Acta Scientific COMPUTER SCIENCES

Volume 6 Issue 7 July 2024

Research Article

Artificial Intelligence in Education: An Automatic Rule-Based Chatbot to Generate Guidance from Lecture Recordings

William Hing[1], Neil Gordon[2]* and Tareq Al Jaber[2]

[1]Artificial Intelligence and Data Science, University of Hull, Kingston Upon Hull, England, HU6 7RX, United Kingdom

[2]School of Computer Science, University of Hull, Kingston Upon Hull, England, HU6 7RX, United Kingdom

***Corresponding Author:** Neil Gordon, School of Computer Science, University of Hull, Kingston Upon Hull, England, HU6 7RX, United Kingdom.

Received: June 11, 2024

Published: June 20, 2024

Abstract

In a new era of educational and research-based chatbots, implementing personalised interactive learning resources is critical in enhancing students' academic experiences [1]. Whilst general purpose chatbots are now available with a range of platforms, there are concerns about the nature of the content, and on the resources required to run the. This research involves an innovative integration of Artificial Intelligence (AI) within an educational context, to develop an advanced bespoke chatbot based on transcriptions and summaries of lecture recordings. This novel tool represents an evolution from passive to active learning resources, likely to improve student learning engagement and comprehension [2].

Leveraging advanced Natural Language Processing (NLP) techniques, this chatbot aims to foster an engaging learning environment by transforming passive lecture content into an interactive, query-answering interface, whilst enabling some measure of how much resource such platforms require.

The chatbot employs several key AI components: Speech recognition for transcriptions, Name Entity Recognition (NER) for topic and keyword extraction, Spacy for processing user queries, Wordnet for synonym generation and language identification and translation to handle non-English queries. Furthermore, the chatbot uses TF-IDF for information retrieval and the T5 transformer model for summarisation which is renowned for its semantic comprehension and ability to produce context-aware responses. This allows the chatbot to provide detailed lecture summaries and define complex terms using API integration.

User engagement is facilitated using PyQt5 to develop a user-friendly graphical interface. The interface offers a variety of features, including adjustable text size, a theme switch feature to transition between light and dark modes, multiple conversation management and conversation deletion options. These features enhance the user's overall experience and allow personalised chatbot interaction.

While there are challenges related to the accuracy of transcriptions, topic modelling reliability and the quality of responses, this project aims to impact the future of personalised, interactive, and educational resources.

Keywords: Artificial Intelligence; ChattBot; Technology Enhanced Learning

Introduction

The background of this paper lies at the intersection of online education, artificial intelligence (AI), and natural language processing (NLP), with a specific focus on chatbot applications in the educational sector. The relationship between artificial intelligence (AI) and education has been a dynamic and critical research area in recent years. AI has introduced promising opportunities for innovative pedagogical techniques that encourage more adaptive, engaging, and personalised learning experiences [3].

In recent years, online education has experienced a flux in popularity and adoption driven by technological advances and the need for more flexible learning opportunities [4]. This shift has sparked interest in using AI and NLP techniques to develop educational tools that can provide more personalised learning experiences, support student engagement, and reduce the workload of educators [5].

Educational chatbots have emerged as a promising application of AI and NLP technologies in online education. These chatbots are designed to interact with learners, provide information or guidance, answer questions, and offer personalised support [6]. However, the effectiveness of these educational chatbots depends highly on the underlying AI and NLP technologies that power them, and these determine their ability to understand and respond to user queries accurately and contextually [7].

Traditional educational resources, such as lecture slides, recordings, and textbooks, are rich in information but often need more interactivity. This reduces the engagement of students and hinders effective learning. The passive consumption of educational content may lead to reduced critical thinking, active learning and comprehension of complex concepts. Therefore, the change from passive to more interactive learning resources is essential to advance the quality of learning and teaching [1].

This project introduces an AI-powered chatbot capable of translating long lecture records into interactive chatbot responses through summaries and definitions of terms, facilitating active learning by changing how students engage with lecture content. Active learning is a student-centric approach that involves activities promoting analysis, synthesis and evaluation and has been proven to improve academic performance and enhance student engagement [2]. When looking at blended and online learning, the transition from passive to active learning is becoming increasingly important [8]. The proposed AI-powered chatbot provides active learning experiences to its users by enabling conversational interaction with lecture content.

The developed chatbot leverages two powerful technologies to enable high-quality conversational experiences: the Text-To-Text Transfer Transformer (T5) and Google's Speech Recognition service. The T5 model is a state-of-the-art model in the field of NLP; it is employed to comprehend and summarise lecture content [9]. The Google Speech Recognition Service plays an essential role in transforming the audio data for each lecture into text. This service accurately transcribes the wav converted mp4 files into written text, allowing the T5 model to interact with the generated text and draw relevant information to form summaries.

The chatbot is designed to handle the user's natural language queries and provide comprehensive explanations for terms and summaries for entire lectures. This functionality is facilitated through a blend of pre-defined intents and user prompts, ensuring a contextually relevant and logical conversation. This design enhances student comprehension and helps foster a deeper understanding of the lecture content, providing a tool that will help to facilitate active learning.

A simple user interface (UI) emphasises simplicity and user-centric design, highlighted by elements such as a typing indicator and distinct colours to differentiate between the user and chatbot responses. The UI follows user-friendly design and interactive learning principles, creating an easy-to-navigate and functional platform. Features such as a typing indicator, conversation history, the ability to enlarge or minimise text and distinct colours to differentiate between the chatbot and user responses contribute to a more interactive and engaging learning environment. To replicate real-time integration, the interface integrates a delay to reinforce the impression of a genuine conversation with the chatbot.

Preliminary evaluations suggest that the AI chatbot has the potential to enhance student comprehension and promote self-paced learning. Nonetheless, challenges remain in areas such as improving the chatbots understanding and response accuracy, adding a voice interface, question and answer model development, improving summary generation and transcription code (using paid services instead of free) and adding tailored responses to different learning styles.

In summary, this AI-powered chatbot makes a significant step towards creating a more interactive, engaging and effective learning environment. This project shows the potential of AI in transforming traditional educational resources into interactive learning tools, thus helping to bridge the gap between technology and pedagogy.

Methodology

This paper outlines a practical approach to building bespoke chatbots to support education. The first step was collecting essential source data - audio files derived from lecture recordings. The video lecture recordings were collected from lecturers at the university. The raw video data was then converted into a usable format for subsequent stages using Python's PyDub library. The library performs several operations on each .mp4 file found in the directory, including loading the file, normalising the audio volume, and then converting the audio to a .wav format with specified parameters. See Figure 1 for the main steps in preparing the source data.

Data collection and preparation

Normalisation involved setting the audio volume to -20dBFS (decibels relative to full scale); this was done to ensure consistent audio volume across all the pre-processed audio files. The outputted .wav files are in PCM 16-bit format with mono audio and a sample rate of 16000 Hz to make the data suitable for further data processing, such as transcription.

The code also contains thorough exception handling to provide error messages for issues such as file not found and denied permis-

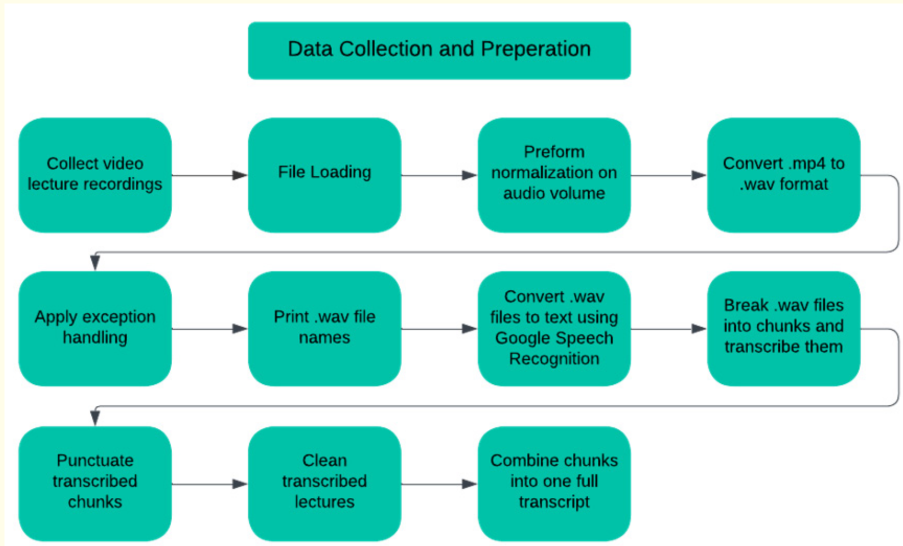

Figure 1: Flow chart outlining the data collection and preparation steps.

sions. The code then confirms the successful conversion and normalisation of each file in the output directory, printing the names of the .wav files. This ensures that the data is completed before the data is passed on to subsequent stages of the project.

The raw .wav data is then converted into a text format using Google Speech Recognition API. After initialising the API, the audio files were loaded for processing. The .wav format lectures are broken up into chunks and then transcribed. The transcribed lectures are then subjected to a cleaning operation which removes irrelevant elements, such as unnecessary spaces and stop words. This operation helps to enhance the efficiency of subsequent stages of the project. By removing these elements, the model can focus on meaningful data, which helps reduce computational complexity and improve the model's performance.

The individual chunks are combined into one full transcript when cleaning is completed, and a punctuation model is used to restore punctuation to the transcribed lectures. This was a crucial step for further analysis as it transformed the raw audio data into a comprehensible format for the summarization model.

Fine-Tuning the T5 model and text summarization pipeline

Given the computational intensity of the task of fine-tuning the T5 model and optimising the text summarisation pipeline, the University of Hull's high-performance computer was used. Figure 2 above outlines the steps to fine-tune the model and create the text summarisation pipeline. Fine-tuning the T5 model and constructing a text summarisation pipeline begins with defining a data generator function that reads data from the given path and yields relevant information for the training process (only papers filtered under the computer science (cs) category are filtered from the JSON object).

Following the data generator function, the ArxivDataset class was set up, extending the Pytorch Dataset class. This class represents the Arxiv papers dataset, employing the defined data generator filter and loading data from the JSON file. Each dataset comprises tokenised versions of a paper abstract (source) and title (target).

Next, the T5 tokenizer (specifically the T5-large variant) is initialised from its pre-trained state; this tokenizer transforms the text data into a format the T5 model can efficiently process. Subsequently, the ArxivDataset is initialised with the path to the data, the T5 tokenizer, and the maximum lengths for the source and target texts. A PyTorch Dataloader is then created, allowing batch-wise iteration over the dataset, thus facilitating the training process.

The T5-large model is downloaded and transferred to the GPU within the established data-loading framework if available. Hyperparameters for the number of epochs, learning rate are identified here. For full details of the hyperparameters contact the authors. An Adam optimiser is then set up for training the model's parameters, with a learning rate of 0.0001. The model is then fine-tuned over three epochs with 160 iterations for each epoch, adjusting the model's weight in response to the calculated loss within each batch. The final loss recorded was 0.00013, indicating that the training was successful, and the model worked well with the data provided.

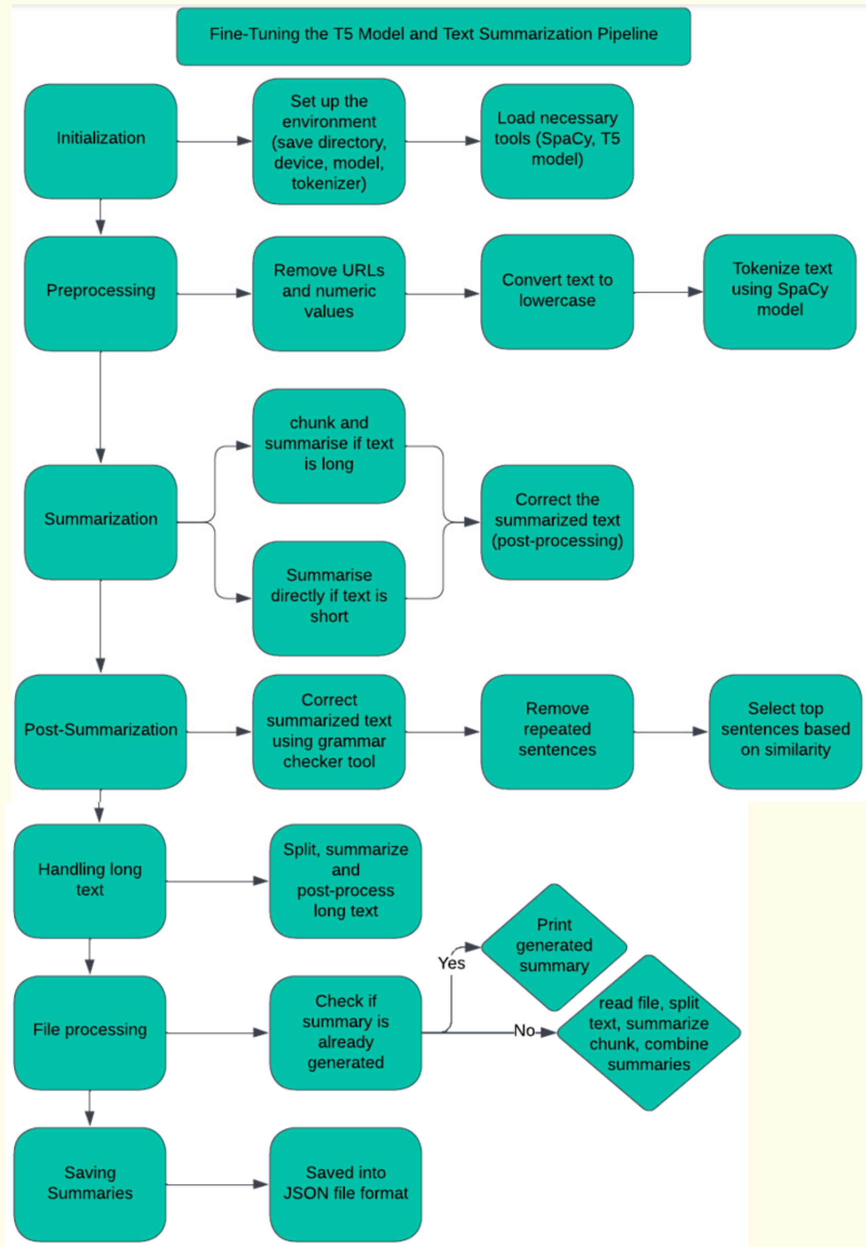

Figure 2: Flow chart outlining Fine-Tuning the T5 model and text summarisation pipeline steps.

Upon fine-tuning the model, the TextSummerizer class is defined. This class incorporates the trained T5 model for generating summaries for the given lecture transcripts. It also comprises text pre-processing and post-processing methods and removing redundancies in the generated summaries. The TextSummerizer class also accommodates short and long texts during the summarisation process; for longer texts, summaries are generated in chunks and are later concatenated. The summary is then post-processed and cleaned of any redundancies. Furthermore, functions are incorporated to save the generated summaries to a file and summarise a batch of texts. This provides an efficient way to handle large volumes of data. The final step in the pipeline is the 'process_file' method. This function takes care of reading a file, splitting it into chunks, generating summaries for each chunk and then saving them to an output file. With this, the fine-tuning and text summarisation pipeline for the T5 model is complete.

Chatbot development

The chatbot class is a central component of the chatbot; Figure 3 above outlines the steps taken during this phase. It begins with defining the necessary methods and attributes for operation. These methods cover a broad spectrum of tasks, including initial setup, user interaction, text processing, information retrieval and response generation.

In the initial setup phase, handled by the init method, all required attributes required by the chatbot are initialised; these attributes include a variety of predefined phrases, paths for data and model loading, instances of required models and other required tools and intent definitions.

Keyword extraction is performed in the 'process_question' method, where important keywords and entities are identified from a user's query. These keywords are further processed by the 'get_synonym' method, enhancing the understanding of user inputs by considering all keyword's synonyms. The method also incorporates caching to optimise the retrieval of synonyms for repeated words.

Data and model handling

Lecture data and summaries form the core information source for the chatbot and are loaded into 'lecture_info' and 'lecture_summaries' dictionaries through the 'load_lecture_data' function. This process paves the way for the chatbot to provide summaries and respond to user queries effectively.

User interaction

The 'chat_loop' function oversees the interaction between the user and the chatbot, mapping user intent to corresponding response-generating functions. These functions, 'respond_to_greeting' and 'respond_to_goodbye', further personalise user interactions with suitable responses to greetings and farewells. Further NLP techniques are added to the 'get_entities' function to extract named entities from the user's input.

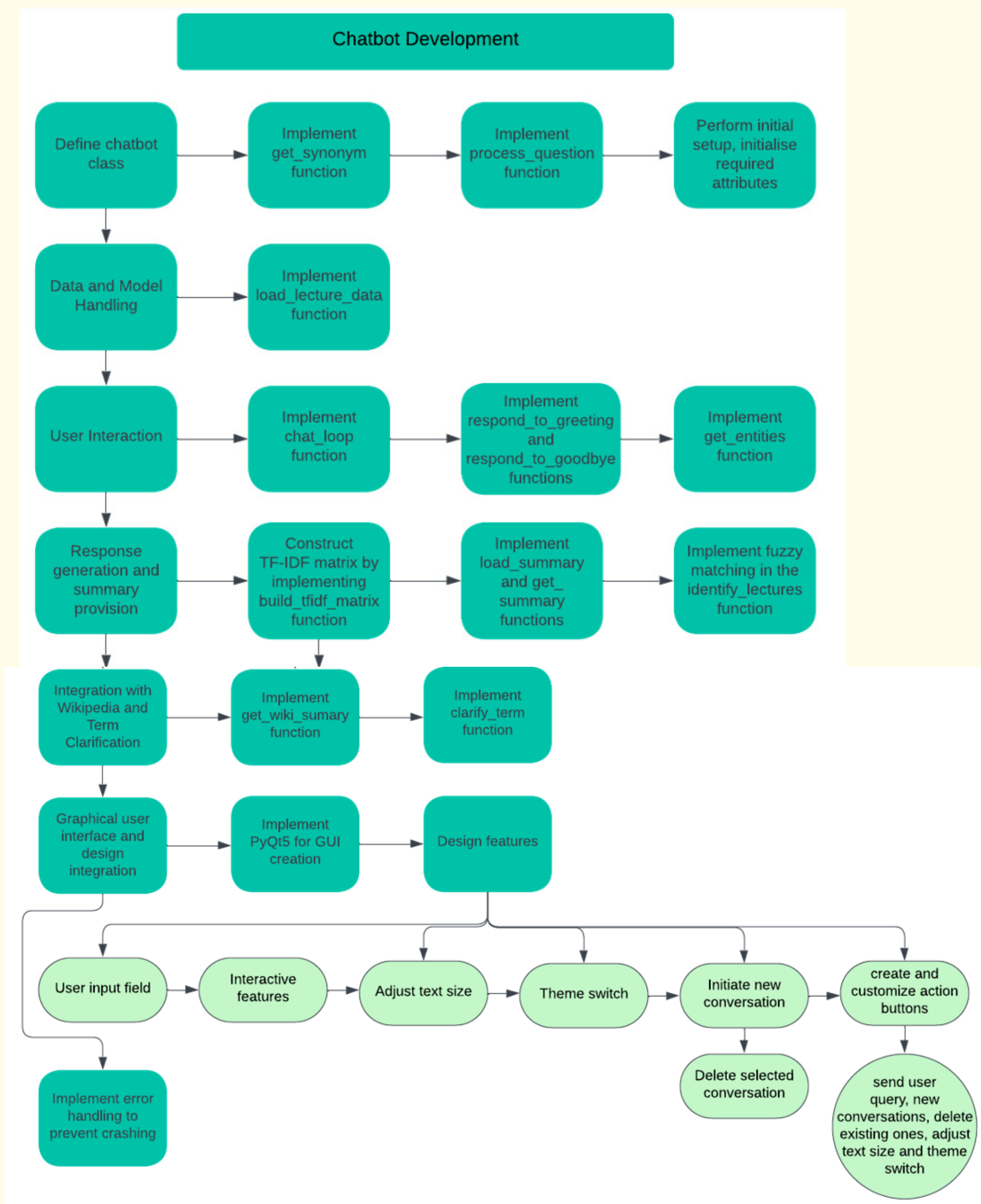

Figure 3: Flow chart outlining the steps undertaken during the chatbot development stage.

Response generation and summary provision

The chatbot uses the 'build_tfidf_matrix' function to construct a TF-IDF matrix to find similar lectures, handled by 'find_similar_lecture' based on user queries. For lecture summarisations, the 'load_summary' and 'get_summary' functions retrieve or create lecture summaries when required. The 'identify_lectures' function uses fuzzy matching to find the most similar lecturer to a user's query, with 'respond_to_summary_request' handling summary requests.

Integration with Wikipedia and term clarification

The chatbot is integrated with Wikipedia API using the 'get_wiki_sumary' to fetch specific terms. The 'clarify_term' function helps define terms from the user's statements, fetch Wikipedia summaries and correct spelling when required.

Graphical user interface (GUI) design and integration

The project's final phase focuses on creating a user-friendly GUI for seamless interaction with the chatbot. PyQt5, a powerful library for creating desktop applications, is used to construct the interface. The GUI comprises a main window displaying the conversation history and input fields for user queries. Additionally, the interface includes various features to facilitate interaction, such as options to initiate new conversations, delete existing ones, and adjust the text size for better readability. It also incorporates a theme switch button, enabling users to toggle between light and dark modes based on preference, enhancing the user's experience [10].

User testing was conducted to ensure the interface was intuitive and easy to use. The application design also takes into consideration aesthetic elements and response visibility. A typing indicator is implemented to inform users when the model is processing a response, making the interaction more engaging and mimicking real-life conversation dynamics.

Results

The project's primary focus was the development of an AI-based chatbot capable of processing and responding to user queries efficiently. The project results regarding the chatbot's functionality and ability to generate accurate query responses are presented.

The chatbot was evaluated by testing its conversational abilities. The user interface designed with PyQt5 made this process straightforward. The GUI includes various features designed to improve the user's experience and usability.

The 'New Conversation' button allows users to initiate a fresh dialogue with the chatbot, resetting the conversation history to ensure a clean slate for the new interaction. This feature helps segregate discussion topics or state a new query thread.

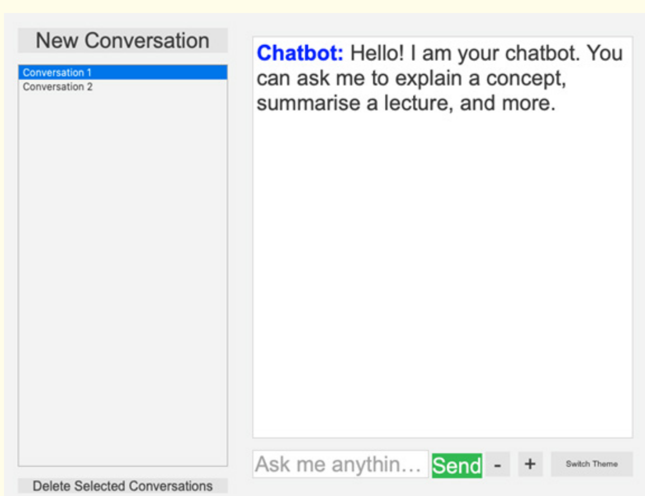

Figure 4a: Image showing the enlarged text size feature

The 'Delete Selected Conversation' button allows users to remove a selected conversation. This feature can be handy when users want to declutter the conversation pane or erase irrelevant or complete queries.

Additionally, the GUI includes a plus and minus button next to the send button, which allows users to adjust the text size to their preference. This ensures that the chatbot interface is accessible to a broader range of users, those who are visually impaired or those who prefer smaller or larger text sizes (See figure 4A).

Finally, the interface also includes a 'Switch Theme' button. This allows users to switch between light and dark themes, catering to different visual preferences and reducing eye stains, particularly in low-light conditions or during long periods of use (See Figure 4B).

The most fundamental test assessed the greetings and goodbye functions built. During this test, the chatbot was greeted with 'Hello' and 'Bye' and could accurately interpret these simple intents and respond correctly with the correct greeting or farewell (see Figure 5).

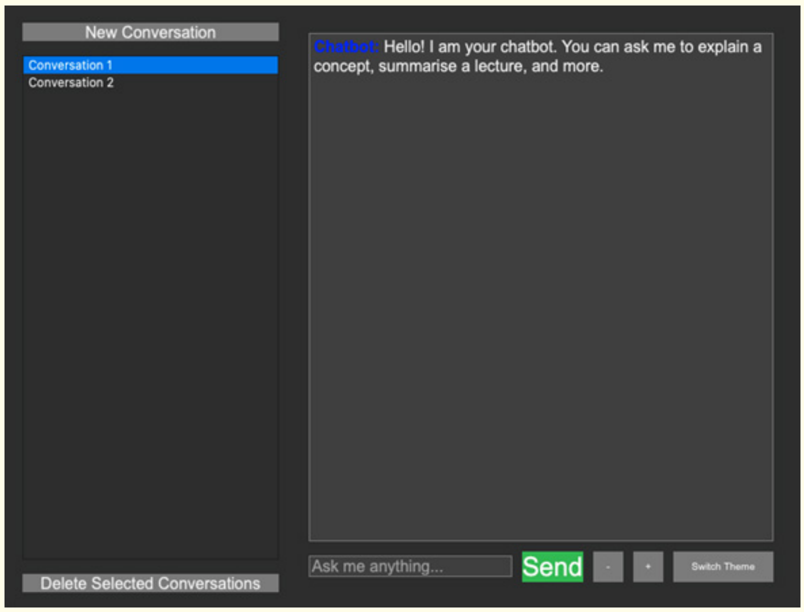

Figure 4b: Image showing the switch team to dark mode feature.

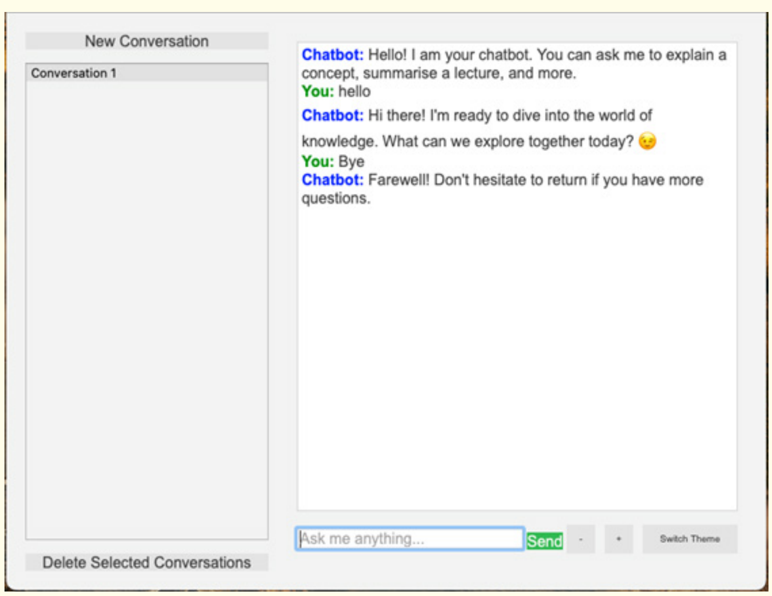

Figure 5: Image showing chatbot greetings and goodbye responses.

The chatbot's capabilities extend beyond basic greetings and goodbyes. When asked to 'list lectures', the chatbot displays a comprehensive list of available summarised lectures, correctly sourcing the information from the 'lecture_info' dictionary. This proves its ability to parse and correctly respond to specific intents, demonstrating effective NLP technique integration (see Figure 6).

The chatbot's ability to summarise the above lectures was another essential aspect tested. When a user requests a 'summarise 'lecture name' lecture', the chatbot accurately pulls the summary from the 'lecture_summaries' dictionary or creates a new summary using the T5 summarise if the pre-existing summary is unavailable

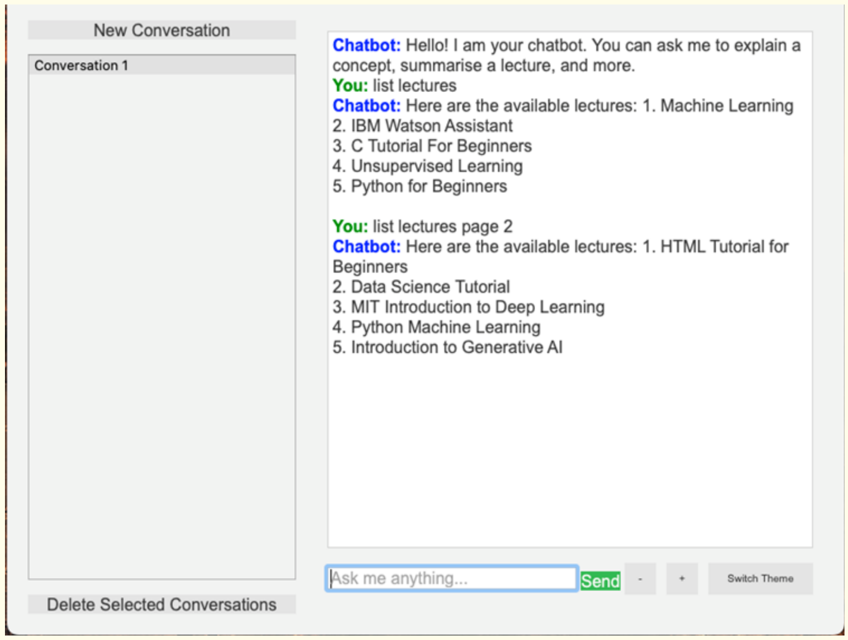

Figure 6: Image showing the initialisation of the chatbot's list_lecture function.

(see Figure 7). This allows quick response for available lectures with a slight wait for newly generated lecture summaries (dependent on computational resources).

The 'clarify_term' function was also evaluated. This function provides the correct definition and explanation of a term or topic from Wikipedia when prompted by the user (see Figure 8). If the term is misspelt, the function corrects the spelling, further expanding its usefulness as an educational tool.

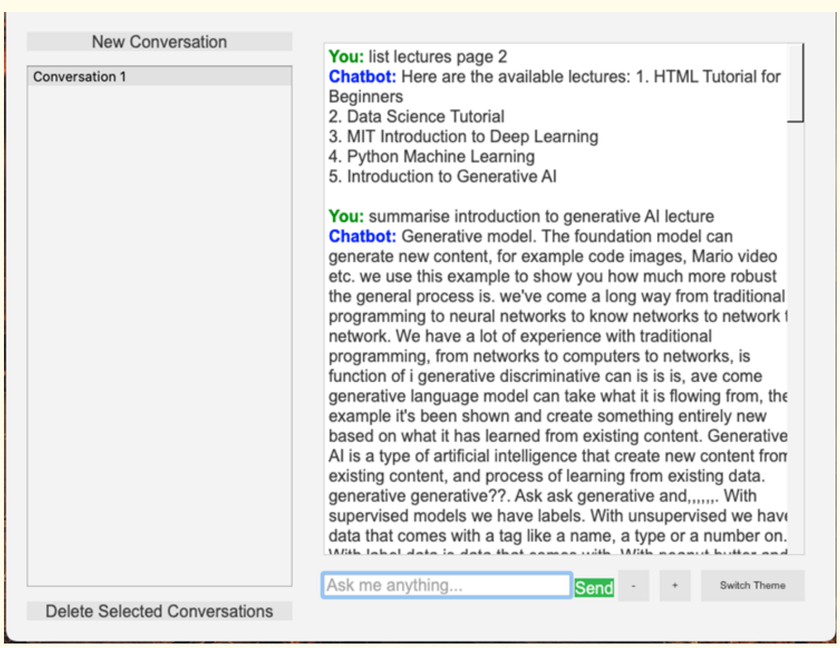

Figure 7: Image showing the chatbot retrieving and printing a summary for a requested lecture.

Discussion

This research highlights the benefits of using AI and NLP technologies in education. By creating a chatbot that can transform re-

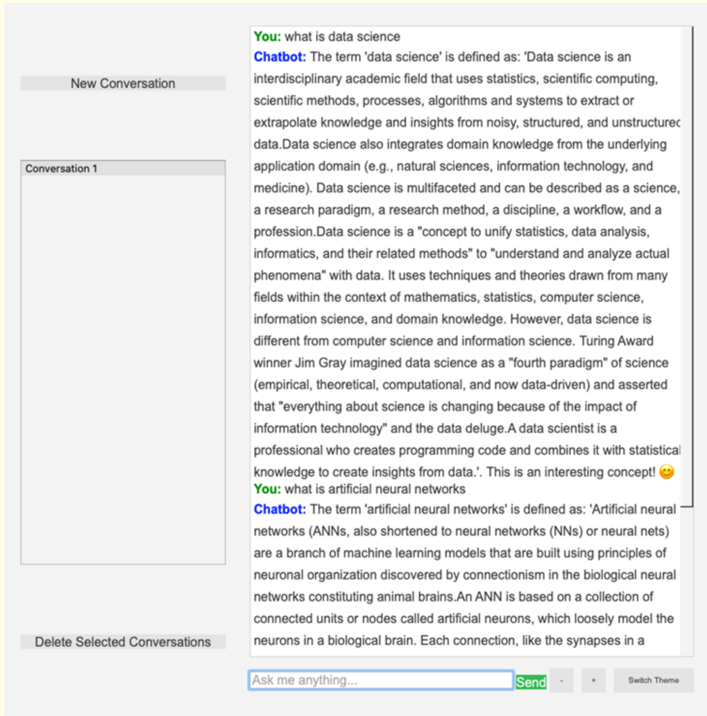

Figure 8: Image showing the chatbot response to defining a term.

corded lectures into interactive conversations, learner engagement can be improved, and information retention may be increased [11]. This innovation represents a move towards more personalised and inclusive learning environments by successfully converting academic content into a dialogue-based format.

This chatbot follows pre-defined rules to ensure consistency in its responses across various user interactions. Unlike more intricate AI models, rule-based systems have explicit instructions enabling the chatbot to answer user inputs accurately. Although these systems may struggle with understanding context, integrating advanced NLP libraries such as SpaCy for language processing and the T5 model for text summarisation can enhance the chatbot's ability to comprehend user inquiries, leading to satisfactory interactions [12].

The chatbot also incorporates WordNet, a lexical database that extends the keyword search in responses. WordNet is used to find synonyms in user responses, enhancing the chatbot's understanding of user inputs, enabling it to identify keywords and better explain requested terms.

However, even using these technologies, the chatbot can improve drastically. While rule-based chatbots are proficient at handling routine tasks, their capacity to comprehend abstract concepts and respond to intricate questions remains challenging. Future improvements for the chatbot include integrating more advanced transcription, summarisation and NLP models.

During the development phase, the investigation and testing of various models and services were pursued. Renowned services such as IBM Watson and Microsoft Azure were assessed for their potential in lecture transcription and chatbot development, as well as BERT and BART for summarisation tasks; these models and platforms are highly recognised for their superior performance in NLP tasks [13,14].

In the case of IBM Watson, testing was conducted to assess the accuracy and quality of produced transcriptions, with the tool initially being chosen to be the primary model option. However, IBM Watsons API imposes certain restrictions, such as character limits for free users. The expansive lecture dataset and the iterative testing approach employed to improve the quality of the transcriptions led to a swift depletion of the monthly character allowance, rendering the service impractical for sustained use in the project. Likewise, an attempt was made to use IBM Assistant for chatbot development. However, complications arose due to recent changes in IBM's service policies and the deprivation of the free service. This saw the service switch to a paid model, presenting additional financial and integration challenges that hindered its adoption [15]. This limitation means that there was only a restricted consideration of accuracy and loss data. The key focus for the paper was on the capability of these tools to enable a bespoke chatbot from existing resources.

Similarly, Microsoft Azure was also explored for its transcription services. However, like IBM Watson, Azure also imposes limits on free-tier users, which became restrictive given the size and diversity of the lecture dataset and the iterative, testing-focused nature of the project.

Various free services and libraries were also tested, such as Python-based libraries like SpeechREcognition and pydub for transcription production, with Google's most reliable and accurate speech-to-text model. For summarization tasks, models like BERT and BART were evaluated, with the T5 model providing the most satisfactory results in the project context [16].

Future improvements for the chatbot can include incorporating adaptive learning to increase the ability of the model to learn from user interactions and improve its accuracy over time [17]. The question and response mechanism of the chatbot could be enhanced to handle more complex queries by integrating more advanced hybrid versions of the rule-based systems or other sophisticated AI models. Other improvements could involve enhancing the quality of transcriptions and summarisations by deploying paid models trained on more comprehensive and diverse datasets, such as IBM Watson for transcriptions and chatbot development.

To further improve the user's experience, future chatbot versions could use a more interactive and user-friendly GUI, developed using HTML or other technologies to create a web-based chatbot. Additional features such as voice recognition, sentiment analysis or multimodal input could also be considered to capture nuances of human communication more effectively [18,19].

Conclusion

In conclusion, this paper has illustrated the feasibility of using AI and NLP technologies to convert lecture content into an engaging and conversational interface. The developed chatbot employs a rule-based system bolstered by advanced NLP libraries, which enable efficient and consistent response generation.

Although the study has shown some successes, challenges still need to be addressed. These include improving the chatbot's ability to understand and respond to complex questions, testing the system in a wider variety of data sets to improve its performance, and utilising more advanced transcription and summarisation models to enhance the data used by the chatbot.

Other significant insights from this project involve the complexities of AI system development, particularly within the constraints of free-tier service usage. The balance between achieving advanced AI functionalities and resource availability posed significant challenges and revealed the need for more accessible and flexible AI tools for academic and research applications.

Regardless of the outlined challenges, this research project underlines the transformative potential of AI in educational practices. Creating an interactive, intelligent tool that can effectively transcribe and summarise lecture recordings represents a stride towards a personalised learning environment. Future work aims to address these challenges while enhancing the scope and versatility of the chatbot, thereby increasing its potential as a potent educational instrument.

Bibliography

1. Bulger M. "Personalised Learning: The Conversations We're Not Having The Promise of Personalized Learning" (2016).
2. Freeman S., *et al*. "Active learning increases student performance in science, engineering, and mathematics". *Proceedings of the National Academy of Sciences* 111.23 (2014): 8410-8415.
3. Luckin R., *et al*. "Intelligence Unleashed An Argument for AI in Education" (2016).
4. Sheetz D and Bonk Curtis J. "The world is open: How web technology is revolutionising education". *The Internet and Higher Education* 12.3-4 (2009): 181.
5. Holmes W., *et al*. "Artificial Intelligence In Education Promises and Implications for Teaching and Learning" (2019).
6. Kumar JA. "Educational chatbots for project-based learning: investigating learning outcomes for a team-based design course". *International Journal of Educational Technology in Higher Education* 18.1 (2021).
7. Munir H., *et al*. "Artificial Intelligence and Machine Learning Approaches in Digital Education: A Systematic Revision". *Information* 13.4 (2022): 203.
8. Hattie J. "Visible learning: A synthesis of over 800 meta-analyses relating to achievement" (2009).
9. Raffel C., *et al*. "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer". *Journal of Machine Learning Research, online.* 21 (2020): 1-67.

10. Summerfield M. "Rapid GUI Programming with Python and Qt: The Definitive Guide to PyQt Programming" (2007).

11. Bates T. "Teaching in a Digital Age: Second Edition" (2019).

12. McTear MF. "Spoken Dialogue Technology". London: Springer London (2004).

13. Devlin J., *et al*. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding" (2018).

14. Lewis M., *et al*. "BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. arXiv (Cornell University)" (2019).

15. IBM. IBM Watson (2023).

16. Raffel C., *et al*. "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer" (2019).

17. Settles B. "Active Learning. Synthesis Lectures on Artificial Intelligence and Machine Learning". Cham: Springer International Publishing (2012).

18. Krippendorff KH. "Content Analysis - 3rd Edition : an Introduction to Its Methodology". Thousand Oaks: Sage Publications, Inc (2013).

19. Li J., *et al*. "Deep Reinforcement Learning for Dialogue Generation". Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (2016).