Introduction to AI COM727

Medical Chatbot

Mehrdad Madadi

Student number: 102607756

Group:

Arash Mehrdad

Vijay Rawat

Parteek Sharma

Professors:

Dr. Shakeel Ahmad

Dr. Kashif Talpur

SOLENT UNIVERSITY SOUTHAMPTON

2024 - 2025

Title page

Contents

Introduction

Artificial Intelligence (AI) chatbots have transformed communication by enabling personalized interactions across various platforms. These systems use natural language processing (NLP) and machine learning to interpret user input, generate meaningful responses, and adapt based on context (Adamopoulou & Moussiades, 2020). Their versatility extends to diverse applications, including customer support, education, and healthcare, where they enhance accessibility and efficiency (Abu Shawar & Atwell, 2007).

AI chatbots are increasingly being integrated into the healthcare industry to streamline patient interactions, assist in diagnostics, and provide basic medical advice and offer preliminary guidance, such as whether to seek medical attention or follow self-care procedures (Lippi et al., 2020). With their ability to process vast amounts of medical data quickly, chatbots can help reduce the burden on healthcare professionals by addressing common queries and performing initial assessments (Zhou et al., 2020). As these systems improve, they may play an even larger role in telemedicine and patient monitoring.

While AI chatbots in healthcare offer numerous benefits, their implementation raises several legal and ethical concerns. One of the primary issues is the potential for AI to provide incorrect or harmful medical advice, which could lead to legal liability for healthcare providers or developers (Vayena et al., 2018). Furthermore, chatbots can inadvertently violate patient privacy by mishandling sensitive health data, leading to breaches of confidentiality and regulatory violations under laws such as the GDPR or HIPAA (Garrido et al., 2019). Ethical concerns also arise regarding the lack of human empathy in AI interactions, which can affect patient trust and satisfaction, especially in sensitive contexts like mental health or end-of-life care (Lloyd et al., 2020). These concerns highlight the need for clear legal frameworks and ethical guidelines to ensure the responsible use of AI in healthcare.

Social implications of AI chatbots in healthcare also require careful consideration. One concern is the risk of exacerbating health disparities, as individuals without access to technology or digital literacy may be excluded from these services (Sicari et al., 2019). Additionally, the use of AI in healthcare may result in job displacement for medical professionals, particularly those in roles like administrative support or initial consultation, creating economic challenges (Challen et al., 2019). These social and professional challenges emphasize the need for equitable AI solutions in healthcare.

Need for your prototype

As mentioned in the introduction, the AI chatbot developed to collect symptoms from users and provide probable diagnoses and advice is essential for improving accessibility to healthcare. By offering preliminary guidance on whether professional medical help is necessary, the chatbot helps streamline patient triage, enabling quicker decision-making and reducing strain on healthcare systems (Chung et al., 2020). This tool can empower users to make informed choices about their health, ensuring timely intervention and optimal use of healthcare resources.

Statement of the problem

The healthcare system faces significant challenges related to accessibility, efficiency, and the burden on healthcare professionals. Many individuals, particularly in underserved or rural areas, struggle to access timely medical advice and may delay seeking care due to long wait times or limited resources (Sicari et al., 2019). Additionally, healthcare systems often become overwhelmed with routine consultations, limiting professionals' ability to focus on more complex cases (Deo, 2015). Current solutions for symptom checking and patient triage are often inadequate, leading to inefficiencies and sometimes incorrect self-diagnosis by users (Chung et al., 2020). The lack of timely, easily accessible medical advice can result in unnecessary hospital visits or, conversely, the failure to seek help for serious conditions (Vayena et al., 2018). An AI chatbot that collects symptoms and provides probable diagnoses, along with guidance on whether to see a doctor, could address these gaps, offering immediate assistance and ensuring users make informed healthcare decisions in a timely manner.

Aims and Objectives

The aim of this project is to develop an AI chatbot that collects user symptoms, provides probable diagnoses, offers basic medical advice, and suggests medical attention when needed, improving healthcare accessibility and decision-making.

Objectives:

- To integrate a medical database for accurate disease suggestions based on symptoms.
- Provide general recommendations based on symptom input
- To suggest when users should seek medical attention.

Proposed Solution

The proposed solution is to train an AI chatbot that collects user-reported symptoms, provides probable diagnoses, offers recommendations, and directs users to seek professional medical advice when necessary. A supervised learning model will be developed using the Mendeley Dataset and the DailyDialog Dataset.

Prototype Design

Our Medical Chatbot will use natural language processing and neural networks to deliver accurate response. The data will be a JSON file , which will train the model. It will employ the Natural Language Toolkit (NLTK) for input processing, and utilise a neural network model, which will include multiple layers. Neural networks are made up from different layers which are used to analyse data to find patterns. The different layers takes the inputs and calculates the sum of weights. The activation function then takes this sum and outputs the result (Sharma, Rai, and Dev 2012) The dataset used includes intents such as symptoms and responses tailored for healthcare use cases.

Prototype Development and AI Algorithms used

Initially, our group considered using more complex models for training the chat bot. However, we understood that this would require very powerful computers so we used the model provided by the teacher.

To develop this chat bot we used two data bases one for medical queries and the other for normal dialogs (like hello how are you etc.). We converted them into JSON files for structuring intents

(intents.json). Then we used NLTK for processing and normalising the language data (Tokenisation and lemmatisation were applied to standardise text input and Stop words were removed to enhance data quality). Sequential neural network with Input layer with 128 nodes and ReLU activation, Hidden layer with 64 nodes and dropout for regularization and Output layer with softmax activation for classification for layers was trained on this JSON file for 40 epochs using Stochastic Gradient Descent (SGD) with a learning rate of 0.01. At the end we got **96.06%** accuracy, with a loss of **0.1046.**

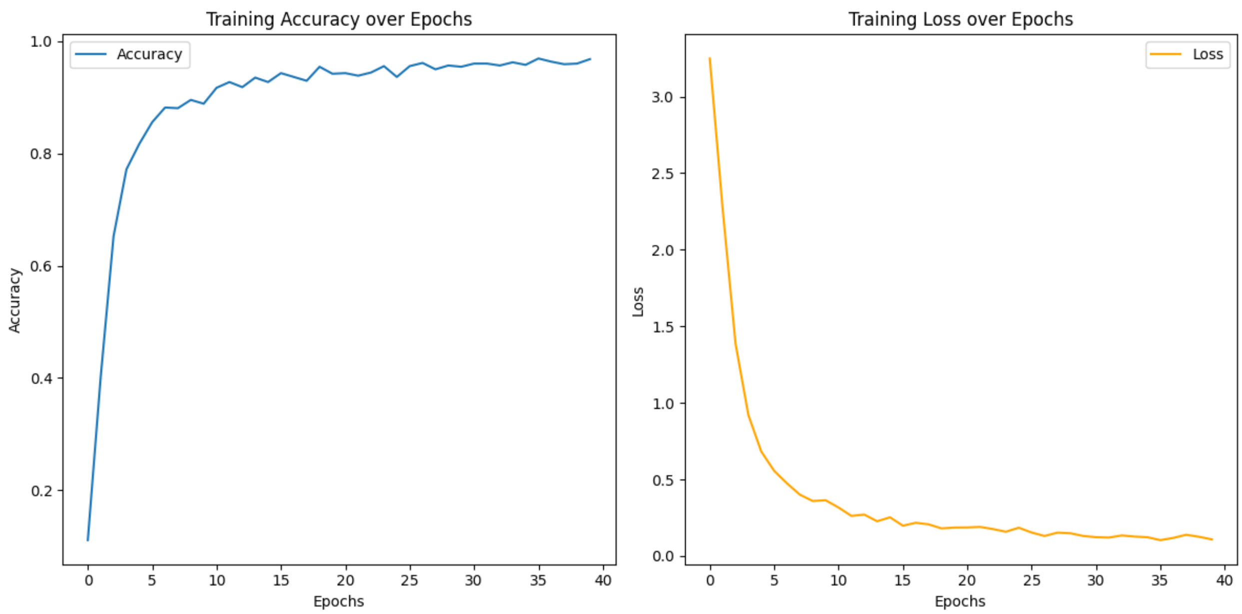

Figure 1 – accuracy and loss over epochs

Table 1 - Packages and Libraries used in the medical chatbot

Library/Package	Use	Used in
JSON	To read from JSON file	Training and chatbot
NLTK	To preprocess text data	Training and chatbot
NumPy	To manipulate data	Training and chatbot
Pickle	To save and load data	Training and chatbot
TensorFlow	To build and train the model	Training and chatbot
Random	To shuffle training data and retrieve random responses	Training and chatbot

Intents.JSON

This file is made from various **intents**, each identified by a unique tag. Each intent has its own **patterns** (examples of user input) and **responses** (chatbot output). When we run the chatbot it uses this dataset to classify user inputs and give relevant response.

```
{
    "intents": [
        {
            "tag": "greetings",
            "patterns": [
                "hello",
                "hey",
                "hi",
                "good day",
                "Greetings",
                "what's up",
                "how is it going"
            ],
            "responses": [
                "Hello!",
                "Hey!",
                "What can I do for you?"
            ]
        },
        {
            "tag": "goodbye",
            "patterns": [
                "See ya",
                "see you later",
                "bye",
                "goodbye",
                "have a nice day",
                "bye bye"
            ],
            "responses": [
                "Goodbye!",
                "See you later.",
                "Please come back soon."
            ]
        },
```

Figure 2 - JSON file code (Greetings)

```
{
    "tag": "bronchial_asthma",
    "patterns": [
        "fatigue, cough, high fever, breathlessness, family history",
        "I have symptoms related to bronchial asthma",
        "I think I have bronchial asthma",
        "My symptoms point to bronchial asthma",
        "What can I do for bronchial asthma?"
    ],
    "responses": [
        "The diagnosis is Bronchial Asthma. For asthma, quick-relief inhalers like salbutamol open airways, and corticosteroids ca
    ]
},
{
    "tag": "migraine",
    "patterns": [
        "acidity, indigestion, headache, blurred and distorted vision, excessive hunger, stiff neck, depression, irritability, vis
        "I have symptoms related to migraine",
        "I think I have migraine",
        "My symptoms point to migraine",
        "What can I do for migraine?"
    ],
    "responses": [
        "The diagnosis is Migraine. For migraines, triptans can help stop an attack, while beta-blockers are often used to prevent
    ]
```

Figure 3 - JSON file code example (asthma and migraine)

Medical_Chatbot.ipynb

This file contains both training and chatbot. First part is training where we preprocess the data by tokenizing, lemmatizing, and filtering out stop words.

```python
with open('intents.json') as file:
    intents = json.load(file)

# Preprocess data
words = []
classes = []
documents = []
ignore_letters = ['?', '!', '.', '/', '@']

for intent in intents['intents']:
    for pattern in intent['patterns']:
        word_list = nltk.word_tokenize(pattern)  # Tokenize the sentence
        words.extend(word_list)
        documents.append((word_list, intent['tag']))
        if intent['tag'] not in classes:
            classes.append(intent['tag'])

# Remove stopwords and lemmatize words
stop_words = set(stopwords.words('english'))
words = [
    lemmatizer.lemmatize(word.lower())
    for word in words
    if word not in ignore_letters and word.lower() not in stop_words
]
words = sorted(list(set(words)))
classes = sorted(list(set(classes)))

# Save words and classes for inference
with open('words.pkl', 'wb') as f:
    pickle.dump(words, f)
with open('classes.pkl', 'wb') as f:
    pickle.dump(classes, f)
```

Figure 4 - Medical_Chatbot.ipynb file (preprocessing)

Then create a bag-of-words model to train a neural network using TensorFlow. This model predicts the user's intent from input text and provides relevant responses.

```
# Prepare training data
training = []
output_empty = [0] * len(classes)

for document in documents:
    bag = []
    word_patterns = document[0]
    word_patterns = [lemmatizer.lemmatize(word.lower()) for word in word_patterns]
    for word in words:
        bag.append(1) if word in word_patterns else bag.append(0)

    output_row = list(output_empty)
    output_row[classes.index(document[1])] = 1
    training.append([bag, output_row])

# Shuffle and convert training data to NumPy arrays
random.shuffle(training)
training = np.array(training, dtype=object)

train_x = list(training[:, 0])
train_y = list(training[:, 1])

# Build the model
model = Sequential()
model.add(Dense(128, input_shape=(len(train_x[0]),), activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(64, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(len(train_y[0]), activation='softmax'))

# Compile the model using SGD optimizer
sgd = SGD(learning_rate=0.01, decay=1e-6, momentum=0.9, nesterov=True)
model.compile(loss='categorical_crossentropy', optimizer=sgd, metrics=['accuracy'])

# Train the model
print("Training the model...")
hist = model.fit(np.array(train_x), np.array(train_y), epochs=40, batch_size=5, verbose=1)
```

Figure 4 - Medical_Chatbot.ipynb file (building and training the model)

The chatbot is designed to operate interactively through the command line, identifying user symptoms and offering advice or remedies for various medical conditions based on predefined responses.

```python
def clean_up_sentence(sentence):
    sentence = sentence.lower()
    sentence_words = nltk.word_tokenize(sentence)
    sentence_words = [lemmatizer.lemmatize(word) for word in sentence_words]
    return sentence_words

# Convert input to bag-of-words format
def bow(sentence, words):
    sentence_words = clean_up_sentence(sentence)
    bag = [0] * len(words)
    for s in sentence_words:
        for i, w in enumerate(words):
            if w == s:
                bag[i] = 1
    return np.array(bag)

# Predict the class of the input
def predict_class(sentence):
    bow_vector = bow(sentence, words)
    try:
        res = model.predict(np.array([bow_vector]))[0]
    except Exception as e:
        print(f"Prediction error: {e}")
        return []
    ERROR_THRESHOLD = 0.1
    results = [[i, r] for i, r in enumerate(res) if r > ERROR_THRESHOLD]
    results.sort(key=lambda x: x[1], reverse=True)
    return [{"intent": classes[r[0]], "probability": str(r[1])} for r in results]

# Get response based on predicted class
def get_response(intents_list, intents_json):
    if not intents_list:
        return "I'm sorry, I couldn't understand. Please try again."
    tag = intents_list[0]['intent']
    for intent in intents_json['intents']:
        if intent['tag'] == tag:
            return random.choice(intent['responses'])
    return "I'm sorry, I couldn't find a response for that."

# Run the chatbot
def chatbot():
    print("Chatbot is ready! Type 'exit' to stop.")
    while True:
        user_input = input("You: ").strip()
        if user_input.lower() == "exit":
            print("Goodbye!")
            break
```

Figure 5 - Medical_Chatbot.ipynb file (chatbot deployment)

Evaluation

In the training part with each epoch the code evaluates the accuracy of the model and prints it.

```
177/177 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.0690 - loss: 3.3890
Epoch 2/40
177/177 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - accuracy: 0.3338 - loss: 2.5989
Epoch 3/40
177/177 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.6287 - loss: 1.4735
Epoch 4/40
177/177 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.7290 - loss: 1.0300
Epoch 5/40
177/177 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.8441 - loss: 0.6779
Epoch 6/40
177/177 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.8331 - loss: 0.6022
Epoch 7/40
177/177 ━━━━━━━━━━━━━━━━━━━━ 1s 1ms/step - accuracy: 0.8776 - loss: 0.4075
Epoch 8/40
177/177 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.8995 - loss: 0.3548
Epoch 9/40
177/177 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.8954 - loss: 0.3683
Epoch 10/40
177/177 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.9238 - loss: 0.2718
```

Figure 6 - Medical_Chatbot.ipynb file (accuracy and loss over each epoch)

Limitation

One of the limitations is that many of the symptoms have different synonyms and chatbot may not recognize all of them, and creating a chatbot that is so powerful and can do so requires a much powerful model and much better system. Another one can be that a disease has multiple symptoms and user should try to give as many as possible to get better response. Finally limited dataset coverage prevented the chatbot's ability to recognize rare symptoms.

Conclusion

The medical chatbot demonstrates the potential of AI in healthcare by providing preliminary diagnoses, medical advice, and guidance on seeking professional help. Despite limitations like handling synonyms, reliance on complete user input, and dataset coverage, the prototype marks a significant step toward improving healthcare accessibility and reducing the burden on professionals. With further refinement, this technology could play a transformative role in telemedicine and patient support.

Reference List

Adamopoulou, E. and Moussiades, L., 2020. An overview of chatbot technology. *IFIP Advances in Information and Communication Technology*, 584, pp.373–383. Available at: https://doi.org/10.1007/978-3-030-49186-4_31 [Accessed 31 Dec. 2024].

Abu Shawar, B. and Atwell, E., 2007. Chatbots: Are they really useful? LDV Forum - Special Issue on Learning-Oriented Natural Language Processing, 22(1), pp.29–49.

Lippi, G., Sanchis-Gomar, F. and Cervellin, G., 2020. Chatbots in health care: The promise and the challenge. The Lancet Digital Health, 2(11), pp.e578–e579. Available at: https://doi.org/10.1016/S2589-7500(20)30197-7 [Accessed 31 Dec. 2024].

Zhou, L., Chou, Y. and Fu, Y., 2020. Artificial intelligence in health care: A review. BMC Medical Informatics and Decision Making, 20(1), pp.1–9. Available at: https://doi.org/10.1186/s12911-020-1073-3 [Accessed 31 Dec. 2024].

Vayena, E., Blasimme, A., and Winickoff, D.E., 2018. Machine learning in health care: A critical appraisal of its legal and ethical implications. JAMA, 320(6), pp.507-508. Available at: https://doi.org/10.1001/jama.2018.10075 [Accessed 31 Dec. 2024].

Garrido, T., et al., 2019. Privacy and security concerns of AI in healthcare. Journal of Medical Internet Research, 21(4), e13535. Available at: https://doi.org/10.2196/13535 [Accessed 31 Dec. 2024].

Lloyd, J., et al., 2020. Ethical considerations for AI in healthcare: The role of human empathy in the development of AI systems. AI and Ethics, 1(2), pp.219-230. Available at: https://doi.org/10.1007/s43681-020-00014-4 [Accessed 31 Dec. 2024].

Sicari, S., et al., 2019. AI in healthcare: Addressing the digital divide. Frontiers in Public Health, 7, pp.245. Available at: https://doi.org/10.3389/fpubh.2019.00245 [Accessed 31 Dec. 2024].

Challen, R., et al., 2019. Artificial intelligence, bias and clinical safety. BMJ Quality & Safety, 28(5), pp.347-351. Available at: https://doi.org/10.1136/bmjqs-2018-008544 [Accessed 31 Dec. 2024].

Chung, J., et al., 2020. AI in healthcare: A brief overview of applications and challenges. *Healthcare*, 8(1), pp.12-19. Available at: https://doi.org/10.3390/healthcare8010012 [Accessed 31 Dec. 2024].

Deo, R.C., 2015. Machine learning in medicine. *Circulation*, 132(20), pp.1920-1930. Available at: https://doi.org/10.1161/CIRCULATIONAHA.115.001593 [Accessed 31 Dec. 2024].

Sharma, V., Rai, S. & Dev, A., 2012. A comprehensive study of artificial neural networks. *International Journal of Advanced Research in Computer Science and Software Engineering*, 2(10), pp. 1-6.