

REQUEST RESPONSE SYSTEM USING CONVOLUTIONAL NEURAL NETWORK IN MEDICAL INDUSTRY

¹S. Sri Lavanya, ²Sreemya Gavini, ³Meda Praneetha, ⁴Kolipaka Gayathri, ⁵Malyala Sai Shreya ¹Assistant Professor, ^{2,3,4,5}UG Students, Dept. of CSE (AI & ML), Malla Reddy Engineering College for Women (Autonomous), Hyderabad, India. E-Mail: sslavanya79@gmailcom

ABSTRACT

You must take care of your health if you want to live a long and healthy life. Many times, people are unaware of all the possible diagnoses, treatments, or signs of a specific illness. For young people, it is quite difficult to get a doctor's appointment for any health issues and managing phone calls, so users with the problem travel to the hospital in person for a time-consuming test. By offering relevant advice on leading a healthy lifestyle, healthcare/medical chatbots can help with these kinds of issues. The idea is to use neural networks to build a medical chatbot that can provide information, identify diseases, and provide basic disease information as well as when and where to seek medical advice. Natural language processing technology, which enables users to post their concerns about any disease as well as their health, is the foundation of the success of medical chatbots. Without physically being at the clinic or hospital, the user can use a chatbot to ask any personal questions about health care. Through a medical chat-bot, this will increase access to medical information while lowering the cost of healthcare. The programme development strategy calls for customer sentiment analysis.

Introduction

In order to maintain a high quality of life, there has been a rising understanding over the past few decades of the significance of prioritising one's own health. However, people frequently find it difficult to offer their healthcare requirements the proper attention as a result of a variety of issues. By creating a healthcare chatbot, we want to address this problem by utilising the strengths of machine learning (ML), natural language processing (NLP), and convolutional neural network (CNN) technologies.

A healthy body is a priceless treasure, and everyone wants to live a better life. We now have the chance to offer people healthcare services conveniently and affordably thanks to advances in ML. The topic of artificial intelligence (AI) is thriving, offering a variety of applications in the research arena, since technology has become an essential aspect of our life. The accuracy of risk classification based on large data volumes is improved by researchers' increased attention on disease prediction using big data analysis.

Even while e-healthcare resources are essential for developing nations, there are still issues with their awareness and infrastructure. Many people turn to the internet for solutions to their healthcare-related questions. In order to meet this demand, we created a platform that helps doctors by providing patients with online medical services. Users of this portal can quickly receive medical advice, learn about various ailments, and investigate the various diagnoses.

We have put an illness prediction chatbot in place to increase communication effectiveness. In order to understand user inputs and respond appropriately, chatbots use ML and NLP techniques to emulate

human communication. In our study, we suggest a chatbot for disease prediction that combines NLP, ML, and the CNN algorithm.

The goal of our project is to create a healthcare chatbot that employs ML, NLP, and the CNN algorithm to deliver individualised medical care. We want to empower people to take charge of their health and enhance their general well-being by combining cutting-edge technologies. It is a healthcare chatbot for improved access to medical guidance.

Existing System

Several of the current systems rely on text-based chats, but these come with certain limitations. One major drawback is the absence of instant responses for patients, as they often have to wait for a significant amount of time to receive acknowledgment from experts. Additionally, the existing systems are constrained by a limited dataset that covers only a specific number of diseases. Moreover, technical issues persist, particularly with the accuracy of voice messages in the current system.

Problem Definition

The hectic pace of modern life has led to a significant decline in individuals' ability to prioritize their health, ultimately jeopardizing their overall quality of life. To address this issue, leveraging machine learning (ML) offers an opportunity to provide healthcare services conveniently and affordably. However, establishing effective e-healthcare facilities, particularly in developing countries, remains challenging due to limited awareness and inadequate infrastructure. Additionally, there is a growing reliance on the internet for healthcare-related queries, highlighting the need for accessible and reliable medical guidance.

In response to these challenges, our project aims to develop a healthcare chatbot that utilizes ML, natural language processing (NLP), and convolutional neural network (CNN) algorithms. The primary objective is to create a platform that assists healthcare professionals in delivering online medical services, enabling users to easily seek medical guidance and gain exposure to various diseases and corresponding diagnoses. By implementing a chatbot for disease prediction, communication with the platform will become more effective and personalized.

Therefore, it revolves around the need to develop an ML-based healthcare chatbot using NLP and CNN algorithms to address the following key challenges:

- a. Lack of prioritization of personal health
- b. Limited access to healthcare services
- c. Inadequate infrastructure and awareness
- d. Dependency on the internet for healthcare queries
- e. Inefficient communication channels

By addressing these challenges through the development of an ML-powered healthcare chatbot, we aim to enhance individuals' access to reliable healthcare services, improve disease prediction accuracy, and ultimately promote better health outcomes and an enhanced quality of life.

Proposed System

In our system, users have the flexibility to interact with the chatbot through text or voice, providing a seamless and user-friendly experience. The chatbot employs a combination of text and voice responses to cater to the user's preferred mode of communication. When users engage in a conversation with the chatbot, it utilizes advanced algorithms and techniques to analyze their queries and identify potential diseases based on the symptoms or information provided. Drawing upon its vast knowledge base, the chatbot then offers valuable suggestions and recommendations specific to the identified disease, empowering users to make informed decisions about their health. Additionally, the system goes beyond disease identification by prescribing specialist doctors who specialize in the relevant field, facilitating

further consultations if necessary. Importantly, our system is designed to accommodate multiple users simultaneously, ensuring smooth performance without any lagging or delays, thereby optimizing the user experience for all.

SYSTEM ARCHITECTURE

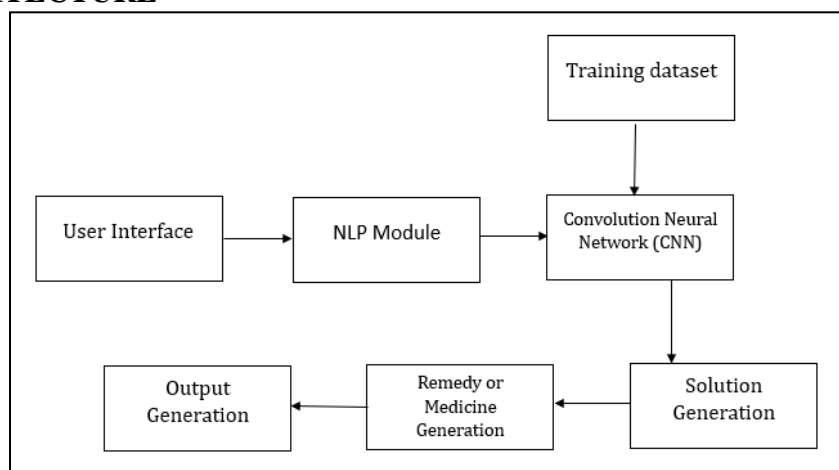


Fig.1. System architecture

User Interface

This is the interface through which users interact with the chatbot.

Users can input their questions and symptoms.

Natural Language Processing (NLP) Module:

This module is responsible for understanding and processing the user's input.

It performs tasks such as tokenization, entity recognition, and sentiment analysis.

NLP techniques like Named Entity Recognition (NER) can be used to extract relevant symptoms from the user's input.

Convolutional Neural Network (CNN):

The CNN is responsible for analyzing the symptoms extracted by the NLP module.

The CNN model is trained using a dataset that contains symptoms, corresponding remedies, and basic medicines.

The CNN architecture may consist of multiple convolutional layers, pooling layers, and fully connected layers to learn patterns and make predictions.

Training Dataset:

This dataset contains labelled examples of symptoms, remedies, and basic medicines.

Each example consists of symptom input, corresponding remedy, and basic medicine output.

Solution Generation:

The CNN model takes the extracted symptoms as input and generates predictions for remedies and basic medicines.

The model outputs a ranked list of potential remedies and basic medicines based on the symptoms provided.

Remedy/Medicine Recommendation:

Based on the predictions generated by the CNN model, the chatbot selects the most appropriate remedies and basic medicines.

The recommendations may be ranked based on relevance or confidence scores assigned by the CNN model.

Output Generation:

The chatbot generates a response to the user, providing the recommended home remedies and basic medicines.

The response may include additional information, instructions, or warnings related to the recommended remedies and medicines.

IMPLEMENTATION**Libraries****TensorFlow**

Google created the open-source machine learning framework known as TensorFlow. It is intended to streamline the creation, training, and application of machine learning models. TensorFlow offers a comprehensive set of tools and libraries that allow developers and researchers to work with large-scale numerical computations efficiently.

Here are some key aspects and features of TensorFlow:

Computational Graph: TensorFlow represents computations as a directed graph called a computational graph. In this graph, nodes represent mathematical operations, and edges represent the data flow between these operations. This graph-based approach allows for efficient parallel execution and optimization of computations.

Automatic Differentiation: TensorFlow includes automatic differentiation capabilities, which enable the computation of gradients automatically. Gradients are essential for optimizing machine learning models through techniques like gradient descent. TensorFlow's automatic differentiation simplifies the process of implementing complex optimization algorithms.

Flexible Architecture: TensorFlow offers a flexible architecture that allows developers to build and deploy machine learning models on various platforms and devices. It supports distributed computing, allowing models to be trained on multiple machines or devices simultaneously.

High-Level APIs: TensorFlow offers high-level APIs with user-friendly interfaces for creating and training deep learning models, such as Keras. Keras simplifies the process of defining and training neural networks by providing abstractions and pre-built layers. It allows for rapid prototyping and experimentation.

NLTK

Popular Python library NLTK (Natural Language Toolkit) was created especially to work with human language data. It offers a wide range of resources, tools, and methods for handling, analysing, and processing textual data. Natural language processing (NLP) research and application development heavily utilise NLTK.

Here are some key aspects and features of NLTK:

Text Corpora and Resources: NLTK includes a diverse collection of lexical resources which cover various languages and domains, providing researchers and developers with pre-processed text data for experimentation and analysis. NLTK also provides access to datasets like the Gutenberg Corpus, WordNet, and various language corpora.

Tokenization: NLTK offers various tokenization methods to break text into smaller units. Tokenization is a crucial step in many NLP tasks, including text classification, information retrieval, and sentiment analysis. NLTK provides ready-to-use tokenizers that can handle different languages and tokenization requirements.

Part-of-Speech Tagging: NLTK includes algorithms and models for part-of-speech (POS) tagging, which assigns grammatical labels to each word in a sentence. POS tagging is essential for many NLP tasks, such as text parsing, named entity recognition, and machine translation. NLTK provides pre-trained models for POS tagging and allows for training custom taggers.

Stemming and Lemmatization: NLTK provides functions for stemming and lemmatization, which help reduce words to their base or root forms. Stemming involves removing prefixes or suffixes from words, while lemmatization maps words to their dictionary or base form. These techniques are useful for tasks like information retrieval, text normalization, and topic modelling.

Sentiment Analysis: NLTK offers tools and resources for sentiment analysis. Sentiment analysis is used in areas like social media monitoring, customer feedback analysis, and opinion mining. NLTK provides pre-trained sentiment analysis classifiers and lexicons to perform sentiment analysis tasks.

Machine Learning and Classification: NLTK includes algorithms like Naive Bayes, decision trees, and maximum entropy classifiers. These algorithms can be trained on labelled data to classify text into predefined categories or labels.

Natural Language Processing Techniques: NLTK incorporates various NLP techniques, such as named entity recognition, chunking, syntactic parsing, and dependency parsing. These techniques allow for deeper linguistic analysis and extraction of structured information from unstructured text.

NumPy:

The NumPy module, short for Numerical Python, is a fundamental package for scientific computing in Python. It provides efficient and flexible data structures, functions, and tools for working with large arrays and matrices of numeric data. numpy is widely used in various fields such as data analysis, machine learning, image processing, and simulation.

Here are some key aspects and features of the NumPy module:

ndarray: The core data structure of NumPy is the ndarray (n-dimensional array). It is a powerful container that allows efficient storage and manipulation of homogeneous data, such as numbers. ndarrays provide fast and vectorized operations, making them suitable for numerical computations. They can have any number of dimensions and support various data types.

Vectorized Operations: NumPy enables efficient vectorized operations on arrays. Vectorization allows mathematical operations to be applied to entire arrays rather than individual elements, which significantly improves performance. With NumPy, you can perform arithmetic operations, element-wise computations, and mathematical functions on arrays without using explicit loops.

Array Creation & Manipulation: NumPy offers a wide range of functions for manipulating arrays. You can reshape arrays, change the dimensions, transpose arrays, concatenate or split arrays, and perform various other transformations. These functions allow you to manipulate the shape and structure of arrays to suit your specific needs.

Mathematical Functions: NumPy includes a comprehensive set of mathematical functions that operate elementwise on arrays. It provides functions for basic arithmetic operations, trigonometry, exponential and logarithmic functions, statistical computations, and much more. These functions are optimized for performance and can be applied to entire arrays efficiently.

Scikit – learn:

Through a standardised Python interface, Scikit-Learn offers a variety of supervised and unsupervised learning techniques. It is distributed under several Linux distributions and is available under a liberal simplified BSD licence, which promotes both academic and commercial use. SciPy (Scientific Python), which must be installed before using scikit-learn, is the foundation upon which the library is based. This stack of items contains:

Flask:

The Flask module is a lightweight and popular web framework for building web applications in Python. It provides a simple and flexible way to create web servers, handle HTTP requests, and generate dynamic web pages. Flask is known for its minimalistic design, ease of use, and extensive community support.

Here are some key aspects and features of the Flask module:

<http://doi.org/10.36893/JNAO.2023.V14I2.0204-0214>

Routing: Flask allows you to define routes, which associate URL patterns with functions called view functions. When a request is made to a specific URL, Flask maps it to the corresponding view function, which generates the response to be sent back to the client. Routing enables you to create different endpoints and handle different types of requests (GET, POST, etc.) easily.

Request Handling: Flask provides convenient mechanisms to handle incoming HTTP requests. You can access request data such as form inputs, query parameters, and request headers. Flask simplifies tasks like form validation, file uploads, and cookie handling. It also provides built-in support for handling common request types, such as JSON and XML.

Response Generation: Flask offers flexibility in generating HTTP responses. You can return plain text, HTML, JSON, or other content types as responses. Flask handles content negotiation automatically, allowing you to easily return different response formats based on the client's request. Additionally, Flask provides convenient methods for setting response headers and status codes.

following RESTful principles. Flask's simplicity and flexibility make it a popular choice for API development.

Colorama:

The Colorama module is a Python library that provides cross-platform support for coloured terminal text. It simplifies the process of printing coloured output in command-line interfaces, making it easier to add visual elements to text-based applications. Colorama is compatible with Windows, macOS, and Linux operating systems.

Here are some key aspects and features of the Colorama module:

Cross-Platform Support: Colorama handles the differences in ANSI escape sequence support among different operating systems. It automatically detects the platform and modifies the text output accordingly, ensuring that coloured text works consistently across Windows, macOS, and Linux.

Foreground and Background Colours: Colorama allows you to set the foreground and background colours of the terminal text. You can choose from a range of predefined colours, including common colours like red, green, blue, and yellow. Setting the colour is as simple as wrapping the text you want to colorize with Colorama's provided functions or by using the ANSI escape sequences directly.

Text Styles: Colorama supports different text styles to enhance the appearance of the output. You can apply styles like bold, dim, underline, and italic to the text, giving it a distinctive look. Similar to setting colours, applying text styles is straightforward using Colorama's functions or ANSI escape sequences.

Colorama is a valuable tool for enhancing the visual appeal of command-line applications. It simplifies the process of adding colour and styles to terminal output, making the output more visually appealing and improving readability. Whether it's highlighting important information, organizing output, or simply adding a touch of creativity, Colorama provides an easy and convenient way to work with coloured terminal text in Python.

Algorithm

Convolutional Neural Networks (CNNs) have revolutionized text classification by leveraging their success in computer vision tasks. By adapting the CNN architecture to process sequential data, such as text, they excel at capturing local patterns and global dependencies within textual content. Through convolutional and pooling layers, CNNs extract informative features from text, allowing them to discern subtle nuances and representations. With fully connected layers and appropriate activation functions, these models learn to classify text accurately across multiple categories. CNNs have become a powerful tool in natural language processing, enabling automated analysis, sentiment classification, topic identification, and more, further advancing our understanding of textual data.

The Convolutional Neural Network (CNN) algorithm is primarily known for its effectiveness in computer vision tasks. However, it can also be applied to text classification tasks with certain modifications. Here is how a CNN algorithm is used for text classification:

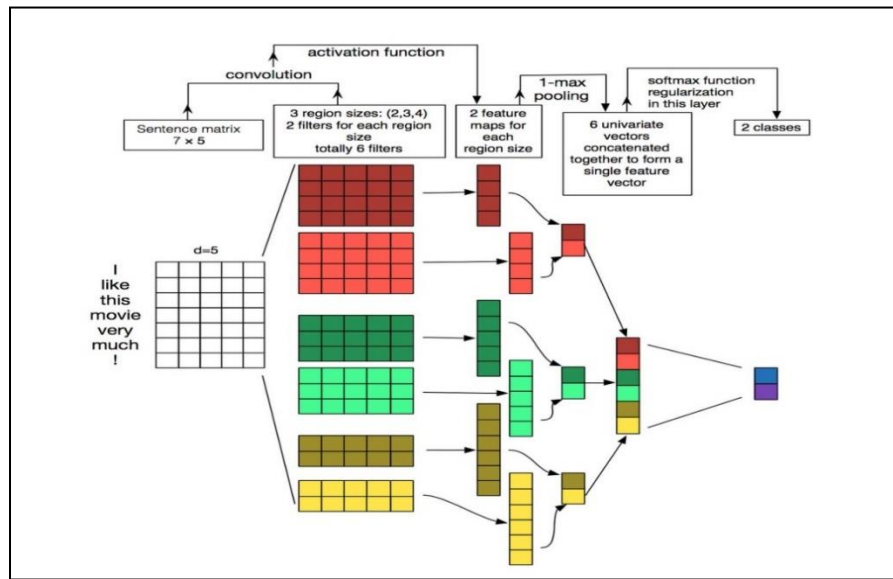


Fig.2. Working process of a CNN algorithm for text classification

Pre processing

Tokenization: The text is split into individual words or tokens.

Vocabulary creation: A vocabulary is created by mapping each unique word to a unique index.

Padding: Text sequences are often of varying lengths, so padding is used to make all sequences of equal length.

Word Embeddings

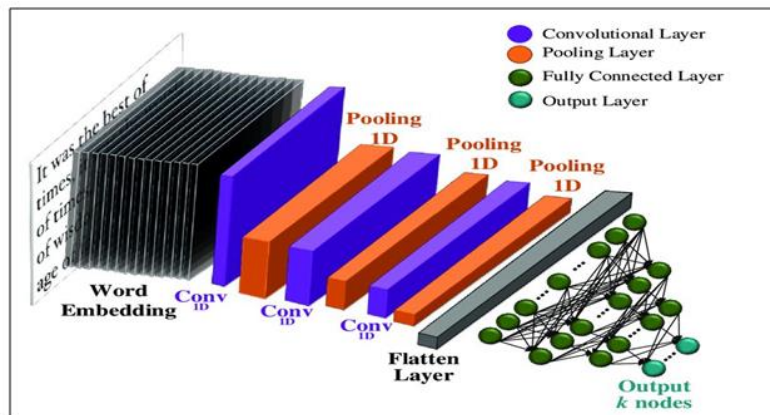


Fig.3. Process of word embedding

Word embeddings are used to represent words in a continuous vector space. Common techniques include Word2Vec, GloVe, or FastText.

Each word in the vocabulary is replaced with its corresponding word embedding vector.

Convolutional Layer:

In computer vision, the convolutional layer applies filters to the input image to extract meaningful features. In text classification, the same concept is applied to the text sequences.

Filters of different sizes are used to capture n-grams (contiguous word sequences) of varying lengths.

The filters move across the text sequences, computing element-wise multiplications and summing the results to obtain feature maps.

The resulting feature maps capture local patterns or features in the text.

Pooling Layer

The pooling layer reduces the dimensionality of the feature maps while preserving the most important information.

Max pooling is commonly used, where the maximum value within each window (e.g., 2x2) of the feature map is selected.

The pooling operation helps to extract the most salient features from the text.

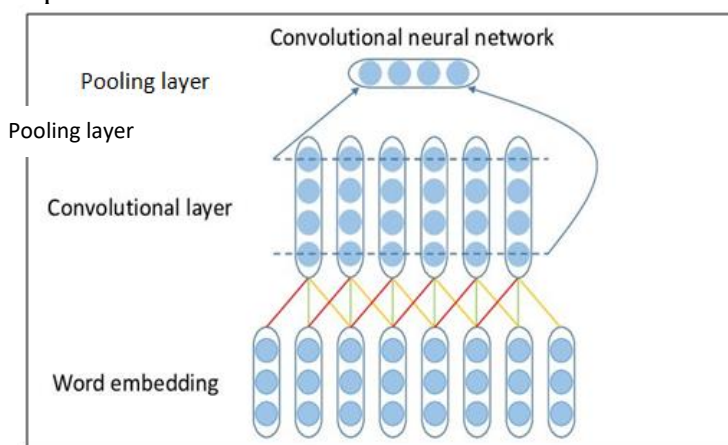


Fig.4. Pooling Layer

Fully Connected Layers

The pooled feature maps are flattened into a 1D vector and passed through one or more fully connected layers.

These layers learn complex relationships between the extracted features and the target labels.

Activation functions (e.g., ReLU, sigmoid, or softmax) are applied to introduce non-linearity.

Output Layer

The output layer consists of one or more units, depending on the number of classes in the classification task.

For binary classification, a single unit with a sigmoid activation function is often used.

For multi-class classification, the number of units matches the number of classes, and a softmax activation function is typically applied.

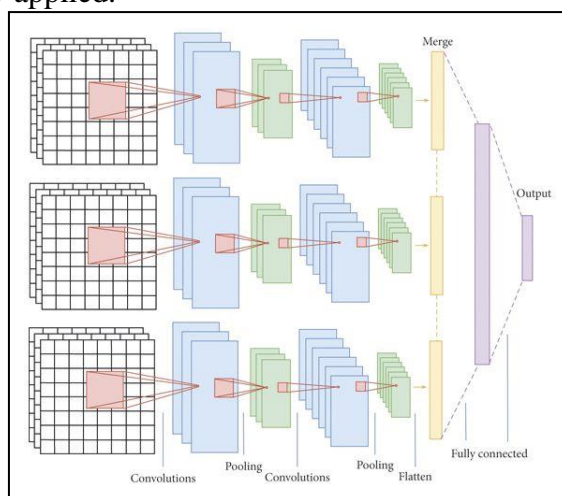


Fig.5. Process of output generation

Training and Optimization:

→ The CNN model is trained using labeled data and during training, the model's weights are updated to minimize a specified loss function, such as cross-entropy loss.

Optimization techniques like stochastic gradient descent (SGD) or Adam are used to iteratively adjust the weights.

The model learns to map the input text to the correct labels by adjusting the weights based on the provided training data.

Prediction:

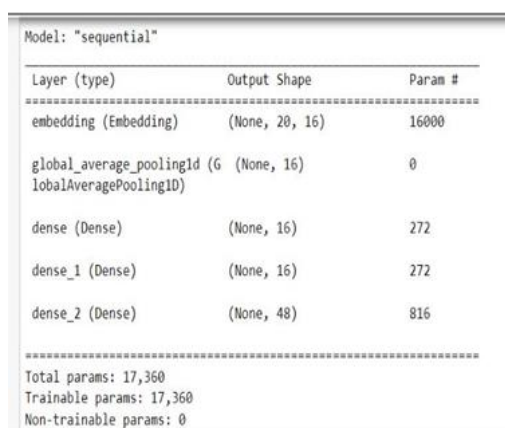
Once the CNN model is trained, it can be used to classify new, unseen text samples.

The input text goes through the same preprocessing steps as during training, including tokenization and word embeddings.

The trained model then assigns a probability distribution over the possible classes, indicating the predicted class for the input text.

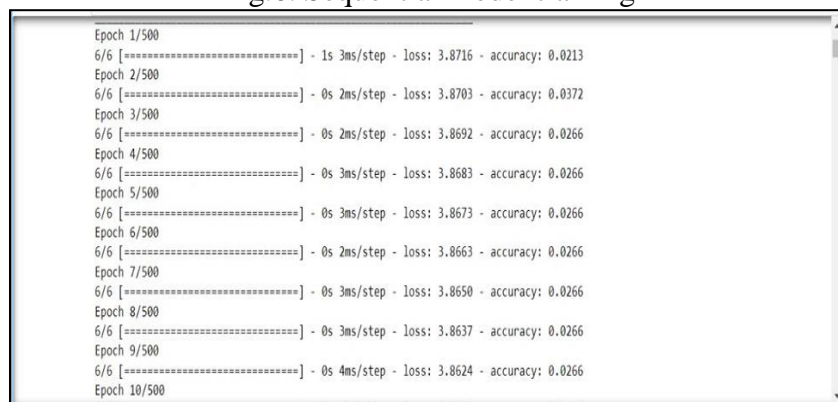
RESULTS

Model Training



Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, 20, 16)	16000
global_average_pooling1d (GlobalAveragePooling1D)	(None, 16)	0
dense (Dense)	(None, 16)	272
dense_1 (Dense)	(None, 16)	272
dense_2 (Dense)	(None, 48)	816
Total params: 17,360		
Trainable params: 17,360		
Non-trainable params: 0		

Fig.6. Sequential model training



Epoch	6/6 [=====]	Time	Loss	Accuracy
Epoch 1/500	6/6 [=====]	1s 3ms/step	loss: 3.8716	accuracy: 0.0213
Epoch 2/500	6/6 [=====]	0s 2ms/step	loss: 3.8703	accuracy: 0.0372
Epoch 3/500	6/6 [=====]	0s 2ms/step	loss: 3.8692	accuracy: 0.0266
Epoch 4/500	6/6 [=====]	0s 3ms/step	loss: 3.8683	accuracy: 0.0266
Epoch 5/500	6/6 [=====]	0s 3ms/step	loss: 3.8673	accuracy: 0.0266
Epoch 6/500	6/6 [=====]	0s 2ms/step	loss: 3.8663	accuracy: 0.0266
Epoch 7/500	6/6 [=====]	0s 3ms/step	loss: 3.8650	accuracy: 0.0266
Epoch 8/500	6/6 [=====]	0s 3ms/step	loss: 3.8637	accuracy: 0.0266
Epoch 9/500	6/6 [=====]	0s 4ms/step	loss: 3.8624	accuracy: 0.0266
Epoch 10/500	6/6 [=====]			

Fig.7. Training the epochs

6/6 [=====] - 0s 2ms/step - loss: 0.0163 - accuracy: 1.0000
Epoch 492/500
6/6 [=====] - 0s 2ms/step - loss: 0.0161 - accuracy: 1.0000
Epoch 493/500
6/6 [=====] - 0s 1ms/step - loss: 0.0160 - accuracy: 1.0000
Epoch 494/500
6/6 [=====] - 0s 2ms/step - loss: 0.0159 - accuracy: 1.0000
Epoch 495/500
6/6 [=====] - 0s 2ms/step - loss: 0.0158 - accuracy: 1.0000
Epoch 496/500
6/6 [=====] - 0s 2ms/step - loss: 0.0156 - accuracy: 1.0000
Epoch 497/500
6/6 [=====] - 0s 2ms/step - loss: 0.0155 - accuracy: 1.0000
Epoch 498/500
6/6 [=====] - 0s 2ms/step - loss: 0.0154 - accuracy: 1.0000
Epoch 499/500
6/6 [=====] - 0s 2ms/step - loss: 0.0153 - accuracy: 1.0000
Epoch 500/500
6/6 [=====] - 0s 2ms/step - loss: 0.0151 - accuracy: 1.0000

Fig.8. Training the epochs (contd.)

Chatbot Interaction

```

Welcome to HealthCheck Services. Please type your query below. To end the conversation, type quit.
User: Hello
1/1 [=====] - 0s 116ms/step
Bot: Hi
User: What should I call you?
1/1 [=====] - 0s 32ms/step
Bot: I'm Medtrix, your bot assistant
User: How old are you?
1/1 [=====] - 0s 16ms/step
Bot: I get reborn after every compilation
User: Can you help?
1/1 [=====] - 0s 17ms/step
Bot: Tell me your problem to assist you
User:

```

Fig.9. Output screen 1

```

Welcome to HealthCheck Services. Please type your query below. To end the conversation, type quit.
User: Is anyone there?
1/1 [=====] - 0s 143ms/step
Bot: Hi there
User: Give me a hand please
1/1 [=====] - 0s 13ms/step
Bot: Tell me your problem to assist you
User: How do you treat a mild Fever?
1/1 [=====] - 0s 8ms/step
Bot: Take acetaminophen (Tylenol, others) or ibuprofen (Advil, Motrin IB, others) in appropriate doses to help bring your temperature down. Drink plenty of fluids, particularly water. Avoid alcohol, tea and coffee as these drinks can cause slight dehydration. Avoid taking cold baths or showers. Get medical help if the fever lasts more than five days in a row.
User: Thank you
1/1 [=====] - 0s 24ms/step
Bot: Glad it helped you
User: Bye
1/1 [=====] - 0s 15ms/step
Bot: See you later
User: quit

```

Fig.10. Output screen 2

```

Welcome to HealthCheck Services. Please type your query below. To end the conversation, type quit.
User: Hi
1/1 [=====] - 0s 64ms/step
Bot: Hi there
User: I want your help
1/1 [=====] - 0s 23ms/step
Bot: Tell me your problem to assist you
User: What to do if I get a sore throat?
1/1 [=====] - 0s 26ms/step
Bot: Make sure you get plenty of rest and drink a lot of fluids. Inhale steam, Run hot water in a sink. Drape a towel over you to trap the steam, and have the person lean over the sink with the water running. Breathe deeply through the mouth and nose for 5 to 10 minutes. Repeat several times a day. Sip chicken broth or warm tea with honey. Don't give honey to a child under 12 months of age. Suck ice cubes, ice lollies or hard sweets but do not give young children anything small and hard to suck because of the risk of choking. In case of serious infection, consume Acetaminophen (Tylenol), Ibuprofen (Advil or Motrin) or Naproxen (Aleve) on appropriate doses.
User: Thank you
1/1 [=====] - 0s 16ms/step
Bot: Happy to help!
User: Also, Which medicine to take if I have a cold?
1/1 [=====] - 0s 16ms/step
Bot: Water, juice, clear broth or warm lemon water with honey is absolutely vital to help 'flush' out the cold, as well as to break down congestion and keep your throat lubricated. Lots of rest is the key treating a cold. You may find you need 12 hours of sleep each night. Decongestant nasal sprays like oxymetazoline (Afrin) can help too. Over-the-counter decongestants, antihistamines and pain relievers, such as ibuprofen or acetaminophen, can reduce some cold symptoms. But use them with caution.
User: That's helpful
1/1 [=====] - 0s 15ms/step
Bot: Any time!
User: Goodbye
1/1 [=====] - 0s 32ms/step
Bot: Feel free to come back again in case you need more help
User: quit

```

Fig.11. Output screen 3

```

Welcome to HealthCheck Services. Please type your query below. To end the conversation, type quit.
User: hello
1/1 [=====] - 0s 48ms/step
Bot: Hi there
User: I need your assistance
1/1 [=====] - 0s 15ms/step
Bot: Tell me your problem to assist you
User: What to do if I get a heart attack?
1/1 [=====] - 0s 31ms/step
Bot: EMERGENCY!!! CONTACT YOUR PHYSICIAN OR CALL 108 IMMEDIATELY
User: Thank you
1/1 [=====] - 0s 35ms/step
Bot: You're most welcome!
User: quit

```

Fig.12. Output screen 4

```

Welcome to HealthCheck Services. Please type your query below. To end the conversation, type quit.
User: Is anyone there?
1/1 [=====] - 0s 60ms/step
Bot: Hi
User: I need your assistance
1/1 [=====] - 0s 16ms/step
Bot: Sure, tell me how can assist you
User: What to do if I get a diabetes(type1 or type2)?
1/1 [=====] - 0s 16ms/step
Bot: EMERGENCY!!! CONTACT YOUR PHYSICIAN OR CALL 108 IMMEDIATELY
User: Ok, thank you
1/1 [=====] - 0s 24ms/step
Bot: Any time!
User: quit

```

Fig.13. Output screen 5

CONCLUSION

In conclusion, the healthcare chatbot project utilizing CNN algorithm, machine learning (ML), and natural language processing (NLP) has demonstrated significant potential in revolutionizing healthcare services. The integration of CNN algorithm in the chatbot's architecture allows it to extract meaningful features from medical images, enabling accurate diagnosis and treatment recommendations. This advancement has the potential to streamline the diagnostic process, reduce human error, and improve patient outcomes.

Furthermore, the integration of ML techniques has empowered the chatbot to learn from vast amounts of medical data, enabling it to make more informed decisions and provide personalized recommendations to users. ML algorithms can analyse patient data, identify patterns, and offer tailored healthcare advice, making the chatbot a valuable tool for both patients and healthcare professionals. This technology has the potential to enhance the efficiency of healthcare systems, optimize resource allocation, and improve overall patient care.

Moreover, the incorporation of NLP in the healthcare chatbot project has enhanced its ability to understand and respond to natural language queries, making interactions more user-friendly and intuitive. The chatbot can interpret complex medical jargon, extract relevant information, and provide accurate and timely responses, improving accessibility to healthcare information. By leveraging NLP techniques, the chatbot becomes a reliable source of medical guidance, empowering individuals to make informed decisions about their health and well-being.

FUTURE SCOPE

We plan to extend this project to accommodate further information about roads such as follows: Enhanced Diagnostic Accuracy: With advancements in AI, machine learning, and natural language processing, healthcare chatbots utilizing CNN algorithm, ML, and NLP will experience significant improvements in their diagnostic capabilities. This will result in more accurate and reliable diagnoses, reducing misdiagnoses and enabling early detection of diseases.

Personalized and Efficient Care: Chatbots are poised to become integral parts of healthcare systems, working alongside healthcare providers to deliver personalized and efficient care. By leveraging ML and NLP, chatbots can analyse patient data, medical history, and symptoms to provide tailored recommendations, treatment plans, and preventive measures, optimizing patient care and outcomes.

Integration with Telemedicine: The integration of chatbots with telemedicine platforms is a promising future prospect. This integration will enable seamless remote consultations, allowing patients to interact with chatbots to receive preliminary assessments, schedule appointments, and access healthcare services from the comfort of their homes. This advancement will greatly enhance healthcare accessibility, particularly for individuals in remote areas or with limited mobility.

Improved Patient Engagement and Satisfaction: As NLP technology advances, chatbots will become more proficient in understanding and responding to natural language queries, fostering improved patient engagement and satisfaction. The ability of chatbots to interpret medical jargon, provide empathetic responses, and address patients' concerns will create a more user-friendly and interactive healthcare experience, promoting better communication and trust between patients and healthcare systems.

REFERENCES

- [1] Hiba Hussain¹, Komal Aswani², Mahima Gupta³, Dr. G.T.Thampi⁴, "Implementation of Disease Prediction Chatbot and Report Analyzer using the Concepts of NLP, Machine Learning and OCR," IRJET, Apr 2020.
- [2] Oh, K.-J., D. Lee, B. Ko, and H.-J. Choi, A chatbot for psychiatric counseling in mental healthcare service based on emotional dialogue analysis and sentence generation. In 2017 18th IEEE International Conference on Mobile Data Management (MDM). IEEE, 2017.
- [3] Kowatsch, T., M. Nißen, C.-H. I. Shih, D. Rügger, D. Volland, A. Filler, F. Künzler, F. Barata, D. Büchter, B. Brogle, et al. (2017). Text-based healthcare chatbots supporting patient and health professional teams: preliminary results of a randomized controlled trial on childhood obesity.
- [4] Lin Ni(B), Chenhao Lu, Niu Liu, and Jiamou Liu, "MANDY: Towards a Smart Primary Care Chatbot Application", SPRINGER, 2017.
- [5] Divya, S., V. Indumathi, S. Ishwarya, M. Priyasankari, and S. K. Devi (2018). A self-diagnosis medical chatbot using artificial intelligence. *Journal of Web Development and Web Designing*, 3(1), 1–7.
- [6] Chung, K. and R. C. Park (2019). Chatbot-based healthcare service with a knowledge base for cloud computing. *Cluster Computing*, 22(1), 1925–1937.
- [7] Ahmed Fadil, Gianluca Schiavo, "Design for healthcare chatbot" Arxiv, 2019.
- [8] Beaudry, J., A. Consigli, C. Clark, and K. J. Robinson (2019). Getting ready for adult healthcare: Designing a chatbot to coach adolescents with special health needs through the transitions of care. *Journal of pediatric nursing*, 49, 85–91.
- [9] Kavitha, B. and C. R. Murthy (2019). Chatbot for healthcare system using artificial intelligence.
- [10] Kandpal, P., K. Jasnani, R. Raut, and S. Bhorge, Contextual chatbot for healthcare purposes (using deep learning). In 2020 Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4). IEEE, 2020.
- [11] A. F. Ur Rahman Khilji, S. R. Laskar, P. Pakray, R. A. Kadir, M. S. Lydia and S. Bandyopadhyay, "HealFavor: Dataset and A Prototype System for Healthcare ChatBot," 2020 International Conference on Data Science, Artificial Intelligence, and Business Analytics (DATABIA), 2020.