**Development of a Medical Diagnosis Chatbot Using Natural Language Processing and Machine Learning Techniques (Prototype)**

Student Name

**Muhammad Rizwan Tahir**

Supervisor Name

**Basma Abdulaimma**

A thesis submitted in partial fulfilment of the requirements of Liverpool John Moores University

Abstract

This research paper presents the development of a medical diagnosis chatbot using natural language processing (NLP) and machine learning techniques. The chatbot aims to provide quick and accurate medical support, making healthcare more accessible and timelier. With the increasing demand for healthcare services and the advancements in AI technologies, chatbots have emerged as a promising solution to address the challenges faced in the healthcare industry. The paper begins by discussing the motivations behind building medical diagnostic chatbots. It highlights the need for accessible and timely medical support, especially in remote areas or during emergencies when immediate access to healthcare professionals may be limited. The evolution of human-computer interactions in healthcare is also explored, emphasizing the shift towards more personalized and patient-centric care. The technological foundations behind modern chatbots are then examined. The paper delves into NLP techniques and machine learning models that form the backbone of chatbot development. These include LSTM (family of RNNs) and Transformers, which have shown great potential in understanding and generating human-like responses. The dataset, MedDialog-EN, from Kaggle is also discussed as a valuable resource for training the chatbot. The research objective is to build an NLP-based chatbot using the MedDialog-EN dataset and sequence-to-sequence models. The methodology section outlines the steps involved in model development, including data collection, preprocessing, and training. The paper highlights the use of LSTM (family of RNNs) and Transformers in capturing the context and generating accurate responses. Among these, the Encoder-Decoder LSTM model performed well, achieving a loss of 0.074, an accuracy of 99.05%, and a ROUGE-1 F1 score of 0.976. The Facebook/BART-base model also showcased promising results with an accuracy of 92.09% and a ROUGE-1 F1 score of 0.951. The deployment of the chatbot through a mobile application is also described, showcasing the integration of AI technologies into user-friendly interfaces. The paper also discusses the limitations of the project. It acknowledges the challenges faced in developing a fully functional and reliable medical chatbot, such as the need for more diverse and comprehensive datasets, the exploration of different NLP techniques, and the improvement of the user experience in future work. Overall, this research paper contributes to the knowledge on medical chatbots by presenting the development of a chatbot using NLP and machine learning techniques.

Keywords

Medical diagnosis chatbot, Natural language processing (NLP), Machine learning (ML), Deep learning, Healthcare, Human-computer interaction, NLP techniques, Sequence-to-sequence models, MedDialog-EN dataset, Recurrent Neural Networks (RNNs), Transformers, Mobile application, React Native, Artificial Intelligence (AI) technologies, User experience, Data preprocessing, Model development, Model deployment, Limitations, Future work, Virtual doctors, Pre-trained models, Generative Pre-trained Transformer (GPT) models, Bidirectional Encoder Representations from Transformers (BERT), COVID-19, Conversational agents, Support Vector Machines (SVM), Long Short-Term Memory (LSTM), GPT-2, BART (Bidirectional and Auto-Regressive Transformers), T5 (Text-to-Text Transfer Transformer), Ethical considerations, Chatbot platforms, Amazon Alexa, Google Assistant, IBM Watson, Doctor-patient conversations, Tokenization, Stop-word removal, Lemmatization, Sentiment analysis, Artificial Intelligence Markup Language (AIML).

Acknowledgement

I want to express my sincere gratitude to Professor Basma Abdulaimma, who served as my research supervisor, for her direction, support, and help throughout the project. I would also like to thank Liverpool John Moores University for letting me use their library and lab facilities. I also want to express my gratitude to my friends for their helpful suggestions and ideas that drove and inspired this study. Not least of all, I want to thank my parents for their continuous support and inspiration during my studies.

Table of Contents

[**1.** **Introduction** 3](#_Toc145643532)

[**1.1. The Rise of Artificial Intelligence in Healthcare** 3](#_Toc145643533)

[**1.2. Motivation for Medical Diagnostic Chatbots** 3](#_Toc145643534)

[**1.3. The Evolution of Human-Computer Interactions in Healthcare** 3](#_Toc145643535)

[**1.4. The Modern Era: Chatbots Transforming Healthcare** 4](#_Toc145643536)

[**1.5. Technological Foundations Behind Modern Chatbots** 4](#_Toc145643537)

[**1.6. Research Objective and Methodology** 5](#_Toc145643538)

[**1.7. Structure of the Dissertation** 6](#_Toc145643539)

[**2.** **Background and literature review** 7](#_Toc145643540)

[**2.1. Chatbots and Potential in Medical Diagnosis** 7](#_Toc145643541)

[**2.2. Historical Roots of Chatbot Technology** 7](#_Toc145643542)

[**2.3. Revolutionary Advancements with Virtual Personal Assistants** 8](#_Toc145643543)

[**2.4. The Impact of Pre-trained Models in Modern Chatbot Development** 8](#_Toc145643544)

[**2.5. The Evolution and Significance of Medical Chatbots** 9](#_Toc145643545)

[**2.6. Literature Review: Artificial Intelligence in Healthcare** 9](#_Toc145643546)

[**3.** **Considerations** 13](#_Toc145643547)

[**3.1. Legal considerations** 13](#_Toc145643548)

[**3.2.** **Social considerations** 14](#_Toc145643549)

[**3.3.** **Ethical considerations** 14](#_Toc145643550)

[**3.4.** **Technical considerations** 14](#_Toc145643551)

[**3.5.** **Data considerations** 14](#_Toc145643552)

[**3.6.** **Machine learning considerations** 15](#_Toc145643553)

[**3.7.** **User considerations** 15](#_Toc145643554)

[**4.** **System Design and resources** 15](#_Toc145643555)

[**4.1.** **System Design** 15](#_Toc145643556)

[**4.1.1.** **Data Collection and Analysis** 16](#_Toc145643557)

[**4.1.2.** **Data Preparation** 16](#_Toc145643558)

[**4.1.3.** **Model Training** 16](#_Toc145643559)

[**4.1.4.** **Model Evaluation** 16](#_Toc145643560)

[**4.1.5.** **Model Validation** 16](#_Toc145643561)

[**4.1.6.** **Model Analysis** 16](#_Toc145643562)

[**4.1.7.** **Version Control** 17](#_Toc145643563)

[**4.1.8.** **CI/CD (Continuous Integration / Continuous Deployment) Pipeline** 17](#_Toc145643564)

[**4.1.9.** **Model Registry** 17](#_Toc145643565)

[**4.1.10.** **Model Serving** 17](#_Toc145643566)

[**4.1.11.** **Mobile Application** 17](#_Toc145643567)

[**4.1.12.** **Backend Development** 17](#_Toc145643568)

[**4.1.13.** **Docker Containerization** 17](#_Toc145643569)

[**4.1.14.** **Database Management** 17](#_Toc145643570)

[**4.1.15.** **Model Monitoring** 17](#_Toc145643571)

[**4.2.** **Available Hardware Resources** 18](#_Toc145643572)

[**5.** **Data collection and data analysis** 18](#_Toc145643573)

[**5.1.** **Data collection** 18](#_Toc145643574)

[**5.2.** **Data description** 19](#_Toc145643575)

[**5.3.** **Advantages of MedDialog-En Dataset** 19](#_Toc145643576)

[**5.4.** **Limitations of MEdDialog-En Dataset** 19](#_Toc145643577)

[**5.5.** **Exploratory Data Analysis** 19](#_Toc145643578)

[**5.5.1.** **Dataset Basic Information** 20](#_Toc145643579)

[**5.5.2.** **Text Data Descriptive Analysis** 20](#_Toc145643580)

[**6.** **Methodology** 22](#_Toc145643581)

[**6.1.** **General Data pre-processing** 23](#_Toc145643582)

[**6.1.1.** **Remove ID Column** 23](#_Toc145643583)

[**6.1.2.** **Remove Duplicate Dialogues** 23](#_Toc145643584)

[**6.1.3.** **Convert to Lowercase** 23](#_Toc145643585)

[**6.1.4.** **Remove Irrelevant Characters** 23](#_Toc145643586)

[**6.1.5.** **Handle Contractions** 24](#_Toc145643587)

[**6.1.6.** **Spell Correction** 24](#_Toc145643588)

[**6.1.7.** **Remove URLs** 24](#_Toc145643589)

[**6.1.7.** **Handle Stop-words** 24](#_Toc145643590)

[**6.1.8.** **Handle Lemmatization** 24](#_Toc145643591)

[**6.2.** **Model Architecture Selection and Justification** 24](#_Toc145643592)

[**6.2.1.** **Encoder-Decoder LSTM (Long Short-Term Memory)** 25](#_Toc145643593)

[**6.2.2.** **GPT-2 (Generative Pre-trained Transformer)** 26](#_Toc145643594)

[**6.2.3.** **BART (Bidirectional and Auto-Regressive Transformers)** 26](#_Toc145643595)

[**6.2.4.** **T5 (Text-to-Text Transfer Transformer)** 26](#_Toc145643596)

[**6.3.** **Model Specific Pre-processing** 27](#_Toc145643597)

[**6.3.1.** **Encoder-Decoder LSTM (Long Short-Term Memory)** 27](#_Toc145643598)

[**6.3.2** **GPT-2 (Generative Pre-trained Transformer)** 28](#_Toc145643599)

[**6.3.3.** **BART (Bidirectional and Auto-Regressive Transformers)** 28](#_Toc145643600)

[**6.3.4.** **T5 (Text-to-Text Transfer Transformer)** 29](#_Toc145643601)

[**6.4.** **Model Training and Hyperparameter Selection** 29](#_Toc145643602)

[**6.4.1** **Encoder-Decoder LSTM (Long Short-Term Memory)** 30](#_Toc145643603)

[**6.4.2.** **Pre-trained models (GPT-2, BART and T5)** 30](#_Toc145643604)

[**6.5.** **Model Evaluation** 31](#_Toc145643605)

[**6.5.1.1.** **Accuracy** 31](#_Toc145643606)

[**6.5.1.2.** **Loss** 31](#_Toc145643607)

[**6.5.1.3.** **Perplexity** 31](#_Toc145643608)

[**6.5.1.4.** **ROUGE (Recall-Oriented Understudy for Gisting Evaluation) score** 31](#_Toc145643609)

[**7.** **Results with Mobile Application** 32](#_Toc145643610)

[**7.1.** **Backend and Model Serving** 32](#_Toc145643611)

[**7.2.** **Securing the Flask Service with Docker (Containerization)** 32](#_Toc145643612)

[**7.3.** **Frontend (Mobile Application)** 32](#_Toc145643613)

[**7.3.1.** **User chat** 33](#_Toc145643614)

[**8.** **Limitations** 33](#_Toc145643615)

[**8.1.** **Time constraints** 33](#_Toc145643616)

[**8.2.** **Hardware constraints** 33](#_Toc145643617)

[**8.3.** **Dataset Limitations for comprehensive medical chatbot** 33](#_Toc145643618)

[**9.** **Discussion** 33](#_Toc145643619)

[**10.** **Future work** 34](#_Toc145643620)

[**11.** **Conclusion** 35](#_Toc145643621)

[**12.** **References** 36](#_Toc145643622)

[**13.** **Appendices** 39](#_Toc145643623)

[**13.1. Figure 1** 39](#_Toc145643624)

[**13.2. Figure 2** 39](#_Toc145643625)

[**13.3. Figure 3** 39](#_Toc145643626)

[**13.4. Figure 4** 40](#_Toc145643627)

[**13.5. Figure 5** 40](#_Toc145643628)

[**13.6. Figure 6** 40](#_Toc145643629)

[**13.7. Figure 7** 41](#_Toc145643630)

[**13.8. Figure 8** 41](#_Toc145643631)

[**13.9. Figure 9** 41](#_Toc145643632)

[**13.10. Figure 10** 42](#_Toc145643633)

[**13.11. Figure 11** 42](#_Toc145643634)

[**13.12. Figure 12** 42](#_Toc145643635)

[**13.13. Figure 13** 43](#_Toc145643636)

[**13.14. Figure 14** 43](#_Toc145643637)

[**13.15. Figure 15** 44](#_Toc145643638)

[**13.16. Figure 16** 44](#_Toc145643639)

[**13.17. Figure 17** 44](#_Toc145643640)

[**13.18. Figure 18** 45](#_Toc145643641)

[**13.19. Figure 19** 45](#_Toc145643642)

[**13.20. Figure 20** 46](#_Toc145643643)

[**13.21. Figure 21** 46](#_Toc145643644)

# **Introduction**

## **1.1. The Rise of Artificial Intelligence in Healthcare**

Artificial intelligence (AI) has made tremendous progress in recent years, bridging industries, and fundamentally changing the healthcare sector. The revolutionary potential of AI has a tremendous impact on the intricate network of healthcare services, from early detection and diagnosis to personalised therapy and complete patient care (Davenport and Kalakota, 2019). The AI has shown to be particularly successful in the medical sector due to its capacity to analyse large datasets and identify small trends in the datasets. In this rapidly evolving environment where AI is having an increasing impact on society, chatbots (intelligent and interactive computer systems) with machine learning and natural language processing capabilities have the potential to transform the way that healthcare is provided and usher in a new era of healthcare support (Thwala et al., 2023). The aim of this research is to create a medical diagnostic chatbot that uses machine learning models based on NLP techniques to provide quick and accurate medical support.

## **1.2. Motivation for Medical Diagnostic Chatbots**

The motivation behind building a chatbot for medical diagnosis is to make healthcare more accessible and timely, meet the growing demand for healthcare services, and leverage advances in machine learning (Hossain et al., 2021). Additionally, by giving instant medical advice and help, the chatbot makes sure that users can get medical help whenever they need it, without having to wait for an appointment or for a doctor to be available. Moreover, it makes it easier for healthcare professionals to focus on the most important cases by giving preliminary evaluations and suggestions (Ashwini et al., 2022). The chatbot built on machine learning approaches, lets the chatbot understand what the user is asking, recognize symptoms, and give accurate advice based on a large database of medical information. (Iroju and Olaleke, 2015). Therefore, the goal is to use technology to make medical services more accessible, efficient, and high-quality.

The development of medical diagnosis chatbot stems from a desire to harness natural language processing techniques to enhance the accessibility and efficiency of healthcare services (Yadav, n.d.). The goal of the chatbot is to meet the growing need for medical advice and support by giving people an easy-to-use, interactive way to get accurate medical information in a timely manner (Anjum et al., 2023). The other reason is that NLP frameworks and machine learning algorithms might be able to analyse user input, recognize symptoms, and make accurate guesses based on a large database of medical knowledge (Iroju and Olaleke, 2015). By automating the initial assessment and triage process, NLP-based chatbots can help reduce the workload of healthcare workers, make the best use of resources, and make sure that people get help quickly, even when it's not an emergency. Overall, the idea behind using NLP-based chatbots to help with medical diagnosis is to improve access, speed, and the quality of healthcare services.

## **1.3. The Evolution of Human-Computer Interactions in Healthcare**

As the potential of artificial intelligence in the healthcare industry becomes clear, it is critical to comprehend the evolutionary background of human-computer interactions that led to the current situation. Historically, early initiatives to automate healthcare support provided insights into the issues and laid the path for today's more advanced solutions. Several conventional methods were used in the early phases of human-computer interaction to respond to user questions and offer automated customer care. Utilizing menu-based phone systems, Interactive Voice Response (IVR) Systems let customers use their keypad to select options from a list. IVR's strict structure frequently annoyed users because of its rigidity and limited engagement (Dillman et al., 2009). Frequently Asked Questions (FAQs) and Knowledge Bases, which provided static information repositories that users could search through, were frequently seen on websites and applications. While they were useful for simple searches, they lacked the flexibility to deal with sophisticated or unusual user inquiries, potentially leading to mismatches between user queries and the information that was made available (Jansen et al., n.d.). Users were given the opportunity to find information online through search engines, but their efficiency depended on their ability to craft specific queries, which could result in gaps in the retrieval of correct and pertinent information (Jansen et al., n.d.). These conventional methods had serious drawbacks while being widely used. Moreover, the ability of Scripted Chat Interfaces to engage in nuanced conversations and respond to specific user queries was frequently hampered by their reliance on prepared responses triggered by specific phrases (Guy, 2016).Nevertheless, they were effective for facilitating online interactions. Although efficient in some circumstances, automated email responses lacked the real-time interaction and dynamic conversation necessary for efficient customer service (Sanjuán and Magallares, 2014). Due to their impersonal and irritating encounters, IVR Systems also have low user satisfaction (Nordholm and Bäck, 2001). Although useful for reference, FAQs and Knowledge Bases struggled to handle the wide range of user inquiries and required manual updates to keep up with evolving material (Wilson, 1997). The static nature, limited personalization, and inability to understand and effectively reply to natural human language requests were the main limitations of these conventional approaches.

## **1.4. The Modern Era: Chatbots Transforming Healthcare**

To address the limitation in conventional approaches, recent advancements in technology have sparked a paradigm shift in the healthcare industry that is encapsulated by chatbots, which are intelligent and interactive computer systems. Through the merger of Natural Language Processing (NLP) and Machine Learning (ML) technologies, these systems, skilled at replicating human-like interactions, present efficient communication ways that have the potential to reinvent medical diagnosis and counselling services.

Chatbots, which are linked into platforms such as Facebook Messenger (Shivam et al., 2018) and WhatsApp, have gone far beyond basic tasks. They've become an essential element of our digital experience, from assisting bank customers (Liu et al., 2018) to answering questions at universities (Ranoliya et al., 2017).They're especially useful in healthcare, where they provide real-time answers, improve patient interaction, and make services more efficient. They excel at handling numerous conversations at once, resulting in enhanced productivity and significant cost savings (Brooke, n.d.)(Cavanagh and Millings, 2013). Users profit from their constant accessibility because it is available around the clock (Brooke, n.d.).Mainly the reasons chatbots seem so intuitive is the integration of NLP and ML technologies, which enable them to offer personalized responses. This makes user interactions feel more genuine and fulfilling (Cavanagh and Millings, 2013). Besides assisting users, chatbots are data collectors, gathering vital insights to refine products and shape marketing strategies (Smutny and Schreiberova, 2020). Their role in healthcare is especially noteworthy. Chatbots assist patients in understanding their health, boosting their engagement (Chen and Decary, 2020). In areas where medical resources are scarce, they provide much-needed information and guidance (Tallyn et al., 2018)(IEEE Staff, 2017). They're also a boon for the elderly and those with disabilities, bridging the communication gap (Tallyn et al., 2018).Moreover, the educational scope of chatbots is growing. They're emerging as tools for language learning, helping users refine their skills with feedback (Foley and Woollard, n.d.).As technology advances, especially in the realms of AI and NLP, chatbots are set to become even more advanced, offering deeper and more complex interactions (Cavanagh and Millings, 2013).Overall, chatbots are evolving rapidly, ready to cater to diverse needs across various sectors. As technology continues to progress, their role will only become more central in our daily lives.

## **1.5. Technological Foundations Behind Modern Chatbots**

While chatbots have been easily integrated into a variety of industries including healthcare, their complex characteristics are based on NLP techniques and machine learning models. To fully grasp their powers and potential, it is necessary to investigate the technological foundations that power these intelligent systems.

Chatbots are technologies designed to mimic human-like conversations. They function by accepting textual input, processing it to detect purpose, and then creating an appropriate answer. Advanced Natural Language Processing (NLP) and Machine Learning (ML) models are largely responsible for the magic behind these operations. Understanding human language can be tricky, and that's where NLP methods assist. NLP techniques such as tokenization, named entity identification, and dependency parsing are at the core of interpreting textual input in chatbots. These mechanisms aid in the breakdown of sentences, the identification of relevant terms such as names or medical diagnoses, and the comprehension of word relationships (Jurafsky and Martin, n.d.). As chatbot technology advances, it's transforming how patients connect with healthcare providers and how health services are delivered.

Machine learning models have advanced and are driving this revolution in healthcare, with decision trees and SVM classification algorithms playing a key role. These algorithms are essential for jobs like classifying inquiries (Zhang and Wallace, 2015). Recurrent neural networks (RNNs) and transformers have brought in a new age for chatbot interactions thanks to the adoption of deep learning models. These models offer an in-depth comprehension of sequential data, ensuring that dialogues with healthcare chatbots are not only educational but also contextually aware and organic (Zhou et al., 2021). Other than this, there's a rising trend towards using pre-trained models, especially those designed for sequence-to-sequence (one sequence as an input and produce another sequence as an output) tasks (like medical chatbot). T5 (Text-to-Text Transfer Transformer) and BART (Bidirectional and Auto-Regressive Transformers) are examples of such models. These models have been trained on vast amounts of text, allowing them to understand language patterns deeply. They can be customized for jobs, like making sense of medical data. In healthcare, where accuracy is vital, these tools are extremely helpful. These seq2seq models can transform one piece of text into another, which is especially useful for tasks like summarizing medical records or translating medical terms. Their design makes them powerful tools in various applications, showing the potential of modern tech in medicine (Grambow et al., 2022) (Raffel et al., 2019). Overall, machine learning innovations, from decision trees to pre-trained models like T5 and BART, are revolutionising the healthcare industry by improving the processing of medical data and contextualising chatbot interactions.

## **1.6. Research Objective and Methodology**

Due to significant advances in machine learning, the objective of this research is to build an NLP-based chatbot that uses modern natural language processing methods to effectively read user input and offer immediate medical diagnostic support. The MedDialog-EN dataset (He et al., 2020), which contains roughly 257,469 dialogues and is 295 MB in size, will be used in this research for the training phase of medical diagnosis chatbot. While this dataset can provide a solid foundation for this research, in further work plan will be to incorporate more data sources, such articles from medical journals, to improve the accuracy and comprehensiveness of the system. In the model training phase, sequence-to-sequence models will be implemented due to the nature of the MedDialog-EN dataset and the requirements for generating diagnostic text based on user queries. There will be two pivotal neural network architectures: RNN (Recurrent Neural Networks) and Transformers. RNN (Recurrent Neural Networks) and Transformers will play key roles in the creation of this medical diagnosing chatbot. Specifically, from the RNN, utilising the encoder-decoder LSTM (Long Short-Term Memory) structure, known for its effectiveness in retaining sequential information. From the Transformer, utilisation of pre-trained models like GPT-2, BART, and T5. These models were chosen because they have a track record of success in understanding context, generating text that makes sense, and their flexibility in handling various tasks, which makes them ideal for use in medical chatbots. They can deliver precise and contextually relevant responses due to their extensive training on a variety of datasets. In the model deployment phase, a mobile application will be developed having a user-centric chat interface, using React Native (javaScript based Cross-platform mobile application development framework). This interface will serve as the primary medium of interaction for end-users with the chatbot. For the integration or make a connection between the machine learning model and the user interface, and to facilitate efficient model serving, Flask (python based) framework will be used. Due to constraints like limited time and limited computational resources, this chatbot's initial version will only be used as a prototype. For the implementation, will be working on using the computational resources, single NVIDIA GeForce RTX 3090 GPU, which offers a memory capacity of 24,268 MB. These limitations could make it difficult to start out with the best accuracy possible. Overall, by using NLP techniques, the MedDialog-EN dataset, RNNs, Transformer models, and a React Native app user interface, this research aims to create a medical diagnostic chatbot. However, due to limitations, the initial version will only be a prototype that uses a single NVIDIA GeForce RTX 3090 GPU.

## **1.7. Structure of the Dissertation**

The format of the rest paper is as follows:

**Chapter 2 (Background and Literature Review):** This chapter delves into the historical development of Natural Language Processing (NLP) in healthcare. It examines the progression of NLP applications and their implications for the medical field. Furthermore, the chapter also delves into previous works on medical chatbots, analysing their methodologies, findings, and limitations.

**Chapter 3 (Considerations):** An exploration into various considerations that influence the creation and deployment of medical chatbots.

**Chapter 4 (System Design and Resources):** This chapter delves deep into the architecture of the proposed system. It showcases the system design diagram, highlighting on how different components interface with each other. In addition, the chapter provides insights into the hardware resources utilized in the system's development, ensuring scalability and robustness.

**Chapter 5 (Data Collection and Analysis):** A breakdown of the data collection, followed by a description of the datasets. Special focus is given to the MedDialog-EN dataset, elaborating on its properties and significance. The chapter concludes with detailed exploratory data analysis, highlighting preliminary insights and patterns.

**Chapter 6 (Methodology):** This chapter discusses the methodologies in data preprocessing, cleaning, and model implementation. A detailed examination of the encoder-decoder LSTM structure from the RNN family is provided. The chapter also brings to light experiments with Transformer-based models like GPT-2, BART, and T5. Lastly, it discusses the evaluation criteria used to gauge model performance.

**Chapter 7 (Results with Mobile Application):** React Native mobile application development, its integration mechanisms with Flask for model serving, and the outputs. This chapter contains the practical results achieved.

**Chapter 8 (Limitations):** A reflection on the challenges faced during the research. This chapter delineates limitations from both the dataset's perspective and the models, providing a view of potential pitfalls and gaps.

**Chapter 9 (Discussion):** A discussion on the study's findings, their significance for the medical domain. This chapter also discuss the broader implications of the study, results with existing literature, and highlighting potential breakthroughs.

**Chapter 10 (Future Work):** This chapter points towards potential areas of exploration, suggesting improvements, new features for the mobile application, and further research that can build upon the current study.

**Chapter 11 (Conclusion):** A summation of the research, covering the major discoveries, insights, and their implications.

# **Background and literature review**

## **2.1. Chatbots and Potential in Medical Diagnosis**

Artificial intelligence (AI), machine learning, and deep learning have all advanced quickly in recent years, giving machines an ever-increasing ability to emulate human interactions. The creation of chatbots, which use artificial intelligence and natural language processing (NLP) to have conversations that resemble those between people, is one significant application of this technology (Davenport and Kalakota, 2019). According to (Hossain et al., 2021), chatbots have various uses in a variety of industries, including education, call centres, online retailers, customer service, and online gaming. They are also being researched for their potential to transform the way that doctors support diagnoses. Chatbots may analyse and understand user inputs through text or speech interfaces and offer responses that mimic human understanding by utilising NLP techniques. This presents an exciting opportunity to enhance the efficiency and accessibility of medical diagnosis support.

## **2.2. Historical Roots of Chatbot Technology**

While developing an NLP-based chatbot for medical diagnosis support, it is important to acknowledge the historical roots of chatbot technology. In 1950, Alan Turing published a paper (Turing, 1950), in which he introduced the Turing Test concept. The Turing Test measures the ability of machines to show intelligent behaviour that is indistinguishable from human behaviour. Turing argued that if a machine can convince an interrogator that it is human during a discussion, it can be "thinking". The Turing Test, which evaluates a machine's ability to mimic human intelligence, has become a benchmark for measuring the progress of AI technology including chatbots. However, it has limitations in assessing true understanding, capturing all aspects of intellect, establishing a clear threshold, and taking embodiment and sensory experience into account. According to (Joseph Weizenbaum, 1966) (Brandtzaeg and Følstad, 2017), first chatbot, it is interesting to note that ELIZA was originally designed for the medical domain. It was just a computer program made to respond to user questions in a way that made the user think they were speaking to a real psychologist, serving as a virtual psychiatrist. It is developed by a German computer scientist Joseph Weizenbaum in 1966. As mentioned in (Sharma et al., n.d.), ELIZA mimicked therapy sessions by rewording user utterances and offering them as questions. It lacked a contextual foundation and relied instead on a "pattern matching" and replacement mechanism to give the appearance of understanding. The most well-known script by ELIZA, mimicked a psychiatrist and asked open-ended inquiries to participants. Despite Weizenbaum's intention to highlight the superficiality of human-machine connection, many users gave ELIZA human-like emotions and even thought it would have a good impact on people's life, particularly those who struggle with psychiatric disorders. However, its conversational capabilities were limited, it was enough to confuse those who were unfamiliar with dealing with computers, prompting the development of other chatbots (Klopfenstein et al., 2017). ELIZA was the early conceptualization of healthcare chatbots after this researchers’ started experimentation in the medical domain to build medical chatbots with improvements.

Following the early experimentation, the personality-based chatbot PARRY, created in 1972, was an upgraded version of Eliza. According to (ZEMČÍK, 2019), a well-known chatbot named PARRY was first introduced in 1972 at Stanford University's Psychiatry Department by psychiatrist and computer scientist Kenneth Mark Colby. Unlike other chatbots like Eliza, PARRY acts like a paranoid schizophrenic patient rather than pretending to be a doctor. Its goal was to serve as a functional model of Colby's theoretical pattern of paranoia functioning as a defective processing of signs in the patient's mind, as well as a didactic tool for young psychiatrists learning how to communicate with patients diagnosed with paranoid schizophrenia. PARRY tries to provoke controversies and elicit more elaborate answers from the participant by engaging in paranoid ramblings and changing topics. However, the main drawbacks of PARRY were its narrow focus on paranoid behaviour, limited conversational scope, and reliance on pre-programmed responses, inability to learn, limited language comprehension, and possibility for improper responses. Nonetheless, PARRY's contributions assisted in the advancement of chatbot technology and inspired more research in natural language processing and artificial intelligence (Colby et al., 1971). However, as the field advanced, increasingly complex chatbots emerged. In 1995, another chatbot named ALICE was built, and it represented a big step forward by utilising a simple pattern-matching algorithm and the Artificial Intelligence Markup Language (AIML). It could imitate user interactions by matching their input with predefined patterns and providing appropriate answers. ALICE's success in simulating human-like conversation earned it the Loebner Prize for "most human computer" in 2000, 2001, and 2004. However, the limitations of ALICE and similar chatbots, such as their reliance on basic pattern-matching algorithms, lack of responsiveness, and trouble handling ambiguity in user input, prevent them from engaging in meaningful and contextually relevant conversations, highlighting the need for more advanced natural language processing techniques in chatbot development (Bruno Marietto et al., 2013). Another chatbot, in the early 2000s, SmarterChild was a chatbot that operated on AOL (America Online), Yahoo, and MSN (Microsoft Network) platforms. It allowed users to ask inquiries and receive information about upcoming sporting events or stock values. SmarterChild functioned by recognising keywords and doing searches to deliver instant responses to user inquiries. However, had limitations because it could only engage in basic interactions and deliver information that could be acquired through search engines. Users were aware that they were conversing with a machine, and the chatbot was incapable of engaging in complex conversations (Molnar and Szuts, 2018). As field of AI progressed, the limitations of early chatbots became clear, emphasising the need for more powerful natural language processing techniques to generate meaningful and relevant conversations. These challenges paved the way for the next evolution in interactive technology. Over time, the push to overcome these limitations led to the development and refinement of more advanced systems.

## **2.3. Revolutionary Advancements with Virtual Personal Assistants**

The development of chatbot technology has made tremendous strides, making increasingly advanced chatbots available in a variety of messenger programmes. Further revolutionising the field of artificial intelligence and natural language processing was the invention of virtual personal assistants like IBM Watson, launched in 2010 (IBM. (n.d.)., 2023), Apple Siri, launched in 2011 (Apple Inc. (n.d.)., 2023), Microsoft Cortana, launched in 2014 (Microsoft Corporation. (n.d.)., 2023), Amazon Alexa, launched in November 2014 (Digital Trends. (n.d.)., 2023), and Google Assistant, launched in 2016 (Google. (n.d.)., 2023). In the paper (Chakrabarti et al., n.d.), author discusses the development of virtual personal assistants (VPAs) that utilize multi-modal dialogue systems to enhance interaction between humans and machines. These VPAs, such as Microsoft Cortana, Apple Siri, Amazon Alexa, and Google Assistant, incorporate technologies like speech recognition, gesture recognition, and image/video recognition to process multiple user input modes. These VPAs have applications in various domains, including education assistance, medical assistance, robotics, home automation, and security access control.

## **2.4. The Impact of Pre-trained Models in Modern Chatbot Development**

One significant advancement in the development of chatbot technology is due to the emergence of pre-trained models (PTMs) has brought natural language processing (NLP) to a new era. According (Qiu et al., 2020), pre-trained models have greatly improved chatbot technology development by providing a solid foundation for natural language processing and development. These models, which were trained on large-scale databases, capture the semantic and syntactic patterns of human language, allowing chatbots to better understand and create content. In addition, pre-trained models have the benefit of being able to acquire universal language representations (Tu et al., n.d.). These models develop a thorough knowledge of language, including word meanings, contextual embeddings, and grammatical structures, by training on a large amount of text data. Chatbots can use this information to enhance their language comprehension skills, enabling them to accurately interpret user inquiries and produce appropriate responses. Furthermore, pre-trained models can be improved on chatbot activities, such question answering or dialogue production, so they adapt their expertise to the target area. (Akbik et al., n.d.). The pre-trained model is fine-tuned by using task-specific data to train it, enabling the model to specialise in the required chatbot functionality. Compared to building a chatbot model from scratch, this transfer learning strategy saves a lot of time and computational resources. Additionally, pre-trained models handle the problem of little training data in NLP activities, such as chatbot development (Tu et al., n.d.). On small datasets, deep neural networks typically overfit, which results in poor generalisation. However, this problem is solved by pre-training on large datasets, which creates a solid foundation of language understanding. The model's understanding of the given domain is then improved by fine-tuning with task-specific data, improving chatbot performance. Moreover, pre-trained models have paved the way for chatbot designs that are more sophisticated. For instance, the Transformer model's self-attention mechanism has completely changed the area of NLP and is frequently employed in pre-trained models like BERT (Bidirectional Encoder Representations from Transformers) and GPT (Generative Pre-trained Transformer) (Howard and Ruder, n.d.). Using this attention mechanism, the model can recognise long-range dependencies in text, leading to more precise language comprehension and generation. The evolution of chatbot technology has been directly driven by these improvements in design, resulting in conversational agents that are more advanced and effective. Overall, pre-trained models' (PTMs') development has considerably boosted chatbot technology by giving NLP a solid foundation and paving the way for more sophisticated and useful conversational bots through better language understanding and generation.

## **2.5. The Evolution and Significance of Medical Chatbots**

In the medical domain, chat-bot development has advanced greatly throughout time, beginning with ELIZA and continuing to the present. According to (Amer et al., 2021), chatbots in healthcare have become essential during pandemics (COVID-19) due to the requirement to handle a large amount of user inquiries. Chatbots are sophisticated communication tools that replicate human-like conversations to deceive users into thinking they are speaking with a person rather than a machine. As said by (Yadav, n.d.), in the healthcare or medical domain, chatbot’s main goal is to fill the communication gap between users and healthcare experts by promptly responding to their inquiries. People might be less likely to prioritize their health issues in the current digital era, where internet addiction is common. They frequently avoid going to hospitals for mild illnesses that might get worse if they are not addressed. Chatbots have evolved as an easy way to answer these queries rather than searching through various web articles. The absence of prompt responses for people in need of medical care, which results in lengthy wait times for professional acknowledgement, is just one issue with the current procedures, though. Additionally, some processes, such as doctor phone consultations or live online chats, may be charged for. However, according to (Laranjo et al., 2018), users can be cautious to completely accept the diagnosis recommendations made by a chatbot. For the technology to be widely used and successfully integrated into healthcare systems, user acceptance and confidence must be built.

## **2.6. Literature Review: Artificial Intelligence in Healthcare**

For the development of medical chatbots, let's delve into the literature review in this field. In prior studies, researchers have been actively exploring and experimenting with a wide range of artificial intelligence (AI) technologies. These studies have covered a variety of AI subfields, such as machine learning, deep learning, and natural language processing (NLP). Researchers have worked to maximize the potential of these AI technologies to create cutting-edge applications and solutions in the field of healthcare. Let’s discuss the literature review.

Traditional machine learning techniques like SVM (Support vector machine) have also been implemented in the development of chatbots for medical diagnosis support in order to supplement the advancements made with NLP and AI technology. In (Omoregbe et al., 2020), study discussed the development of CUDoctor, a chatbot service that evaluates tropical disease symptoms in Nigeria using natural language processing (NLP) techniques and fuzzy logic rules. Users can enter their symptoms into the chatbot, which is a component of the Covenant University Doctor (CUDoctor) telehealth system, and the symptoms are subsequently sent to the system for diagnosis support. To build the knowledge basis for the system, the study gathered information from a medical database and conducted interviews with specialists and people who were knowledgeable about the various disorders. Based on inclusion criteria, such as recent diagnosis or familiarity with the disorders under investigation, respondents were chosen via snowball sampling. In addition, the disease is predicted using a fuzzy support vector machine (SVM) based on the symptoms entered by the users. The study used the Recall-Oriented Understudy for Gisting Evaluation (ROUGE-L) measure and the Bilingual Evaluation Understudy (BLEU) score to assess how well the produced service performed. The ROUGE-L metric evaluates the lengthiest common subsequence between them, whereas the BLEU score checks how similar the chatbot's generated responses are to the reference responses. ROUGE-L and BLEU-2 scores were 25.29 and 31.56, respectively, according to the study. The System Usability Scale (SUS) questionnaire, a commonly used instrument for assessing a system's usability, was employed in the study's usability testing. Higher SUS scores indicated better usability; they ranged from 0 to 100. The mean SUS score for CUDoctor was 80.4, which meets the SUS evaluation requirements and is regarded as excellent. The user module, diagnostic interface, knowledge base, and fuzzy inference engine were all components of the system's architecture. For connectivity between the chatbot and the system as well as with SMS subscribers, Twilio and the Telegram Bot API were utilised. However, the system's limitations include the potential for false positives, as the final diagnosis must still be verified by a physician. The study also noted that the diagnosis system may be improved by being made even more automated, by adding voice interaction for increased engagement, and by expanding functionality to include treatment recommendations and drug adherence. Overall, the study demonstrated that a medical diagnosis system could be successfully developed combining NLP, fuzzy logic, and machine learning techniques, with good usability outcomes. However, it acknowledged the need for additional improvements and changes that would increase utility and accuracy.

In this paper (Yadav, n.d.), researchers built system for healthcare applications using a variety of AI technologies and machine learning methods. In this study scientist used a cutting-edge method that uses Deep Neural Network (DNN) classifier technology and Natural Language Processing (NLP) techniques to provide medical diagnoses and recommendations for patients based on their symptoms. People frequently ignore their health in today's fast-paced society because of their busy schedules. The CBMD system intends to facilitate early intervention for health issues and enhance patients' access to medical information. The architecture of the system includes a web-based chatbot interface that enables users to communicate with the system via text or voice. The text is pre-processed using NLP techniques including stemming, stopping words, and tokenization when a patient submits their symptoms to extract relevant symptoms. To ensure correct diagnoses, the system of this study used a knowledge base, a collection of medical information, and labels to identify quality information. In this study, scientists used DNN (Deep Neural Network) classifier, which finds patterns and connections between symptoms and illnesses, is essential to this process. The DNN recommends the best course of action for the indicated illness by analysing the input. Several metrics, including precision, recall, F1-score, and accuracy were utilised to assess the system's performance. These measurements reveal information about the system's capacity to offer precise medical diagnoses and effective cures. It is possible to evaluate the system's success in enhancing healthcare outcomes by making comparisons with currently used techniques. However, the CBMD system has drawbacks despite its apparent advantages. The quality and depth of the knowledge base have a significant impact on its correctness. Inaccurate information can result in diagnoses that are wrong. Additionally, even while the DNN classifier can handle a variety of symptoms and diseases, it can have trouble classifying complex or rare medical disorders that call for specialised knowledge. Overall, the CBMD system offers a cutting-edge approach to improving medical diagnostic and treatment recommendations. The system's accuracy, reliability, and ethical compliance must be ensured, though.

In this article (Zhou et al., 2021), In the context of online medical consultations, the authors suggest using a machine learning model called DP-CRNN (Deep CNN-RNN) to model and analyse patient clinician generated data. The purpose was to offer patients knowledgeable clinic advice and pre-diagnosis options. A total of 173,184 patient inquiries from six distinct medical departments made up the data utilised for evaluation, which was gathered via the online medical service "39WenYiSheng". Convolutional and recurrent neural networks are combined in the DP-CRNN model, which is an integrated framework. It uses an RNN to record sequential information and a CNN to extract semantic features from the patient's questions. To generate recommendations for the medical department, the model then incorporates these features using pooling layers and feeds them into a classification layer. The author conducted experiments by comparing DP-CRNN with four different neural network models (CNN, RNN, CRNN, and RCNN), using criteria like precision, recall, accuracy, and F1 score, . In terms of recommendation accuracy, precision, recall, and F1 score, the results demonstrate that DP-CRNN performs better than the baseline models. Due to the complexity of symptoms in these fields, the model struggles to appropriately support situations in internal medicine and gynaecology, two fields where it performs exceptionally well. They identified hidden semantic similarities from the data and improved the learning process by using a clustering approach. The findings demonstrate that the clustering model considerably enhances the performance of recommendations, offering more accurate pre-diagnosis recommendations. Overall, in terms of providing intelligent pre-diagnosis support in online medical environments, the suggested DP-CRNN model and the intelligent diagnostic module demonstrate promising results. The method makes use of machine learning techniques to efficiently handle patient enquiries and assist patients in making wise medical decisions. However, it has some significant limitations, including the need to manage inconsistent data dissemination in various medical departments and the potential difficulties of managing more complicated symptoms that would call for a deeper understanding of semantics.

In (Bi et al., 2020), the research proposed a Bi-directional LSTM Model with Symptoms-Frequency Position Attention (BLSTM-SFPA) to answer medically questions. The objective is to improve the comprehension, representation, and accuracy of symptom descriptions in medical queries. Three essential elements were included in the proposed model: the framework of BLSTM-SFPA, semantic normalisation based on kernel word-common word substitution, and symptoms-frequency position attention (SFPA). Assigning higher weights to frequently occurring symptoms and their impact on nearby terms, SFPA considers the frequency and placement of symptom words in users' descriptions. Semantic normalisation replaces common terms with semantic kernel words using word2vec and a Chinese Ci-Lin lexicon, increasing semantic representation. The framework is divided into two stages: traditional attention mechanisms produce initial representations of historical descriptions and diagnostic responses, and BLSTM-SFPA produces final representations based on position-aware influence spread by symptom words. This study shows that research on the MED-QA (Medical question answers) demonstrates that BLSTM-SFPA outperforms reliable baselines, delivering cutting-edge performance in answering medical questions. Measures of model performance include MAP (Mean Average Precision) and MRR (Mean Reciprocal Rank). Parameters are fine-tuned via Bayesian optimisation, and the best values are determined by experimental testing. When the propagation range parameter () is set between 10 and 25, the model performs better. In contrast to conventional attention mechanisms, the proposed SFPA mechanism concentrates on relevant symptom phrases and their surrounding context, offering useful guidance for medical question answering. The lexical gap and location information in symptom descriptions are handled well by the suggested BLSTM-SFPA model, resolving issues with medical question-answering systems. However, in other circumstances, it can be necessary to have a large amount of historical data with symptoms that are identical. Overall, the study proposes a novel method for effectively answering medical questions and shows that it works through experiments and evaluations. The SFPA mechanism is essential for improving attention and semantic representation, which improves the effectiveness of systems that answer medical questions.

In (Sirriani et al., n.d.), study experimented how two generative pre-trained transformer (GPT) models, GPT-2 and GPT-Neo, perform in the context of medical text prediction using a dataset of 374,787 free-text dental care clinical notes. Given the lack of readily available medical datasets and the high expense of creating and maintaining large datasets, the goal is to investigate the potential of these pre-trained models to aid in medical charting tasks. The paper explained the techniques used, such as optimising the GPT-2 and GPT-Neo models for next word prediction on the dental dataset, with a split of 80% for training, 10% for validation, and 10% for testing. Both models' performance metrics, such as prediction accuracy and loss, are shown. According to the study's results, GPT-2 has a testing accuracy of 76% compared to GPT-Neo's 53%. Particularly when it comes to predicting names, abbreviations, and punctuation, GPT-2 is better than GPT-Neo. However, for predicting the subsequent word in the dental notes, both models have acceptable accuracy scores. The study also discussed the limitations, which includes the dataset's limited regional focus and limited scope to dental notes. Although, future research has been advised to examine complex language use and tasks, expand the study to cover additional medical domains. Overall, the study shows how GPT models were first applied in the medical field and shows how they have the potential to help in medical charting.

In the paper (Iftikhar et al., n.d.), to develop content about the use of AI in healthcare, tests were conducted using OpenAI's ChatGPT-3. The researchers employed the "regenerate response" feature of ChatGPT-3 to acquire more information regarding the supply of health services and types of Internet of Orthopaedic Things (IOT). Following that, the generated content was summarised and cross-confirmed by a literature study. The results of the studies showed that ChatGPT-3's natural language processing (NLP) technology has the potential to revolutionise the healthcare business. It is capable of analysing patient notes, medical publications, and research articles, as well as assisting in the creation of medical reports, documentation, and drug discoveries. ChatGPT-3 can also be used in medical education as a learning tool, to improve communication between patients and healthcare workers, and to convert clinical language into organised clinical data. However, there are some limitations to consider. The study acknowledges that the information gathered by ChatGPT-3 was left un-cited in order to minimise confusion and repetition. Furthermore, while ChatGPT-3 has showed promise, more research and development are required to fully realise its potential benefits in healthcare. When applying AI models like ChatGPT in health care, ethical and privacy concerns must also be addressed. Overall, the studies showed the potential of ChatGPT-3 in revolutionising health services using NLP technology, however more study and development is required to overcome limitations and fully use its benefits.

In (Amer et al., 2021), study introduces an innovative chatbot system that can communicate with users and offer them COVID-19-related information. For question-answering, the system uses the pre-trained Google BERT language model. The model is divided into two phases: text classification using BERT to categorise input into groups according to meaning, and using the BERT model to respond to user inquiries. The BERT token limitations, which is typically 512 tokens, is bypassed by categorising responses according to the type of inquiry asked. This makes the answering process more effective by enabling the BERT classifier to select the proper response category. On the Stanford University SQuAD V2.0 question-answering dataset, the BERT model is trained and tested. When classifying questions into their appropriate categories using BERT, excellent accuracy was attained. The answering stage makes advantage of BERT's capacity to comprehend the context and give precise answers to user inquiries. The system's precision and testing outcomes showed promise, proving how effective it is at giving trustworthy replies to frequent COVID-19-related queries. Despite the success, controlling text strings that go beyond the BERT token limit presents difficulties. By organising questions into categories and skilfully choosing the most appropriate environment for responses, the article offers a practical solution. This study shows that the BERT model excels at many NLP tasks and provides advantages over traditional machine learning models by capturing sequential material in text. The suggested chatbot concept has produced encouraging results and has room for improvement to increase its sturdiness and capabilities. Overall, a smart chatbot system built on BERT for queries pertaining to COVID-19. It shows how well BERT performs in text classification and response tasks. By utilising appropriate datasets and pre-processing methods, the proposed procedure can be utilised in the medical diagnosis support.

This paper (Xygi et al., 2023), proposed a strategy for using chatbots in the field of biomedicine to effectively assist doctors and the public in accessing and understanding difficult biomedical concepts. For tasks involving natural language processing, the authors developed the chatbot using a variety of BERT models (RoBERTa, XLM-R, BERT Large, and BioBERT). They compare the effectiveness of the CountVectorizer method and the BERT models as well. The author discussed that the BERT models are language models that have already been trained and have shown successful in several NLP tasks. When doing tasks like question-answering and language inference, they make use of an attention mechanism to learn the contextual relationships between words. This enables them to evaluate the semantic similarity of texts. Due to their training on biomedical text data, the chosen BERT models offer a variety of features and complexities that make them suited for biomedical domain tasks. The chatbot used in this study was developed as a question answer chatbot, retrieving responses from a collection of predefined responses rather than making incorrect medical recommendations. The paper describes the algorithm and the evaluation metrics (Cosine similarity, precision, recall, and F1-score) used to assess the performance of the chatbot. According to experiments conducted by the authors of this study, RoBERTa is more sensitive to the similarity threshold, and performs better at higher similarity values. Even with lower similarity thresholds, the other BERT models still perform well. Surprisingly, with fewer response attempts, the CountVectorizer performs better in terms of F1-score. The authors argued that this is due to CountVectorizer's usage of context-specific word embeddings rather than BERT's thorough sentence parsing. However, the paper not mentioned, the detail on the machine learning methods that were employed to implement the system. Although similarity measures and the F1-score are mentioned, the precise procedures and methodologies are not covered in detail. Overall, the study offers a technique for using chatbots that have BERT models for biomedical domain tasks. In the context of the chatbot implementation, it demonstrates the benefits and drawbacks of the CountVectorizer and BERT models.

In the paper (Grambow et al., 2022), the conversion of doctor-patient conversations into clinical notes is explored by the authors using Natural Language Processing (NLP) and machine learning. By evaluating the possible benefits of improved pre-training utilising specialised medical dialogues, this study expands on prior research. The researchers experimented with models like BART (Bidirectional and Auto-Regressive Transformers), LED (Longformer Encoder-Decode), and DialogLED using transformer topologies. By enhancing these models with a dataset on conversation between doctors and patients, initial benchmarks were established. Their investigation included comparing models that were pre-trained on medical domain data to those that weren't. Traditional criteria including accuracy, recall, and F1-score were used to objectively evaluate model efficacy. Moreover, a clinical Named Entity Recognition (NER) model was included in a novel method to concept-centric evaluation. The mapping of identified entities to UMLS concept unique identifiers (CUIs) made possible by this model when used in conjunction with QuickUMLS allowed for the thorough computation of precision, recall, and F1-score based on anticipated types and UMLS CUI sets. The results of the research were enlightening. Models who had received pre-training in the medical domain clearly outperformed those who hadn't been exposed to such a training setting. It's interesting to note that when long-sequence transformers were compared to more traditional models like BART, the former shown to be far better at capturing clinical discussions. However, there were a few challenges with the research. Although promising, the produced models still had flaws that required human interaction to fix. The complex medical terminology presents significant challenges for widely used pre-trained algorithms. Even though the study used doctor-patient connections for the pre-training phase, it advised using a larger dataset that included clinical notes for possibly better outcomes. The authors also suggested possible directions for additional research, like adjusting model outputs to keep a constant length. Overall, study contributes to function of NLP in transcribing of clinical interactions. It emphasises the necessity of pre-training in a certain topic and highlights the complex opportunities and problems unique to the medical industry.

In conclusion, the creation of chatbots using AI technology and machine learning techniques has yielded promising results in the field of medicine. These developments have produced more complex chatbots that can respond to medical queries, analyses patient-generated data, and make diagnosis and suggestions based on symptoms. Pre-trained model offers a strong foundation for linguistic comprehension and has significantly enhanced chatbot development. To increase user acceptability and confidence in chatbot diagnosis, more study and development are still required. This research also discussed the limitations and the need for continued advancements in this area, even while they show how AI and machine learning might improve medical diagnosis and treatment suggestions.

# **Considerations**

There are various considerations that should be considered when developing and deploying (deployment so the end-user can use) a NLP-based chatbot for medical diagnosis support. Let's break down and discuss each point individually.

## **3.1. Legal considerations**

There are a few legal considerations to keep in mind when making and using an NLP-based chatbot for medical analysis. First, it is very important to make sure that privacy and data security laws are followed, such as the General Data Protection Regulation (GDPR) in the European Union. Protecting user data, getting informed consent, and putting in place the right security steps are all important parts of staying legal (General Data Protection Regulation, 2023). Second, users should be given clear disclaimers that explain the limits of the chatbot's advice and stress how important it is to talk to a trained medical professional for an accurate evaluation and treatment. Moreover, it's important to follow medical ethics and professional rules to avoid legal problems that could arise from a wrong diagnosis or bad advice (WMA - The World Medical Association-WMA International Code of Medical Ethics, 2023). By thinking about these legal issues, make sure that NLP-based chatbots for medical analysis are used in a responsible and ethical way.

## **Social considerations**

When developing and employing an NLP-based chatbot for medical analysis, it's important to think about how its use will affect society. One important thing to think about is how it might affect the relationship between the doctor and the patient. Even though chatbots can help with healthcare in a convenient and easy-to-reach way, they should not be used instead of human interaction and personalized care from healthcare experts (Laranjo et al., 2018). Another social consideration is to think about whether the chatbot's suggestions are biased. Care must be taken to make sure that the chatbot doesn't make healthcare biases or differences in evaluation and treatment worse or make them worse (Rajkomar et al., 2018). Additionally, it's important to handle accessibility issues to make sure that the chatbot can be used by a wide range of people, including those who are disabled or don't have easy access to technology (WHO Global Observatory for eHealth., 2011). By taking these social factors into account, NLP-based robots can be made that add to healthcare services, keep care focused on the patient, and promote equal access.

## **Ethical considerations**

When developing and employing an NLP-based chatbot for medical analysis, it is important to think about ethics to make sure the technology is used in a responsible way. Protecting the safety and confidentiality of patients is a very important part of being ethical. To protect private medical information, the chatbot should follow strict data protection and security rules (WMA - The World Medical Association, 2023). Transparency and informed agreement are two other important ethical factors to think about. Users should know exactly what the system can do and what it can't do, and they should give their permission for data to be collected and used (Vaidyam et al., 2019). Moreover, the chatbot should be made so that it does not harm the health of the patient, making sure that the advice and suggestions it gives are based on evidence and are correct and reliable. By taking these ethics concerns into account, can make sure that NLP-based chatbots used in medical diagnosis are trusted, private, and used in a responsible way.

## **Technical considerations**

For a medical diagnosis chatbot, scalability, integration, and latency are essential technical considerations. Scalability ensures that the chatbot can handle multiple users inquiries at a time. This requires a strong infrastructure and a design that allows for concurrent processing. Integration emphasises the chatbot's smooth integration with existing medical systems, particularly EHRs (Electronic health records), to give personalised responses while also maintaining data privacy. Finally, concerns about latency highlight the need of real-time answers, particularly in emergencies, necessitating optimised algorithms and smart server placements to reduce response time.

## **Data considerations**

The quality and diversity of the data utilised for training a medical diagnosis chatbot is critical to its reliability and effectiveness. The foundation of reliable diagnostic results is high-quality data that is both relevant and validated. Equally important is the diversity of this data, which covers a wide range of medical conditions and demographics and areas, promoting a complete and inclusive diagnostic capability. The chatbot's database should evolve in conjunction with the medical industry. Regular updates based on the most recent medical research, paired with a feedback loop involving medical professionals, ensure that the chatbot remains relevant and up to date, delivering insights based on the most recent discoveries and standards. This mix of quality, diversity, and regular updates of data underlies the chatbot's capacity to deliver reliable and fast medical advice (Tamang et al., 2023).

## **Machine learning considerations**

Several factors must be considered when developing machine learning for medical chatbots. The use of algorithms such as LSTM or GPT-2 must be tailored to the specific needs of medical diagnosis. According to the (Lee and Shin, 2020), trust requires clarity in how machine learning algorithms arrive at their findings (model interpretability). Regular real-world validation ensures that the model remains relevant, while vigilance against overfitting and bias is essential for accuracy. The model must be realistically effective, not merely theoretically valid, and capable of adapting to the ever-changing medical the situation. Finally, strong safety standards are required to avoid and correct potentially dangerous chatbot suggestions, ensuring patient safety remains a priority.

## **User considerations**

In the development and refinement of medical chatbots, user considerations are critical. The interface should prioritise user-centric design, emphasising simplicity of use and accommodating diverse devices, while also using NLP for natural interactions. Prioritising accessibility guarantees inclusion for all users, complies with World Health Organisation (2020) global standards (Global strategy on digital health 2020-2025, 2021), and includes features that accommodate to varied user needs. Furthermore, a robust feedback system is critical, offering crucial insights for iterative changes and ensuring the chatbot is always aligned with changing user preferences and demands.

In conclusion, developing an NLP-based chatbot for medical diagnosis necessitates careful consideration of legal, social, ethical, technical, data, machine learning, and user considerations. Compliance with privacy rules, a patient-centred approach, ethical standards, and the use of robust data and algorithms are all critical. By combining these features with a user-centric design, you can create a chatbot experience that is dependable, transparent, and effective. As medical technology advances, it is critical that these systems not only meet to strict standards but also continuously adapt to serve consumers responsibly and efficiently.

# **System Design and resources**

## **System Design**

The process of defining the components or modules for a system to meet defined criteria is referred to as system design. The purpose is to decide on the high-level structure and behavior of the system. A well-defined system design can be visualized as a plan for the system's development, outlining its components, and integration (Yau and J-p Tsai, 1986).

The development of a medical diagnosis chatbot involves several modules or components, from data collection to serving the model on a React Native mobile application. In the system design, MLOPs (Machine Learning Operations) will also be incorporated. The system will include the MLOPs process of hosting, monitoring, and replacing the ML model versions. In this system design, containerize solution for the development of medical chatbot to ensure scalable, portable, and consistent environment.

[Figure 1](#_13.1._Figure_1) demonstrates the end-to-end functionality of medical chatbot diagnosis system from data collection to ML (Machine Learning) model monitoring with docker containerization. The system has two life cycles, development life cycle and ML operations. From the [figure 1](#_13.1._Figure_1), system has data collection phase which is the initial part of this system design. In data collection phase, data can be collected from different source such as snowflake, databricks, mysql, mongo DB etc. For this medical chatbot diagnosis system, the doctor-patient dialogues dataset has been collected from Kaggle datasets (DiagNoise-Me) to train two different type of deep learning neural networks.

The components of development life cycle are data preparation, model training, model evaluation and validation, and model analysis. On the other hand, ML operations has version control, CI/CD pipelines, model registry, model serving and model monitoring. Let’s delve into each component.

### **Data Collection and Analysis**

The initial phase of medical chatbot development involves data collection. The dataset of doctor-patient dialogues will be extracted from the data source named Kaggle. The dataset will be then loaded into the python environment for EDA (Exploratory Data Analysis). EDA is critical for understanding the dataset's features, potential abnormalities, and patterns. Following an intensive investigation, becomes the primary input for the next data preparation or pre-processing phase.

### **Data Preparation**

From the EDA, the raw data is cleaned during this phase. It entails thorough cleansing of the doctor-patient dataset to ensure that inconsistencies, outliers, or missing values are addressed appropriately. Furthermore, the data is prepared to meet the requirements of the selected machine learning models for example, data format and tokenization according to the selected machine learning model. The result of this phase is a well-prepared dataset, which serves as the foundation for model training.

### **Model Training**

With the data prepared, the focus shifts to model development. The proposed system leverages two types of neural network architectures: RNNs and Transformers. TensorFlow is used to create an Encoder-Decoder LSTM model, and pre-trained models such as GPT-2, BART, and T5 are drawn from the Transformer package. Once the training process completes, the models are saved, leading us to the model evaluation phase.

### **Model Evaluation**

In this phase the performance of the trained models will be evaluated. To assess the model's ability, metrics such as perplexity, BLEU score, ROUGE score, accuracy, and loss value are used. This evaluation provides useful information by assessing if the model is ready for validation or needs further development.

### **Model Validation**

Validation serves as a checkpoint to ensure that the model generalises correctly. The model's predictions are verified against actual scenarios using a separate validation dataset. Depending on the results, the process may proceed to model analysis or model serving phase.

### **Model Analysis**

This is an introspective phase in which the model's performance during validation is evaluated. To further optimise the model, techniques such as quantization or pruning may be used. If there are any inconsistencies or poor performance issues, there are two options: start retraining or alter the existing models, with the latter generally avoiding the requirement for training from scratch.

### **Version Control**

It is critical to manage multiple model iterations in a dynamic context. GitHub can be an excellent platform for version control, with capabilities such as rollback that allow you to revert to prior model versions if necessary.

### **CI/CD (Continuous Integration / Continuous Deployment) Pipeline**

The pipeline enables the transition from development to deployment easier. Processes like as unit testing and integration testing assure the system's robustness. The models are packed and deployed once they have been cleared. GitHub Actions can be useful for automating this pipeline.

### **Model Registry**

It is now possible to manage model versioning and trace model lineage easily. In this system design TensorFlow Hub will be utilised for this phase. By using this, registry keeps an organised record of model iterations and their respective performance metrics.

### **Model Serving**

The model enters the serving phase after successfully training and validating it. Serving is where the end-users interact with the system for the purpose of use. In this medical diagnosis chatbot, TensorFlow Serving, the "SaveModel" format, will be used to make the model available to end users.

### **Mobile Application**

In this component of the system, the chat interface that end-users interact with will be implemented using React Native application development framework. This application will connect with the backend via REST APIs. The user will be able to ask queries by using this interface and will get the responses of the model from the backend (develop with flask).

### **Backend Development**

The backend acts as a middleman between the mobile application and model serving layers. It handles the data flow by delivering user queries to the model, receiving processed data, and ensuring the front end receives timely and accurate results. Flask python framework will be used for the backend development of the system.

### **Docker Containerization**

Both the frontend (mobile application) and backend (a Flask application) are containerized using Docker to create a consistent, scalable, and portable environment. This ensures that the system remains consistent and functioning regardless of where it is deployed.

### **Database Management**

A database is a structured collection of data that is electronically stored and accessed. Databases are meant to store data in a structured manner, making it simple to obtain, manipulate, and save data. For the medical chatbot, MySQL relational database management system will be used to store and retrieve user conversations, offering insights into user behaviour and facilitating continuous model improvements.

### **Model Monitoring**

This is the last phase or component of medical diagnosis system design. As the system goes live, it is critical to monitor the model's real-world performance. Continuous monitoring helps in the detection of anomalies or decreases in performance. If problems arise, it functions as a trigger, pushing the system back into the development phase and ensuring the model continues to perform optimally.

In conclusion, system design is the design plan for developing systems, detailing components, structure, and behaviour. The design of the medical chatbot diagnosis system encompasses everything from data collection, model training and deployment via a React Native application. For excellent real-time performance, the focus is on developing models, evaluation, validation, serving, and continuous monitoring.

## **Available Hardware Resources**

In machine learning, developing, or training a model requires hardware resources, which plays a pivotal role. There are mainly two reasons behind it, computational intensity, and data size. For the development of medical chatbot, NLP tasks, often involves millions, or billions of parameters. Training such models requires great amount of computational power. Moreover, training on large datasets can also be memory intensive. The dataset consists of patient-doctor dialogues can be vast and may requires a large memory to be fit in.

This research possesses, single GPU with a total memory of 24,268 MB and a CPU memory of 15,888.04 MB, which shows the powerful machine for training small-scale machine learning models. However, in NLP, large volume of data cannot be used rather than a subset of data. Moreover, this limitation also restricts to the smaller batch size during the training of the model, which can significantly take more training time, might take days or even months.

# **Data collection and data analysis**

In this research, a dataset containing doctor-patient dialogues has been used to develop a chatbot to facilitate doctors for medical diagnosis. The dataset's main goal is to make it easier to create intelligent dialogue systems that can help doctors diagnose patients more accurately. AI models can learn from real-world medical interactions, increase the precision of diagnoses, and improve patient care by examining these dialogues.

The dataset is important because it has the potential to transform the healthcare sector by utilising cutting-edge natural language processing (NLP) and dialogue modelling methods. Doctors can benefit from useful decision-support from dialogue systems trained on this data, which can help them identify complex medical diseases and prescribe appropriate treatment recommendations.

## **Data collection**

A large and diverse collection of conversations between doctors and patients is necessary for training data to create efficient medical dialogue systems. However, obtaining such data while ensuring data privacy is a challenging task. The creation of reliable and adaptable medical dialogue systems is made difficult by the size limitations or disease-bias of the existing medical dialogue datasets. To address this issue, as written in (He et al., 2020), the researchers of this study, created two large-scale medical dialogue datasets: MedDialog-EN in English and MedDialog-CN in Chinese. For the development of medical diagnosis based chatbot, MedDialog\_EN has been used containing 257,454 doctor-patient dialogues. However, MedDialog-CN represents an extensive collection of 1,145,231 Chinese dialogues between patient and doctors (He et al., 2020). The choice was made because English is a universal language and can target users globally. I tried to translate the "MedDialog-CN" dataset from Chinese to English using Python and the Google, Microsoft, and Yandex APIs. However, these APIs only offer a limited number of free characters for translation, I ran across serious limits. Furthermore, it turned out to be extremely expensive, costing more than $6000, to translate a large amount of Chinese data.

The dataset "DiagNoise-Me" available on Kaggle (https://www.kaggle.com/datasets/dsxavier/diagnoise-me?select=en\_medical\_dialog.json) is a valuable resource, used to collect MedDialog-EN data for medical dialogue chatbot system. This dataset is provided in the JSON (JavaScript Object Notation) format.

## **Data description**

According to (He et al., 2020), MedDialog-EN contains 257,469 English patient-doctor dialogues. The dataset contains 514,938 (Patient and Doctor) utterances in all, equally split between patients and doctors. The information was carefully gathered from websites like [iCliniq - Online Doctor 24/7 | Expert MDs Answer Your Questions](https://www.icliniq.com/) and [HealthCareMagic | Online Doctor Consultation Clinic | Online Doctor Appointment 24x7](https://www.healthcaremagic.com/) that provide a variety of healthcare services like symptom checkers, video consultations, and online doctor chats.

Each dialogue in MedDialog-EN consists of two fundamental components.

* **Description of Medical condition:** The patient's medical condition, symptoms, medical history, medications, allergies, and other significant information pertaining to the medical situation are all covered in detail in this section.
* **Conversation between Patient and Doctor:** The main conversation between the patient and the doctor is shown, resembling the normal progression of a medical consultation. Patients discuss their medical issues with clinicians, who then provide professional guidance, possible diagnosis, treatment suggestions, and further questions to gather more information.

## **Advantages of MedDialog-En Dataset**

MedDialog-EN provides exceptional benefits because to its large number of answers questions and utterances, making it one of the most complete medical dialogue datasets available. With a massive amount of data at its disposal, the dataset enables the construction of strong and complex medical dialogue systems through training and fine-tuning. Furthermore, the datasets reveal comprehensive coverage of medical specialisations, including a vast variety of healthcare disciplines ranging from general internal medicine to highly specialised branches. This variety ensures that the dialogue systems can efficiently manage a wide range of medical conditions and events, catering to the different demands of patients and medical professionals alike. Furthermore, the MedDialog-EN records represent patients from various areas and backgrounds, capturing varied nations, ethnicities, ages, genders, occupations, education levels, and incomes.

## **Limitations of MEdDialog-En Dataset**

When using the MedDialog-EN dataset for the development AI-driven medical dialogue systems, various constraints must be recognised. The dataset's acquired primarily from specialised internet platforms such as [iCliniq - Online Doctor 24/7 | Expert MDs Answer Your Questions](https://www.icliniq.com/) and [HealthCareMagic | Online Doctor Consultation Clinic | Online Doctor Appointment 24x7](https://www.healthcaremagic.com/). This narrow scope may not adequately represent the entire range of medical interactions, which may limit the generalizability of dialogue systems trained on this dataset to real-world medical scenarios. Another limitation is, the information spans just 2008 to 2020, it may not include the most current advancements in medical practise and patient-doctor connections. As the healthcare landscape is always changing, new medical diseases, treatments, and communication patterns that have evolved in recent years may be not represented in this dataset (He et al., 2020).

## **Exploratory Data Analysis**

#### Exploratory Data Analysis (EDA) is a process that involves analyzing datasets to identify and summarize their primary characteristics, frequently with the help of statistical and visualization techniques. EDA seeks to find patterns, trends, anomalies, relationships, and insights in the data as well as possible issues and areas suitable for additional investigation. Python is a popular language for doing EDA because it is easy to use, flexible, and have many libraries for data processing and visualization. Python has been used in this study to check the insights of the doctor-patient dialogues. Let’s check the insights by importing the dataset in python. I have converted the dataset from the JSON format to pandas DataFrame for better analysis.

### **Dataset Basic Information**

In [Figure 2](#_13.2._Figure_2), shows the summary of the doctor-patient dialogues dataset after using the info() method of pandas DataFrame in python. This DataFrame provides the summary of the structure, including columns, datatypes, non-null counts, and memory usage. It's especially beneficial for quickly gaining an overview of the dataset and checking for missing values. From above it can be seen that there are **4** columns in the dataset, id, Description, Patient and Doctor. The “id” column is the unique identifier, description is the overall summary of each dialogue, patient and doctor column contains the conversation between doctor and a patient in each dialogue. There are **257,469** dialogues or entries in this dataset.

[Figure 3](#_13.3._Figure_3), shows the top 5 entries in this dataset, which has been retrieved using the head() method of the pandas DataFrame. By default, this method or function displays the first **5** entries of the dataset. From the above table, each entry has conversation between doctor and patient. If you look at the second entry in doctor columns and fourth entry in the doctor’s column, you may see the duplication of records. It is essential to remove duplicate records during pre-processing to guarantee the model's accuracy and effectiveness throughout training. Duplicates can increase the training time, cause overfitting, and skew the outcomes of data analysis.

[Figure 4](#_13.4._Figure_4), shows that the id values are likely sequential integers, starting from **0** and going up to **257,468**. The mean is roughly half of the maximum value, and the standard deviation suggests a uniform distribution. However, in this case there is no need of “id” column, will be removed in the pre-processing section.

#### **Missing Values**

There are no missing values or missing cells in the datasets, as shown in [Figure 5](#_13.5._Figure_5), hence no missing values handling strategies are required.

#### **Check Repetitive Dialogues**

The custom function has been written to check repetitive dialogues in the dataset. From the [Figure 6](#_13.6._Figure_6), it can be seen that the function visualize\_sentence\_occurrences() counts how many times a specific sentence appears in three fields (Description, Doctor, and Patient) of each entry (or conversation) in the dataset. From the output the sentence "What does abutment of the nerve root mean?" occurred **758** times in the dataset. There might other sentences which are more than **1** in the data and might be possible there are some which occurred only at once. Removal of duplication of records is necessary to avoid overfitting of the model and avoid unnecessary increase in training time.

From the [figure 7](#_13.7._Figure_7), one contraction can be seen "cant" in the dataset. There are also some other formal and informal contractions in the dataset have also been observed. It is necessary to handle these types of contractions from the dataset because some of them do not convey any meaning to the model and can affect the context.

### **Text Data Descriptive Analysis**

In this section I will calculate various statistics and characteristics from a text dataset. The purpose of text data descriptive analysis is to get the insights from the data and identify potential data quality issues. It will also help in the selection of vectorization techniques selection (like word2vec). Moreover, also tells removal of some words just like stopwords (like "and", "the", "is", etc.) as for some of the NLP tasks there is no need of stopwords, however, for some might need it. In the context of medical diagnosis support we need to stop words because by removing it we might lose the context of the conversation between doctor and patient.

#### **Text analysis tasks**

Following is the list of text analysis tasks which you will see in this section.

* **Unique Word Count:** The number of unique words across each column.

**Reason:** It helps in understanding the diversity of the vocabulary in the dataset. Many unique words might suggest the need for dimensionality reduction techniques or word embeddings like word2vec when vectorizing text. On the other hand, less unique word count indicates a narrow or domain specific text data.

* **Average Word Length:** The average length of words when considering all the words in the dataset.

**Reason:** To check the nature of the dataset. Some URLs and technical or scientific text might give longer average word length. This can also influence decisions related to tokenization and stemming**.**

* **Stopwords Count:** The number of commonly used words (like "and", "the", "is", etc.)

**Reason:** Stop words do not carry significant meaning in many NLP tasks but for some tasks like I am going to build "medical chatbot or medical dialogue system" it may carry contextual information. However, for the tasks like text classification, semantic analysis we do not need it.

* **Frequency Distribution of words:** The ten most frequently occurring words in the dataset, along with their respective frequencies.

**Reason**: This can give a quick idea of the topics in the dataset.

* **Total Word Count:** The cumulative count of all words across all documents.

**Reason:** Overall volume of the textual content. High word count needs more computational resources for the processing.

* **Punctuation Frequency:** The count of each punctuation frequency across all documents.

**Reason:** It is necessary to check the punctuation frequency to simplify the text to process and analyse, can help standardize the input, to focus on semantic meaning and to improve token embeddings.

[Figure 8](#_13.8._Figure_8) shows the total number of entries or rows or records or documents in each column. All the three columns having the same number of entries which are **257,468.**

The density of unique words in the total number of words is shown in [figure 9](#_13.9._Figure_9). There are **5,27,01870** total words and **6,42,374** in the dataset of doctor-patient dialogues.

[Figure 10](#_13.10._Figure_10) shows the word length analysis in the dataset. The average length calculated in the dataset is **4.18**, minimum word length is **1** and the maximum word length is **222**. The maximum word length is because of some URLs. In the pre-processing, the URLs will be removed as these don’t contribute meaningful information to the model.

[Figure 11](#_13.11._Figure_11) depicts the stop words count vs total words count in the dataset. There are **19,128860** stop words out of **5,27,01870** total words., which is **36.30%** of the dataset. Dealing with the stop words is critical, some of the NLP tasks don’t need stop words because of the little meaning in the context and to reduce the noise in the data. However, in the case of medical chatbot stop word removal may loss the contextual information. Therefore, in the pre-processing stage, there will be no stop words removal techniques will be performed.

[Figure 12](#_13.12._Figure_12) is the analysis of top 10 frequent words in the dataset. In this case, mostly the stop words are the most top 10 frequent words in the dataset. This shows that more than topic words there are stop words.

The bar plot in [figure 13](#_13.13._Figure_13), represents the punctuation marks on horizontal axis and their frequency on the vertical axis. The total number of punctuations in the dataset are **7,162,754** and the punctuation mark **dot (.)** used the most, around **3,800,000**. Furthermore, we can also see other punctuation marks like **“? , / ‘”** etc. It is necessary to remove punctuation marks to simplify the dataset, consistency of dataset and for the noise reduction. However, sometimes removal of punctuation marks especially “**?”** may loss the context.

In the context of medical chatbot and the type (Patient-Doctor Dialogues) of dataset, better is not to remove the punctuations, as the presence of “**?”** can provide the contextual information. For example: Flu is a virus? or Flu is a virus. In these both cases context has been changed due to the presence of punctuations. Moreover, some pre-trained models like GPT, BART etc. was trained with complete sentences that included punctuation. When you remove punctuation and then fine-tune on this modified data, there might be a misalignment between what the model originally learned and the new training data.

Understanding the characteristics and statistics of the dataset is critical when creating deep learning models, particularly transformers like GPT-2 or BART. The dataset's size of **257,469** (each column) entries or documents or rows is critical in establishing fine-tuning effectiveness and computational resource needs. The total word count of **5,27,01870** offers information about content volume and memory requirements. A vocabulary of **642,374** unique words indicates the need of aligning with the vocabulary of the pre-trained model to mitigate out-of-vocabulary issues. The average word length of **4.18** suggests language complexity, with longer words implying technical discussions common in medical situations. Furthermore, the stop words count, which stands at **19,128860**, influences preprocessing decisions about whether to filter out "noise" or keep it for the contextual meaning in the doctor-patient dialogues dataset. Finally, the top ten most frequent words provide a measure of the dataset's domain relevance, with medically relevant words emphasising its richness in medical content.

#### **Distribution of text lengths**

[Figure 14](#_13.14._Figure_14) depicts the distribution of text lengths in the dataset of doctor-patient dialogues. From the visualizations, the text length for descriptions is shorter compared to both doctors' responses and patients' queries. The maximum text length for a description column peaks at about **300** words. In contrast, doctor responses approximately **11,000** words, and patient queries approximately **17,500** words. This suggest that while descriptions are concise and to the point, patients' queries and doctors' responses in the medical dialogue system tend to be much more detailed and extensive.

From the [figure 15](#_13.15._Figure_15), boxplots illustrate the presence of outliers in the dataset. There is need to remove these outliers during the preprocessing stage before training the model. Furthermore, some pre-trained models have a preset maximum text length, so we must modify the content to fit inside these restrictions.

Additionally, it makes sure that machine learning models are trained without biases towards text lengths and helps spot potential data abnormalities, including unusually short or long entries.

In conclusion, python was used to do exploratory data analysis (EDA) on a dataset of doctor-patient conversations. The dataset, comprising 257,469 dialogues, highlighted potential duplicate records and repetitive dialogues. Initial analyses did not identify any missing values; however, they did identify text length outliers and possible data anomalies. While there are 5,27,01870 total words with 6,42,374 being unique, stop words account for 36.30% of the dataset, and their retention is important for preserving context in medical dialogues. Moreover, punctuations, especially the question mark, plays an important role in maintaining contextual meaning.

# **Methodology**

The methodology section of the paper is divided into five parts. The first part, General Data Pre-processing, focuses on basic tasks such as cleaning and standardizing the dialogues, preparing them for the machine learning modelling. The second part, Model Architecture Selection and Justification, delves into the reasons behind the selection of model architectures suitable for medical-based chatbots. The third part, Model Specific Pre-processing, highlights additional steps required to adjust the data with the specific model's requirements. The fourth part, Model Training and Hyperparameter Selection, gives the training process and best possible hyperparameters of each model. The last part of the methodology, Model Evaluation, detailing the methods used to assess the medical chatbot's capability in handling medical queries.

## **General Data pre-processing**

Data pre-processing or preparation in machine learning is the process of getting the raw data cleaned and organise so that it can be used for training and developing machine learning models. It is very useful process in order to utilise computational resources effectively and get better results especially in terms of performance and accuracy of the machine learning models.

Pre-processing datasets, like conversations between doctors and patients, is crucial for enhancing data quality, relevance, and model accuracy. The text must be cleaned up and standardised, tokenized, dealing with stop-words and stemming or lemmatization must be handled. The context must be preserved, textual data must be encoded for machine learning applications, the dataset must be balanced. Proper pre-processing ensures meaningful analysis, especially in sensitive contexts like healthcare.

Text data in NLP (Natural Language Processing) can be divided into many types, each with its own linguistic characteristics, such as posts on social media, literature, academic publications, news, user reviews and conversations. Depending on the type of dataset and machine learning technique, several preprocessing techniques are used, including as tokenization, lowercasing, stop-word removal, lemmatization, and handling of URLs, numerals, emojis and much more. For example, in the context of medical dataset information are so important, it is important to retain stop-words during doctor-patient conversations while in text classification, better to remove stop-words.

Upon analysing the data through EDA, the following methods has been applied to ready the dataset for machine learning model development.

### **Remove ID Column**

[Figure 2](#_13.2._Figure_2), shows the presence of “id” column in the dataset, used as a unique identifier. The "id" column in the dataset is just and identifier of each dialogue and does not provide any semantic meaning. Moreover, it can introduce the noise without benefiting the performance of the model so better is to remove the "id" column from the dataset. The “id” column from the dataset has been removed by using drop ("id", axis=1) method.

### **Remove Duplicate Dialogues**

From [figure 6](#_13.6._Figure_6), duplication of dialogues can be seen in the dataset. To ensure effective training and avoid overfitting, removing duplicate dialogues from doctor-patient datasets is crucial for the development of medical chatbot. The model's understanding may be biased because of duplicates, which also limits its ability to generalise to new scenarios. The custom python function has been written to remove the duplicate dialogues. It iterates through each dialogue and save the unique dialogue in the unique dialogues list if the dialogue appears again than not saving anywhere.

[Figure 16](#_13.16._Figure_16) shows that the statement which occurred 758 times in the data now been appearing only once.

### **Convert to Lowercase**

Converting to the lowercase is a common pre-processing technique which helps in text normalization, case-insensitive matching and reduce the dimensionality of machine learning models. The reduction in dimensionality means two same texts will be treated as one e.g., "TEXT" and "text" will be treated as one token, ultimately results in less use of computational resources and the model will perform better.

The lower() method associated with string objects in Python has been used to convert all the text into lowercase.

### **Remove Irrelevant Characters**

Irrelevant characters in the dataset can introduce inconsistencies, removing them can ensure data cleanliness and consistency which can improve the accuracy of the machine learning model. Moreover, unnecessary characters can increase the use of computational resources as well.

As described earlier, the presence of “**?”** andsomepunctuationscan provide the contextual information. Therefore, a regular expression pattern has been used to remove and retain the specific punctuation. The “re” package from python has been used to remove the irrelevant characters with the usage of "[^a-zA-Z0-9\s.,;?!'\"-]” pattern. The pattern removes all the irrelevant characters but keeps letters, numbers, whitespaces, and common punctuation marks like "?".

### **Handle Contractions**

In the EDA part, an example has been shown of contraction "cant", these type of contractions can cause the data inconsistency, ambiguity, and can reduce model accuracy. Therefore, the custom function has been written with the use of ‘re’ package to replace the contractions to their original form, with the list of ‘replacements’.

### **Spell Correction**

Incorrect spellings can result in data inconsistency and misinterpretation of the model. In the context of medical chatbot it is mandatory to generate appropriate responses from the system. Therefore, TextBlob, a python package or library has been used to correct the spellings in the dataset.

### **Remove URLs**

In the medical chatbot dialogue system, URLs are often removed to maintain the data's cleanliness and the model's focus on the semantic content of the dialogue because URL's do not carry semantic meaning related to the medical content. The presence of URL's can introduce the noise, therefore, better is to remove the URLs. The “re”, python package or library has been used with the regular expression pattern “'https?://\S+|www\.\S+|\S+\.com\S\*'” to remove the URLs from the dataset.

### **Handle Stop-words**

Removing stop-words could result in the loss of some contextual meaning, particularly in a dataset of doctor-patient dialogue format where each word could sentiment or crucial information. However, it can consume a lot of computational resources as the dataset have **19128860** stop-words but retaining the sentiment is more important. Therefore, it has been decided not to remove the stop words from the dataset.

### **Handle Lemmatization**

Lemmatization is the process in NLP (Natural Language Processing) in which words are converted to their root form, e.g "going" to "go" , "is" "am" "are" to "be". Lemmatization can help in reducing the dimensionality of the data by converting words to their base form and can be computationally effective. The usage of lemmatization depends upon the type of problem, dataset, and selection of machine learning model. It might result in loss of context, especially in a medical chatbot and the type of dataset (conversational; doctor-patient dialogues) preparing.

[Figure 17](#_13.17._Figure_17) depicts, one machine learning model has been trained with lemmatization while inferencing some grammatical incorrectness has been seen, “flu be a viral”. Lemmatization has also resulted in loss of tense and tone. Therefore, for training the machine learning models lemmatization has not been considered in pre-processing. Moreover, neural architectures (sequence to sequence) have been implemented in the development of medical chatbot which can handles the context and can often handle various forms of a word.

Above are the pre-liminary pre-processing techniques used in this research for all the machine learning architectures that have been used. There are **model-specific pre-processing** techniques which have been described in the part C of this methodology section after giving the justification of architectures selection.

## **Model Architecture Selection and Justification**

When using NLP for the creation of medical chatbot choosing the appropriate model is necessary. There are various types of machine learning models which can handle NLP tasks, the common three types include One-to-Sequence, Sequence-to-One, and Sequence-to-Sequence.

**One-to-Sequence:** This type of model accepts a single data point as input and returns a sequence as output. Image captioning is an example where, given an image (a single input), the model outputs a descriptive caption (a sequence of words) (Xu et al., 2022).

**Sequence-to-One:** In this type, the model accepts a sequence as input and outputs a single data point. Sentiment analysis is a common example of this use case where a sentence or paragraph (a sequence of words) is evaluated to establish its overall sentiment, which can be classified as positive, negative, or neutral (a single output) (Chan et al., 2023).

**Sequence-to-Sequence:** This model type takes a sequence as input and output a sequence. Machine translation is an example (Sutskever et al., 2014). When an English sentence (an input sequence of words) is translated into for example Spanish (an output sequence of words), the model produces another series of words in Spanish.

For medical chatbot, the machine learning model's task will be to receive user queries (input sequences) and produce appropriate responses (output sequences). According to the nature of interaction, where both the input (user query) and the desired output (chatbot's response) are sequences, the **Sequence-to-Sequence** modelling approach is appropriate for the system. This is due to the ability of Sequence-to-Sequence models to turn one set of ordered data into another, making them perfect for tasks such as dialogue systems where the goal is a meaningful and contextually relevant communication. Furthermore, in the medical domain, where user queries can be varied and complex, it's essential for the chatbot to interpret the context and produce responses that are not only accurate but also coherent and contextually relevant.

Deep learning architectures often use a combination of recurrent neural networks (RNNs), long short-term memory networks (LSTMs), and the transformer architecture to handle sequence-to-sequence tasks. In sequence-to-sequence models, for example, an encoder processes the input sequence, and a decoder generates the output sequence. Two different types of neural architectures have been chosen for the development of medical chatbot, LSTM and Transformer. From the LSTM, encoder-decoder LSTM type has been selected and from Transformer three different pre-trained models have been selected (GPT-2, BART and T5).

* Encoder-Decoder LSTM (Long Short-Term Memory)
* GPT-2 (Generative Pre-trained Transformer)
* BART (Bidirectional and Auto-Regressive Transformers)
* T5 (Text-to-Text Transfer Transformer)

### **Encoder-Decoder LSTM (Long Short-Term Memory)**

LSTM (Long Short-Term Memory) is a type of recurrent neural network (RNN), which functions similarly to a brain in identifying patterns in a series of data. Unlike typical RNNs, which can forget items from a long time in the past, LSTMs feature a unique design with three gates (input gate, forget gate, and output gate) that help with memory. This makes LSTMs ideal for tasks such as language translation and comprehending spoken speech.

The Encoder-Decoder LSTM architecture is made up of two primary parts. The encoder processes each item in the input sequence and generates a "context" vector that represents the prominent aspects of the sequence. The context vector is then used by the decoder to construct an output sequence, which can be word-by-word or character-by-character in language tasks (Alakkari et al., 2023).

**Justification**

The Encoder-Decoder LSTM is suitable for medical chatbots because it can handle both short and comprehensive doctor-patient conversations. It is effective in recalling the context of health conversations because of its design, which guarantees that it remembers important details. Its versatility makes it appropriate for comprehensive medical conversations and can be utilised for a variety of jobs. It's able to provide appropriate responses even with limited data. In addition, it can be modified to bring attention to important sections in dialogues, which is particularly helpful when some medical terminology or phrases are more crucial than others.

### **GPT-2 (Generative Pre-trained Transformer)**

The transformer architecture serves as the foundation for the OpenAI model GPT-2. It is intended to provide text over long periods that is coherent and contextually relevant. GPT-2 can produce writing in a range of styles and on a variety of topics due to pre-training on a sizable corpus of internet-sourced content. The term "pre-trained" refers to the initial phase of training on a large amount of data before any specific fine-tuning (Schneider et al., 2021).

**Justification**

GPT-2 is a language model that can be fine-tuned using conversations between patients and doctors. It's good at creating clear answers, can talk about a wide range of medical topics, and is smart at understanding the context from conversations, like a patient's symptoms or history. Since it was also trained on the medical domain, it can also function well even when there is a lack of medical conversation data. Nevertheless, care is still necessary despite these advantages. To ensure that GPT-2's responses are accurate and secure, extensive checks and professional assessments are required.

**Version**

GPT-2 medium, version has been used in this project for the training on patient-doctor dialogues. GPT-2 medium has approximately 355 million parameters. The training time is shorter as compared to the models (like BART) having the same number of trained parameters. This version of gpt-2 primarily also designed for generative tasks.

### **BART (Bidirectional and Auto-Regressive Transformers)**

BART is a model created by Facebook AI that uses transformers to work on both the encoder and decoder sides, like how GPT-2 and BERT do. BART combines the strengths of models like BERT, which are excellent at decoding coherent sequences of text with models like GPT-2, which are excellent at comprehending the context from supplied text (encoder tasks) (Lewis et al., n.d.).

**Justification**

BART is a model that specialised at both comprehending and generating text. It excellent at taking details from conversations, especially because of its capacity to consider data from both the start and end of a dialogue. This ability enables it to produce accurate and relevant responses, which makes it an excellent choice for medical chatbots. Its versatility and ability to provide in-depth responses allow it to answer queries as well as write text, ensuring that patients receive information that will be precise and clear.

**Version**

Facebook/bart-base, version has been used in this project for the training on patient-doctor dialogues. Facebook/bart-base has approximately 140 million parameters. The training time can be time consuming, but it can handle variety of tasks due its architecture.

### **T5 (Text-to-Text Transfer Transformer)**

The T5 model, which was developed by Google Research, is based on the idea that practically all NLP problems can be reduced to text-to-text issues. In other words, the input and output can both be treated as text for any given purpose, be it translation, summarization, question answering, or another. The model is uniformly trained to translate textual inputs into textual outputs (Ni et al., 2021).

**Justification**

T5, with its text-to-text design, offers a streamlined approach suitable for building medical chatbots using doctor-patient dialogues. It's pre-trained on vast texts and can be further refined for medical conversations, making it both versatile and effective. Whether it's giving brief advice or going into detailed topics, T5 is adept due to its proven performance and deep contextual understanding, important for sensitive medical conversations.

**Version**

T5-small, version has been used in this project for the training on patient-doctor dialogues. T5-small has approximately 60 million parameters.

## **Model Specific Pre-processing**

Model-specific preprocessing is the process of getting input data prepared and formatted in a way that's unique to a specific model architecture or type. The shape, type, and format of the input data that they accept may differ according to different models or architectures. Let’s discuss the specific pre-processing techniques for the models that has been used in this research or project.

### **Encoder-Decoder LSTM (Long Short-Term Memory)**

Following are the pre-processing techniques has been performed for the encoder-decoder LSTM model after cleaning the dataset.

#### **Data Formatting**

When developing a medical chatbot with an encoder-decoder LSTM model, the encoder reads the patient's questions and turns them into a sort of summary or "context". Then, the decoder uses this context to produce appropriate answers from the doctor.Special tokens like "start" and "end" added to the patient's questions and the doctor's answers. These tokens make it easier for the encoder and decoder to distinguish between different dialogue interactions in our data by helping them in identifying the start and end of sequences.

#### **Tokenization**

The raw text data cannot be processed directly by neural networks. Therefore, the text data, in this case the patient's questions and the doctor's answers, transformed into numerical sequences so that the LSTM can understand it.

For the LSTM model “Tokenizer()” method from keras package been utilised. This tokenization step allows the model to analyse patterns and relationships between different queries and responses, which forms the foundation for generating relevant answers.

#### **Handle sequence length**

Neural networks require input data with a consistent shape. However, in natural language sentences can vary to different length, to resolve this I added pad sequences to the shorter ones and truncated those having greater lengths, results in same length of sentences. This ensures that each input fed into the neural network has same shape.

For the LSTM model I capped sequence at a length of 256 due to limitation of computational resources. It tried with more, but kernel was dying due to increase in length.

#### **Split dataset**

The training set and validation/test set are two sets that are typically separated from the available dataset in machine learning and deep learning. This separation helps in evaluating the model's performance on unseen data. The main goal is to prevent the model from overfitting the training set of data. If a model performs exceptionally well on training data but poorly on validation data, it is obviously overfitted.

Data splitting has been performed using the “train\_test\_split” method of scikit learn library. The parameter “test\_size” set to “0.2”, indicates that 20% of the dataset used as the validation set, while the remaining 80% used for training.

### **GPT-2 (Generative Pre-trained Transformer)**

Following are the pre-processing techniques has been performed for the GPT-2 medium model after cleaning the dataset.

#### **Data Formatting**

For the GPT-2, it's important to tell the model context and roles. I have re-format the data by adding "Patient:" or "Doctor:" before each dialogue, this makes the model understand the difference between the two roles. This helps the model learn and reproduce these roles better during training.

#### **Tokenization**

GPT2Tokenizer, from the transformer library has been used for the tokenization of the dataset for gpt-2 medium training.

#### **Handle sequence length**

To limits the number of tokens in each training example “block\_size” parameter of “TextDataset” method (from transformer library) utilized. The block size of 128 length passed to ensure the consistent input lengths, which is important for batching in the network.

#### **Split dataset**

To evaluate a model's ability to generalize, data splitting has been performed using the “train\_test\_split” method of scikit learn library. The parameter “test\_size” set to “0.2”, indicates that 20% of the dataset used as the validation set, while the remaining 80% used for training.

### **BART (Bidirectional and Auto-Regressive Transformers)**

Following are the pre-processing techniques has been performed for the "facebook/bart-base" model after cleaning the dataset.

#### **Data Formatting**

For BART, dataset converted into input and output list sequences as it requires a clear distinction between input and output sequences.

#### **Tokenization**

BartTokenizer,from the transformer library has been used for the tokenization of the dataset for BART training. Like other tokenizers this also converts text into a sequence of integers which represents each token.

#### **Handle sequence length**

For batching, BART, like other transformer models, needs input sequences to be a fixed length. This is handled by truncating and padding sequences. Truncation, trim sequences that exceed the model’s maximum length. Padding, sequences are padded to match the required length. The maximum length of “facebook/bart-base” is 1024 tokens.

#### **Attention Masking**

BART uses attention masks to distinguish between an actual sequence content and padding. This ensures the model doesn't pay attention to the padded tokens.

#### **Split dataset**

To evaluate a model's ability to generalize, data splitting has been performed using the “train\_test\_split” method of scikit learn library. The parameter “test\_size” set to “0.2”, indicates that 20% of the dataset used as the validation set, while the remaining 80% used for training.

### **T5 (Text-to-Text Transfer Transformer)**

Following are the pre-processing techniques has been performed for the "t5-small" model after cleaning the dataset.

#### **Data Formatting**

Just like BART, for T5, dataset also converted into source and targets list sequences as it requires a clear distinction between input and output sequences.

#### **Tokenization**

T5Tokenizer, from the transformer library has been used for the tokenization of the dataset for T5-small training. T5 uses a “SentencePiece” tokenizer, which is a data-driven, unsupervised text tokenizer and de-tokenizer primarily utilised for tasks involving the generation of text using neural networks.

#### **Handle sequence length**

To limits the number of tokens in each training example, the maximum length of 128 length passed to ensure the consistent input lengths, which is important for batching in the network.

#### **Attention Mask**

Including T5, attention mask is crucial for many transformer-based models. The attention mask ensures that the model doesn’t pay attention to the padded tokens which are not required.

#### **Split dataset**

To evaluate a model's ability to generalize, data splitting has been performed using the “train\_test\_split” method of scikit learn library. The parameter “test\_size” set to “0.2”, indicates that 20% of the dataset used as the validation set, while the remaining 80% used for training.

## **Model Training and Hyperparameter Selection**

In the context of machine learning and deep learning, model training is the process by which a model identifies patterns in data. In this process, data are fed into the model, predictions are made, the error is calculated using a loss function, and the model's parameters are modified to reduce the error (Wei et al., 2019).

To improve the performance of the model, hyperparameter selection refers to the optimisation of external parameters that are established before the training process starts. Grid search, random search, and Bayesian optimisation are a few techniques for hyperparameter tuning which help in determining the optimal combination of these parameters for best model performance (Luo, 2016). However, as these techniques are computationally intensive, for this project or prototype none of the hyperparameter techniques has been used due to the limitations of computational resources. I tried once with the small subset of the dataset but couldn’t proceed it as data size also plays a role in hyperparameter selection. For example, in the case of huge dataset there is less chance of overfitting so dropout value can be lower given by the hyperparameter tuning technique, however, in case of small dataset the dropout value should be higher to avoid overfitting. Additionally, in NLP tasks embedding layers, high dimensionality of text data and using pre-trained models can make a huge number of trainable parameters which requires substantial computational power for hyperparameter tuning.

Let’s delve into the discussion of model training and chosen hyperparameters of all the four models.

### **Encoder-Decoder LSTM (Long Short-Term Memory)**

#### **Hyperparameters**

The hyperparameters used are Dropout and Recurrent Dropout (both set at 0.3), Embedding Dimension (100), and 128 LSTM Units (2 layers both for encoder and decoder), batch size of 4 and 6 epochs for training.

#### **Configuration**

* The model is based on an encoder-decoder LSTM setup for sequence learning.
* Vocabulary size is sourced from the tokenizer and affects both embedding and output configurations.
* The encoder consists of two LSTM layers, uses embedding to transform token indices into 100-dimensional vectors, and added a 0.3 dropout rate for regularization.
* The decoder uses Bahdanau attention to weigh encoder outputs, consists of two LSTM layers to process the weighted data, and output with a dense layer having softmax activation to predict the next word from the vocabulary.

#### **Training**

The model uses the Adam optimizer and calculates loss using sparse categorical crossentropy. An early stopping technique is built up using a 4-epoch patience throughout training, giving attention to the validation loss. The training is stopped to prevent overfitting and returns to the most efficient weights if the validation loss does not improve over four consecutive epochs. Training is set for 6 epochs, but it might conclude earlier because of the early stopping mechanism.

### **Pre-trained models (GPT-2, BART and T5)**

Let's discuss the pretrained models' hyperparameters and training techniques. Because they all used the same transformer library, there are shared components amongst them.

#### **Hyperparameters**

* Batch Size: For the GPT-2 model, a batch size of 4 samples is selected. In contrast, for both BART and T5 models, the batch size is set at 2 due to memory limitations.
* Number of epochs: 2 epochs for training. Limited to 2 due to computational time considerations. As, increasing in number of epochs can take even weeks when fine tuning pre-trained models.
* Early stopping patience: Waits for 2 epochs without improvement before halting training. Interestingly, the patience period is the same as the total number of training epochs. This choice was made because, during training, there were noticeable high loss values. Thus, it seemed sensible to have the early stopping criteria align with the total epochs.
* Learning Rate: Used the default learning rate of the pre-trained models 0.001.

#### **Training**

A "Trainer" method from the transformer library has been used to set up the model with training arguments, data collator, training data, an early stopping mechanism, and a validation dataset. The training process than starts with the "trainer.train()" function, refining the model based on the given data.

## **Model Evaluation**

Model evaluation is the process of determining how well a trained machine learning model performed. The main objective is to evaluate how effectively the model predicts new, unforeseen data (Raschka, 2018). Understanding the performance of NLP models on certain tasks requires evaluation. Frequently, the NLP task at hand determines the proper evaluation metric. In the context of medical chatbot, where the problem type is text generation, perplexity and ROUGE score evaluation metrics are important (Omoregbe et al., 2020).

* + 1. **Metrics**

### **Accuracy**

Accuracy is a metric used to determine how well a model performs while answering questions using a dataset of conversations between a doctor and a patient. It specifically determines the proportion of times the model responds correctly to a patient's or a doctor's enquiry. Although accuracy is a simple metric, it's important to take the specifics and context of the dataset into account.

### **Loss**

Loss measures how incorrect a model's predictions are. In question-answering tasks using doctor-patient dialogues, a low loss means the computer's answers closely match real doctor responses. The aim is to reduce this score for better model performance.

### **Perplexity**

Perplexity checks how good a model is at guessing the next word. For medical chatbots, a lower value means the bot can chat more smoothly and make sense.

### **ROUGE (Recall-Oriented Understudy for Gisting Evaluation) score**

The ROUGE score looks at how much the predicted text matches the reference text using different measures like precision, recall, and F1-score. It's particularly useful for tasks like summarization to see how much key information the model includes in its output. In the context of medical chatbot, ROUGE can help determine how closely the generated response matches a desired or reference answer, indicating the system's ability to provide accurate and relevant information.

In a medical chatbot, simply using accuracy to measure its performance isn't enough. Medical chats can have several correct answers, and overlooking vital details isn't ideal. For instance, a bot might perform well with common medical issues but struggle with rare ones, making its accuracy misleading. To get a better understanding, should also check its efficiency on rarer conditions. Metrics like perplexity and ROUGE can be beneficial here. They compare the chatbot's output to expert opinions. While ROUGE assesses the overlap of words considering precision and recall. These scores give an idea of how closely the chatbot's responses align with a medical expert's advice.

* + 1. **Comparative analysis of evaluation metrics**

From the table in [figure 18](#_13.18._Figure_18), presents a comparison of various models based on key performance indicators. The Encoder-Decoder LSTM shows the highest accuracy at 0.990507 and a minimal loss of 0.074382. GPT-2 Medium, while having the highest perplexity, manages a Rouge-1 F-score of 0.511501. The Facebook/BART-base model displays a balanced performance with an accuracy of 0.920898 and Rouge-1 F-score of 0.950963. Conversely, the T5-small model exhibits the lowest accuracy of 0.004141 and a significantly high perplexity. The Rouge metrics further explain the capability of each model in generating coherent and relevant textual responses.

# **Results with Mobile Application**

The combination of machine learning and mobile applications is becoming common. In the context of medical chatbot, patients can get quick medical advice by having a conversation with a bot, almost like they would with a doctor, using an application from their phones. The trained machine learning models have been used to generate responses for the user queries, mobile application used to input the user query and display the generated response. The trained model has been served using backend (Flask; python-based framework) and utilised for the communication between machine learning model and mobile application. Let’s discuss briefly about model serving with flask and react native (mobile application development framework) before diving into the results.

## **Backend and Model Serving**

Flask python-based framework has been utilised for model serving to make it possible for mobile application and the medical chatbot to communicate with one another. Flask is also used to host the trained machine learning model, assuring its accessibility for real-time generation of texts. When a user of the application submits a question about their health, Flask forwards it to the hosted chatbot model. After the model has processed the query and determined an appropriate answer, it uses Flask to communicate the data back to the application. Flask is ideally suited for delivering and integrating our chatbot into the mobile application environment due to its Python foundation and ability to integrate with a variety of machine learning systems.

To handle all the models (Encoder-decoder LSTM, GPT-2, BART and T5), four separate endpoints or APIs have been set up. The reason for this is to generate the responses of each model on the front-end side or mobile application.

## **Securing the Flask Service with Docker (Containerization)**

While using Flask to serve the machine learning model across different environments, docker has been used to avoid environment specific issues. Docker packages the Flask service in a container, which ensures that whether it is set up on a local machine or a server, it works the same way without any dependencies conflict. The other reason for using docker is to easily create more instances of Flask service, making sure everything runs smoothly while keeping things secure.

## **Frontend (Mobile Application)**

React native, developed by Facebook, JavaScript based framework employed for building mobile applications both for Android and iOS platform. By using react native, don’t need to write platform specific code. This means that developers can write a single codebase in JavaScript and can deploy it across multiple platforms (Android and iOS), results in saving time and cost (Kaushik et al., n.d.).

There are some other frameworks like flutter, Ionic, Vue and Xamarin etc., however, react native has been chosen because of mature ecosystem, performance, code reusability, hot reloading, and native modules availability.

From the [figure 19](#_13.19._Figure_19) or above mobile application screenshot simple mobile application has been developed using react native. In this interface users can chat with “Medical Assistant”, both user’s and medical assistant’s text have been saved in a list format in respective properties for conversation display. User can ask query using the input field and send by pressing the send button. The send button triggers the function which has been written for the communication with flask server, send query and retrieve the chatbot’s or trained machine learning mode’s response.

## **User chat**

[Figure 20](#_13.20._Figure_20) displays a screenshot of a mobile app, showing a chat between a user and a digital medical assistant. This assistant is based on a model fine-tuned using the Facebook/BART-base architecture. The model is highly proficient, providing notably accurate responses. Evaluation metrics, such as an accuracy of 0.950963 and an F1 score of 0.937755, support this performance. However, the model is effective, there is scope for improvement. Even if they are correct, some responses may be structured more logically. Although its performance right now is excellent, with more fine-tuning, it may perform much better.

# **Limitations**

In this section several limitations have been discussed which impacted the outcomes of the project. However, these limitations highlight the areas for future enhancements.

## **Time constraints**

During the development of the medical chatbot prototype, there were challenges mainly training the models took a lot of time. Using natural language processing techniques with a dataset of 0.26 million doctor-patient dialogues, which is 295MB, made it even harder. Because of the large data size, the models took a long time to train. With a tight deadline for the project, some tough decisions had to be made. For many models, only a quarter of the dataset was used, and for the BART model, only a tenth was used. This has affected how well models performed, especially the pre-trained ones. Moreover, due to the short time, not all possible models for medical chatbots were looked at. Even though there are many models out there, only the ones that seemed to fit best with the data and goals of the project were chosen.

## **Hardware constraints**

During the model's training, adjusting hyperparameters was challenging. However, with only a single NVIDIA GeForce RTX 3090 GPU available, thorough testing of hyperparameters and using hyperparameters tuning techniques was difficult. Moreover, this single GPU setup prevented training multiple models simultaneously, extending the time needed for experiments. This hardware constraint made refining the model more time-consuming and posed additional challenges to the project's progress.

## **Dataset Limitations for comprehensive medical chatbot**

The doctor-patient dataset, with its 0.26 million dialogues and 295MB size, is still not broad enough for a medical chatbot aiming to cover all medical areas. The whole data is from doctor-patient dialogues, so adding other sources like medical books or research papers could help the bot understand better. However, this also means longer training times; for instance, training the GPT-2 model with only half of this dataset was estimated to take 338 hours, which can be seen in [figure 21](#_13.21._Figure_21).

Despite the limitations, the encoder-decoder LSTM and facebook/bart-base both performed well, suggesting it could do even better with more time and refining.

# **Discussion**

Machine learning and mobile apps are merging in our daily lives, popular examples are Facebook, uber and google translate. Machine learning models can also be integrated with the mobile application for getting medical assistant. For the interaction between mobile applications and machine learning models there are plenty of backend languages that can be used like Java, Nodejs and python etc. which act as middleware. However, in this project I have used python because it is popular for building machine learning models, and Flask Python-based framework used for writing endpoints for model serving. When user has a health query, Flask will communicate it to the machine learning model, model generates the response, and then user will get the response on the mobile application. I have utilized Flask to work with various models like LSTM, GPT-2, BART, and T5. This provides diverse response options, enhancing user experience. Additionally, I have containerized the flask application using Docker to ensure Flask's consistent functioning across different devices. This consistency is especially crucial for mobile apps handling such intricate models.

For the development of mobile applications, I have utilised React native, cross-platform application development framework, by Facebook. This enables the development of apps for both Android and iOS with one set of code, ensuring speed, code efficiency, and quick updates. Among choices like Flutter and Ionic, React Native has been chosen, primarily for the seamless chat experience with the "Medical Assistant".

During the implementation of medical chatbot, I have utilised four different models. Encoder-Decoder LSTM, GPT-2 Medium, Facebook/BART-base, and T5-small. The dataset used to create this chatbot included a size of 295 MB and consists of 0.26 million doctor-patient conversations. However, due to computational and temporal limitations, only a portion of the data was utilised. In particular, the Encoder-Decoder LSTM, a type of LSTM model, demonstrated remarkable performance across all metrics, although it was just trained on 6 epochs. Its low loss (0.07) and high accuracy (0.990), along with good ROUGE scores, point to its potential as a robust model for this application. However, this model required pre-trained word embedding like word2vec for better text generation. Similarly, Facebook/BART-base showed better loss (0.39) and accuracy (0.92), especially when comparing its metrics to the other pre-trained models. The GPT-2 Medium and T5-small versions, on the other hand, failed. Despite their success in a variety of NLP tasks, they failed in this application and under the current training conditions (limited to only two epochs). This implies that these models might require further training or modifications suited to the medical field.

Developing a medical chatbot, considering the limitations of using just a single NVIDIA GeForce RTX 3090 GPU, presents certain challenges. To transform this prototype into a fully functional project, several steps need to be undertaken. Firstly, the training durations should be extended to ensure the model learns optimally. Additionally, incorporating more diverse datasets, not limited to just doctor-patient dialogues, can provide a broader understanding and better performance. Moreover, it might be beneficial to explore other pre-trained models, especially those that aren't open source, to potentially enhance the chatbot's capabilities. Lastly, fine-tuning of hyperparameters is essential to maximize the efficiency and effectiveness of the model.

In conclusion, I have combined machine learning and mobile apps to create a medical chatbot. I used different tools and techniques to make a helpful health assistant, even with some tech and data challenges. To make this chatbot even better in the future, need more training, varied data, and some adjustments to the methods.

# **Future work**

Creating a medical chatbot has been both an exciting and challenging task. The current version of the chatbot does its job, however, there is room to make it even better. One of the ways to improve it is by adding more data. This data can be sourced from different sources like medical books, research papers, and case studies. By doing this, the chatbot can have a wider knowledge base, which means it can provide more accurate and helpful answers to users. The world of Natural Language Processing (NLP) is vast. There are many techniques in NLP that can help improve how the chatbot understands and responds to questions. By exploring more of these techniques, the chatbot can communicate more clearly and effectively. Another area to explore is machine learning models. There are many different types of machine learning models, sequence to sequence models, to solve this type of problem, some might be better suited for medical questions and answers. Trying out different models can help find the best one for this chatbot. A challenge faced during the chatbot's creation was adjusting model hyperparameter settings. This was due to limited hardware resources. However, as technology improves and becomes more available, it's possible to overcome this challenge. To overcome computational issues, future efforts will use accelerated machine learning techniques and consider collaborating with cloud platforms for better model training and deployment.

The user experience is also crucial, making the chatbot's interface user-friendly and adding features like user accounts can make it more helpful. User accounts can save past questions of specific users and provide more personalized answers. This feature needs database management to save the user’s chat, Firebase (NoSql Database) will be incorporated. Moreover, it's essential to ensure that everyone can use the chatbot, so adding features for people with disabilities will also be considered.

In the future work, the integration of MLOps (Machine Learning Operations) practices will also be incorporated. These practices include continuous integration, automated testing, and continuous deployment. This integration aims to enhance the procedures for building up the chatbot, maintaining its smooth operation, and continually monitoring its performance to address any difficulties as soon as they arise. Adopting these procedures will simplify processes and improve the chatbot system's overall effectiveness.

# **Conclusion**

In conclusion, this research paper discusses the development of a medical diagnosis chatbot using natural language processing and machine learning techniques. The chatbot aims to provide quick and accurate medical support, making healthcare more accessible and timelier. The paper explores the motivations for building medical diagnostic chatbots and the evolution of human-computer interactions in healthcare. It also discusses the technological foundations behind modern chatbots, such as NLP techniques and machine learning models. The research objective is to build an NLP-based chatbot using the MedDialog-EN dataset and sequence-to-sequence models. The paper outlines the methodology, including the use of LSTM (family of RNNs) and Transformers, and the deployment of the chatbot through a mobile application using react native. The system design involves data collection, model development, evaluation, and validation, as well as ML operations such as version control, CI/CD pipelines, model registry, model serving, and model monitoring. The paper concludes by discussing the limitations of the prototype and the structure of the dissertation. It highlights the potential of AI technologies in developing effective medical chatbots, however also acknowledges the need for further research and development to fully realize their benefits. The paper emphasizes the importance of combining machine learning and mobile apps for creating a medical chatbot and suggests future work such as adding more data, exploring different NLP techniques, and improving the user experience. Overall, the research paper demonstrates the potential of chatbots in healthcare and provides insights into their implementation and challenges.

# **References**

Akbik, A., Blythe, D. and Vollgraf, R., (n.d.) *Contextual String Embeddings for Sequence Labeling*. [online] Available at: https://github.com/zalandoresearch/flair.

Alakkari, K., Subhi, A.A., Alkattan, H., Kadi, A., Malinin, A., Potoroko, I., Abotaleb, M. and El-Kenawy, E.S.M., (2023) Forecasting covid-19 infection using encoder-decoder lstm and attention lstm algorithms. *Journal of Intelligent Systems and Internet of Things*, 82, pp.20–33.

Amer, E., Hazem, A., Farouk, O., Louca, A., Mohamed, Y. and Ashraf, M., (2021) A Proposed Chatbot Framework for COVID-19. In: *2021 International Mobile, Intelligent, and Ubiquitous Computing Conference, MIUCC 2021*. Institute of Electrical and Electronics Engineers Inc., pp.263–268.

Anjum, K., Sameer, M. and Kumar, S., (2023) AI Enabled NLP based Text to Text Medical Chatbot. In: *2023 3rd International Conference on Innovative Practices in Technology and Management (ICIPTM)*. [online] IEEE, pp.1–5. Available at: https://ieeexplore.ieee.org/document/10117966/.

Anon (2021) *Global strategy on digital health 2020-2025*. [online] Available at: http://apps.who.int/bookorders.

Apple Inc. (n.d.)., (2023) *Siri - Apple*. [online] Available at: https://www.apple.com/siri/. [Accessed 25 Jul. 2023].

Ashwini, S., Rajalakshmi, N.R., Victer Paul, P. and Jayakumar, L., (2022) Dynamic NLP Enabled Chatbot for Rural Health Care in India. In: *2022 2nd International Conference on Computer Science, Engineering and Applications, ICCSEA 2022*. Institute of Electrical and Electronics Engineers Inc.

Bi, M., Zhang, Q., Zuo, M., Xu, Z. and Jin, Q., (2020) Bi-directional LSTM Model with Symptoms-Frequency Position Attention for Question Answering System in Medical Domain. *Neural Processing Letters*, 512, pp.1185–1199.

Brandtzaeg, P.B. and Følstad, A., (2017) Why people use chatbots. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Springer Verlag, pp.377–392.

Brooke, J., (n.d.) *SUS - A quick and dirty usability scale*.

Bruno Marietto, M. das G., Aguiar, R.V., Barbosa, G. de O., Botelho, W.T., Pimentel, E., Franca, R. dos S. and da Silva, V.L., (2013) Artificial Intelligence Markup Language: A Brief Tutorial. *International Journal of Computer Science & Engineering Survey*, 43, pp.1–20.

Cavanagh, K. and Millings, A., (2013) (Inter)personal computing: The role of the therapeutic relationship in E-mental health. *Journal of Contemporary Psychotherapy*, 434, pp.197–206.

Chakrabarti, S., Saha, H.N., University of Nevada, Institute of Engineering & Management, IEEE-USA, Institute of Electrical and Electronics Engineers. Region 6 and Institute of Electrical and Electronics Engineers, (n.d.) *2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC) : 8th-10th January, 2018, University of Nevada, Las Vegas, USA*.

Chan, J.Y. Le, Bea, K.T., Leow, S.M.H., Phoong, S.W. and Cheng, W.K., (2023) State of the art: a review of sentiment analysis based on sequential transfer learning. *Artificial Intelligence Review*, 561, pp.749–780.

Chen, M. and Decary, M., (2020) Artificial intelligence in healthcare: An essential guide for health leaders. *Healthcare Management Forum*, 331, pp.10–18.

Colby, K.M., Weber, S. and Hilf, F.D., (1971) *Artificial Paranoia’*. *Artificial Intelligence*, .

Davenport, T. and Kalakota, R., (2019) *DIGITAL TECHNOLOGY The potential for artificial intelligence in healthcare*. *Future Healthcare Journal*, .

Digital Trends. (n.d.)., (2023) *What is Amazon’s Alexa, and what can it do?* [online] Available at: https://www.digitaltrends.com/home/what-is-amazons-alexa-and-what-can-it-do/ [Accessed 25 Jul. 2023].

Dillman, D.A., Phelps, G., Tortora, R., Swift, K., Kohrell, J., Berck, J. and Messer, B.L., (2009) Response rate and measurement differences in mixed-mode surveys using mail, telephone, interactive voice response (IVR) and the Internet. *Social Science Research*, 381, pp.1–18.

Foley, T. and Woollard, J., (n.d.) *The digital future of mental healthcare and its workforce*.

General Data Protection Regulation, (2023) *GDPR compliance checklist - GDPR.eu*. [online] Available at: https://gdpr.eu/checklist/ [Accessed 26 May 2023].

Google. (n.d.)., (2023) *Google Assistant, your own personal Google*. [online] Available at: https://assistant.google.com/ [Accessed 25 Jul. 2023].

Grambow, C.A., Zhang, L. and Schaaf, T., (2022) *In-Domain Pre-Training Improves Clinical Note Generation from Doctor-Patient Conversations*.

Guy, I., (2016) Searching by talking: Analysis of voice queries on mobile web search. In: *SIGIR 2016 - Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*. Association for Computing Machinery, Inc, pp.35–44.

He, X., Chen, S., Ju, Z., Dong, X., Fang, H., Wang, S., Yang, Y., Zeng, J., Zhang, R., Zhang, R., Zhou, M., Zhu, P. and Xie, P., (2020) MedDialog: Two Large-scale Medical Dialogue Datasets. [online] Available at: http://arxiv.org/abs/2004.03329.

Hossain, M.M., Krishna Pillai, S., Dansy, S.E. and Bilong, A.A., (2021) Mr. Dr. Health-Assistant Chatbot. *International Journal of Artificial Intelligence*, 82, pp.58–73.

Howard, J. and Ruder, S., (n.d.) *Universal Language Model Fine-tuning for Text Classification*. [online] Available at: http://nlp.fast.ai/ulmfit.

IBM. (n.d.)., (2023) *IBM Watson*. [online] Available at: https://www.ibm.com/watson [Accessed 25 Jul. 2023].

IEEE Staff, (2017) *2017 International Conference on Inventive Communication and Computational Technologies (ICICCT).* IEEE.

Iftikhar, L., Feras Iftikhar, M. and Hanif, M.I., (n.d.) *Cronicon ChatGPT-3 on Health Services as a Virtual Doctor*.

Iroju, O.G. and Olaleke, J.O., (2015) A Systematic Review of Natural Language Processing in Healthcare. *International Journal of Information Technology and Computer Science*, [online] 78, pp.44–50. Available at: http://www.mecs-press.org/ijitcs/ijitcs-v7-n8/v7n8-7.html.

Jansen, B.J., Spink, A. and Saracevic, T., (n.d.) *Real life, real users, and real needs: a study and analysis of user queries on the web*. [online] Available at: www.elsevier.com/locate/infoproman.

Joseph Weizenbaum, (1966) computational\_linguistics. 9, pp.36–45.

Jurafsky, D. and Martin, J.H., (n.d.) *Speech and Language Processing An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition Third Edition draft Summary of Contents*.

Kaushik, V., Gupta, K. and Gupta, D., (n.d.) *React Native Application Development*. [online] Available at: https://ssrn.com/abstract=3330011.

Klopfenstein, L.C., Delpriori, S., Malatini, S. and Bogliolo, A., (2017) The rise of bots: A survey of conversational interfaces, patterns, and paradigms. In: *DIS 2017 - Proceedings of the 2017 ACM Conference on Designing Interactive Systems*. Association for Computing Machinery, Inc, pp.555–565.

Laranjo, L., Dunn, A.G., Tong, H.L., Kocaballi, A.B., Chen, J., Bashir, R., Surian, D., Gallego, B., Magrabi, F., Lau, A.Y.S. and Coiera, E., (2018) *Conversational agents in healthcare: A systematic review*. *Journal of the American Medical Informatics Association*, .

Lee, I. and Shin, Y.J., (2020) Machine learning for enterprises: Applications, algorithm selection, and challenges. *Business Horizons*, 632, pp.157–170.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., Zettlemoyer, L. and Ai, F., (n.d.) *BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension*. [online] Available at: https://huggingface.co/transformers.

Liu, B., Xu, Z., Sun, C., Wang, B., Wang, X., Wong, D.F. and Zhang, M., (2018) Content-Oriented User Modeling for Personalized Response Ranking in Chatbots. *IEEE/ACM Transactions on Audio Speech and Language Processing*, 261, pp.122–133.

Luo, G., (2016) A review of automatic selection methods for machine learning algorithms and hyper-parameter values. *Network Modeling Analysis in Health Informatics and Bioinformatics*, 51.

Microsoft Corporation. (n.d.)., (2023) *Cortana - Microsoft*. [online] Available at: https://www.microsoft.com/en-us/cortana/ [Accessed 25 Jul. 2023].

Molnar, G. and Szuts, Z., (2018) The Role of Chatbots in Formal Education. In: *SISY 2018 - IEEE 16th International Symposium on Intelligent Systems and Informatics, Proceedings*. Institute of Electrical and Electronics Engineers Inc., pp.197–201.

Ni, J., Ábrego, G.H., Constant, N., Ma, J., Hall, K.B., Cer, D. and Yang, Y., (2021) Sentence-T5: Scalable Sentence Encoders from Pre-trained Text-to-Text Models. [online] Available at: http://arxiv.org/abs/2108.08877.

Nordholm, S. and Bäck, R., (2001) On the role of nonergodicity and slow IVR in unimolecular reaction rate theory - A review and a view. *Physical Chemistry Chemical Physics*, 312, pp.2289–2295.

Omoregbe, N.A.I., Ndaman, I.O., Misra, S., Abayomi-Alli, O.O. and Damaševičius, R., (2020) Text messaging-based medical diagnosis using natural language processing and fuzzy logic. *Journal of Healthcare Engineering*, 2020.

Qiu, X.P., Sun, T.X., Xu, Y.G., Shao, Y.F., Dai, N. and Huang, X.J., (2020) *Pre-trained models for natural language processing: A survey*. *Science China Technological Sciences*, .

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W. and Liu, P.J., (2019) Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. [online] Available at: http://arxiv.org/abs/1910.10683.

Rajkomar, A., Hardt, M., Howell, M.D., Corrado, G. and Chin, M.H., (2018) Ensuring fairness in machine learning to advance health equity. *Annals of Internal Medicine*, 16912, pp.866–872.

Ranoliya, B.R., Raghuwanshi, N. and Singh, S., (2017) Chatbot for university related FAQs. In: *2017 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2017*. Institute of Electrical and Electronics Engineers Inc., pp.1525–1530.

Raschka, S., (2018) Model Evaluation, Model Selection, and Algorithm Selection in Machine Learning. [online] Available at: http://arxiv.org/abs/1811.12808.

Sanjuán, P. and Magallares, A., (2014) Coping Strategies as Mediating Variables Between Self-serving Attributional Bias and Subjective Well-Being. *Journal of Happiness Studies*, 152, pp.443–453.

Schneider, E.T.R., De Souza, J.V.A., Gumiel, Y.B., Moro, C. and Paraiso, E.C., (2021) A GPT-2 language model for biomedical texts in Portuguese. In: *Proceedings - IEEE Symposium on Computer-Based Medical Systems*. Institute of Electrical and Electronics Engineers Inc., pp.474–479.

Sharma, V., Goyal, M. and Malik, D., (n.d.) *An Intelligent Behaviour Shown by Chatbot System*. [online] Available at: www.ijntr.org.

Shivam, K., Saud, K., Sharma, M., Vashishth, S. and Patil, S., (2018) Chatbot for College Website. *IJCAT-International Journal of Computing and Technology*, [online] 56. Available at: https://www.researchgate.net/publication/362546056.

Sirriani, J., Sezgin, E., Claman Dds, D. and Linwood, S.L., (n.d.) Medical text prediction and suggestion using generative pre-trained transformer models with dental medical notes. [online] Available at: https://doi.org/10.1101/2022.04.29.22274513.

Smutny, P. and Schreiberova, P., (2020) Chatbots for learning: A review of educational chatbots for the Facebook Messenger. *Computers and Education*, 151.

Sutskever, I., Vinyals, O. and Le, Q. V., (2014) Sequence to Sequence Learning with Neural Networks. [online] Available at: http://arxiv.org/abs/1409.3215.

Tallyn, E., Fried, H., Gianni, R., Isard, A. and Speed, C., (2018) The ethnobot: Gathering ethnographies in the age of IoT. In: *Conference on Human Factors in Computing Systems - Proceedings*. Association for Computing Machinery.

Tamang, S., Humbert-Droz, M., Gianfrancesco, M., Izadi, Z., Schmajuk, G. and Yazdany, J., (2023) *Practical Considerations for Developing Clinical Natural Language Processing Systems for Population Health Management and Measurement*. *JMIR Medical Informatics*, .

Thwala, E.K.I., Adegun, A.A. and Adigun, M.O., (2023) Self-Assessment Chatbot for COVID-19 prognosis using Deep Learning-based Natural Language Processing (NLP). In: *2023 International Conference on Science, Engineering and Business for Sustainable Development Goals, SEB-SDG 2023*. Institute of Electrical and Electronics Engineers Inc.

Tu, M., Huang, K., Wang, G., He, X. and Zhou, B., (n.d.) *The Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI-20) Select, Answer and Explain: Interpretable Multi-Hop Reading Comprehension over Multiple Documents*. [online] Available at: www.aaai.org.

Turing, A.M., (1950) *M I N D A QUARTERLY REVIEW OF PSYCHOLOGY AND PHILOSOPHY I.-COMPUTING MACHINERY AND INTELLIGENCE*. [online] Available at: https://academic.oup.com/mind/article/LIX/236/433/986238.

Vaidyam, A.N., Wisniewski, H., Halamka, J.D., Kashavan, M.S. and Torous, J.B., (2019) *Chatbots and Conversational Agents in Mental Health: A Review of the Psychiatric Landscape*. *Canadian Journal of Psychiatry*, .

Wei, J., Chu, X., Sun, X.Y., Xu, K., Deng, H.X., Chen, J., Wei, Z. and Lei, M., (2019) *Machine learning in materials science*. *InfoMat*, .

WHO Global Observatory for eHealth., (2011) *MHealth : new horizons for health through mobile technologies.* World Health Organization.

Wilson, T.D., (1997) *Trends in . . . A critical review: Information behaviour: An interdisciplinary perspective*. *Information Processing and Management*, .

WMA - The World Medical Association, (2023) *WMA Declaration of Helsinki – Ethical Principles for Medical Research Involving Human Subjects – WMA – The World Medical Association*. [online] Available at: https://www.wma.net/policies-post/wma-declaration-of-helsinki-ethical-principles-for-medical-research-involving-human-subjects/ [Accessed 27 May 2023].

WMA - The World Medical Association-WMA International Code of Medical Ethics, (2023) *WMA International Code of Medical Ethics – WMA – The World Medical Association*. [online] Available at: https://www.wma.net/policies-post/wma-international-code-of-medical-ethics/ [Accessed 26 May 2023].

Xu, Y., Li, L., Xu, H., Huang, S., Huang, F. and Cai, J., (2022) Image Captioning In the Transformer Age. [online] Available at: http://arxiv.org/abs/2204.07374.

Xygi, E., Andriopoulos, A.D. and Koutsomitropoulos, D.A., (2023) Question Answering Chatbots for Biomedical Research Using Transformers. In: *5th International Conference on Artificial Intelligence in Information and Communication, ICAIIC 2023*. Institute of Electrical and Electronics Engineers Inc., pp.25–29.

Yadav, A., (n.d.) *Chatbot-Based Medical Diagnosis Using Natural Language Processing and Classifier*.

Yau, S.S. and J-p Tsai, J., (1986) *A Survey of Software Design Techniques*. *IEEE TRANSACTIONS ON SOFTWARE ENGINEERING*, .

ZEMČÍK, Mgr.T., (2019) A Brief History of Chatbots. *DEStech Transactions on Computer Science and Engineering*, aicae.

Zhang, Y. and Wallace, B., (2015) A Sensitivity Analysis of (and Practitioners’ Guide to) Convolutional Neural Networks for Sentence Classification. [online] Available at: http://arxiv.org/abs/1510.03820.

Zhou, X., Li, Y. and Liang, W., (2021) CNN-RNN Based Intelligent Recommendation for Online Medical Pre-Diagnosis Support. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 183, pp.912–921.

# **Appendices**

## **13.1. Figure 1**

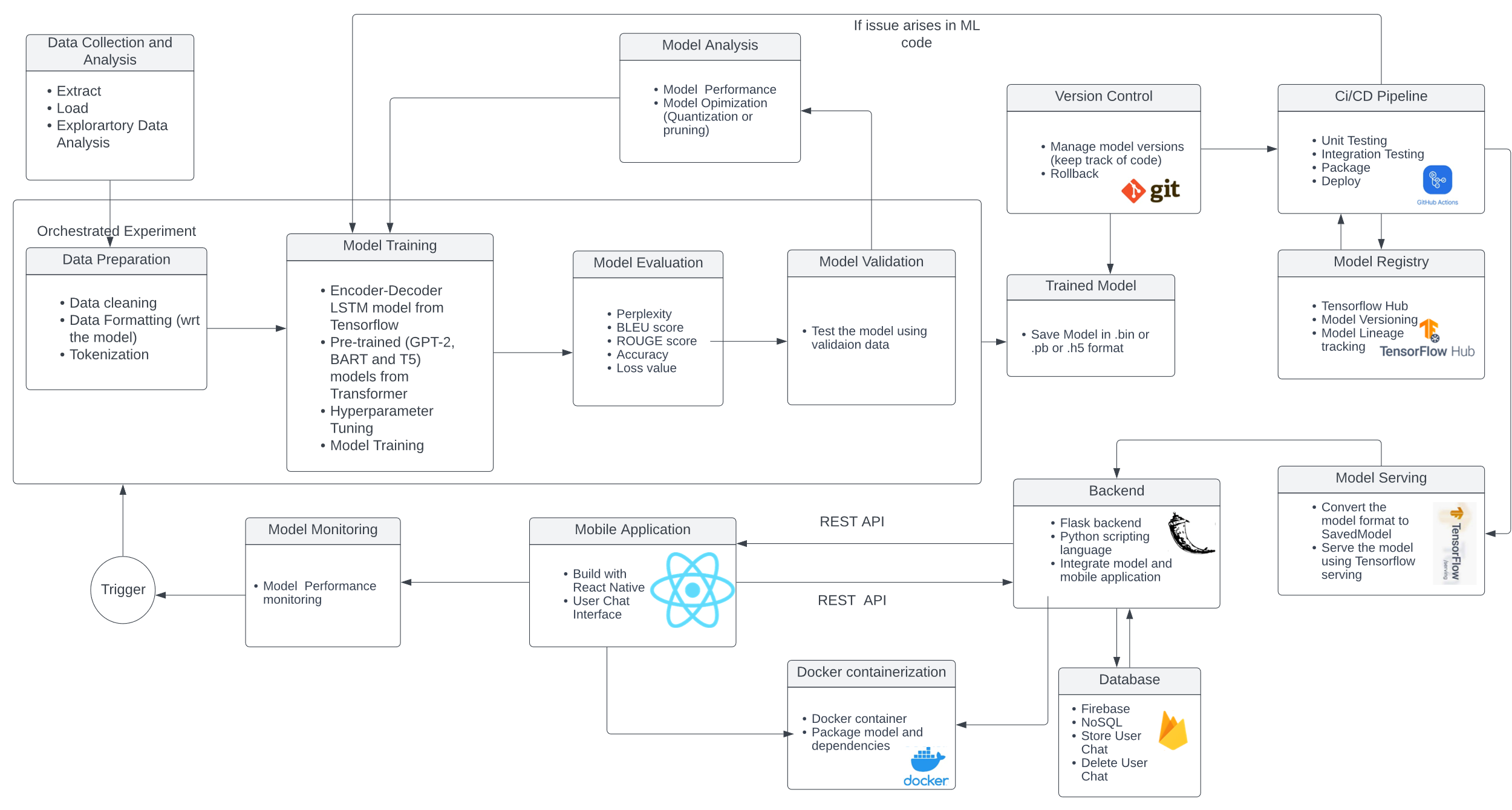


Figure 1–System Design of medical chatbot

## **13.2. Figure 2**

**A screenshot of a computer

Description automatically generated**

Figure 2 – Dataset columns and datatypes list

## **13.3. Figure 3**

A screenshot of a chat

Description automatically generated

Figure 3 – Top 5 rows or entries in this dataset

## **13.4. Figure 4**

A screenshot of a computer

Description automatically generated

Figure 4 – Descriptive statistics of the dataset

## **13.5. Figure 5**

A screen shot of a graph

Description automatically generated

Figure 5 – Missing values in the dataset

## **13.6. Figure 6**

**A screenshot of a computer

Description automatically generated**

Figure 6 – Check repetitive dialogues in the dataset.

## **13.7. Figure 7**

A screenshot of a computer

Description automatically generated

Figure 7 – contractions in the dataset

## **13.8. Figure 8**

A chart of a document length

Description automatically generated with medium confidence

Figure 8 – Entries or documents in each column

## **13.9. Figure 9**

A blue rectangular object with white background

Description automatically generated

Figure 9 – Unique words vs total words

## **13.10. Figure 10**

A red rectangle with different colors

Description automatically generated

Figure 10 – Word length analysis

## **13.11. Figure 11**

A green and blue rectangles

Description automatically generated

Figure 11–Stop words count.

## **13.12. Figure 12**

A graph of blue bars

Description automatically generated

Figure 12 – Top 10 Most frequent words

## **13.13. Figure 13**

A graph of a number of blue bars

Description automatically generated

Figure 13 – Punctuation Frequency

## **13.14. Figure 14**

**A collage of graphs

Description automatically generated**

Figure 14 – Distribution of text lengths

## **13.15. Figure 15**

A diagram of a graph

Description automatically generated

Figure 15 – Distribution of text lengths

## **13.16. Figure 16**

A screenshot of a computer code

Description automatically generated

Figure 16 – After removing the duplicates.

## **13.17. Figure 17**

A screenshot of a computer

Description automatically generated

Figure 17 – Model response with lemmatization

## **13.18. Figure 18**

**A screenshot of a computer

Description automatically generated**

Figure 18 – Comparison of the trained models

## **13.19. Figure 19**

A screenshot of a chat

Description automatically generated

Figure 19 – Mobile application chat interface

## **13.20. Figure 20**

Screens screenshot of a chat

Description automatically generated

Figure 20 – User chat using BART finetuned model.

## **13.21. Figure 21**

A screenshot of a computer error

Description automatically generated

Figure 21– GPT-2 with half dataset