International Journal On Engineering Technology and Sciences – IJETS
ISSN (P):2349-3968, ISSN (O): 2349-3976 Volume XI - Issue IV, April – 2024

AI BASED MEDICAL CHATBOT MODEL FORINFECTIONS DISEASE PREDICTION

[1]Abisheik S, [2]Manikarthick K, [3]Rajapaul J, [4]Syed Abuthakir S, [5]Mr.Syed Mohamed Ali. M
[1,2,3,4]Student, [5]Assistant Professor, Department of Information technology,
Loyola Institute of Technology andScience, Kanyakumari, Tamil Nadu, India

Abstract - In the realm of healthcare, the integration of artificial intelligence (AI) and natural language processing (NLP) technologies has significantly enhanced the accessibility and efficiency of medical assistance. This study presents the development of a novel medical chatbot designed to streamline the process of disease identification and classification. The chatbot utilizes disease information provided in JSON format, employing advanced NLP techniques to comprehend user queries effectively. Through the implementation of a Long Short-Term Memory (LSTM) classifier, the system categorizes the input data into pertinent disease categories with high accuracy and reliability. The methodology encompasses several key stages. Initially, the JSON input containing disease details is parsed and preprocessed to extract relevant features and ensure data quality. Subsequently, NLP algorithms are applied to interpret user inputs, facilitating seamless interaction between the user and the chatbot. The LSTM classifier, trained on a comprehensive dataset of disease patterns, effectively categorizes the input information, enabling rapid and accurate diagnosis. Furthermore, the study addresses the user interface aspect by incorporating a tablet interface for intuitive interaction. This interface serves as a user-friendly platform for inputting queries and receiving informative responses from the chatbot. The system's design prioritizes accessibility and user experience, ensuring that individuals with varying levels of technological proficiency can benefit from its functionalities. The efficacy of the developed medical chatbot is evaluated through comprehensive testing procedures, including performance assessment and user feedback analysis. Results demonstrate the system's ability to accurately identify and classify diseases, thereby enhancing medical diagnosisand decision-making processes.

I. INTRODUCTION

In recent years, the fusion of healthcare with cutting-edge technologies has brought forth transformative innovations, reshaping the landscape of medical assistance and accessibility. Among these innovations, medical chatbots stand out as promising solutions that leverage artificial intelligence (AI) and natural language processing (NLP) to provide immediate and personalized healthcare support. This comprehensive introduction will delve into the evolution of medical chatbots, their significance in modern healthcare, the challenges they aim to address, and the potential impact they hold in revolutionizing patient care. The concept of chatbots traces its roots back to the early days of computing, where primitive forms of conversational agents were developed to simulate human-like interactions. Over the years, advancements in AI, NLP, and machine learning have propelled the evolution of chatbots, transforming them from rudimentary assistants to sophisticated virtual agents capable of understanding and responding to natural language queries. In the context of healthcare, the evolution of medical chatbots has been particularly notable, driven by the increasing demand for accessible and efficient medical assistance. In today's fast-paced world, access to timely and personalized healthcare services is paramount.However, traditional healthcare delivery models oftenface challenges such as long wait times, limited availability of healthcare professionals, and geographic barriers to access. Medical chatbots offer a promising solution to these challenges by providing immediate and round-the-clock support to individuals seeking medical advice, information, or assistance. With the ability to engage in natural conversations and interpret user queries, chatbots serve as virtual healthcare companions, empowering individuals to take control of their health and well-being.

Despite the promise of medical chatbots, several challenges exist in their development and implementation. One of the primary challenges is ensuring the accuracy and reliability of medical information provided by chatbots. As healthcare is a complex and nuanced domain, chatbots must be equipped with robust algorithms and up-to-date medical knowledge to deliver accurate responses and recommendations. Additionally, ensuring user privacyand data security is crucial, particularly when dealing with sensitive health information. Furthermore, achieving seamless integration with existing healthcare systems and workflows presents a significant challenge, as chatbots must be able to collaborate effectively with healthcare professionals and complement existing medical services.

The potential impact of medical chatbots on patient care is immense. By providing immediate access to medical assistance, chatbots can help individuals make informed decisions about their health and seek timely treatment when needed. Moreover, by offering personalized recommendations and guidance based on individual health profiles and preferences, chatbots can empower individuals to adopt healthy behaviors and adhere to prescribed treatment regimens. This proactive approach to healthcare management has the potential to improve patient outcomes, reduce healthcare costs, and enhance overall quality of life.

In summary, medical chatbots represent a groundbreaking innovation in modern healthcare, offering immediate, personalized, and accessible medical assistance to individuals worldwide. While challenges exist in their development and implementation, the potential impact of chatbots on patient care is undeniable. By leveraging AI, NLP, and machine learning technologies, chatbots have the power to transform the way healthcare is delivered, making it more efficient, effective, and patient-centered. As we continue to explore the possibilities of medical chatbots, it is imperative to prioritize accuracy, privacy, and integration to ensure their successful adoption and widespread use in the healthcare industry.

II PROPOSEDSYSTEM

Traditionally, medical chatbots have focused on interpreting user queries and providing information or guidance based on predefined algorithms. However, these systems often fall short in offering personalized medication recommendations tailored to the user's specific health condition. Moreover, there is a lack of integration with tablet interfaces, which could serve as a convenient medium for delivering such recommendations. Our proposed system aims to bridge these gaps by leveraging advanced technologies such as natural language processing (NLP) and LSTM classification to analyze user inputs effectively. By processing disease-related information provided in JSON format, the system can accurately classify and analyze the user's health condition. Additionally, the integration of a tablet interface enables the chatbot to recommend suitable medications based on the identified disease, ensuring a holistic approach to medical assistance. Through this innovative integration, our system not only provides users with valuable insights into their health status but also offers personalized recommendations for medication, thereby enhancing the overall quality of healthcare delivery.

Modules:

Data Preprocessing: Parse the JSON input containing disease information and preprocess it for further analysis. This may involve extracting relevant features, cleaning the data, and preparing it for input into the LSTM classifier.

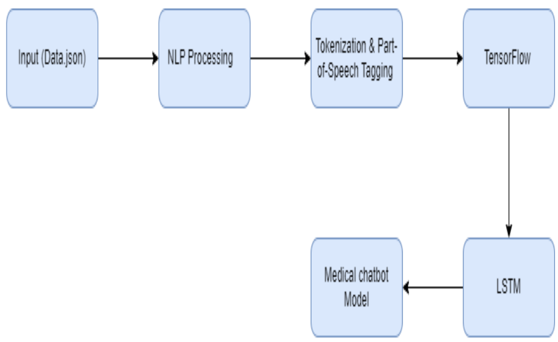
NLP Processing: Apply NLP techniques to understand the user's input or query. This could involve tasks such as tokenization, stemming, or lemmatization to extractthe meaning from the text.

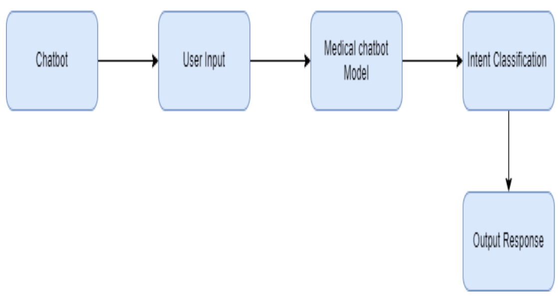
LSTM Classification: Train or use a pre-trained LSTMclassifier to classify the input data based on the disease information provided. This classifier should be able tocategory size the input into relevant disease categoriesor labels.

Response Generation: Based on the classification results, generate appropriate responses to the user's input. These responses should be informative and helpful in addressing the user's query or concern.

UI Interface: Design and implement a tablet interface that allows users to interact with the chatbot. This interface should provide a user-friendly experience forinputting queries and receiving responses.

Block diagram





Convolution Neural Network (CNN)

A multi-layer feed-forward neural network, or CNN, can reduce computation time and complexity while improving error in a backpropagation network (BP). As a result of its ability to recognise local characteristics using a convolution kernel and automatically learn these features for classification purposes, it has recently been employed for sentiment classification. Convolution layer, pooling layer, and fully connected layer are the three basic layers that make up a CNN model. The convolutional layer receives sentences as input in the form of a matrix of numbers. Each token in a sentence, which is made up of words, corresponds to a row or vector on the matrixtable.

Neural networks are a subset of machine learning, and they are at the heart of deep learning algorithms. They are comprised of node layers, containing an input layer, one or more hidden layers, and an output layer. Each node connects to another and has an associated weight and threshold. If the output of any individual node is above the specified threshold value, that node is activated, sending data to the next layer of the network. Otherwise, no data is passed along to the next layer of the network.

While we primarily focused on feedforward networks in that article, there are various types of neural nets, which are used for different use cases and data types. For example, recurrent neural networks are commonly used for natural language processing and speech recognition whereas convolutional neural networks (ConvNets or CNNs) are more often utilizedfor classification and computer vision tasks. Prior to CNNs, manual, time-consuming feature extraction methods were used to identify objects in images. However, convolutional neural networks now provide a more scalable approach to image classification and object recognition tasks,leveraging principles from linear algebra, specifically matrix multiplication, to identify patterns within an image. That said, they can be computationally demanding, requiring graphical processing units (GPUs) to train models.

Convolutional neural networks are distinguished from other neural networks by their superior performance with image, speech, or audio signal inputs. They have three main types of layers, which are:

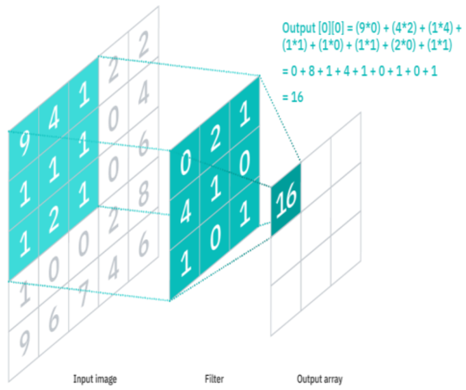
- Convolutional layer
- Pooling layer
- Fully-connected (FC) layer

The convolutional layer is the first layer of a convolutional network. While convolutional layers can be followed by additional convolutional layers or pooling layers, the fully-connected layer is the final layer. With each layer, the CNN increases in its complexity, identifying greater portions of the image. Earlier layers focus on simple features, such as colors and edges. As the image data progresses through the layers of the CNN, it starts to recognize large elements or shapes of the object until it finally identifies the intended object.

Convolutional Layer

The convolutional layer is the core building block of a CNN, and it is where the majority of computation occurs. It requires a few components, which are input data, a filter, and a feature map. Let's assume that the input will be a color image, which is made up of a matrix of pixels in 3D. This means that the input will have three dimensions—a height, width, and depth—which correspond to RGB in an image. We also have a feature detector, also known as a kernel or a filter, which will move across the receptive fields of the image, checking if the feature is present. This process is known as a convolution.

The feature detector is a two-dimensional (2-D) array of weights, which represents part of the image. While they can vary in size, the filter size is typically a 3x3 matrix; this also determines the size of the receptive field. The filter is then applied to an area of the image, and a dot product is calculated between the input pixels and the filter. This dot product is then fed into an output array. Afterwards, the filter shifts by a stride, repeating the process until the kernel has swept across the entire image. The final output from the series of dot products from the input and the filter is known as a feature map, activation map, or a convolved feature.



As you can see in the image above, each output value in the feature map does not have to connect to each pixel value in the input image. It only needs to connect to the receptive field, where the filter is being applied. Since the output array does not need to map directly to each input value, convolutional (andpooling) layers are commonly referred to as "partially connected" layers. However, this characteristic can also be described as local connectivity.

Note that the weights in the feature detector remain fixed as it moves across the image, which is also known as parameter sharing. Some parameters, like the weight values, adjust during training through the process of backpropagation and gradient descent. However, there are three hyperparameters which affect the volume size of the output that need to be set before the training of the neural network begins. These include:

1. The **number of filters** affects the depth of the output. For example, three distinct filters would yield three different feature maps, creating a depth of three.

2. **Stride** is the distance, or number of pixels, that the kernel moves over the input matrix. While stride values of two or greater is rare, a larger stride yields a smaller output.

3. **Zero-padding** is usually used when the filters do not fit the input image. This sets all elements that fall outside of the input matrix to zero, producing a larger or equally sized output. There are three types of padding:

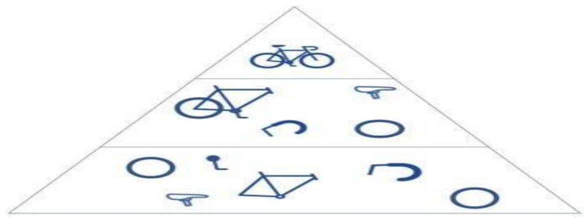
- **Valid padding:** This is also known as no padding. In this case, the last convolution is dropped if dimensions do not align.
- **Same padding:** This padding ensures that the output layer has the same size as the input layer
- **Full padding:** This type of padding increases the size of the output by adding zeros to the border of the input.

After each convolution operation, a CNN applies a Rectified Linear Unit (ReLU) transformation to the feature map, introducing nonlinearity to the model.

As we mentioned earlier, another convolution layer can follow the initial convolution layer. When this happens, the structure of the CNN can become hierarchical as the later layers can see the pixels within the receptive fields of prior layers. As an example, let's assume that we're trying to determine if an image contains a bicycle. You can think of the bicycle as a sum of parts. It is comprised of a frame, handlebars, wheels, pedals, et cetera. Each individual part of the bicycle makes up a lower-level pattern in the neural net, and the combination of its parts represents a higher-level pattern, creating a feature hierarchy within the CNN

Ultimately, the convolutional layer converts the image into numerical values, allowing the neural network to interpret and extract relevant patterns.

Pooling Layer



Pooling layers, also known as down sampling, conducts dimensionality reduction, reducing the number of parameters in the input. Similar to the convolutional layer, the pooling operation sweeps a filter across the entire input, but the difference is that this filter does not have any weights. Instead, the kernel applies an aggregation function to the values within the receptive field, populating the output array.There are two main types of pooling:

- **Max pooling:** As the filter moves across the input, it selects the pixel with the maximum value to send to the output array. As an aside, this approach tends to be used more often compared to average pooling.
- **Average pooling:** As the filter moves across the input, it calculates the average value within the receptive field to send to theoutput array.

While a lot of information is lost in the pooling layer, it also has a number of benefits to the CNN. They help to reduce complexity, improve efficiency, and limit risk of overfitting.

Fully-Connected Layer

The name of the full-connected layer aptly describes itself. As mentioned earlier, the pixel values of the input image are not directly connected to the output layer in partially connected layers. However, in the fully-connected layer, each node in the output layer connects directly to a node in the previous layer.

This layer performs the task of classification based on the features extracted through the previous layers and their different filters. While convolutional and pooling layers tend to use ReLu functions, FC layers usually leverage a softmax activation function to classify inputs appropriately, producing a probability from 0 to1.

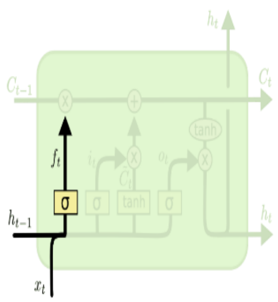
The Word2Vec and GloVe model are two common embedding techniques that produce these vectors. The CNN model uses filters to extract local feature from input vectors. Convolutional layer, the most crucial layer in CNN, performs the majority of feature computations. Utilizing a process known as convolution kernel, convolutional layer generates feature maps.

The most crucial features are extracted by the pooling layer following the convolution procedure. Calculated by the pooling layer are local enough statistics. Through this procedure, the feature dimensions of the pooling layer are decreased, CNN's computational time and cost are decreased, and the overfitting issue is avoided. In order to categorise sentiment results, the fully connected layer creates a probability distribution.**Long Short Term Memory**

One of the deep learning algorithms, RNN, is mostly used in natural language processing (NLP) to anticipate the next word based on the words that have already been given in a phrase.

The first step in our LSTM is to decide what information we're going to throw away from the cell state. This decision is made by a sigmoid layer called the “forget gate layer.” It looks at ht−1ht−1 and xtxt, and outputs a number between 00 and 11 for each number in the cell state Ct−1Ct−1. A 11 represents “completely keep this” while a 00 represents “completely get rid of this.”

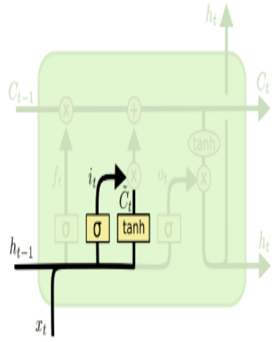
Let's go back to our example of a language model trying to predict the next word based on all the previous ones. In such a problem, the cell state might include the gender of the present subject, so that the correct pronouns can be used. When we see a new subject, we want to forgetthe gender of the old subject.



$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right)$$

The next step is to decide what new information we'regoing to store in the cell state. This has two parts. First, a sigmoid layer called the “input gate layer” decides which values we'll update. Next, a tanh layer creates avector of new candidate values, C~tC~t, that could be added to the state. In the next step, we'll combine thesetwo to create an update to the state.

In the example of our language model, we'd want to add the gender of the new subject to the cell state, to replacethe old one we're forgetting.



$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] + b_i\right)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

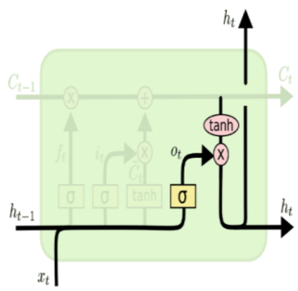
It's now time to update the old cell state, Ct−1Ct−1, into the new cell state CtCt. The previous steps already decided what to do, we just need to actually do it.

We multiply the old state by ftft, forgetting the things we decided to forget earlier. Then we add it∗C~tit∗C~t. This is the new candidate values, scaled by how much we decided to update each state value.

In the case of the language model, this is where we'd actually drop the information about the old subject's gender and add the new information, as we decided inthe previous steps.

Finally, we need to decide what we're going to output. This output will be based on our cell state, but will be a filtered version. First, we run a sigmoid layer which decides what parts of the cell state we're going to output. Then, we put the cell state through tanhtanh (to push thevalues to be between −1−1 and 11) and multiply it by theoutput of the sigmoid gate, so that we only output the parts we decided to.

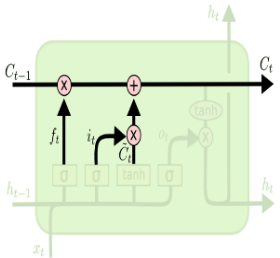
For the language model example, since it just saw a subject, it might want to output information relevant to a verb, in case that's what is coming next. For example, it



$$o_t = \sigma\left(W_o \, [h_{t-1}, x_t] + b_o\right)$$

$$h_t = o_t * \tanh\left(C_t\right)$$

might output whether the subject is singular or plural, sothat we know what form a verb should be conjugated intoif that's what follows next.



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

III RESULT AND DISCUSION

Certainly! Here's a brief explanation of the results and discussion section for a project on an AI-based medical chatbot model for infectious diseases:

1. Summary of Results: Begin by summarizing the key findings and outcomes of your project. Highlight the performance metrics of your AI-based medical chatbot model, such as accuracy, precision, recall, F1 score, etc., in diagnosing infectious diseases.
2. Comparison with Existing Methods: Discuss how your AI-based chatbot model compares with existing methods of diagnosing infectious diseases. Highlight any advantages or improvements your model offers over traditional methods or other AI-based approaches.
3. Evaluation Metrics: Provide a detailed analysis of the evaluation metrics used to assess the performance of your chatbot model. Explain how these metrics were calculated and what they indicate about the model's effectiveness in diagnosing infections accurately.
4. Case Studies or Examples: Include case studies or examples of real interactions between users and your chatbot model. Discuss how the chatbot responded to different scenarios and whether it accurately diagnosed infections based on the symptoms provided by users.
5. User Feedback: Present any feedback received from users who interacted with the chatbot. Discuss their experiences, satisfaction levels, and any suggestions for improvement they may have provided.
6. Challenges and Limitations: Acknowledge any challenges or limitations encountered during the development and deployment of the chatbot model. This could include issues related to data quality, model performance, user acceptance, or ethical considerations.
7. Future Directions: Outline potential future directions for improving the chatbot model. This could involve refining the model architecture, incorporating additional data sources, expanding its capabilities to diagnose a wider range of infections, or enhancing theuser interface for better usability.
8. Conclusion: Summarize the main findings and implications of your study. Reinforce the significance of your AI-based medical chatbot model for diagnosing infectious diseases and its potential impacton healthcare delivery and patient outcomes.

By covering these key points in your results and discussion section, you can provide a comprehensive overview of your project's findings and insights into the development and evaluation of your AI-based medical chatbot model for infectious diseases.

IV CONCLUSION

In conclusion, the proposed system represents a significant advancement in the field of medical assistance, particularly in disease analysis and medication recommendation. By integrating a chatbot interface with tablet recommendation capabilities, the system offers a holistic approach to addressing user health concerns. Throughout the development process, we focused on leveraging cutting-edge technologies such as natural language processing (NLP) and LSTM classification to ensure accurate interpretation of user inputs and disease classification. The system's abilityto process disease-related information provided in JSON format enables it to effectively analyze the user's health condition and provide personalized recommendations for medication.

REFERENCES

[1] Nie Y, Williams J, Dinan E, Weston J. Dialogue natural language inference. arXiv Prepr; 2020. arXiv:2005.07421. 24. Chen X, Qian J, Lu H, Zhu H. BERT for joint intent classification and slot filling. arXiv Prepr; 2019. arXiv:1902.10909.

[2] Safi Z, Abd-Alrazaq A, Khalifa M, Househ M. Technical aspects of developing chatbots for medical applications: scoping review. J Med Internet Res. 2020;22(12), e19127.

[3] Topol E. Deep medicine: how artificial intelligence can make healthcare human again. Hachette UK; 2019.

[4] Bao Q, Ni L, Liu J. HHH: an online medical chatbot system based on knowledge graph and hierarchical bi-directional attention. In: Proceedings of the Australasian computer science week multiconference. 2020, February:1–10.

[5] Haug CJ, Drazen JM. Artificial intelligence and machine learning in clinical medicine, 2023. N Engl J Med. 2023;388(13):1201–1208.

[6] Battineni Gopi, Chintalapudi Nalini, Amenta Francesco. AI Chatbot design during an epidemiclike the novel coronavirus. Healthcare. 2020;8(2):154.

[7] Soufyane A, Abdelhakim BA, Ahmed MB. An intelligent chatbot using NLP and TFIDF algorithm for text understanding applied to the medical field. In: Emerging trends in ICT for sustainable development: The proceedings of NICE2020 international conference. Cham: Springer International Publishing; 2021, January:3–10.

[8] Athota L, Shukla VK, Pandey N, Rana A. Chatbot for healthcare system using artificial intelligence. In: 2020 8th International conference on reliability, infocom technologies and optimization (trends and future directions)(ICRITO). IEEE; 2020, June: 619– 622.

[9] Ayanouz S, Abdelhakim BA, Benhmed M. A smart chatbot architecture based NLP and machine learning for health care assistance. In: Proceedings of the 3rd international conference on networking, information systems & security. 2020, March: 1–6.

[10] Vaira L, Bochicchio MA, Conte M, Casaluci FM, Melpignano A. MamaBot: a System based on ML and NLP for supporting Women and Families during Pregnancy. In: Proceedings of the 22nd international database engineering & applications symposium. 2018, June:273–277.

[11] Darcy AM, Louie AK, Roberts LW. Machine learning and the profession of medicine. Jama. 2016;315(6):551–552.

[12] Shree R, Rastogi A, Kalaiarasan C. Machine learning-driven cutting-edge approach for designing a healthcare Chatbot. Int J Intell Syst Appl Eng. 2023;11(8s):198–205.

[13] Fonna MR, Widyantoro DH. Tutorial system in learning activities through machine learning- based Chatbot applications in pharmacology education. In: 2021 8th International Conference on Advanced Informatics: Concepts, Theory and Applications (ICAICTA). IEEE; 2021, September:1–6.

[14] Kim Y, Kim JH, Kim YM, Song S, Joo HJ. Predicting medical specialty from text based on a domain-specific pre-trained BERT. Int J Med Inform. 2023;170, 104956.

[15] Thwala EKI, Adegun AA, Adigun MO. Self-assessment Chatbot for COVID-19 prognosis using deep learning-based natural language processing (NLP). In: In 2023 international conference on Science, Engineering and Business for Sustainable Development Goals(SEB-SDG)Vol. 1

[16]. 16. Chakraborty S, Paul H, Ghatak S, et al. An AI-based medical Chatbot model for infectious disease prediction. Ieee Access. 2022;10:128469–128483.

[17] 17. Tamizharasi B, Livingston LJ, Rajkumar S. Building a medical chatbot using support vector machine learning algorithm. J Phys Conf Ser. 2020, December;1716(1). p. 012059). IOP Publishing.

[18]18. Badlani S, Aditya T, Dave M, Chaudhari S. Multilingual healthcare chatbot using machine learning. In: 2021 2nd International Conference For Emerging Technology (INCET). IEEE; 2021, May:1–6.

[19] El Zini J, Rizk Y, Awad M, Antoun J. Towards a deep learning question-answering specialized chatbot for objective structured clinical examinations. In: 2019 International Joint Conference on Neural Networks (IJCNN). IEEE; 2019, July:1–9.

[20] Serban IV, Sankar C, Germain M, et al. A deep reinforcement learning chatbot. arXiv Prepr; 2017. arXiv:1709.02349.

[21]21. Brown TB, Mann B, Ryder N, et al. Language models are few-shot learners. arXiv Prepr; 2020. arXiv:2005.14165.

[22] Dinan E, Urbanek J, Szlam A, Kiela D, Weston J. TransferTransfo: a transfer learning approach for neural network based conversational agents. arXiv Prepr; 2019. arXiv:1 901.08149.