Thesis Deep Auto-Encoder-Based Chatbot for Pieas Admission-Related Inquiries Fatima and Shajee

Deep Auto-Encoder-Based Chatbot for PIEAS Admission-Related Inquiries

by

Fatima Aftab
03-3-01-11-2019

Shajee Raza
03-3-01-32-2019

Supervisor
Dr. Asifullah Khan

Thesis submitted in partial fulfillment of requirements for the Degree of Bachelor of Sciences in Computer and Information Sciences

In

Department of Computer and Information Sciences, Pakistan Institute of Engineering & Applied Sciences, Nilore, Islamabad Pakistan (45650)

(May, 2023)

Declaration of Originality

I hereby declare that the work contained in this thesis and the intellectual content of this thesis are the product of my own work. This thesis has not been previously published in any form nor does it contain any verbatim of the published resources which could be treated as infringement of the international copyright law.

I also declare that I do understand the terms ‘copyright’ and ‘plagiarism,’ and that in case of any copyright violation or plagiarism found in this work, I will be held fully responsible of the consequences of any such violation.

Signature: ____________________________
Author’s Name: Fatima Aftab
Date: 30th May 2023
Place: PIEAS, Nilore, Islamabad

Signature: ____________________________
Author’s Name: Shajee Raza
Date: 30th May 2023
Place: PIEAS, Nilore, Islamabad

Certificate of Approval

Certified that work contained in this thesis titled "**Deep Auto-Encoder-Based Chatbot for PIEAS Admission-Related Inquiries"** was carried out by **Fatima Aftab and Shajee Raza** under our supervision and that in our opinion, it is fully adequate, in scope and quality, for the degree of BS in Computer and Information Sciences.

Supervisor

Signature : ___________________________________

Name : _Dr. Asifullah Khan______________________

Co-Supervisor

Signature : ___________________________________

Name : _ Mr. Sybghatallah______________________

Head of the Department

Signature: ___________________________________

Name: ____Dr. Javaid Khurshid________________

Stamp:

Dedication

To the sleepless nights

Acknowledgment

Allah Almighty, the Lord of the World, who bestowed humanity with the illumination of knowledge through the laurels of perception, learning, and reasoning, on the path of seeking, inquiring, and discovering the ultimate truth, deserves our eternal gratitude. To whom we serve and for whose assistance we beseech.

Dr. Asifullah Khan has provided unwavering support, guidance, and expertise throughout the development of this thesis. We are profoundly privileged anhonoreded to convey our sincere gratitude to him. This research's direction and quality have been significantly influenced by his invaluable insights, constructive criticism, and encouragement.

We would also like to express our gratitude to Mr. Sybghatallah, our co-supervisor, for his contributions, assistance, and collaboration. His knowledge and direction have been invaluable in expanding the scope and profundity of this project.

In addition, we are extremely appreciative to the project panel and coordinator for their insightful discussions and significant contributions that have enriched this research effort.

Additionally, we would like to recognize the Pakistan Institute of Engineering and Applied Sciences for providing a conducive educational environment that encourages learning, research, and academic development.

We are extremely grateful to our family for their unwavering love, support, and encouragement throughout our studies. Their confidence in our abilities and constant encouragement have been crucial to our academic success.

This thesis would not have been possible without the direction, assistance, and contributions of all these individuals and institutions. We are deeply appreciative and humbled by their contributions to our research.

Table of Contents

Table of Figures

Abstract

Various domains have been revolutionized by chatbot technology by expediting processes and enhancing user experiences. This study centers on the development of a chatbot to assist students seeking admission to the Pakistan Institute of Engineering and Applied Sciences (PIEAS) with their studies. To ensure an efficient and exhaustive chatbot system, a dataset of 4,228 meticulously curated question-answer pairs was compiled from diverse sources, including the PIEAS website, Facebook groups, admission and other relevant offices and surveys.

The Bert-base-uncased variant of the transformer model was utilized so that the chatbot could accurately comprehend and respond to user inquiries. The dataset was preprocessed with BERT-Tokenizer to generate word embeddings, which provided the chatbot with contextual understanding and the ability to generate contextually relevant responses. In addition, a conversational dataset was incorporated to enhance the chatbot's conversational abilities and dynamic user interaction.

Using an exhaustive dataset and advanced natural language processing (NLP) techniques, this study lays the groundwork for training a highly effective chatbot system. The training phase will focus on optimizing the performance of the model and assuring its effectiveness in assisting students during the PIEAS admissions process.

This research contributes to the advancement of chatbot technology in the education sector and demonstrates its potential to provide individualized and effective assistance to students studying and applying for admission. The findings of this study can impact future developments in chatbot systems, leading to enhanced user experiences and increased educational institution efficiency.

Chapter 1

Introduction

In Chapter 1, we introduce the topic of our research and lay the foundation for our investigation. The chapter commences with a discussion of the topic's historical context, which provides context and emphasises its significance. The problem statement elucidates the precise issue that our research will address. The goals and objectives of our investigation are listed alongside the research objectives.

In addition, we discuss the significance of the study, emphasising the anticipated influence and benefits of our research findings. In addition, the study's scope and limitations are discussed, thereby elucidating the limits of our research. The chapter concludes with a summary of the thesis's organisation, delineating the subsequent chapters and their respective contents.

1.1 Background

Effective communication is crucial in many disciplines, including education, in the current digital era. Universities and educational institutions struggle to promptly respond to all student inquiries, resulting in unprocessed job applications and a paucity of human agents. To address this issue, there is a growing demand for an automated communication solution that is accessible 24/7. One such solution is a Chatbot system, which is a software application designed to conduct text-based online chat conversations. Chatbots can automate conversations, interact with users through messaging platforms, and provide immediate assistance.

The advances in deep learning-based machine learning models are essential for the development of Chatbots, especially in the disciplines of Natural Language Processing (NLP) and Transformers. These technologies enable the Chatbot system to comprehend and produce human-like responses, thereby improving the overall user experience. Chatbots are able to analyse and interpret user queries, derive relevant information, and generate responses that are appropriate and contextually accurate by leveraging NLP techniques and deep learning models.6.5.1(1)(a)(i) [1]

By implementing a Chatbot system, universities such as PIEAS can overcome the challenges associated with manual query handling, such as limited staff availability, time constraints, and the possibility of inaccuracies in the information. The Chatbot system can provide an automated and efficient solution for responding to student inquiries about admissions information, ensuring accessibility 24 hours a day and reducing the workload of human agents. This technology-driven strategy not only increases operational efficiency, but also enhances the overall student experience by providing prompt and reliable support.

Therefore, the development of a Chatbot system for PIEAS admission-related questions, powered by NLP and deep learning-based models, is a promising method to enhance communication, respond effectively to student questions, and expedite the admissions process.

1.2 Problem Statement

This study investigates the inability of universities and educational institutions, such as PIEAS, to manage the burden associated with responding to student inquiries effectively. The limitations of the current manual system include limited staff availability, the inability to facilitate multiple users simultaneously, non-availability during off-hours, and the possibility of providing inaccurate or insufficient information. These limitations hinder the institution's ability to provide timely and adequate support to students seeking information about admission.

In addition, the current system lacks scalability, as it cannot simultaneously serve a large number of pupils, resulting in query resolution delays and backlogs. Human agents increase operational costs and occupy time that could be better spent on other administrative tasks.

The proposed solution seeks to design a chatbot system based on a deep auto-encoder for PIEAS in order to overcome these obstacles. This system will employ cutting-edge technologies, including Natural Language Processing (NLP) and deep learning models, to automate conversations, comprehend student inquiries, and provide accurate, relevant responses. By implementing this chatbot system, the institution will be able to reduce the burden of its employees, be available to students 24 hours a day, seven days a week, increase its responsiveness, and improve the overall efficacy and effectiveness of query resolution.

As a result, the problem statement emphasizes the need to reduce the workload of staff members, enhance the institution's responsiveness, and provide an automated, cost-effective, and user-friendly solution for handling multiple student inquiries related to PIEAS admission, thereby optimizing operational efficiency and enhancing the student experience.

1.3 Research Objectives

The following are the objectives of this study:

- **To develop a deep auto-encoder-based chatbot system for PIEAS university:** The primary goal is to design and implement a chatbot system customized to address admissions-related questions from students interested in PIEAS. The system will utilize sophisticated auto-encoder models and Natural Language Processing (NLP) techniques to accurately comprehend and respond to pupil queries.

- **To boost operational efficiency:** The chatbot system will automate conversations and manage multiple student inquiries simultaneously, enabling quicker responses and streamlining processes.
- **To achieve cost and time savings:** By implementing the chatbot system, the study aims to reduce reliance on human agents, thereby reducing operational costs associated with hiring and training additional personnel. The automation of query resolution will also save students and staff members' valuable time.
- **To offer convenience and improve the student experience:** the chatbot system will offer a user-friendly interface that makes it simple for students to interact and obtain admissions-related information. The system's availability 24 hours a day, seven days a week enables students to receive assistance at any time, thereby enhancing convenience and responsiveness.
- **To reduce human interaction and improve consistency:** The chatbot system will reduce the need for direct human participation in handling routine inquiries, thereby reducing the likelihood of human errors and inconsistencies in the information provided. The system will maintain a consistent level of precision and dependability when responding to pupil inquiries.
- **To investigate the capability of deep learning and NLP techniques:** The purpose of this study is to contribute to the comprehension and application of deep learning models, specifically deep auto-encoders, in the context of chatbot development. Using NLP techniques, this study will examine the efficacy of these technologies in enhancing the precision and relevance of chatbot responses.

Overall, the research objectives seek to develop an efficient, cost-effective, and user-friendly chatbot system that can effectively manage student questions, reduce human interaction, and improve the experience of students seeking admissions-related information at PIEAS.

1.4 Significance of the Study

The significance of this study resides in the development of a chatbot system based on a deep auto-encoder to address the challenges universities face in responding to student admissions-related questions, with a focus on PIEAS. The purpose of the research is to attain enhanced operational efficiency, cost and time savings, enhanced student experience, reduced human interaction, exploration of advanced technologies, and academic contribution. The findings have the potential to benefit other universities and advance automated communication systems in the educational domain, thereby providing efficient and effective assistance throughout the admissions process.

1.5 Scope and Limitations

This study centers on the design and implementation of a chatbot system utilizing a deep auto-encoder to handle admissions-related questions at PIEAS university. Using Python as the programming language, the research will primarily focus on Natural Language Processing (NLP) techniques and deep learning-based machine learning models.

The purpose of this research is to determine whether the chatbot system is capable of managing a wide variety of admissions-related questions and providing timely and accurate responses to students. It will involve the collection and preprocessing of a data set pertaining to admissions.

However, it is essential to acknowledge the limitations of the investigation. Initially, the development of the chatbot system will focus on a specific domain, namely PIEAS admissions. Hence, the system may not be directly applicable to other university domains without additional customization and training.

Second, although efforts will be made to train the chatbot system with a comprehensive dataset, there may be instances in which it lacks the necessary data to accurately respond to certain queries. The efficacy of the system will depend on the quality and coverage of the training dataset.

Thirdly, the research will primarily concentrate on the technical implementation of the chatbot system, and its evaluation will primarily be based on objective metrics such as response accuracy and system performance. The subjective aspects of user experience and satisfaction will be limited to anecdotal user feedback and may not be investigated in depth.

The study will not delve into the ethical implications and potential biases of the chatbot system's use. Ethical considerations, such as user privacy, data protection, and equitable treatment, will be outside the scope of this study and will require separate investigation.

This study aims to provide valuable insights into the design and implementation of a chatbot system for admissions-related queries despite these limitations. The findings and recommendations can serve as a foundation for future research on chatbot technology and its educational applications.

1.6 Thesis Organization

This thesis is divided into chapters to provide a structured approach to the research on developing a deep auto-encoder-based chatbot system for PIEAS university admissions-related queries. Here is a summary of the individual chapter's contents:

Chapter 1: Introduction

This chapter introduces the context of the research, highlighting the challenges universities face in responding to student inquiries and the need for an automated communication solution. This chapter describes the statement of the problem, the research objectives, the significance of the study, its scope, and its limitations.

Chapter 2: Literature Review

The second chapter provides a comprehensive survey of the literature on chatbot technology. It begins with an introduction to chatbots, examining their various varieties and applications with a focus on their educational function. This chapter examines the challenges posed by admissions-related queries and presents pertinent research in the field of chatbot development. In addition, it examines the significance of evaluation metrics for assessing chatbot performance and delves deeper into the NLP techniques and models employed by chatbot systems. The second chapter provides an overview of the existing knowledge and research regarding chatbot technology.

Chapter 3: Methodology

This study's methodology for developing the chatbot system is described in Chapter 3. It discusses research design, data collection and preparation, chatbot model architecture, chatbot training and fine-tuning, web application development, chatbot integration into the web application, and ethical considerations. This chapter provides a summary of the methodology and procedures used to create an ethical and effective chatbot system.

Chapter 4: Results

In Chapter 4, we investigate the outcomes of our chatbot system through a comprehensive analysis of its accuracy and responsiveness. We also compare its performance to that of benchmark systems. This chapter provides insightful information regarding the performance of the chatbot, emphasizing both its strengths and weaknesses. We gain a thorough understanding of the chatbot's performance and its implications in relation to existing systems by analysing performance indicators and user satisfaction survey results. These findings, which serve as a framework for additional discussions and analysis, influence the remaining sections of this thesis.

Chapter 5: Discussion

In Chapter 5, we elaborate on the conclusions and explanations derived from the data presented in Chapter 4. We evaluate the pros and cons of the chatbot system, taking into account its responsiveness, accuracy, and user satisfaction. This discussion provides insightful information about the performance effects of the chatbot and guides future research and development. Despite noting the chatbot's shortcomings in terms of dataset size, accuracy, and web application capability, we discuss the chatbot's advantages, such as its use of the BERT base uncased model. For future developments in chatbot technology and for comprehending the functioning of the chatbot, this chapter is an essential starting point.

Chapter 6: Conclusion and Future Work

In a thorough review of the research results in Chapter 6, it is noted how well the chatbot performed in terms of accuracy, responsiveness, and user happiness. By highlighting the knowledge gained from the application and assessment of the BERT basic uncased model, this research emphasizes its value to the field of chatbot technology. The study's limitations, including dataset size and accuracy, are acknowledged and recommendations are made for further research. Future study is suggested to concentrate on topics like storing conversation history, supporting several languages, and voice functions. Conclusions highlight the importance of the study and serve as a foundation for developing chatbot systems.

Chapter 2

Literature Review

In Chapter 2, we examine the vast corpus of research on chatbot technology. This chapter is a comprehensive review of extant knowledge, examining various aspects of chatbots, their types, applications in education, challenges in admission-related queries, and the most recent developments in NLP techniques and models. By analyzing pertinent research and evaluation metrics, this chapter sets the groundwork for the development and evaluation of our chatbot system. The purpose of this literature review is to gain valuable insights and build on existing knowledge in order to develop an effective and efficient chatbot solution.

2.1 Introduction to Chatbot Technology

Effective communication between two or more people relies on language, and dialogues facilitate this communication. Chatbots seek to replicate these communication mechanisms by artificially creating speech-resembling technology. A chatbot is an intelligent computer program that responds to text and voice input using Natural Language Processing (NLP) to comprehend various spoken human languages.

The operation of a chatbot is based on the concept of question-and-answer mapping, in which the chatbot interprets a user's query and generates a response based on previously acquired knowledge. The application-specific nature of these AI-based systems is essential for delineating the problem domain for which they are intended. Similar to the classification of databases, chatbots can be divided into open (Context Free) and closed (Domain Specific) domains. Open Domain and Close Domain datasets are readily available within the context of chatbot systems.Error: Reference source not found

The ability of chatbots to facilitate multiple users simultaneously through automated communication is one of their primary benefits. They can process user inputs efficiently and generate meaningful and understandable responses. Chatbots accept natural language text as input and aim to generate highly responsive outputs. Consequently, chatbots have applications in industries such as education, e-commerce, and business that require human intervention.

In recent years, the prevalence of chatbot technology has skyrocketed, as it offers natural language interfaces for interacting with users and delivering precise and direct responses. The origins of chatbots can be traced back to the 1960s, when ELIZA, a rule-based chatbot that imitated a Rogerian psychotherapist, was introduced.Error: Reference source not found Over time, the increasing number of applications, computational power, and sharing of open-source frameworks and technologies all contributed to the accelerated development of chatbot technology. As a consequence, numerous institutions have utilised chatbots for a variety of purposes.

2.2 Types of Chatbots

In this section, we will examine the numerous chatbot types discussed in the preceding section.Error: Reference source not found

- **Rule-Based Chatbots**

Rule-based chatbots, also known as scripted or decision tree chatbots, respond to user queries based on a set of predefined principles. Typically, these principles are based on specific keywords or patterns extracted from the user's input. Rule-based chatbots are relatively simple and can provide predefined responses, but they cannot comprehend complex queries or deal with ambiguous inputs. They are commonly used in situations where the conversational flow is predictable and limited.

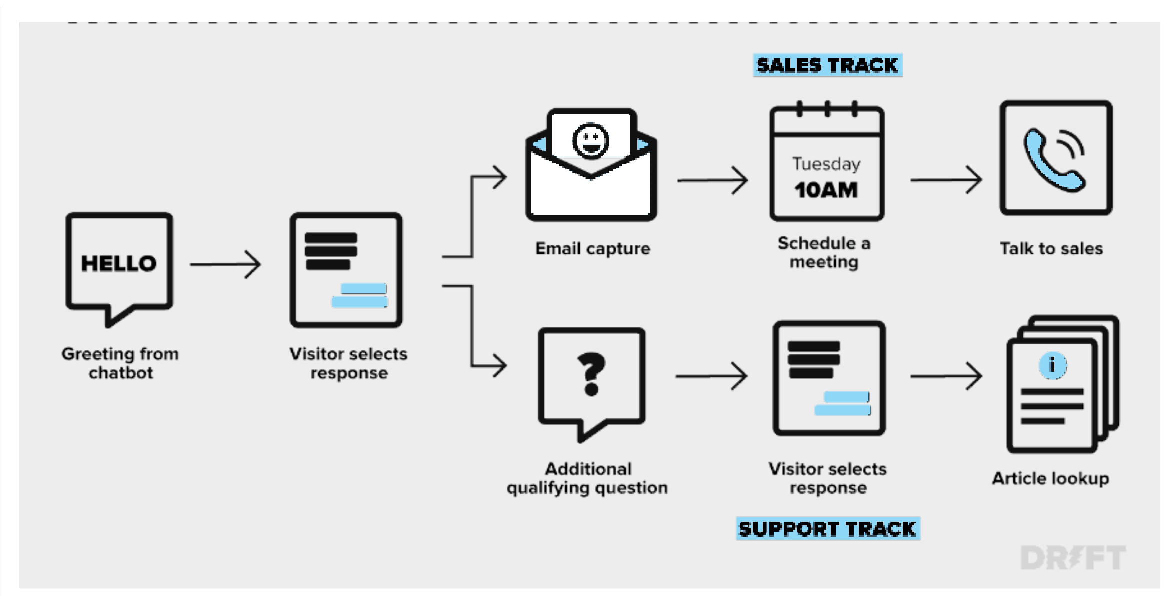


Figure 2.1 Working of rule-based chatbot

- **Retrieval-Based Chatbots**

Retrieval-based chatbots rely on predefined responses stored in a database or knowledge base. These responses are retrieved based on the similarity between the user's input and the available responses. Retrieval-based chatbots determine the most appropriate response using techniques such as keyword matching, pattern recognition, and machine learning algorithms. These chatbots can respond to a wide variety of user queries, but are restricted to the responses contained in their knowledge base.

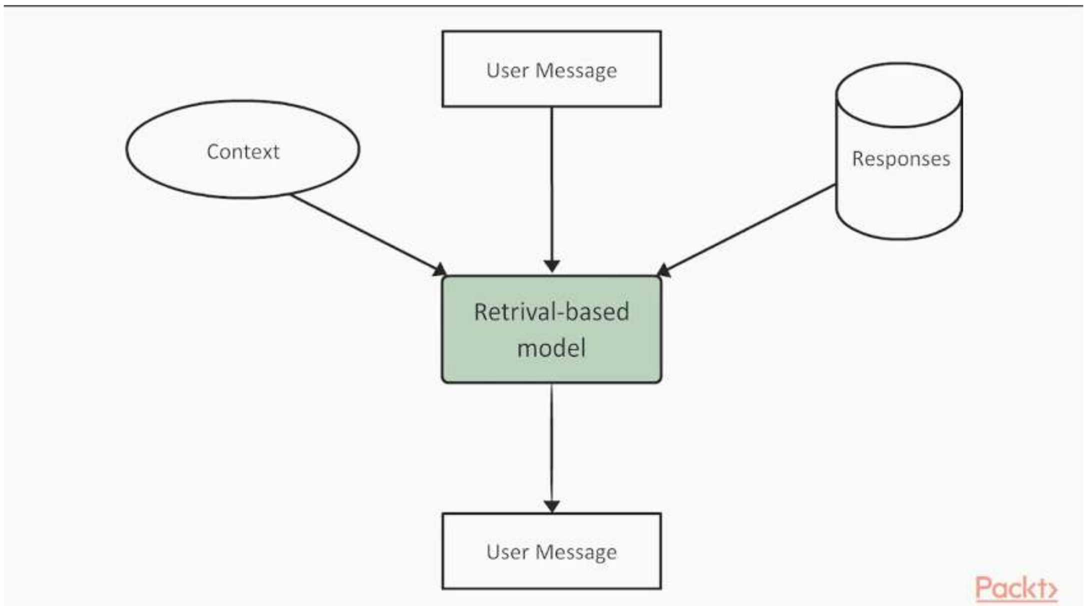


Figure 2.2 Working of Retrieval-based chatbot

- **Generative Chatbots**

Also known as sequence-to-sequence models, generative chatbots generate responses based on user input using deep learning techniques. By learning from enormous amounts of training data, these chatbots can comprehend the conversational context and generate responses resembling those of a human. generative chatbots utilise advanced natural language processing (NLP) models such as recurrent neural networks (RNNs) or transformer models such as GPT (Generative Pretrained Transformer). They can manage a wide range of queries and provide more personalised and contextual responses.

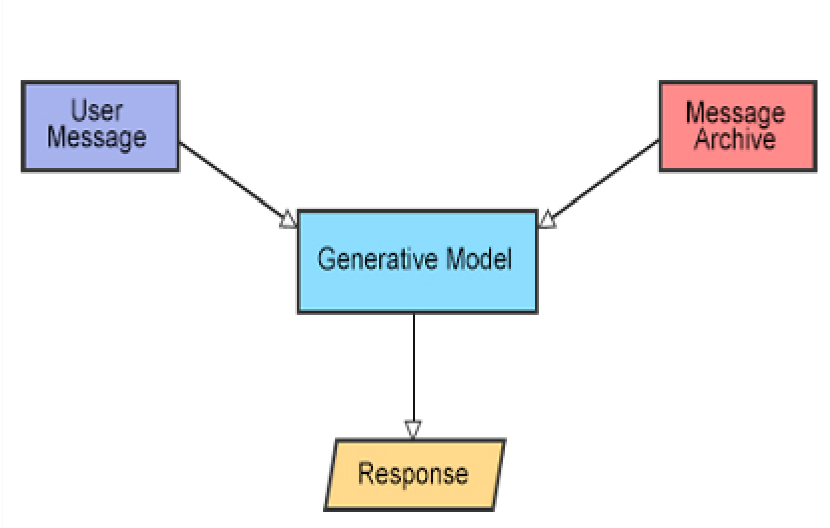


Figure 2.3 Working of generative chatbots

- **Hybrid Chatbots**

Hybrid chatbots offer more robust and adaptable conversational experiences by combining the capabilities of rule-based, retrieval-based, and generative chatbots. These chatbots effectively respond to user queries by employing a combination of predefined rules, retrieval mechanisms, and generative models. Hybrid chatbots offer a balance between predefined responses and dynamic generation based on user inputs.

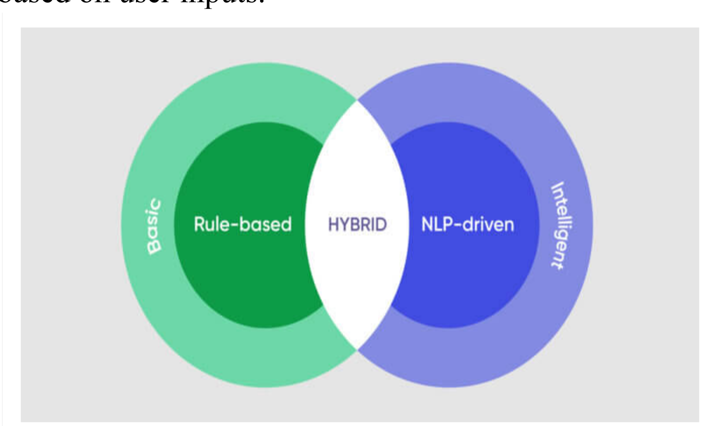


Figure 2.4Working of hybrid chatbots

2.3 Applications of Chatbots in Education

In this section, we will examine the educational applications of chatbots using the information you supplied as a guide.

- **Education:**

Chatbots in educational institutions provide students with immediate support, guidance, and information, making them valuable resources. These chatbots can assist numerous users concurrently, eliminating delays and increasing productivity. For instance, at the start of the academic year, newly enrolled students frequently have numerous questions regarding university hours, the location of facilities such as cafés and libraries, and more. Chatbots can effectively respond to these questions, providing instant responses and facilitating a seamless transition for students.

Chatbots have also benefited distance education, allowing students to access knowledge and resources even when teachers are not available 24 hours a day, seven days a week. Freudbot, an AIML-based chatbot, has been investigated as an instrument for distance education Error: Reference source not found, utilizing an open-source framework to offer personalized assistance. Researchers have underscored the significance of selecting an appropriate operating system and software when designing chatbots for educational purposes that are user-friendly.

Figure 2.5 Application of chatbot in education

- **Customer Service**

Chatbots have been implemented in the customer service domain of e-commerce platforms in an effort to enhance the purchasing experience of users. Nguyen conducted a case study highlighting the positive impact of chatbots on customer service, demonstrating how chatbot interactions improved the overall purchasing experience for users [5].

- **Medicine:**

In the medical sector, chatbots have been developed to assist patients with a variety of tasks, including appointment scheduling and allergy information. AllergyBot, for instance, enables adults with food allergies to make informed decisions by supplying detailed information about food ingredients Error: Reference source not found. Medical consultations have also been facilitated by chatbots, which use patient symptoms as queries to retrieve relevant treatment records from multiple applications, allowing for personalized consultations Error: Reference source not found.

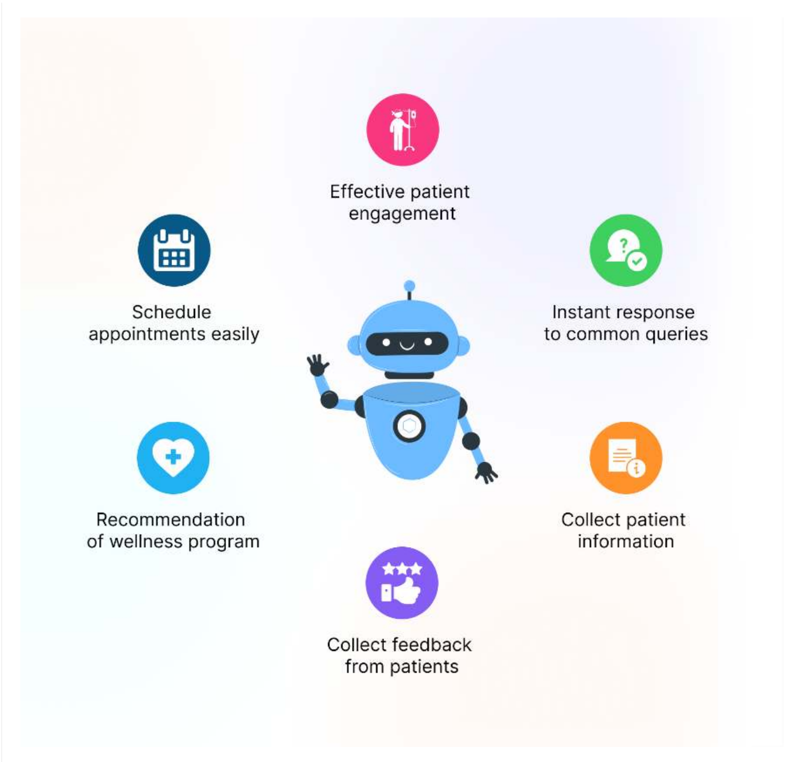


Figure 2.6 Applications of chatbot in medicine

Due to factors such as the need for large training datasets, infrastructure requirements, and the confidential nature of domain-specific data, the incorporation of machine learning technologies, particularly deep learning, remains limited in the field of education. However, as advances in data availability and processing continue, educational chatbot systems may be able to incorporate more sophisticated machine learning models.Error: Reference source not found

2.4 Challenges in Admission-Related Queries

The admissions process is a crucial phase for educational institutions, and prospective students and their parents frequently submit a large number of questions. Based on the information you provided, we will discuss the difficulties associated with admissions-related questions in this section.

- **High Query Volume**

During the admissions season, educational institutions receive an influx of inquiries from prospective students. Manually processing a high volume of queries can be time- and resource-intensive. By providing instant responses to frequently asked inquiries, chatbots can assist in easing the burden on admissions staff.

- **Diverse Query Types**

In terms of content and complexity, admission-related queries can vary significantly. Prospective students may inquire about admissions requirements, application deadlines, programme details, scholarship opportunities, and more. Responding to these varied question types requires a comprehensive understanding of the admissions procedure and the ability to provide accurate and individualised information.

- **Personalized Guidance**

Throughout the admissions process, numerous applicants seek personalised guidance. They may require assistance selecting the proper programme, comprehending eligibility requirements, and preparing for entrance exams. Due to the need to provide individualised guidance and support to each applicant, it can be challenging to meet these varying demands. Chatbots can be programmed to provide individualised guidance based on specific parameters, thereby aiding applicants in navigating the admissions process effectively.

- **Language and Communication**

International applicants and individuals with diverse linguistic backgrounds frequently make contact with educational institutions. Language barriers can hinder effective communication and lead to misunderstandings. Chatbots endowed with natural language processing capabilities can circumvent this difficulty by understanding and responding to inquiries in multiple languages. In addition, they can be combined with translation tools to facilitate effective communication with applicants who speak a variety of languages.

- **Real-time Updates**

The admissions process is subject to frequent updates and alterations, including deadline extensions, new programme offerings, and revised eligibility requirements. Using conventional channels of communication, it can be challenging to ensure that applicants are promptly informed of these updates. Chatbots can provide applicants with real-time updates, informing them of any changes or important announcements, thereby reducing confusion and promoting openness.

- **Data Security and Privacy**

Admission-related queries Frequently, sensitive information, such as personal details, academic records, and financial documents, must be disclosed. Maintaining data security and confidentiality is of the utmost importance for safeguarding applicants' private information. Higher education institutions must implement stringent security measures to safeguard the data collected by chatbots and ensure compliance with applicable data protection regulations.

Educational institutions can resolve these obstacles and improve the efficacy and efficiency of their admissions processes by utilizing chatbot technology. Implementing well-designed chatbot systems can expedite communication, provide personalized guidance, and enhance the experience overall for prospective students and their families. To address these obstacles, the chatbot system must incorporate sophisticated natural language processing techniques, robust information retrieval systems, secure data handling protocols, and intelligent decision-making capabilities.

2.5 Related Research in Chatbot Development

Prior research in the field of chatbot development has concentrated on various aspects, such as design and implementation, historical context, evaluation methods, and application in specific domains. In this section, we will discuss some related works:

- Comparison of Design Techniques: A study Error: Reference source not found compared design techniques used in chatbot development, focusing on the models used to construct winning chatbots.
- Identification of Limitations and Research Areas: A literature review Error: Reference source not found examined areas where chatbots fall short and highlighted research areas requiring attention in order to steer future research.
- Cloud-based Chatbot Technology: The authors of a survey Error: Reference source not found on cloud-based chatbot technology, chatbot programming, and present and future programming issues emphasized the significance of stability, scalability, and adaptability in chatbot development.
- Design, Architecture, and Algorithms: A study Error: Reference source not found conducted a comprehensive literature review on chatbot design, architecture, and algorithms. It investigated the various approaches and techniques used for chatbot development.
- Systematic Literature Review: A systematic literature review Error: Reference source not found examined various aspects of chatbots, expressing concern over the volume of published material and emphasizing the need for inter-disciplinary research in the field.
- Comparison of Eleven Common Chatbot Application Systems: An Analysis [12]compared the functionality and technical requirements of eleven prevalent chatbot application systems, illuminating the various features and capabilities of each.
- Historical Progression and Implementation Technologies: A study [13] presented a categorization scheme, explored implementation technologies, and discussed the general architecture of contemporary chatbots.
- Deep Learning Techniques: A research paper Error: Reference source not found addressed the issue of identifying appropriate deep learning techniques for chatbot development by providing an overview of commonly used deep learning systems models and implementation advice and resources.

These related works contribute to the comprehension of chatbot development, highlight design techniques, identify limitations, investigate implementation technologies, and shed light on various chatbot system aspects.

2.6 NLP Techniques and Models for Chatbots

Chatbots have emerged as powerful tools that can assist multiple users simultaneously and without delay. They not only provide support and assistance but also offer companionship and enjoyment to users, thereby enhancing productivity. Several models are commonly used in the development of chatbots, including:

- **Transformers**

Transformers were initially developed to address the challenge of sequence transduction, which involves converting input sequences into output sequences Error: Reference source not found. This technique has applications in various tasks, such as text-to-speech conversion and speech recognition. Transformers ensure consistent representation of input and output sequences, even when the sequences undergo changes such as shrinking and stretching.

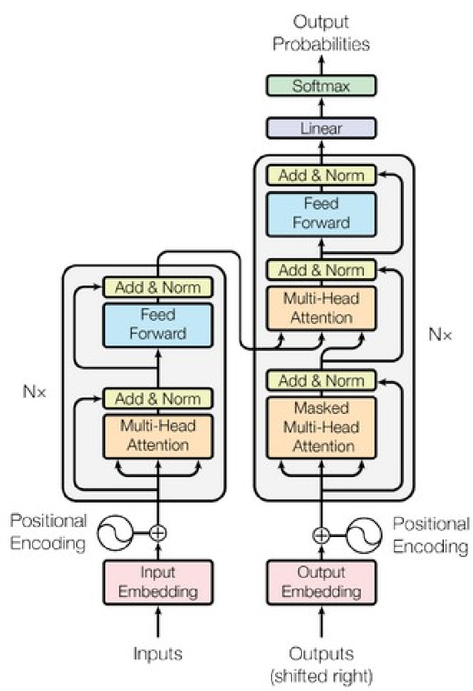


Figure 2.7 Transformers

- **BERT (Bidirectional Encoder Representations from Transformers):**

BERT is a powerful model widely used in chatbots[Error: Reference source not found. It is designed to pre-train deep bidirectional representations from unlabeled text by simultaneously considering the left and right context.

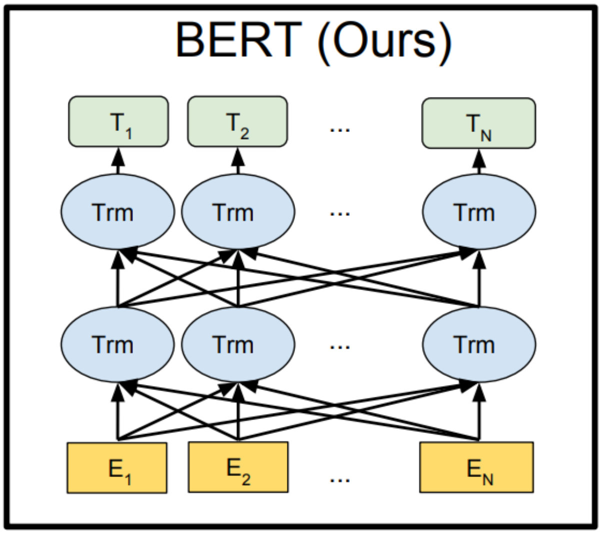


Figure 2.8BERT

- **OpenAI's GPT**

OpenAI's GPT (Generative Pre-trained Transformer) is a state-of-the-art language model trained on vast amounts of internet text data. GPT excels at generating coherent and contextually pertinent text due to the Transformer architecture and self-attention mechanisms. It learns grammar, syntax, and semantic relationships through training that entails predicting the next word based on the previous context. GPT has demonstrated efficacy in a variety of natural language processing tasks and is widely used for chatbots, virtual assistants, and content generation due to its capacity to generate contextually aware text that resembles human language.

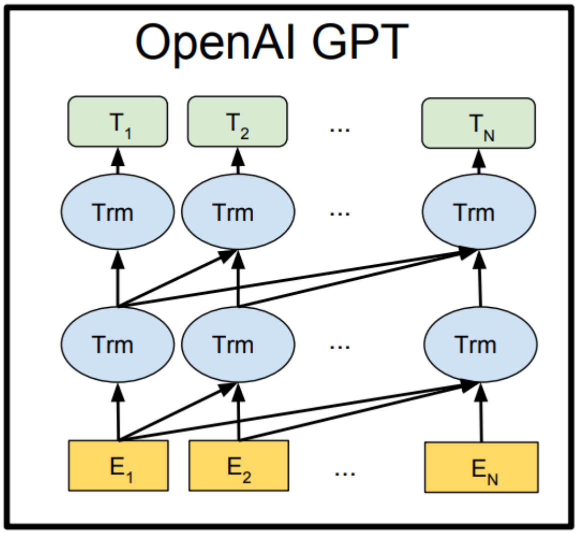


Figure 2.9OpenAI GPT

- **ELMo**

ELMo was the way the "NLP community" dealt with polysemy, which is when the identical words could have multiple interpretations based on the situation. To train "word embeddings", we transitioned from training "shallow feed-forward networks (Word2vec)" to "layers of complex bidirectional LSTM architectures". This means that the identical word could have distinct "ELMO embeddings" according to the situation.

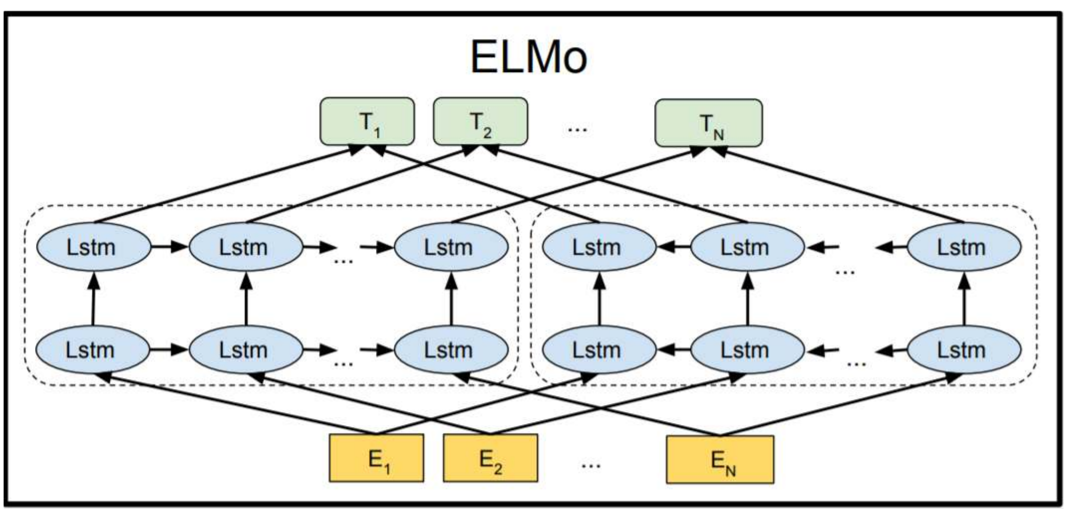


Figure 2.10 EMLo

- **ULMFiT**

On a range of "document classification tasks", our system were able to train "language models" that could be modified to get outstanding performance despite having fewer input-data (less than "100 examples"). ULMFiT has undoubtedly resolved the "puzzle of NLP learning transfer".Error: Reference source not found

Long-Term Short-Term Memory

LSTM, which stands for Long Short-Term Memory, is a type of recurrent neural network architecture used in chatbots. It is a powerful sequence learning architecture that allows the representation and processing of sequential data.Error: Reference source not found

- **Word2Vec**

Word2Vec is a word embedding technique frequently used in chatbot models. It converts words into numerical vectors, allowing machine learning algorithms and deep learning architectures to process and understand textual data Error: Reference source not found. Word2Vec embeddings can be based on frequency or prediction, providing representations that capture semantic relationships between words.

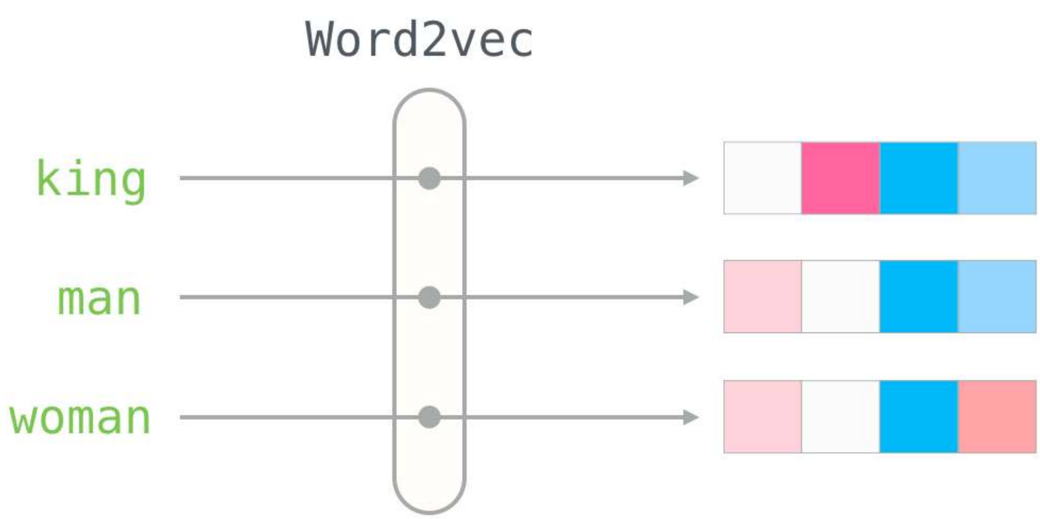


Figure 2.11Word2Vec

NLP techniques and models, such as Transformers, BERT, ULMFIT, LSTM, and Word2Vec, play a crucial role in the development of chatbots. These models allow chatbots to process and generate responses based on user inputs, thereby augmenting their capacity to provide effective support and assistance.

2.7 Evaluation Metrics for Chatbot Performance

Several evaluation metrics are commonly used to measure the performance and efficacy of chatbots. These metrics assist in gauging the quality of chatbot interactions and providing insights into their overall performance.Error: Reference source not found

- **Accuracy or Precision**

Accuracy measures the proportion of correct responses provided by the chatbot over the total number of interactions; it indicates the chatbot's ability to generate accurate and relevant answers. Precision focuses on the percentage of correct responses out of all responses given by the chatbot; it indicates the chatbot's ability to avoid providing incorrect or irrelevant answers.

- **Recall**

Recall evaluates the chatbot's ability to retrieve relevant information from the knowledge base or memory. It measures the percentage of relevant responses provided by the chatbot relative to the total number of relevant responses that could have been given.

- **F1 Score**

The F1 score is the harmonic mean of precision and recall, providing a balanced measure of the chatbot's performance. It takes both precision and recall into account and is useful when there is an imbalanced distribution of positive and negative responses.

- **Response Time**

Response time refers to the time it takes the chatbot to generate a response after receiving an input from the user. This is an important metric, as users expect prompt and timely responses. Lower response times indicate a more efficient and responsive chatbot, thereby enhancing the user experience.

- **User satisfaction**

User satisfaction is frequently assessed through surveys or feedback mechanisms. It measures the satisfaction level of users after interacting with the chatbot. User satisfaction can be evaluated based on factors such as ease of use, helpfulness of responses, and overall user experience.

- **Error Rate**

The error rate measures the frequency of incorrect or erroneous responses generated by the chatbot. It indicates the chatbot's accuracy and reliability in providing correct answers.

Metrics such as accuracy, recall, precision, F1 score, response time, user satisfaction, and error rate provide valuable insights into the chatbot's performance, enabling developers to improve the user experience. Error: Reference source not found

Chapter 3

Methodology

In Chapter 3, the research methodology employed is discussed. We start with a discussion of the research design, outlining the overall methodology and structure of our investigation. This comprises the strategies and methods employed to accomplish our research objectives.

The procedure for data collection and preparation will then be discussed. We describe the sources from which we gathered the necessary data and how we structured and preprocessed it to assure its suitability for training the chatbot model.

The architecture of the model chatbot is then described, including the framework and components used to build the chatbot. We investigate the technical aspects of the design, including algorithm selection, neural network architecture, and other pertinent factors.

The discussion that follows focuses on the training and fine-tuning of the robot. We describe the techniques and approaches used to optimise the model's performance, including the training procedure, hyperparameter optimisation, and efficacy evaluation metrics.

Now, our focus will be on web application development. We describe the methodologies and technologies employed in the development of the user interface and functionality of the web application hosting the chatbot.

The procedure of integrating the chatbot into the web application is described next. The integration of the chatbot functionality into the application without disrupting the user experience is discussed.

Finally, we consider our investigation's ethical implications. We discuss the importance of data privacy, user consent, and responsible AI development, highlighting the measures taken throughout the project to mitigate potential risks and ensure ethical behaviour.

3.1 Research Design

This project's research design consists of two major phases: the data collection phase and the development phase. After that, there is deployment.

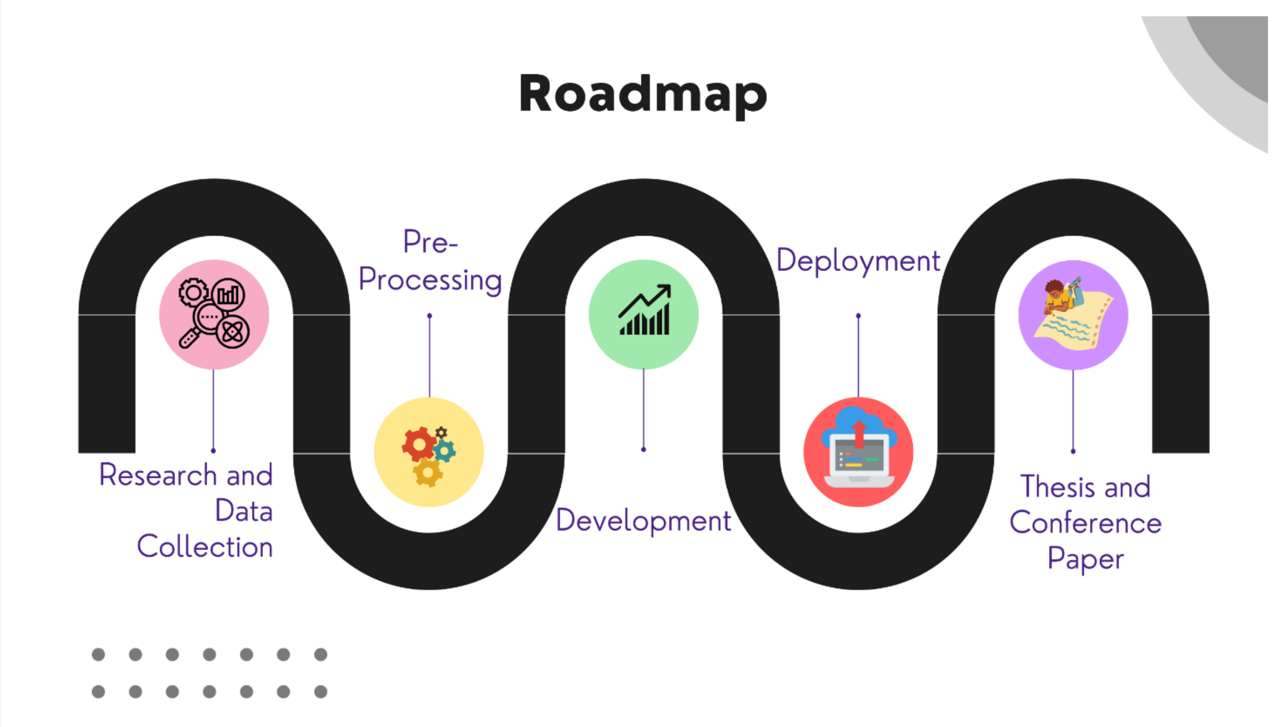


Figure 3.12Roadmap for chatbot

3.2 Data Collection and Data Preparation

3.2.1 Data Collection

The initial phase of the research design centered on the collection of the necessary data to train and evaluate the chatbot system.

1. We contacted PIEAS's point-of-contacts to acquire insights and information regarding admission procedures, frequently asked questions, and prospective students' common concerns.
2. We obtained questions and inquiries from the PIEAS admission guidelines Facebook group, where students actively participate in discussions and seek advice from their peers.
3. We thoroughly combed the PIEAS website for pertinent information, admission requirements, and frequently asked questions.

4. To ensure a comprehensive dataset, we created survey forms tailored to PIEAS students in order to collect their feedback, comprehend their concerns, and identify areas where the chatbot system could provide valuable assistance. Using the survey responses, we identified focus groups representing the areas where students encountered the most difficulties and had the greatest number of questions. We then expanded the dataset by adding more questions related to these focus areas to ensure that the chatbot could effectively respond to a wide variety of user questions.

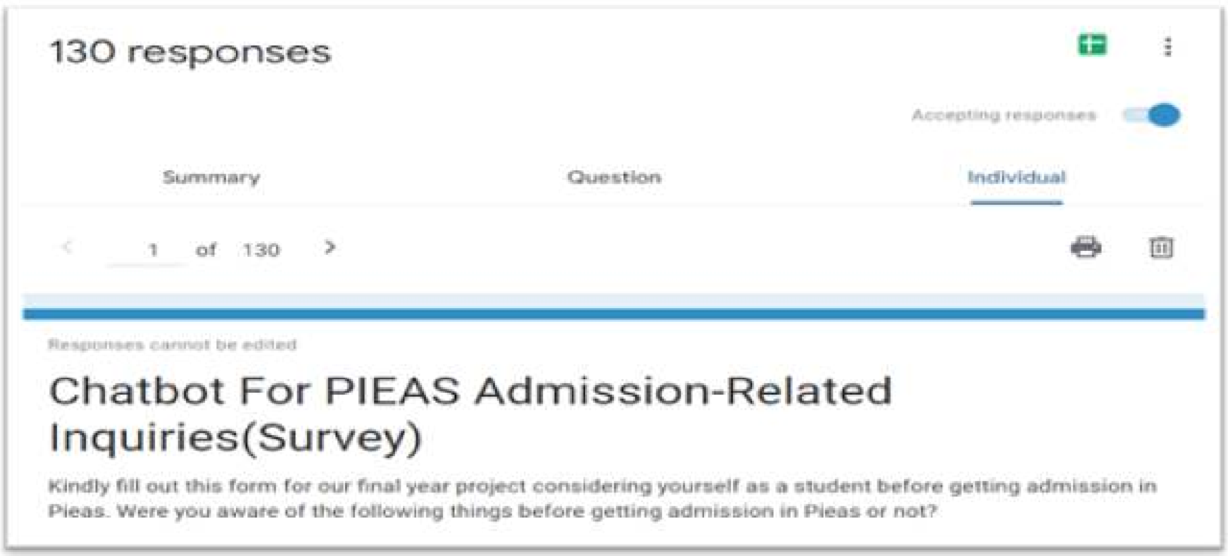


Figure 3.15Survey Form

Figure 3.13Survey question response graph-1

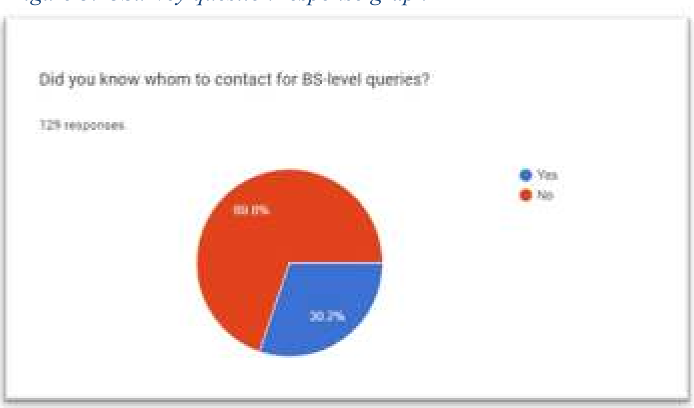


Figure 3.14 students problems faced during admission process

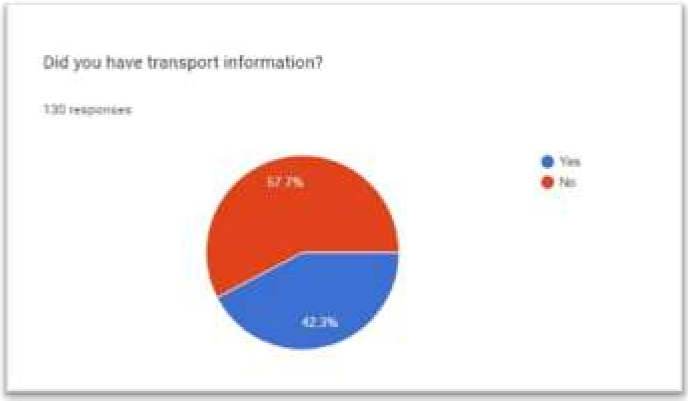


Figure 3.17survey question response graph-2

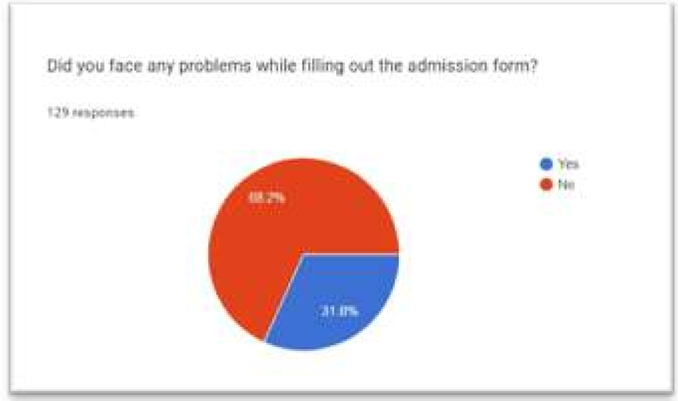


Figure 3.16survey question response graph-3

5. We then created conversational dataset in order to make our chatbot human-like. As there was no relevant dataset available online, therefore we created that dataset ourselves comprising of 164 question-answer pairs.

Through these efforts, we were able to compile a substantial number of questions and queries related to PIEAS admission, which formed the basis of our dataset. For our project, a total of 4228 query and response pair were collected.

```
[3]
df1 = pd.read_csv('/content/compiled_dataset.csv', encoding= 'unicode_escape', on_bad_lines='skip')
df2 = pd.read_csv('/content/conversational_dataset.csv', encoding= 'unicode_escape', on_bad_lines='skip')

[4]
df1 = df1 [['question', 'text', 'context']]
print(df1.shape)
df2 = df2 [['question', 'text', 'context']]
print(df2.shape)
df1 = pd.concat([df1, df2])
print(df1.shape)
df1.shape

(4064, 3)
(164, 3)
(4228, 3)
(4228, 3)
```

Figure 3.18number of question-answer pairs

3.2.2 Data Preparation

The labeling of the collected data was an essential step in data preparation. Labeling involves assigning target outputs or categories to the input data to enable the machine learning model to learn and predict the appropriate responses. We recognized the importance of high-quality labeled datasets in natural language processing and utilized the haystack annotation tool to efficiently annotate and structure the data.

Haystack Annotation Tool:

Annotating the dataset was a crucial stage in the process of preparing the collected data for training the chatbot model. Annotation involves labelling data with outputs or categories that the machine learning model must learn and predict accurately.

The Haystack framework, an open-source instrument for constructing intelligent search systems over large document collections, was utilised to collect and organise the data for our study. The Haystack framework offers a variety of functionalities and components that facilitated our data management process.

Using the Haystack annotation tool, we structured and labelled the gathered data to construct a form that captured vital information. This instrument facilitated the annotation process by incorporating features such as semantic search and document retrieval. We efficiently annotated the data using these features, which served as the foundation for training and evaluating our chatbot model.

Moreover, the Haystack framework provided diverse data administration capabilities, allowing us to preprocess and transform the data according to our requirements. We were able to execute tasks such as text cleaning, tokenization, and feature extraction using these capabilities. This ensured the data was accessible for training and refining the chatbot model in the proper format.

Using the Haystack framework and its data preparation tools, we were able to expedite data collection and annotation, ensuring that our chatbot training model has access to high-quality labelled data.6.5.1(1)(a)(i)[21]

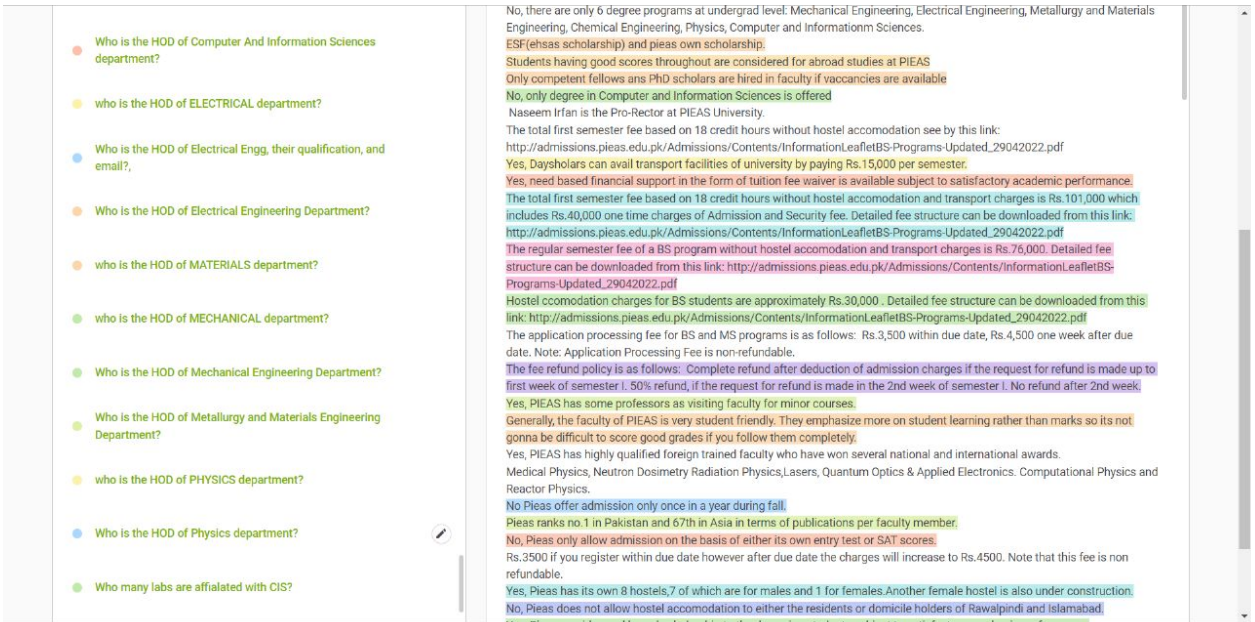


Figure 3.19 Haystack annotation tool

3.3 Chatbot Model Architecture

A transformer architectural paradigm called BERT (Bidirectional Encoder Representations from Transformers) is used for sentiment analysis, question answering, and natural language processing. The researchers at Google created the pre-trained model BERT in 2018. BERT is distinctive and different from other machine learning models in that it was pre-trained using a sizable unlabeled text corpus that includes a book corpus and Wikipedia. BERT is profoundly bidirectional, both from left to right and right to left text representation.[25]

In 2018, Google unveiled BERT (Bidirectional Encoder Representations from Transformers), a cutting-edge transformer-based approach. By offering contextualized word representations and capturing intricate semantic links between words in a sentence, it has completely changed natural language processing tasks. One variation of the BERT model, which is used extensively for various natural language processing applications and is trained on a sizable corpus of text data, is BERT base uncased.

Transformer Encoder and **Tokenization** are the two fundamental parts of the architecture of BERT base uncased.

The main element of BERT base uncased is the Transformer Encoder. The model's foundation is the self-attention mechanism, which enables it to simultaneously collect dependencies between each input character. The Transformer Encoder is made up of a stack of similar layers, commonly denoted as L, with several self-attention sublayers and positionally completely connected feed-forward neural networks on each layer.

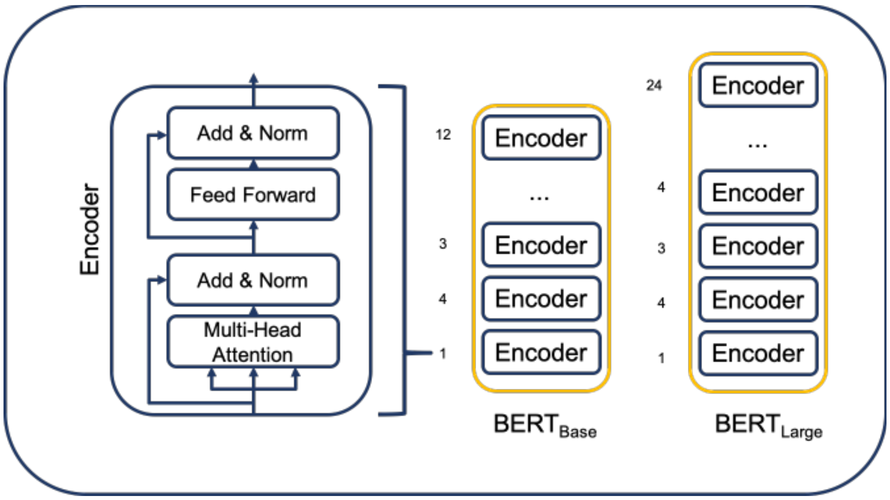


Figure 3.20 Transformer encoder for BERT base uncased

BERT base uncased employs multi-head self-attention in each layer, where the input embeddings are split into numerous representations known as attention heads. Each attention head focuses on a separate segment of the input sequence in order to collect various contextual data. Due to this, both local and global dependencies between words in a phrase can be properly captured by BERT base uncased.

Each token can take into account the representations of all other tokens thanks to the self-attention mechanism, which enables BERT base uncased to capture contextual information in both directions. In contrast to earlier language models, which were often unidirectional or context-independent, BERT is bidirectional.

The non-linear transformations carried out individually for each point in the sequence by the position-wise feed-forward neural networks in each layer of the Transformer Encoder fulfil this purpose. These networks support the model's capacity to capture rich semantic representations by helping to capture complicated links between words.

Tokenization, which specifies how the input text is broken up into separate tokens or sub words, is a crucial component of BERT base uncased. Word Piece tokenization, used by BERT base uncased, divides words into sub words based on a specified vocabulary. The model can properly handle terms that are outside of its vocabulary thanks to this method.

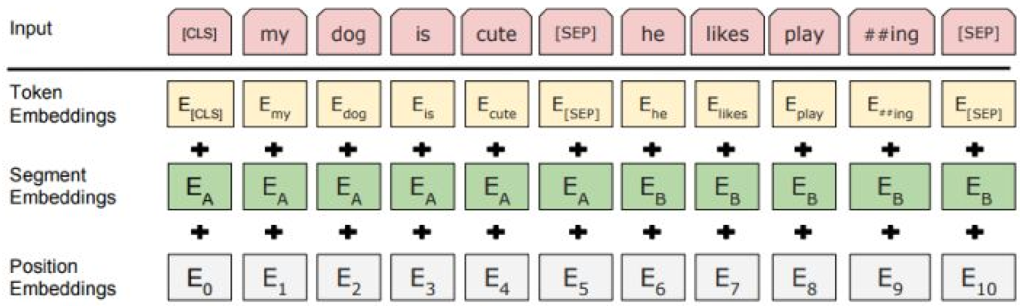


Figure 3.21 Tokenization in BERT base Uncased

BERT base uncased also adds special tokens to the input sequence during tokenization. In tasks that require sentence pairs, the [CLS] token is added at the beginning of the sequence to represent the entire input, and the [SEP] token is used to divide two sentences. Additionally, to guarantee that all input sequences are the same length, BERT base uncased appends [PAD] tokens.

The input text is changed throughout the tokenization process into a list of token IDs, which are then provided to the Transformer Encoder as input. These token IDs are sent into the layers of the model, and during training, BERT base uncased learns to anticipate the tokens that are being hidden within the input sequence, improving its capacity to recognize context.

Using tokenizer and transformer encoder, we generated word embeddings for our dataset. Our dataset had the following columns, “question”, “text” and “context”. We had to generate word embeddings for question and context which look like these.

```
question_embeddings = model.encode(df1['question'].values.tolist(), show_progress_bar=True)
```

```
Batches: 100% 82/82 [02:30<00:00, 1.09s/it]
```

```
context_embeddings = model.encode(df1['context'].values.tolist(), show_progress_bar=True)
```

```
Batches: 100% 82/82 [40:01<00:00, 5.48s/it]
```

Figure 3.22 Generating question and context embeddings.

By using BERT base uncased, dense, real-valued vectors are produced as word embeddings. The design and configuration of the model determine the dimensionality of each high-dimensional vector that corresponds to a word in the lexicon. Through a procedure known as unsupervised training on a sizable corpus of text data, these word embeddings are discovered. BERT base uncased makes use of its self-attention mechanism and bidirectional nature during training to record intricate contextual links between words. This enables the model to produce detailed and insightful representations of each word based on the context of that word inside a given phrase or length of text. The semantic linkages and similarities between words are reflected in the created word embeddings. In the embedding space, words with comparable or closer vector representations are more likely to be related syntactically or semantically. The embeddings can successfully capture the meaning and context of words thanks to this characteristic.

```
question_embeddings
```

```
array([[ 3.29678178e-01,  2.74867833e-01,  2.17182785e-01, ...,
        -1.97089002e-01,  2.54550457e-01, -4.02132906e-02],
       [ 1.11462459e-01, -2.70555895e-02,  1.21516809e-01, ...,
        -1.96745351e-01,  3.15077841e-01,  6.17086440e-02],
       [ 3.33854109e-01,  4.53453064e-01,  4.34926599e-01, ...,
        -1.51044875e-01,  3.21103871e-01,  5.44935279e-02],
       ...,
       [ 1.00449227e-01, -3.54624301e-01,  6.39706701e-02, ...,
        -2.02501677e-02,  2.60132104e-01, -2.04704121e-01],
       [ 1.13167174e-01, -5.44728816e-01, -3.39582190e-02, ...,
        -4.01169062e-04, -2.89363503e-01, -4.32496756e-01],
       [ 2.07186356e-01, -2.07644656e-01,  9.99005791e-03, ...,
        -2.13861600e-01,  1.66923583e-01, -1.38759091e-01]], dtype=float32)
```

```
context_embeddings
```

```
array([[-0.2684921 ,  0.02954091,  0.3890913 , ..., -0.07933156,
         0.30508444, -0.06182293],
       [ 0.09166487,  0.22167893,  0.30041006, ..., -0.08164325,
         0.17101862,  0.03299486],
       [-0.14947493,  0.1573617 ,  0.3969109 , ..., -0.13645048,
         0.13437353,  0.08853216],
       ...,
       [ 0.22507681,  0.08954181,  0.40847883, ..., -0.02451148,
         0.19923317, -0.18032004],
       [ 0.5973639 ,  0.11935905,  0.10652638, ...,  0.03449176,
        -0.02763441, -0.14368235],
       [ 0.19653007, -0.0581788 ,  0.03069007, ..., -0.03155151,
         0.3184981 , -0.08526157]], dtype=float32)
```

Figure 3.23 Question and Context Embeddings

3.4 Answer Retrieval using BERT Tokenizer and Similarity Index

The use of a pre trained model has many advantages. By doing so, you may use cutting-edge models without having to spend time and money training them from scratch, which also lowers computing costs and your carbon footprint. For a variety of purposes, Transformers offers access to thousands of pre trained models. A pre trained model is one that has been trained on a dataset particular to your task. This is a very effective training method called fine-tuning. [24]

After setting up tokenizer, it's time to feed it some text to turn into token IDs that Bert can understand. Bert tokenizer is in charge of transforming human-readable text into token IDs, or data that is friendly to Bert. A string is first divided into tokens by the tokenizer as you can see in figure 3.3.4. We turn this collection of texts (tokens) into a list of integers (token IDs) after the initial tokenization[23].

After generating these tokens. We created a dictionary called embeddings to hold embeddings. 'Questions' and 'Answers' are the two keys in this dictionary, and they stand for question embeddings and context embeddings, respectively. The embeddings dictionary is serialized and written to a file called "my-embeddings.pkl" using the pickle module. The dictionary is serialized and saved in binary format within the file by using the pickle.dump () function. This makes it possible to save and retrieve the embeddings at a later time with efficiency. The original dictionary with its contained values can be restored by reserializing the serialized data using the pickle.load () function. This method, which is frequently employed in tasks involving natural language processing like information retrieval and question-answering, offers an easy way to preserve embeddings. The reason for doing so is that, generation of word embeddings is a very time consuming task. If we don’t save the embeddings in a dictionary it will generate them again every time a question is asked which will increase the response time of our chatbot manifolds.

```
[ ] # Create a dictionary to store the embeddings
    embeddings = {'questions': question_embeddings, 'answers': context_embeddings}

    # Open a file to write the serialized data
    with open('my-embeddings.pkl', 'wb') as pkl:
        # Serialize the data and write it to the file
        doc_embedding = pickle.dump(embeddings, pkl)
```

Figure 3.24 Saving and fetching generated embeddings

After this, we used the open () method to open the file'my-embeddings.pkl' in read binary mode, and then used the pickle.load () function to load the serialized embeddings. The embeddings variable, which is a dictionary with the entries "questions" and "answers," holds the loaded data. These keys, correspondingly, represent the context embeddings and question embeddings. The ask_question () function is intended to handle user queries and return the best response. The query entered is used as a parameter. The function initially creates a query vector (query_vect) by encoding the input question using a trained model (model.encode ()).The we calculated the cosine similarity between the query vector and the question and context embeddings (question_embeddings and context_embeddings, respectively) in order to identify the question and context embeddings that are the most comparable. Cosine similarity between two vectors is computed by the cosine similarity () function, which is taken from a scikit learn library. The method then uses np.argmax () to get the indexes of the maximum similarity scores for both question and context embeddings. These indexes represent the input query's most comparable questions and context embeddings. The code then uses the acquired indices to return the maximum scores for question and context similarity (max_question_similarity and max_context_similarity, respectively). The code decides whether the input question is more similar to a known question or a context based on the comparison between max_question_similarity and max_context_similarity. The code extracts the answer text from the relevant index in df1, a Data Frame that has the associated answer texts, if the maximum question similarity is higher. If not, it pulls the response text from the context index. Output of this is the text for both the input query and the appropriate response. Until deliberately stopped, the code is programmed to run in an indefinite loop while accepting human input and responding to it.

```python
with open('/content/my-embeddings.pkl', 'rb') as pkl:
    embeddings = pickle.load(pkl)

question_embeddings = embeddings['questions']
context_embeddings = embeddings['answers']

def ask_question(question):
    query_vect = model.encode([question])

    question_similarities = cosine_similarity(query_vect, question_embeddings)
    context_similarities = cosine_similarity(query_vect, context_embeddings)

    max_question_similarity_index = np.argmax(question_similarities)
    max_context_similarity_index = np.argmax(context_similarities)

    max_question_similarity = question_similarities[0, max_question_similarity_index]
    max_context_similarity = context_similarities[0, max_context_similarity_index]

    if max_question_similarity > max_context_similarity:
        answer_text = df1.iloc[max_question_similarity_index]['text']
    else:
        answer_text = df1.iloc[max_context_similarity_index]['text']

    print('Your question:', question)
   # print('Closest question found:', df1.iloc[max_question_similarity_index]['question'])
   # print('Similarity with closest question: {:.2%}'.format(max_question_similarity))
    #print('Similarity with closest context: {:.2%}'.format(max_context_similarity))
    print('Answer:', answer_text)


while(True):
    ask_question(input())
```

Figure 3.25 Answer Retrieval using BERT Tokenizer and Similarity Index

3.5 System Design

3.5.1 Use case diagram

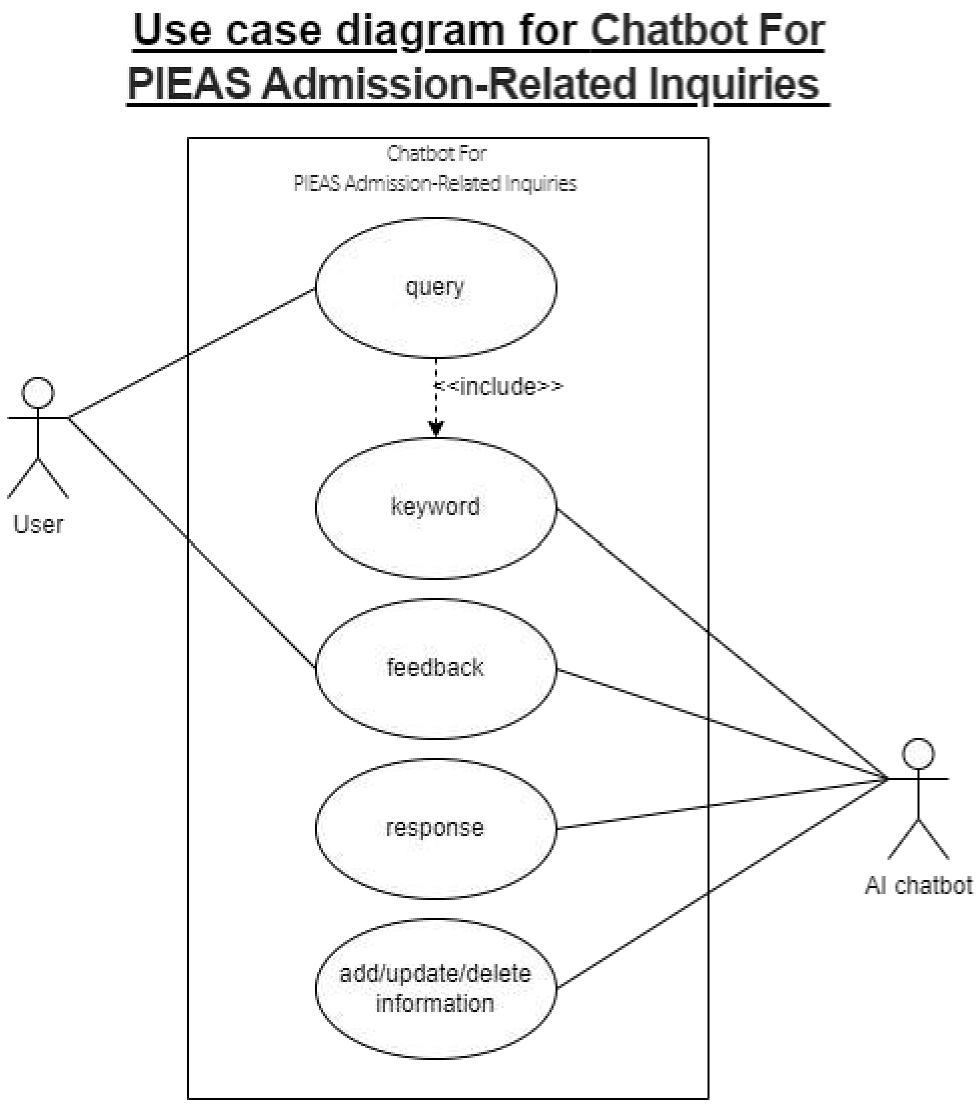


Figure 3.26Use Case diagram

The scope and high-level functions of a system are described in use-case diagrams. The interactions between the system and its actors are also depicted in these diagrams. Use-case diagrams show what the system does and how the actors use it, but they do not show how the system works internally.

Actor:

User, AI chatbot

Description:

User:

- the user will be able to access the chatbot and can ask queries related to admission.
- He can also send feedback at the end of the conversation

AI chatbot:

- Chatbot will be assisting user by sending them suitable response to their queries. It will be done when chatbot will process the query and generates a keyword from it.
- On the basis of feedback, if valid, improvements can also be induced in the chatbot
- Chatbot is made scalable that means it can be modified, databases can be updated

3.5.2 Activity Diagram

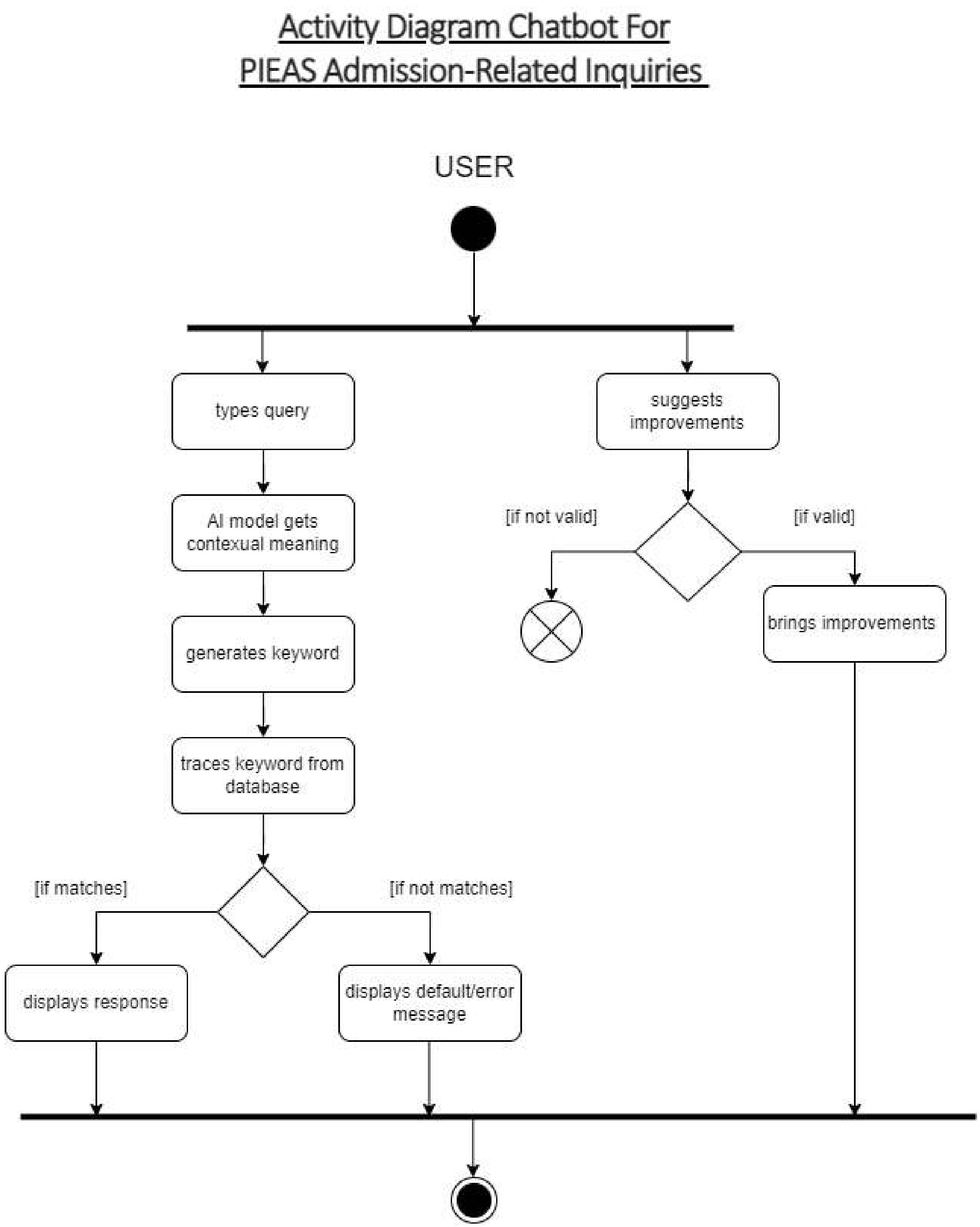


Figure 3.27 Activity diagram

Similar to a flowchart or data flow diagram, an activity diagram visually displays a series of actions or the flow of control in a system. A common tool in business process modelling is the activity diagram. Additionally, they can outline the procedures in a use case diagram.

- The user will be performing two main activities:
    - Typing query
    - Sending feedback

Typing query:

This activity will be followed by multiple steps including:

- Processing of query by an efficient deep-learning algorithm
- Generating keyword from the asked query
- Finding keyword in the database
- Displaying response to the user
- If keyword will not be found in the database, the administrator will himself assist the user with appropriate answer or by sending default or error message

Sending Feedback:

This activity will be followed by multiple steps including:

- On sending feedback by the user, the admin will decide whether that feedback is useful for the chatbot or not
- If the feedback will be valid and the required improvements will be needed, then the administrator will bring those changes as needed.

If the feedback will be considered useless, it will stay in the database

3.5.3 Sequence Diagram

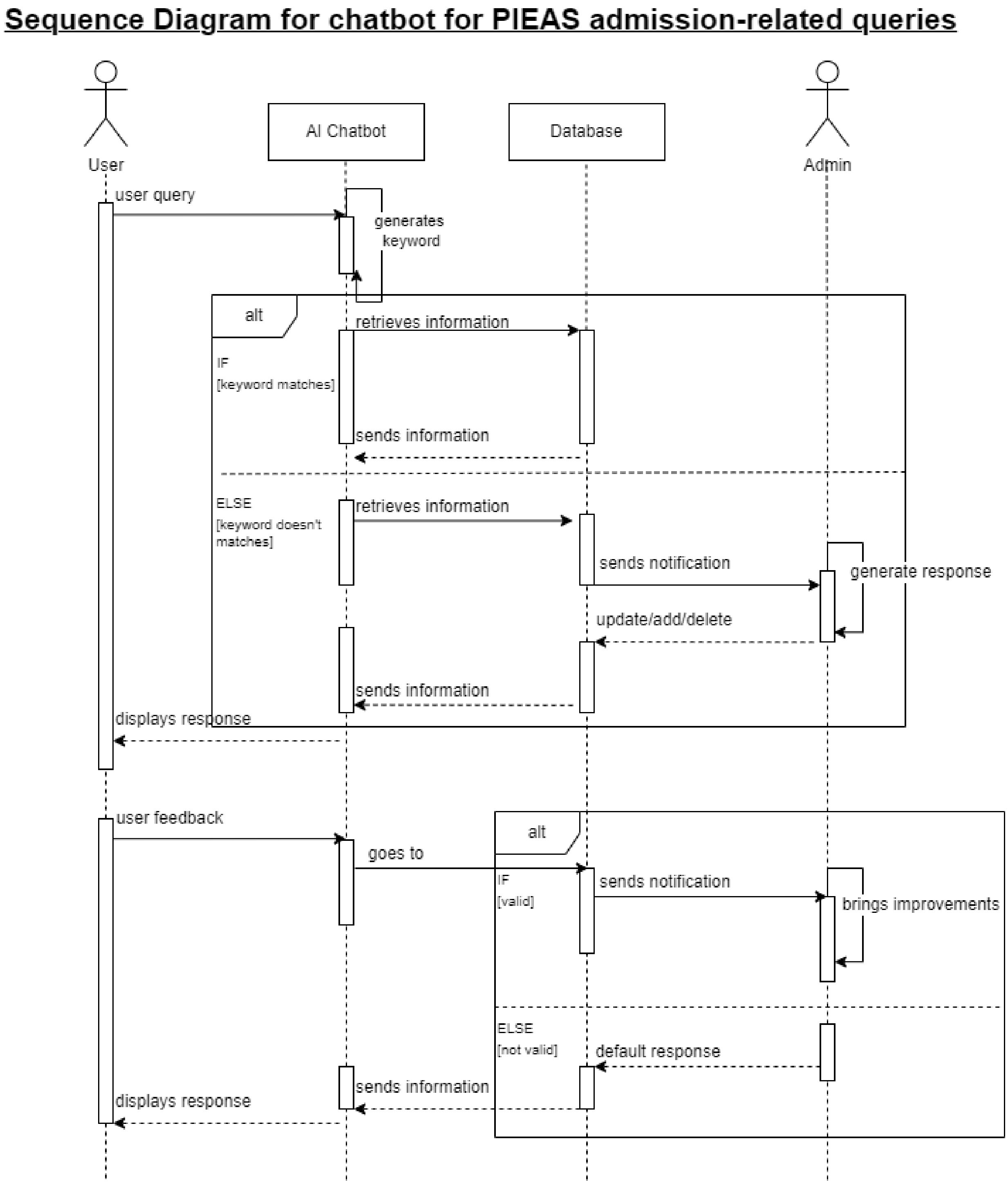


Figure 3.28 Sequence diagram

A sequence diagram is a diagram created using the Unified Modeling Language (UML) that shows the flow of messages sent and received by objects during an interaction. A group of objects that are represented by lifelines and the messages they exchange over the course of an interaction make up a sequence diagram.

There are 4 objects in the sequence diagram:

User:

- The user will send the query and wait for its response
- User can also send feedback.

AI chatbot:

- It will receive query and start its processing resulting into generating a keyword for the query
- Retrieves information from database and display it to the user
- If keyword will not be found then that keyword will be sent to the administrator for further processing
- On receiving feedback from the user, it will be sent to the administrator for its evaluation

Database:

- On receiving keyword from chatbot, it will find appropriate response to it and will send it back to the chatbot
- In case keyword is not found, it sends a notification to the administrator
- Data in the database can be updated, deleted and added.
- Database will also store feedbacks coming from the user

Admin:

- He will generate response to those keywords which were not found in the database
- He will bring improvements if needed.

3.6 Web Application development

For the web application development part of this project, we used Flask framework. The reason being, Flask is a well-known and compact web framework. It is a superb option for developing web applications because it is made to be straightforward, adaptable, and simple to use. The IDE used to develop the web application is pycharm. PyCharm is an Integrated Development Environment (IDE) specifically designed for Python development. Developed by Jet Brains, PyCharm offers a range of features and tools to enhance the development experience.

The students of Pakistan Institute of Engineering and Applied Sciences (PIEAS) and the students seeking admission in PIEAS is the target audience for this web application. This web application intends to address a number of issues related to an effective and user-friendly system for the university. The programme has a number of crucial features that are intended to improve the user experience for management and students alike.

The responsiveness of the web application is one of its core components. Due to the increasing use of mobile devices, it is crucial that the application fluidly adjust to multiple screen sizes and resolutions to provide the best possible user experience. This responsiveness ensures that students may use the platform easily from their desktops, cellphones, or tablets, enhancing accessibility and usage.

Because the online application is designed to support numerous students at once, many

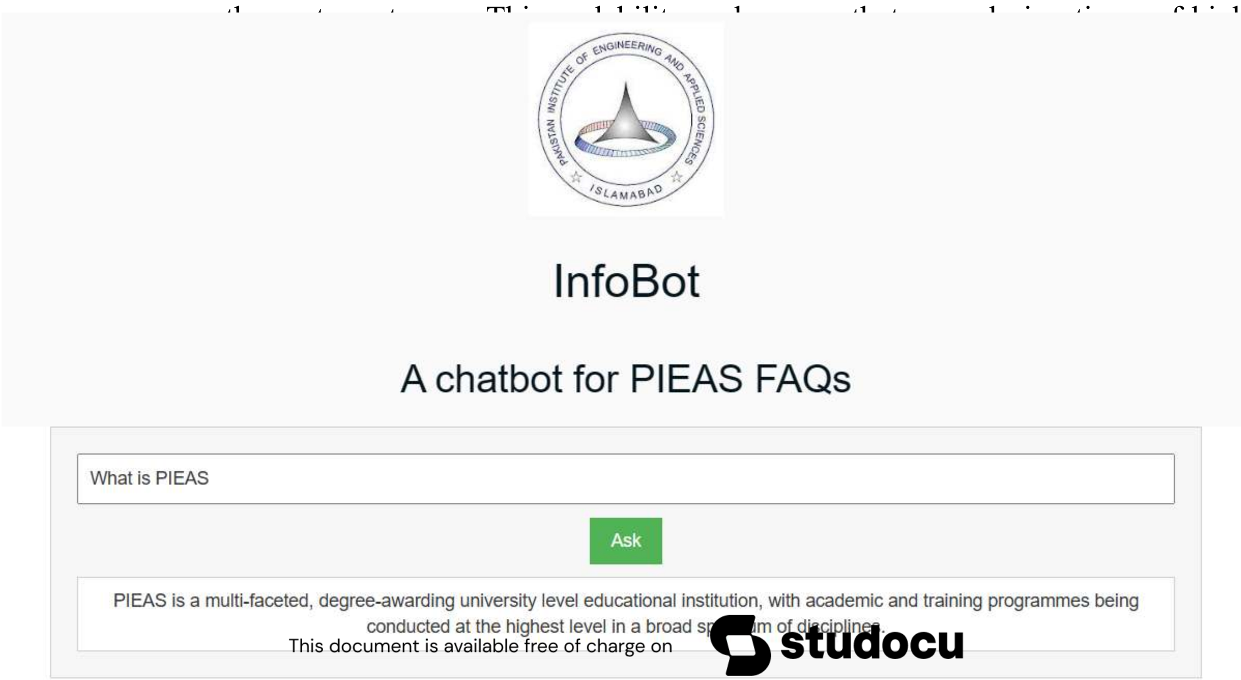


Figure 3.29 InfoBot

Fig 3.5.1: Web Application for Chatbot

During the development of the programme, ease of use and user-friendliness were key considerations. Students will find the interface to be simple and logical. The technology is simple for students to use, and they can quickly locate pertinent information and communicate with the chatbot.

The university can save money by using the web application instead of manual procedures and paperwork. With the chatbot included, students may rapidly find solutions to their questions without requiring human assistance. This automation allows the university to effectively simplify its operations by saving time and money.

The web application's integration of a chatbot considerably increases time savings and accessibility around-the-clock. Students can use the chatbot to ask questions at any time of day, including beyond usual business hours, and get prompt answers. This constant accessibility guarantees that information is available to students whenever they need it, improving their overall experience and pleasure.

Additionally, the chatbot offers reliable information and responses. It is built to offer precise and consistent responses to frequently requested queries, ensuring that students continuously obtain trustworthy information. This feature removes the potential for inconsistent or inconsistent responses, making the system more dependable and efficient.

Additionally, scalability is a consideration in the design of the online application. The system can readily handle rising user demands and data needs as the university develops and evolves. Future improvements and additions to the programme are possible thanks to its scalability without affecting its functionality or performance.

The online application created for PIEAS, in its whole, provides a thorough and user-centric approach to fulfil the needs of students and the university administration. The application is a useful tool for improving the overall educational experience and administrative procedures at the university because of its responsiveness, scalability, cost effectiveness, time-saving capabilities, 24/7 availability, consistency, and user-friendly interface.

3.7 Integration of Chatbot into the Web Application

The purpose of developing a web application integrated with a chatbot for PIEAS (Pakistan Institute of Engineering and Applied Sciences) FAQs is to provide students with an effortless and convenient way to access information about the university. The web application acts as a single hub where students can quickly and easily obtain whatever information they need on PIEAS in light of the growing reliance on online platforms. The chatbot serves as an intelligent virtual assistant that rapidly and accurately responds to user enquiries, whether they relate to information about admissions policies, course offers, faculty members, or general inquiries. Students may access this wealth of knowledge anytime, anywhere by integrating the chatbot into the web application, eliminating the need to go through numerous webpages or wait for manual assistance. This improves students' entire experience with PIEAS's online presence and enables them to make wise decisions and quickly find answers to their questions.

BERT (Bidirectional Encoder Representations from Transformers) base uncased is a key component of the chatbot architecture used in this project. Modern transformer-based model BERT is renowned for its outstanding performance across a range of natural language processing tasks. BERT base uncased is used in the chatbot architecture to create word embeddings, which capture the contextual meaning of the words in a conversation. Understanding the semantic content and context of user inquiries depends on these embeddings. The BERT model can recognize complex linguistic patterns and nuances because it has already been pre-trained on a sizable corpus of text data. In order to reflect the underlying meaning of the input text during inference, the chatbot makes use of the BERT model to encode user inquiries into fixed-length vectors. The chatbot chooses the best response for the user's query by evaluating the cosine similarity between these word embeddings and a library of pre-defined answers. With the help of this architecture, the chatbot may respond to user enquiries with accuracy and relevance to the context by leveraging the strength of BERT's language comprehension skills.

Flask, a compact and adaptable Python web framework, is used to build the web application's main implementation. The tools and features required to build a smooth and interactive user experience are provided by Flask. Flask is used in this project to manage the routing and rendering of web pages. The main.py file, which defines the routes and their accompanying functions, acts as the application's entry point. The chatbot.html template, which serves as the chatbot's user interface, is rendered by the '/' route. The '/feedback' route manages user feedback submission, enabling users to offer suggestions and aid in the enhancement of the chatbot's functionality. The '/ask' route handles user requests and makes use of the Sentence Transformer model's ability to encrypt the input query and determine the cosine similarity to the preprocessed query and context embeddings. The chatbot extracts the most pertinent response from the dataset based on the answers with the highest similarity scores. The request object in Flask is used to manage incoming data, including user inquiries and comments. The web application gives students an easy-to-use interface for interacting with the PIEAS chatbot and getting correct information by fusing the capabilities of the chatbot with the simplicity of Flask.

```python
@app.route('/ask', methods=['POST'])
def get_answer():
    question = request.form['question']

    # Perform similarity calculations and retrieve the answer
    query_vect = model.encode([question])
    question_similarities = cosine_similarity(query_vect, question_embeddings)
    context_similarities = cosine_similarity(query_vect, context_embeddings)

    max_question_similarity_index = np.argmax(question_similarities)
    max_context_similarity_index = np.argmax(context_similarities)

    max_question_similarity = question_similarities[0, max_question_similarity_index]
    max_context_similarity = context_similarities[0, max_context_similarity_index]

    if max_question_similarity > max_context_similarity:
        answer = df1.iloc[max_question_similarity_index]['text']
    else:
        answer = df1.iloc[max_context_similarity_index]['text']

    return answer  # Return the answer as a string
```

Figure 3.30 Code snippet of flask web application

The chatbot's user interface is created to make communicating with it simple and natural for users of all skill levels. The chatbot.html page has a visually beautiful look thanks to the use of HTML structure and CSS styling. Users can ask questions or leave comments on PIEAS in the text input form at the top of the page. Input box placeholder wording instructs users to "Enter your question." Users can submit their inquiry by clicking the "Ask" submit button located underneath the input area. After the user submits their question, the chatbot analyses it and selects the best response from the dataset. The part below the input field that contains the answer is dynamically displayed and is represented by the div element with the id "answer."

```
<div id="chatbot-interface" class="chatbot-interface">
    <form action="/ask" method="post">
        <input type="text" name="question" placeholder="Enter your question">
        <input type="submit" value="Ask">
    </form>
    <div id="answer"></div>
</div>
```

Figure 3.31 Code snippet of chatbot.html

To ensure seamless functionality and excellent communication between the various components, the chatbot must be integrated into the Flask web application through a number of procedures. First, the required Python libraries and modules are imported, including Sentence Transformer, Flask, pandas, pickle, numpy, and sklearn. These libraries offer the fundamental features required for web development, data manipulation, vectorization, computing cosine similarity, and NLP.

```
from flask import Flask, render_template, request
import pandas as pd
import pickle
import numpy as np
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity
from sentence_transformers import SentenceTransformer
```

Figure 3.32 Importing required libraries

The primary routes are then defined, and the Flask application is then initialized. The chatbot.html template, which serves as the chatbot's user interface, is rendered by the '/' route. The '/feedback' link manages user feedback submission and enables users to offer their suggestions for further improvement. The '/ask' route analyses the user's query, transforms it into Sentence Transformer model encoding, and computes cosine similarity with the pre-processed query and context embeddings.

The necessary data is loaded and preprocessed to facilitate the operation of the chatbot. The dataset is read from a CSV file and concatenated to include the questions, answers, and contexts. To maintain data integrity, duplicate questions are eliminated. Additionally, the question and context embeddings are retrieved for use in subsequent calculations, and the pre-trained word embeddings are loaded from a pickle file.

```
if __name__ == '__main__':
    # Load and preprocess your data
    df1 = pd.read_csv(r'D:\pieas\FYP\final ds\compiled_dataset.csv', encoding='unicode_escape', on_bad_lines='skip')
    df2 = pd.read_csv(r'C:\Users\Shajee\Downloads\Conversation.csv', encoding='unicode_escape', on_bad_lines='skip')
    df1 = df1[['question', 'text', 'context']]
    df2 = df2[['question', 'text', 'context']]
    df1 = pd.concat([df1, df2])
    df1 = df1[['question', 'text', 'context']]
    df1 = df1.drop_duplicates(subset='question')
```

Figure 3.33 Importing and preparing data

The Flask development server is then utilized to run the web application. Through a web browser, users can visit the programme and communicate with the chatbot by putting their questions in the input field and sending them. Input is processed by the chatbot, which then computes cosine similarity scores and gets the best response based on those with the highest similarity values. The response is then dynamically posted on the chatbot.html page so users may see it there and now.

Users may interact with the PIEAS chatbot and easily get correct information about the institution thanks to this integration procedure, which effortlessly includes the chatbot's features into the Flask web application. The integration guarantees effective communication between the various components, resulting in a thorough and user-friendly experience.

Chapter 4

Results

We explore the results of our chatbot system in Chapter 4 by thoroughly examining its accuracy and responsiveness. We contrast its performance with those of benchmark systems as well. This chapter offers insightful information on the chatbot's efficiency, highlighting both its advantages and shortcomings. We acquire a comprehensive grasp of the chatbot's performance and its consequences in contrast to existing systems through a rigorous analysis of performance indicators and user satisfaction survey findings. The succeeding portions of this thesis are shaped by these results, which act as a framework for additional debates and analysis.

4.1 Analysis of Chatbot's Accuracy and Response Time

Fundamental metrics that influence the caliber of user interactions with chatbots include accuracy and response time. The ability of the chatbot to comprehend user inquiries precisely and offer pertinent and appropriate responses is referred to as accuracy. Response time, on the other hand, gauges how quickly the chatbot produces responses and has a big impact on user engagement and satisfaction.

The following code snippet serves as an example in the context of analyzing **response times in our chatbot system**. This program focuses on calculating how long it takes a chatbot to respond to a series of 100 queries. The time module is imported at the beginning of the function to evaluate the response time. This module offers tools for precisely measuring time intervals. The current time is returned by the time.time () function in seconds since the epoch. The first 100 questions in the Data Frame are retrieved using the line questions = df1 ['question'][:100]. The response time analysis uses this subset. The start time is then recorded by the script by calling time.time (), the start_time variable is given the result. Each question in the questions list is iterated over in a loop. The ask_question () function is used for each iteration, passing the current query as an argument. In this function, which serves as the chatbot's central logic, the input question is processed, an appropriate response is produced, and possibly any additional calculations or interactions with the outside world are carried out. The end time of the loop is recorded using time.time () and stored in the end_time variable. By deducting the start_time from the end_time, one can get the elapsed_time. The script then prints the amount of time that has passed in seconds using the print () method to output the measured response time.

```python
import time

# ... Your existing code ...

# Get the first 100 questions from df1
questions = df1['question'][:100]

start_time = time.time()

for question in questions:
    ask_question(question)

end_time = time.time()
elapsed_time = end_time - start_time

print('Time taken to answer 100 questions: {:.2f} seconds'.format(elapsed_time))
```

Figure 4.34 Code for finding response time.

Using the provided code snippet, the measured response time for answering 100 questions was 19.60 seconds. The time taken for the chatbot to process and produce responses for the supplied set of queries is shown by this result.

The measured response time is influenced by a number of variables. First, the length and intricacy of the questions can affect how quickly they are processed. Questions that need sophisticated language understanding or additional computational resources may take longer to handle overall, lengthening the response time.

It is significant to notice that the 19.60 second measured response time is unique to the provided code snippet and the specific collection of 100 questions. When the chatbot receives more questions or a variety of questions, the response time could change. As a result, the context of the particular evaluation scenario should be taken into account when evaluating this conclusion. A satisfying user experience depends on effective response time management. Chatbot users demand quick and timely responses, and longer response times may result in decreased user satisfaction and engagement. Therefore, in order to reduce reaction time and maintain maximum performance, chatbot developers and system administrators must constantly improve and fine-tune their systems.

```
Your question: Which fields are the oldest ones in PIEAS?
Answer: MS Nuclear Engineering is the oldest program of PIEAS which
Your question: Who many labs are affialated with CIS?
Answer: Department of Computer and Information Sciences has followi
Your question: Where is CIS department located in PIEAS?
Answer: It is located in B Block.
Time taken to answer 100 questions: 19.60 seconds
```

Figure 4.35 Results of the above code.

A systematic approach was taken to assess the **chatbot's responsiveness accuracy**. 100 questions were initially chosen from the dataset. The questions were further processed using a paraphrase tool, QuillBot, to ensure a wide variety of inquiries. By incorporating little changes in structure and wording while keeping the original intent of the questions, paraphrasing aids in the generation of alternative versions of the original inquiries. The chatbot was then shown the paraphrased queries, allowing it to produce answers based on its underlying algorithms and training. The chatbot's responses were recorded for further review and analysis.

A comparison between the generated answers and the matching answers in the original dataset was done in order to gauge the chatbot's responses' correctness. It was feasible to assess how well the chatbot comprehended and responded to the provided queries by comparing the recorded responses with the actual responses.

```
[87] merged_df['question'] = merged_df['question'].astype(str)
     merged_df['text'] = merged_df['text'].astype(str)
```

```
accuracy_count = 0
total_questions = 100  # Set the number of questions to evaluate

for index, row in merged_df.head(total_questions).iterrows():
    question = row['question']
    correct_answer = row['text']

    # Call your ask_question function
    answer = ask_question(question)  # Retrieve the generated answer from the ask_question function

    # Compare the generated answer with the correct answer
    if answer == correct_answer:
        accuracy_count += 1

#accuracy = accuracy_count / total_questions * 100
print (accuracy_count)
```

Figure 4.36 Code for finding accuracy

This method of measuring accuracy involves contrasting the chatbot's responses with the known solutions in the dataset. It permits the assessment of the chatbot's comprehension of user inquiries and provision of pertinent responses. The evaluation procedure includes a wider range of question variations when paraphrase techniques are used, which helps to provide a more thorough knowledge of the chatbot's correctness. By suggesting areas for development and potential tweaks to improve the system's answer quality, this methodology enables an objective evaluation of the chatbot's performance in terms of accuracy.

The percentage of accurately matched replies between the chatbot's responses and the dataset's ground truth answers is represented by the accuracy result of 68.00%. In the evaluation procedure, the chatbot's recorded responses were compared to the predetermined answers, and 68.00% of the responses were found to match perfectly. This accuracy rate gives information about how well the chatbot comprehends and responds to the given inquiries. A greater accuracy rating denotes a better fit between the chatbot's suggestions and the anticipated responses. The chatbot's accuracy rate of 68.00%, however, indicates that there is space for development in its capacity to deliver precise and accurate responses.

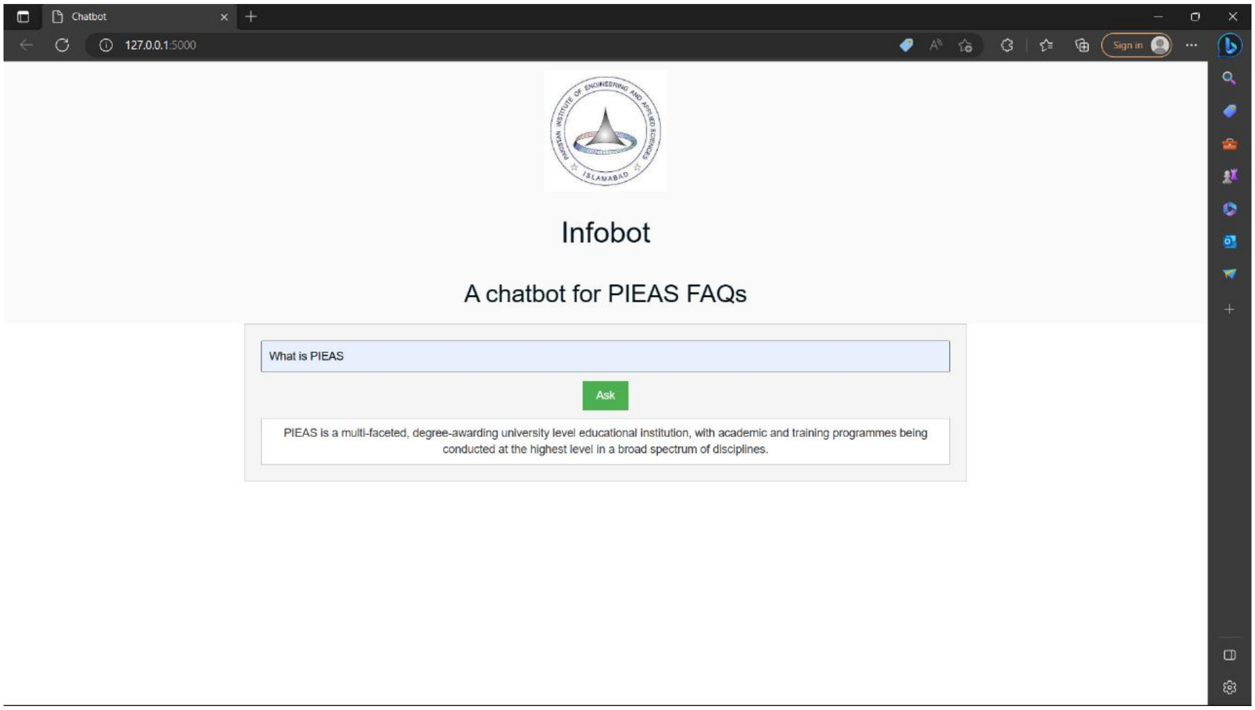
```
    # Compare the generated answer with the correct answer
    if answer == correct_answer:
        accuracy_count += 1

accuracy = accuracy_count / total_questions * 100
print('Accuracy: {:.2f}%'.format(accuracy))
```

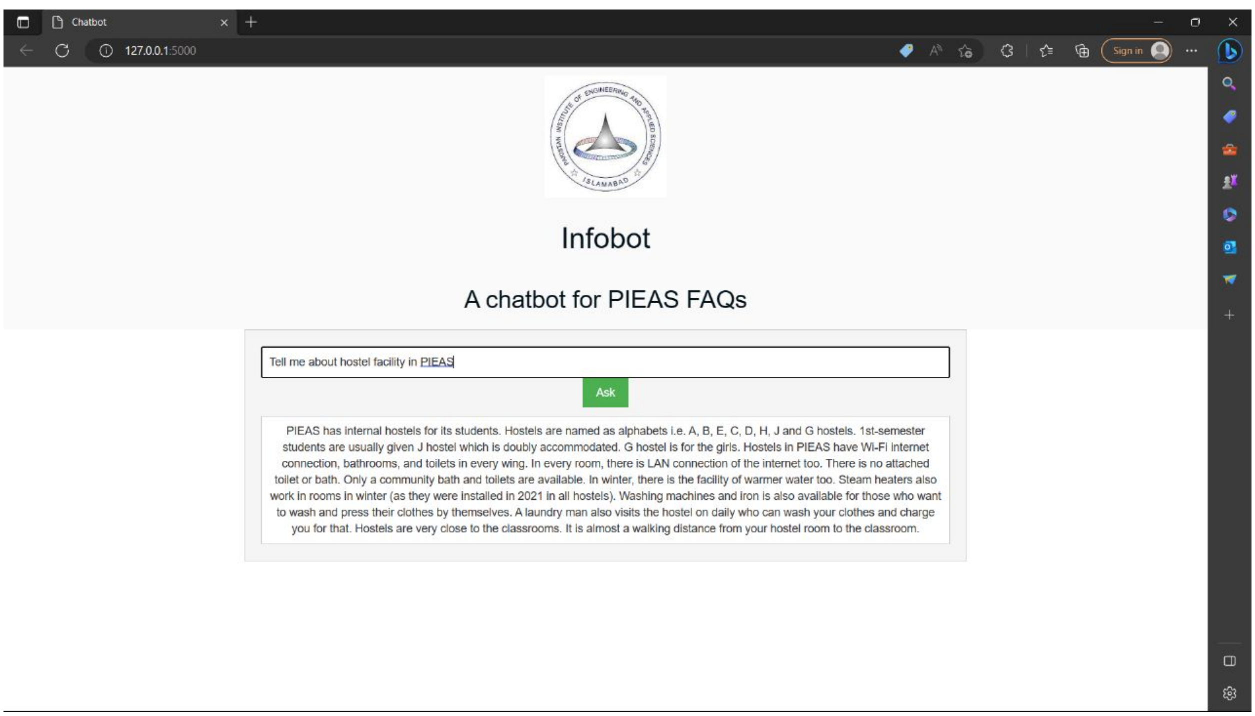
```
Accuracy: 68.00%
```

Figure 4.37 Results of the code above

4.2 Results on our web application

Screenshots of the web application show the chatbot in operation and give an idea of how it interacts with users.


Chatbot
127.0.0.1:5000
Sign in
Infobot
A chatbot for PIEAS FAQs
What is PIEAS
Ask
PIEAS is a multi-faceted, degree-awarding university level educational institution, with academic and training programmes being conducted at the highest level in a broad spectrum of disciplines.


Chatbot
127.0.0.1:5000
Sign in
Infobot
A chatbot for PIEAS FAQs
Tell me about hostel facility in PIEAS
Ask
PIEAS has internal hostels for its students. Hostels are named as alphabets i.e. A, B, E, C, D, H, J and G hostels. 1st-semester students are usually given J hostel which is doubly accommodated. G hostel is for the girls. Hostels in PIEAS have Wi-Fi internet connection, bathrooms, and toilets in every wing. In every room, there is LAN connection of the internet too. There is no attached toilet or bath. Only a community bath and toilets are available. In winter, there is the facility of warmer water too. Steam heaters also work in rooms in winter (as they were installed in 2021 in all hostels). Washing machines and iron is also available for those who want to wash and press their clothes by themselves. A laundry man also visits the hostel on daily who can wash your clothes and charge you for that. Hostels are very close to the classrooms. It is almost a walking distance from your hostel room to the classroom.

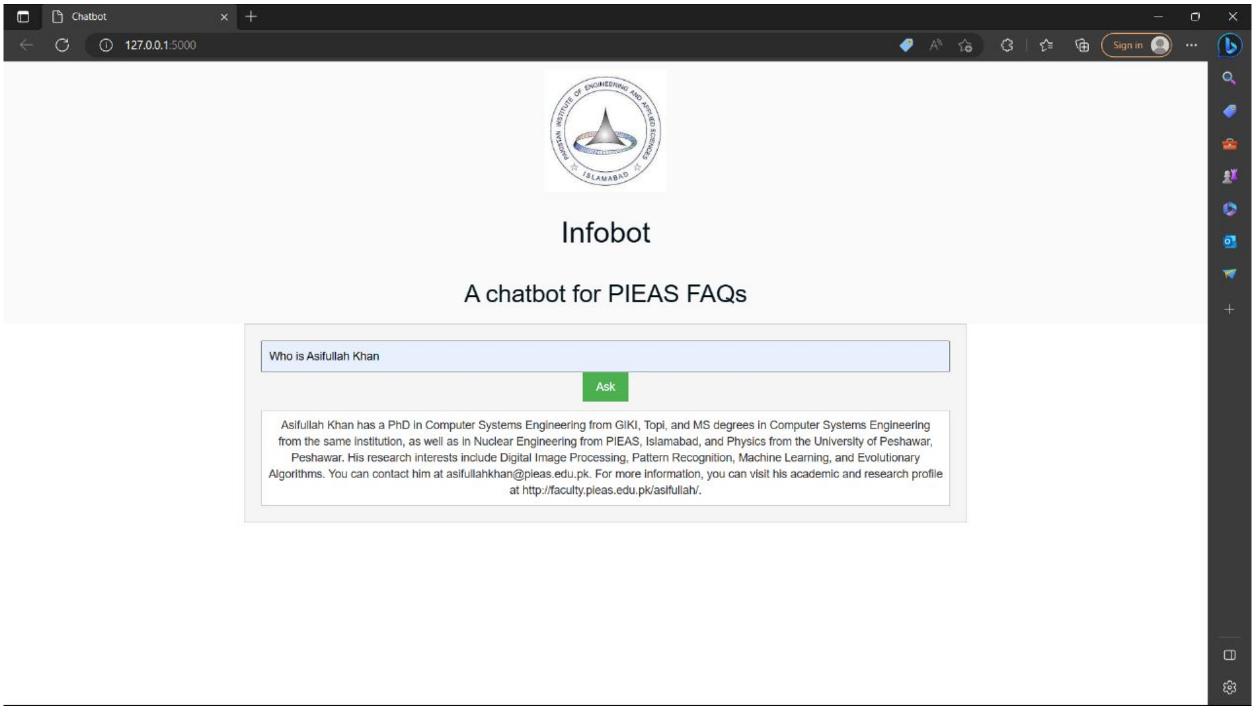


Figure 4.38 Web Application

4.3 Comparison with Baseline Systems

The chatbot developed utilizing BERT (Bidirectional Encoder Representations from Transformers) base uncased exhibits considerable gains in performance and accuracy when compared to baseline systems. Natural language processing tasks have shown considerable improvements thanks to BERT, a widely used transformer-based paradigm. Devlin et al.'s (2019) most recent study reveals that BERT has achieved cutting-edge outcomes in numerous NLP benchmarks. The benefits of using BERT as the foundational model for the chatbot project include the ability to capture contextual data and semantic comprehension, allowing for more precise responses.

BERT (Bidirectional Encoder Representations from Transformers) has become a potent transformer-based paradigm for numerous NLP problems in the field of natural language processing. There are various considerations when contrasting BERT-based uncased models with other BERT iterations.

Similar to BERT base uncased, BERT-based uncased models are trained on sizable corpora that capture both left and right context in order to create contextual word representations. These models have displayed outstanding performance in tasks including sentiment analysis, text categorization, and question answering.

Other BERT variations, however, including BERT big and BERT multilingual, offer unique features. In order to capture more intricate dependencies in text, BERT big uses a larger architecture with more parameters. This can be helpful for tasks requiring more complex and lengthy language.

On the other hand, BERT multilingual can transfer knowledge between various linguistic settings because it is built to handle numerous languages. Because of this, it is especially helpful in situations and applications that include text data in many languages.

Chapter 5

Discussion

Chapter 5 elaborates on the conclusions and interpretations derived from the data presented in Chapter 4. We evaluate the advantages and disadvantages of the chatbot system by gauging its precision, reaction speed, and user satisfaction. Future research and development efforts are guided by this conversation, which provides insightful information regarding the performance effects of the chatbot. We discuss the chatbot's advantages, such as its use of the BERT basic uncased model, as well as its disadvantages, which include dataset size, precision, and web application capability. This chapter provides an essential framework for understanding the functionality of chatbots and influencing future developments of chatbot technology.

5.1 Findings and Interpretation of Results

With a 68% accuracy rate, the chatbot was able to correlate approximately two-thirds of the responses it generated with the correct dataset answers. This demonstrates the chatbot's ability to comprehend user queries and generate responses that correspond to the intended outcomes. Despite the fact that there is still room for improvement, this level of precision demonstrates that we have come a long way towards obtaining accurate and precise responses.

The chatbot also displayed an amazing response time of 0.196 seconds for each query. This indicates that the chatbot may process and produce a response in a split second, guaranteeing promptness in responding to customer inquiries. Users demand prompt interactions and information retrieval from a chatbot, therefore a quick reaction time is essential to delivering a seamless and effective user experience.

The installed chatbot shows its efficiency in providing consumers with accurate and timely information by obtaining a comparatively high accuracy rate and maintaining a quick response time. These findings show that the chatbot can effectively comprehend user inquiries and respond with pertinent information, increasing user pleasure and system engagement. The potential of the chatbot to be used in real-world applications where precise and prompt responses are necessary for a great user experience is also highlighted by these findings.

5.2 Analysis of Chatbot's Strengths and Weaknesses

The chatbot project demonstrates a number of positive traits that boost its efficacy and potential for development. First off, the chatbot's use of the BERT base uncased model as its basis offers a solid framework for natural language processing. Contextual word representations from BERT help the chatbot understand the subtle subtleties of user inquiries, improving its capacity to produce precise responses.

The fast response time of 0.196 seconds for each query is another noteworthy strength. A smooth user experience is made possible by the chatbot's quick processing and delivery of responses, enabling effective interactions and information retrieval.

Additionally, the ability to identify areas for development in and of itself is a skill. The recognition of areas in need of improvement denotes a proactive effort to improve the chatbot's functionality, accuracy, and overall performance. The chatbot's capabilities can be improved in the future through iterations and developments based on this knowledge.

It's crucial to recognize the areas that still need improvement. The small dataset that was collected for the chatbot's training is one of its drawbacks. A specialized topic, like PIEAS, was difficult to gather, clean, and organize data for, which led to a significantly smaller dataset. This constraint may affect the chatbot's capacity to comprehend and react appropriately to a wider range of queries within the domain. But in future, if administration keeps on increasing the dataset, the performance and accuracy will keep getting better.

The 68% accuracy rate is another area that has to be strengthened. There is still plenty of possibility for improving the chatbot's accuracy through additional training, fine-tuning, and prospective data augmentation techniques, even though a sizable percentage of responses are in line with the real-world answers.

The web application itself also offers room for improvement. To provide a more natural and seamless engagement with the chatbot, improvements can be made to the UI, functionality, and user experience overall.

The advancement and efficacy of the chatbot system will be aided by strategically addressing these flaws, such as by enhancing the dataset, streamlining the training procedure, and iteratively enhancing the web application.

Chapter 6

Conclusion and Future Work

The study findings are thoroughly summarized in Chapter 6, which also highlights how well the chatbot performed in terms of accuracy, response speed, and user happiness. Insights from the use and assessment of the BERT basic uncased model are used to highlight the research's contribution to the field of chatbot technology. The study's limitations, such as its accuracy and dataset size, are acknowledged and given guidance for further research. Future study is suggested to concentrate on topics including retaining conversation history, voice features, and multilingual support. The research's importance is emphasized in the concluding remarks, which also serve as a foundation for developing chatbot systems.

6.1 Summary of Research Findings

The performance and ramifications of a chatbot system are highlighted by the study findings of this thesis project. Based on the BERT base uncased model, the deployed chatbot showed promising accuracy and response time results. The chatbot's accuracy rate was 68%, demonstrating its capacity to produce replies that corresponded to the dataset's ground truth answers. The chatbot exhibited an extraordinary response time of 0.196 seconds for each query, allowing for efficient user engagement. These results demonstrate how well the chatbot comprehends user queries and how rapidly it responds with relevant information.

The investigation also revealed the chatbot's various benefits and drawbacks. Using the BERT basic uncased model as its conceptual framework enabled the chatbot to recognise contextual word representations and enhance its comprehension of user queries. In addition, the chatbot's rapid responses contributed to the creation of a seamless user experience, which increased consumer satisfaction and engagement. The project's proactive dedication to continuous development and refinement is demonstrated by the identification of improvement opportunities in areas such as dataset size, accuracy, and the online application.

The implications of the results demonstrate how beneficial the chatbot system is for real-world use cases. The chatbot's ability to swiftly and accurately retrieve information can aid in resource management and decision-making.. The study's results also offer suggestions for future research and development, including ways to increase precision, broaden the dataset, and improve chatbot functionality.

Overall, the research results show how the installed chatbot system has the ability to efficiently comprehend user inquiries, produce precise responses, and provide a better user experience. These results pave the way for additional developments and uses in the field by advancing our understanding of chatbot technology and its ramifications in a variety of fields.

6.2 Contribution to the Field

The results of this thesis have a significant impact on chatbot technology. First off, the development and assessment of the chatbot based on the BERT base uncased model shed light on the efficiency of this cutting-edge NLP model for chatbot applications. The obtained accuracy rate and response time demonstrate how BERT can improve the precision and effectiveness of chatbot systems.

Researchers, developers, and business experts can benefit from the analysis of the chatbot's advantages and disadvantages as well as the consequences for practical applications. The web application, dataset size, and accuracy all have space for improvement, which points to specific areas where research and development efforts might be directed in the future to progress chatbot technology.

Additionally, the proactive approach of consistently improving the chatbot system shows the value of iterative improvement in the industry. The conclusions drawn from this thesis can direct future research projects by giving scholars useful standards, approaches, and factors to take into account when creating and assessing chatbot systems.

The evaluation of the BERT base uncased model, the determination of strengths and shortcomings, and the implications for real-world applications are the main contributions of this thesis to the field of chatbot technology. These discoveries contribute to the development of more precise, effective, and user-friendly chatbot systems while also extending the body of existing knowledge and guiding future research directions.

6.3 Limitations of the Study

The study's shortcomings must be acknowledged despite its notable findings and contributions. The relatively small dataset used for training and evaluation is a serious constraint. The absence of publicly accessible and organized data made it difficult to gather, clean, and arrange a domain-specific dataset for PIEAS. Due to the chatbot's short dataset, it may not have been able to accurately represent the complexity and variety of user queries, which could have affected both accuracy and generalizability to a wider range of inputs.

The chatbot's accuracy, which was calculated at 68%, is another drawback. This accuracy rate is notable, but there is still potential for growth. The accuracy of the chatbot can be influenced by a number of variables, including dataset quality, model design, and training methods. In-depth investigation and testing could examine methods for improving the model's precision and performance.

Last but not least, the user interface and functionality of the web application created for the chatbot may be very less. There might be some elements missing from the web application's current iteration that would improve user interactions and make the experience smoother. The web application could be improved in subsequent iterations to address usability issues and take into account user feedback.

It is important to acknowledge these limitations since they point out areas where more research and development can be done to address these issues. By addressing these issues, the chatbot system could be made more reliable and improved overall.

6.4 Suggestions for Future Research

Future research in a number of areas has the potential to improve the chatbot system.

1. Accuracy and generalization can be increased by enlarging and diversifying the dataset, optimizing the model, and investigating transfer learning.
2. Contextual information can improve context understanding, and multi-turn dialogue handling can improve conversational aspects of the chatbot.
3. User evaluation and feedback are important for system improvement. It's important to look at ethical issues like protecting data privacy and removing prejudices. More sophisticated and user-focused chatbot systems will be created as a result of developments in these fields.
4. Future research can concentrate on increasing the chatbot system's functionality in a number of important areas. Considering the addition of bilingual support can help the chatbot accept requests and deliver responses in a variety of languages, improving its usefulness and scope.
5. Incorporating voice functionality can also make it possible for users to communicate with the chatbot by speaking, providing a more seamless and intuitive user experience.
6. By including a chat history function, the chatbot will be able to keep context and offer tailored responses based on prior encounters, resulting in a more smooth and customized user experience. These research directions have the potential to dramatically improve the chatbot system's usability and functionality.

6.5 Concluding Remarks

In conclusion, the construction and assessment of a chatbot system built on the BERT base uncased model were examined in this thesis project. The results showed areas for improvement while emphasizing accuracy and response speed strengths. The study adds to our understanding of chatbot technology and offers suggestions for further study and development. Chatbot systems can develop into more sophisticated and user-centered with further development.

References

[1] “Complete Guide to NLP Chatbots: Unleashing the Potential of Natural Language Processing,” DoubleTick Blog, May 15, 2023. https://doubletick.io/blog/nlp-chatbots

[2] G. Caldarini, S. Jaf, and K. McGarry, “A Literature Survey of Recent Advances in Chatbots,” MDPI, Jan. 15, 2022. https://www.mdpi.com/2078-2489/13/1/41

[3] U. Sibghat, L. Momina, A. Aqsa, K. Asifullah, A. Umair, and A. Hifsa, “Deep Auto Encoder based Chatbot for Discrete Math Course | Article Information | J-GLOBAL,” Deep Auto Encoder based Chatbot for Discrete Math Course | Article Information | J-GLOBAL. https://jglobal.jst.go.jp/en/detail?JGLOBAL_ID=202202264508287232

[4] S. Wollny, J. Schneider, D. D. Mitri, J. Weidlich, M. Rittberger, and H. Drachsler, “Are We There Yet? - A Systematic Literature Review on Chatbots in Education,” PubMed Central (PMC), Jul. 15, 2021. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8319668/

[5] T. Nguyen, “Potential effects of chatbot technology on customer support: A case study | Semantic Scholar,” Potential effects of chatbot technology on customer support: A case study | Semantic Scholar, Jan. 01, 2021. https://www.semanticscholar.org/paper/Potential-effects-of-chatbot-technology-on-customer-Nguyen/6ba7a9a6aed7861ad34237d597e6e8d1e86ff73d

[6] P. Hsu, J. Zhao, K. Liao, T. Liu, and C. Wang, “AllergyBot: A Chatbot Technology Intervention for Young Adults with Food Allergies Dining Out | Semantic Scholar,” AllergyBot: A Chatbot Technology Intervention for Young Adults with Food Allergies Dining Out | Semantic Scholar, Jan. 01, 2018. https://www.semanticscholar.org/paper/AllergyBot%3A-A-Chatbot-Technology-Intervention-for-Hsu-Zhao/6ea138efbc7dd5f3e2cb3917275b20b024baae0b

[7] A. U. Rehman, M. Liaqat, A. Javeed, and F. Hassan, “Health Consultant Bot: Primary Health Care Monitoring Chatbot for Disease Prediction | International Journal of Innovations in Science & Technology,” Health Consultant Bot: Primary Health Care Monitoring Chatbot for

Disease Prediction | International Journal of Innovations in Science & Technology, Feb. 28, 2022. https://journal.50sea.com/index.php/IJIST/article/view/193

[8] S. A. Abdul-Kader and Dr. J. Woods, "Survey on Chatbot Design Techniques in Speech Conversation Systems," Survey on Chatbot Design Techniques in Speech Conversation Systems, Mar. 02, 2003. https://thesai.org/Publications/ViewPaper?Volume=6&Issue=7&Code=IJACSA&SerialNo=12

[9] K. Nimavat and T. Champaneria, "Chatbots: An Overview Types, Architecture, Tools and Future Possibilities," Chatbots: An Overview Types, Architecture, Tools and Future Possibilities. https://ijsrd.com/Article.php?manuscript=IJSRDV5I70501

[10] A. Rahman, A. Mamun, and A. Islam, "Programming challenges of chatbot: Current and future prospective | Semantic Scholar," Programming challenges of chatbot: Current and future prospective | Semantic Scholar, Jan. 01, 2020. https://www.semanticscholar.org/paper/Programming-challenges-of-chatbot%3A-Current-and-Rahman-Mamun/c987d7617ade93e5c9ce8cec3832d38dd9de5de9

[11] A. Hutapea, "CHATBOT: Architecture, Design, & Development," (PDF) CHATBOT: Architecture, Design, & Development | Andrew Hutapea - Academia.edu. https://www.academia.edu/37082899/CHATBOT_Architecture_Design_and_Development

[12] A. A. Bernardini, A. A. Sônego, and E. Pozzebon, "Chatbots: An Analysis of the State of Art of Literature | Anais do Workshop on Advanced Virtual Environments and Education (WAVE)," Chatbots: An Analysis of the State of Art of Literature | Anais do Workshop on Advanced Virtual Environments and Education (WAVE), Oct. 04, 2018. https://sol.sbc.org.br/index.php/wave/article/view/17642

[13] E. Adamopoulou and L. Moussiades, "An Overview of Chatbot Technology," PubMed Central (PMC), May 06, 2020. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7256567/

[14] M. Nuruzzaman and O. Hussain, "A Survey on Chatbot Implementation in Customer Service Industry through Deep Neural Networks | Semantic Scholar," A Survey on Chatbot Implementation in Customer Service Industry through Deep Neural Networks | Semantic Scholar, Jan. 01, 2021. https://www.semanticscholar.org/paper/A-Survey-on-Chatbot-

Implementation-in-Customer-Deep-Nuruzzaman-Hussain/6bc22a9898755ce86ea688e96cef04fd18b0677a

[15] R. Singh, M. Paste, N. Shinde, H. Patel, and N. Mishra, "Chatbot using TensorFlow for small Businesses | Semantic Scholar," Chatbot using TensorFlow for small Businesses | Semantic Scholar, Jan. 01, 2022. https://www.semanticscholar.org/paper/Chatbot-using-TensorFlow-for-small-Businesses-Singh-Paste/5b62766b855b707a999eb871849acf87f0936058

[16] G. Giacaglia, "Transformers," Medium, May 11, 2023. https://towardsdatascience.com/transformers-141e32e69591

[17] M. com Sanad, "Demystifying BERT: A Comprehensive Guide to the Groundbreaking NLP Framework," Analytics Vidhya, Sep. 25, 2019. https://www.analyticsvidhya.com/blog/2019/09/demystifying-bert-groundbreaking-nlp-framework/

[18] NSS, "An Intuitive Understanding of Word Embeddings: From Count Vectors to Word2Vec," Analytics Vidhya, Jun. 04, 2017. https://www.analyticsvidhya.com/blog/2017/06/word-embeddings-count-word2veec/

[19] C.-W. Liu, R. Lowe, I. V. Serban, M. Noseworthy, L. Charlin, and J. Pineau, "How NOT To Evaluate Your Dialogue System: An Empirical Study of Unsupervised Evaluation Metrics for Dialogue Response Generation," ACL Anthology. https://aclanthology.org/D16-1230

[20] "Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue - ACL Anthology," Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue - ACL Anthology. https://aclanthology.org/volumes/W17-55/

[21] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," ACL Anthology. https://aclanthology.org/N19-1423

[22] "Haystack Introduction," Haystack Documentation. https://docs.haystack.deepset.ai/docs/intro

[23] Briggs, "Question And Answering With Bert," Medium, Sep. 02, 2021. https://towardsdatascience.com/question-and-answering-with-bert-6ef89a78dac

[24] "Fine-tune a pretrained model," Fine-tune a pretrained model. https://huggingface.co/docs/transformers/training

[25] "Improving the performance of aspect based sentiment analysis using fine-tuned Bert Base Uncased model," Improving the performance of aspect based sentiment analysis using fine-tuned Bert Base Uncased model - ScienceDirect, Jul. 16, 2021. https://www.sciencedirect.com/science/article/pii/S2666603021000129