as a Network Operations Centre Engineer at a multinational company and simultaneously running a personal project based on creating awareness of safe internet in his country. His research focuses on Cyber Security, Networking, and Artificial Intelligence.

Rashidul Hasan Nabil received a BSc degree in Computer Science & Engineering, and an MSc degree in Computer Science with specialization in Intelligent Systems, from the American International University-Bangladesh.

He is currently working as a Lecturer in the Department of Computer Science under the Faculty of Science and Technology at American International University-Bangladesh (AIUB). His research interest includes Human-Machine Interaction, Human-Computer Interaction, Machine Learning, and Deep Learning.

How to cite this paper: Sabbir Hossain, Rahman Sharar, Md. Ibrahim Bahadur, Abu Sufian, Rashidul Hasan Nabil, "MediBERT: A Medical Chatbot Built Using KeyBERT, BioBERT and GPT-2", International Journal of Intelligent Systems and Applications(IJISA), Vol.15, No.4, pp.53-69, 2023. DOI:10.5815/ijisa.2023.04.05