applied system innovation

Article

Centralized Database Access: Transformer Framework and LLM/Chatbot Integration-Based Hybrid Model

Diana Bratić [1,*], Marko Šapina [1], Denis Jurečić [1] and Jana Žiljak Gršić [2]

[1] Faculty of Graphic Arts, University of Zagreb, Getaldićeva 2, 10000 Zagreb, Croatia; marko.sapina@grf.hr (M.Š.); denis.jurecic@grf.unizg.hr (D.J.)

[2] Department of Informatics & Computing, Zagreb University of Applied Sciences, 10000 Zagreb, Croatia; jana@tvz.hr

* Correspondence: diana.bratic@grf.unizg.hr

Abstract: This paper addresses the challenges associated with the centralized storage of educational materials in the context of a fragmented and disparate database. In response to the increasing demands of modern education, efficient and accessible retrieval of materials for educators and students is essential. This paper presents a hybrid model based on the transformer framework and utilizing an API for an existing large language model (LLM)/chatbot. This integration ensures precise responses drawn from a comprehensive educational materials database. The model architecture uses mathematically defined algorithms for precise functions that enable deep text processing through advanced word embedding methods. This approach improves accuracy in natural language processing and ensures both high efficiency and adaptability. Therefore, this paper not only provides a technical solution to a prevalent problem but also highlights the potential for the continued development and integration of emerging technologies in education. The aim is to create a more efficient, transparent, and accessible educational environment. The importance of this research lies in its ability to streamline material access, benefiting the global scientific community and contributing to the continuous advancement of educational technology.

Keywords: centralized database; educational materials; transformer framework; NLP; API implementation; LLM/chatbot

Citation: Bratić, D.; Šapina, M.; Jurečić, D.; Žiljak Gršić, J. Centralized Database Access: Transformer Framework and LLM/Chatbot Integration-Based Hybrid Model. *Appl. Syst. Innov.* **2024**, *7*, 17. https://doi.org/10.3390/asi7010017

Academic Editor: Christos Douligeris

Received: 20 December 2023
Revised: 25 January 2024
Accepted: 6 February 2024
Published: 15 February 2024

1. Introduction

The problem of an inconsistent process of searching for and finding materials during study serves as the basis for this investigation. The process of collecting materials, which involves different sources of information from different sources, often encounters obstacles because there is no central database for all materials. The success and efficiency of a student's learning can be measured by their ability to access the materials they need quickly and accurately. If a student cannot easily access the material they need, their motivation and perceived quality of education can be affected. Implementing a system of centralized storage and retrieval creates a clear structure that allows quick access to key resources, defines key learning areas, and better guides students to the materials they need. With such a planned approach, students can expect better coordination and organization of materials as well as greater transparency in terms of deadlines, content, and assignments. To achieve optimal organization of materials, it is also necessary to understand the goals and needs of students so that a digital tool and system can be created that supports the learning process quickly and with quality. The main objective of this work is therefore to develop a software support that integrates all teaching materials and ensures that they are easily and quickly accessible to users. The process involves the implementation and customization of an API (application programming interface) for an existing LLM/chatbot (large language model/chatbot) model [1] that aims to provide precise answers to questions based on a comprehensive database of educational materials. Different types of educational materials, such as scripts, presentations, and articles, can provide different information to users. To achieve this, knowledge, the correct application of text processing algorithms, and an understanding of the semantics and context within the materials are essential to create an efficient and user-friendly search system. Defining the functions of the program and integrating them into an advanced LLM/chatbot model and understanding the basic needs of the users are crucial for effective communication between users and the program.

In response to the complex demands of modern learning, many authors have addressed the integration of AI (artificial intelligence) tools into various learning systems. However, this is an area that is rapidly evolving technologically, almost daily. Therefore, only recent work was considered in the literature review, and a few examples are highlighted here. A collection of open educational approaches influenced by both the Open Educational Resources (OER) movement and the digital collaboration practices can assist educators in navigating, and potentially excelling, in an era marked by the rapid advancement of AI [2]. A support system was developed for an education center by using a chatbot that provides quick answers and is accessible at any time via a web platform. Technologies such as the Dialogflow platform were used for the user interface component and the dialog management component while Firebase was used for the database component in the architecture of the chatbot [3].

By examining the impact of ChatGPT (Chat Generative Pre-trained Transformer) and large language model (LLM) services on learning and instruction in higher education and mapping the system's capabilities, 13 AI-based implications for students' learning in higher education have been defined [4]. The use of ChatGPT to improve students' programming skills was also tested. This included tasks such as generating code from problem descriptions, creating pseudocode for algorithms based on text, and debugging code [5]. In another study, the educational and cultural environment of computer use in general and the additional effects of ChatGPT on this, particularly in e-learning, were worked out. Ease of use was examined through system interactions during e-learning sessions, focusing on aspects such as knowledge presentation, clarity of responses, and the appropriateness of the information provided. The study also examined the limitations and errors encountered with ChatGPT and emphasized the need for critical thinking when using ChatGPT [6]. Focusing on AI-driven e-learning methods, a group of authors presented their EDUBOT, a student chatbot powered by AI and a learning management system (LMS). The system provides students with resources for each topic and solves their problems. This product offers features such as a dynamic front end, interactive query search, personalized student-centric assessment, and Super User [7]. The use of generative language models to enhance interactive learning with social learning robots is proposed in a model that combines the technological capabilities of generative language models with the educational tasks of a social robot that acts as a tutor and learning partner. The social robot generates explanations, asks questions, corrects errors, and gives answers based on a pre-trained GPT-3 model to explore the potential of generative language models for interactive learning with social robots at different levels of abstraction [8]. It is interesting to mention a chatbot that processes voice messages in addition to written communication, which is particularly practical for students with disabilities [9].

Additionally, numerous comprehensive overviews of current studies on the implementation of AI and chatbots in education were compiled, with a particular emphasis on the field of e-learning [10–15].

Furthermore, an innovative hybrid model based on the transformer framework and using an API for an existing large language model (LLM)/chatbot is proposed. This model successfully handles the complexity of centralized storage for educational materials. Considering the challenges posed by a fragmented and disparate database, the model provides an elegant solution that integrates precise, mathematically defined functions using sophisticated algorithms to enable in-depth analysis and text processing. The architecture of the model is based on the integration and customization of the API for the existing LLM/chatbot model.

Chatbots can be broadly categorized into two types: rule-based and AI-based. Rule-based chatbots provide users with different options that they can explore. They are most often used for simple tasks such as answering frequently asked questions (FAQs). AI-based chatbots use artificial intelligence (AI), natural language processing (NLP), and machine learning (ML) technologies and algorithms to understand the various keywords that users type while chatting with them. These chatbots are trained over time and learn which responses they should provide according to user queries. A hybrid chatbot is a blend of rule-based and AI-driven chatbots. It follows pre-set commands for fallback scenarios when the AI is unsure of the answer, avoids misinformation by signaling its uncertainty, and uses machine learning for complex queries. As the name suggests, hybrid chatbots are a mixture of simple, rule-based bots and advanced, context-aware bots. They combine the strengths of both types of chatbot technology. They can handle common, repetitive requests to provide immediate assistance, but they can also understand questions that require some judgment. A hybrid chatbot combines the strengths of rule-based and AI-based chatbots, providing a more balanced and versatile tool. In the landscape of AI-driven education models, the chatbot for education system model proves to be an automated framework that utilizes artificial intelligence and machine learning to address user queries swiftly and accurately in the educational domain. While it facilitates real-time access to information for both students and educators, its reliance on pattern recognition and data analysis may pose challenges for precision and adaptability. In contrast, the proposed hybrid model takes a new approach by integrating advanced natural language processing techniques with a comprehensive database of educational materials. This design enables the hybrid model to provide accurate and adaptable answers and demonstrate its ability to handle a variety of user queries and requests. The education support system for student model focuses on an interactive learning experience within e-learning platforms and emphasizes immediate and informative feedback for students. However, its potential problem of scalability becomes apparent as interactions with students need to be handled individually. This is where the hybrid model comes into play, strategically designed to avoid this limitation by efficiently managing user requests locally. Leveraging a transformer framework and the LLM/chatbot API, the hybrid model excels in processing user requests and retrieving relevant information from a user-provided educational material database.

The main goal is to provide accurate responses to user queries based on a comprehensive database of educational materials. Key components of this approach include text processing algorithms that are able to understand the semantics and context of the materials—a crucial prerequisite for creating an efficient and user-friendly search system.

The methodology of the model focuses on the integration of functionalities into the advanced LLM/chatbot model, considering the basic needs of users to enable effective communication between users and the program. The central goal is to provide a centralized source of information that quickly, relatively accurately, and interactively addresses the challenges of a fragmented and disparate database. In essence, this paper aims to process data from a centralized database in such a way that the LLM can easily access this data and provide relevant responses using the exact information sought. This is achieved by carefully processing and segmenting the text into smaller parts that make it easier to access and interpret the information.

The motivation to incorporate the (LLM)/chatbot into the proposed hybrid model stems from the desire to create a state-of-the-art system that not only efficiently manages educational materials but also ensures accuracy in responding to user queries. This innovative system uses natural language processing (NLP) and machine learning so that it can understand the subtleties of human language, recognize patterns in the data, and continuously improve its accuracy over time.

The (LLM)/chatbot system is characterized by its ability to extract pertinent information from an extensive database of educational materials and provide users with not only prompt but also highly accurate answers. What makes this approach unique is the departure from the traditional practice of preparing and programming responses for different types of queries separately.

The implementation of the OpenAI API eliminates the need for such manual adjustments, thanks to the integration of a pre-trained model that skillfully understands contextual nuances. In essence, the hybrid model seamlessly combines a chatbot model strategically designed for seamless integration into e-learning platforms with an automated framework based on artificial intelligence and machine learning. This synergy results in a system that not only responds quickly and accurately to user queries about education, but also contributes to a holistic and interactive learning experience. The hybrid model provides students with a better and more efficient way to manage learning materials, helping to improve the overall educational journey of learners.

2. Modeling Background

Chatbots, virtual assistants that communicate with users, have become an integral part of the digital age. These programs, which are ubiquitous in many industries, facilitate interaction between humans and technology and provide fast and efficient solutions for various tasks. A computer program, algorithm, or artificial intelligence that communicates with a person or another participant in the communication is called a chatbot. The chatbot symbolizes a dialog between humans and machines through natural language [16–18]. Some chatbots work with generative models and use statistical machine translation (SMT) methods to translate input sentences into corresponding responses. Among them, Seq2Seq (Sequence to Sequence) stands out, an SMT (sequence-to-sequence machine translation) algorithm that uses recurrent neural networks (RNNs) for encoding and decoding inputs into responses—a predominant example of today's practice [19,20].

OpenAI's GPT technology focuses on an innovative approach to improving language understanding capabilities in machine learning models. It utilizes a dual process that exploits the potential of generative pre-training (GPT) [21]. First, the model is trained with a large amount of text data without a specific task and then tuned to specific tasks. This approach has proven to be effective, as the model outperforms the competition on many language tasks [22,23]. The underlying generative pre-training technology takes advantage of the abundance of unlabeled text data available on the Internet and provides a scalable solution to the challenges of understanding language [24]. The core idea of this technology lies in generative pre-training, where the model is exposed to a large amount of text data without an assigned task. The goal is to enable the model to predict the next word in a sentence. Through this process, the model acquires the structure of the language, including grammar, semantics, and some context. This helps the model to understand the nuances of language and build a basic and comprehensive understanding of language [24,25]. The architecture of the novel hybrid model is based on the transformer framework. Transformers use attention mechanisms that allow the model to focus on specific parts of the input data that are relevant to the task. This includes dynamically evaluating the meaning of different words in a sentence, capturing the context and relationships between words in a more efficient way [19,23].

After generative pre-training of the model, where basic linguistic understanding is acquired, the model is tailored to specific tasks through a process called fine-tuning. This involves making precise adjustments to the model on smaller datasets targeted at these tasks. For example, if the objective is to develop a system for answering questions, the pre-trained model is customized using a dataset specifically designed for questions and answers. This bidirectional approach ensures the retention of the general understanding acquired during pre-training while emphasizing its effectiveness for specific tasks [24].

Education is a sector that is constantly evolving, and as technology advances, the integration of chatbots into education is becoming more commonplace. Whether they help with learning, tutoring, or administrative tasks, chatbots offer new possibilities in the educational context. There are numerous examples of the use of AI and chatbots in this field [26], two of which are highlighted in this study.

The first is a chatbot model, the chatbot model described in the paper "Chatbot: An Education Support System for Student" by Clarizia et al. [27], which has been adapted for integration into e-learning platforms. Its purpose is to provide students with an enriching and interactive learning experience that allows them to ask questions and receive immediate informative feedback. The architecture is built to be scalable and capable of managing many interactions with students simultaneously. The chatbot model used in this study consists of four different modules: the front end, the back office, the knowledge-based module, and the e-learning bot module. Following the design philosophy of fluidity and adaptability, the e-learning bot module is designed for continuous learning and refinement and facilitates the simultaneous handling of many interactions with students [27].

The second model, presented by the group of authors around Hiremath et al. [28] In the paper "Chatbot for Education System", introduces an automated framework that uses artificial intelligence and machine learning to answer user queries about education quickly and accurately. The chatbot is designed to provide students and educators with simultaneous access to information and real-time solutions to their queries. The automated system is configured to process user queries using the robust capabilities of artificial intelligence and machine learning. It is equipped with knowledge from various sources and uses techniques such as pattern recognition, natural language processing, and data analysis to compare user queries with existing knowledge in the database. Artificial intelligence enables the chatbot to learn from user queries and improve its responses over time. Machine learning is used to train the chatbot to recognize patterns in data and improve accuracy. Natural language processing (NLP) makes it easier for the chatbot to understand and interpret human language. Pattern recognition is used to compare user requests with information in the database. Data analysis is used to extract relevant information from the knowledge base and provide accurate answers to user queries. When a user enters a query, the chatbot extracts the relevant keyword from the given query and generates a response. If the data are not available in the static database, the chatbot retrieves it from online sources. The chatbot is designed to be scalable, user-friendly, and highly interactive. It can be a valuable tool to provide free advice to students, suggest different courses to them, and offer information about the education system [28].

In the context of developing a forthcoming hybrid model based on the fundamental findings of two key papers, it is important to recognize the significant contributions of various research initiatives in the field of artificial intelligence (AI) within the educational domain. These studies play a crucial role in enhancing understanding of the challenges and functionalities associated with the integration of AI in education. From the wealth of relevant research, the following contributions stand out, offering a systematic exploration of the aforementioned field. The outline follows a logical sequence that begins with innovations in machine learning environments, expands to applications in education, addresses challenges, highlights positive aspects, and concludes with a detailed analysis of performance and system optimization. The first research paper by Park and Shin [29] introduces a revolutionary programming environment designed to enable students to manage the complexity of training machine learning models. By using a block-based programming language, this innovation addresses the inherent limitations of existing learning platforms. A key highlight of the study lies in the practical application of this environment. Two notable examples are the development of a question-answering chatbot and a celebrity look-alike program, both of which were carefully trained on extensive datasets. These practical implementations demonstrate the effectiveness of the proposed programming environment in real-world scenarios and highlight its potential to bridge the gap between theoretical learning and practical application. The next research work is concerned with exploring the transformative capabilities of large language models (LLMs) in the field of education. The study focuses on the concept of prompt engineering and underlines the central role of content knowledge, critical thinking, and iterative design as fundamental elements for unlocking the full potential of LLM AI. By analyzing the transformative aspects, the study sheds light on how these models can revolutionize the educational landscape. The emphasis on critical components such as content knowledge emphasizes the need for a holistic approach to the integration of LLMs that provides a nuanced understanding of their application beyond superficial functionalities [30].

A comprehensive investigation of the impact of ChatGPT in higher education probes into the far-reaching influence of ChatGPT on the dynamics of learning and instruction within higher education. By meticulously mapping the capabilities of ChatGPT, the study offers a nuanced understanding of how this AI model can reshape the educational experience. Beyond a mere exploration of technical functionalities, the research delves into the implications of seamlessly incorporating ChatGPT into university courses. This includes insights into potential effects on students' learning experiences, providing a well-rounded perspective on the tangible impacts of this AI model within higher education settings [31].

The collaboration between educators and AI in educational research provides a deep insight into the evolving relationship dynamics between educators and ChatGPT, a representative of large language models (LLMs). A study by the authors' group [32] identifies four different roles for the chatbot and three roles for educators in an educational setting, providing valuable insights into the collaborative potential of AI in the classroom. Significantly, the study highlights the critical importance of educators' pedagogical expertise and emphasizes that successful integration requires a synergistic approach that leverages both the capabilities of AI and human educators. This nuanced understanding of educator–AI collaboration lays the groundwork for considering the future use of LLM-powered chatbots in education, with a focus on maximizing the strengths of each party. Another study explores the transformative potential of ChatGPT in redefining problems rooted in probability theory and statistics for various academic domains. The study focuses on enhancing interdisciplinary learning and demonstrates ChatGPT's ability to adapt problems and make them more accessible in different fields such as biology, economics, law, and engineering. Using expert opinion and insights from 44 computer science students, the study highlights the effectiveness of the model in maintaining theoretical relevance and promoting engagement and emphasizes the practicality of using ChatGPT for interdisciplinary education [33].

The effectiveness of different types of embedding for recommending learning content, a crucial aspect of AI-enhanced learning, is described in an extensive research paper by Xiu et al. [34]. Three embedding models are evaluated: (i) static embeddings from a concept-based knowledge graph, (ii) contextual embeddings from a pre-trained language model, and (iii) contextual embeddings from a large language model (LLM). The study investigates different ensembles and fusion strategies and evaluates their performance on digital textbooks in Swedish for three subjects and two types of exercises. Results indicate that contextual embeddings from an LLM outperform other models and demonstrate their potential for linking digital learning materials. The study highlights the feasibility of using semantic text similarity based on text embeddings for recommending educational content.

The following article looks at the growing impact of large language model (LLM)-based chatbots on education and examines their positive and negative impact on teaching, learning, and assessment. Conversations with ChatGPT and relevant documents form the basis for discussing the challenges and potential of these chatbots. The article highlights the impact on deep learning, outlines the pedagogical drawbacks, and identifies key components of school culture that can help or hinder the potential of a chatbot. The theoretical framework contrasts assessment for learning and assessment for grading and highlights patterns of chatbot use that are consistent with each culture. Quotes from ChatGPT conversations reinforce the insights gained from the chatbot experience [35]. Focusing on the integration of large language models (LLMs), such as OpenAI's GPT-4, into higher education investigation emphasizes the transformative opportunities they offer across all disciplines. While the full integration of LLMs into curricula is discussed, the paper argues for a more nuanced approach that emphasizes the establishment of transparent and comprehensive guidelines. The paper argues for the involvement of stakeholders, including faculty, administrators, and students, and emphasizes the importance of clear rationales and effective implementation strategies for broad acceptance. Emphasis is placed on the process of creating appropriate policies that ensure the legitimacy of and compliance with LLM integration decisions [36]. In their research, Sajja et al. present an AI-powered framework based on GPT-3 [37] that addresses miscommunication in postsecondary education. A virtual intelligent teaching assistant (TA) promotes self-directed learning and creative thinking by providing easy access to course-related information. It aims to improve student engagement, remove barriers to learning, and reduce the logistical workload of instructors. The study describes the GPT-3-based knowledge discovery component, the architecture of the system, and evaluates accuracy and performance.

A group of authors [38] investigating the effectiveness of large language models (LLMs) in spoken language learning introduce a new dataset of multiple-choice questions in their study to evaluate their performance in areas such as phonetics, phonology, and second language acquisition. The study examines different prompting techniques, including zero- and few-shot methods, chain-of-thought (CoT) prompting, in-domain examples, and external tools. The evaluation of 20 different LLMs shows their strengths and weaknesses in tasks related to the extraction of conceptual knowledge and the application of spoken language knowledge. The results provide valuable insights into the application of LLMs in spoken language learning in educational. In the context of predicting academic success using data mining techniques, feature engineering and selection methods were compared for regression and classification tasks [39]. For regression, the Boruta algorithm and Lasso regression are evaluated, while for classification, Recursive Feature Elimination (RFE) and Random Forest Importance (RFI) are investigated. The results show that Gradient Boost with Boruta selection achieves the lowest Mean Absolute Error (MAE) and the smallest Root Mean Square Error (RMSE) in the regression. In classification, RFI stands out with an accuracy rate of 78%. The study highlights the importance of using appropriate feature engineering and selection methods to improve the efficiency of machine learning algorithms and provides insights into effective strategies for predicting study success. A transformative initiative to implement a chatbot on a higher education institution's web portal that aims to revolutionize student interaction and access to information and support services was also investigated [40]. Leveraging natural language processing, machine learning, and artificial intelligence, the chatbot conducts dynamic conversations with students and provides instant responses and personalized guidance. The chatbot caters to the diverse needs of students in different areas of student life, including information retrieval, academic support, campus services, and scheduling. The chatbot was developed with a user-centered approach that incorporates feedback from students, faculty, and administrators, and rigorous testing ensures accuracy, reliability, and security. The functional, medium-computational chatbot enhances real-time interaction on the institution's web portal and represents a significant advance in student support services.

Taken together, these research contributions enrich the understanding of the role of AI in education and include innovative programming environments, the transformative potential of LLMs, the impact of specific AI models such as ChatGPT in higher education, and the evolving collaboration between educators and AI. This comprehensive background provides a cohesive narrative describing the development of chatbots, their integration into education, and recent advances in AI and machine learning in the educational context.

3. Modeling a Hybrid Model

The following section describes the modeling of the hybrid model, which aims to integrate different technologies in order to synergize their strengths.

3.1. Model Functionality and Structure

A comprehensive understanding of how the chatbot works and how it is structured is crucial for assessing its efficiency and capabilities. From integration with other platforms and user interface design to data analysis and processing, each segment is critical to the final product. The software interface should be designed in such a way that it directly presents users with methods to use. Intuitiveness reduces the need for detailed instructions and allows users to interact with the application right from the start. This feature is essential because, regardless of the technical capabilities, the usefulness of the application will be compromised if users are reluctant to engage with it. In the digital age, when everything is fast, users expect quick responses to their requests. The application must be tuned to guarantee prompt responses to user requests. Longer waiting times can cause the user to lose concentration or become restless. Certain sensitive data, such as API keys, are embedded directly in the source code. Such an approach can reveal vulnerabilities to security threats. Protection protocols such as external variables or cryptography should be implemented to ensure the invulnerability of this information. In addition, constant updates of all packages and libraries are essential to circumvent potential security flaws.

The new hybrid model of the chatbot integrates important modules, including the aforementioned front-end user interface (UI), the back office, the knowledge-based module (database), and the e-learning module (chatbot).

The front-end module focuses on creating an intuitive interface and an easily navigable user interface that allows users to interact with the chatbot across different devices.

The back-office module, on the other hand, manages the operations in the background, orchestrates the business logic, and stores the data in the database while monitoring the interactions with the users.

The knowledge-based module serves as a specialized repository for information about users, educational objects, and other relevant data, managing and maintaining the chatbot's knowledge base. This module is crucial for providing accurate answers to students' questions. Its architecture has been carefully designed to allow flexibility and adaptability.

The e-learning bot module recognizes students' queries and provides accurate answers using natural language processing and domain ontology. As the core of the model, this module evaluates students' queries and provides relevant answers based on the knowledge-based module.

The second integrated model is an automated framework that uses machine learning to train the chatbot to improve its ability to recognize patterns in data and increase accuracy. In addition, natural language processing enables the chatbot to understand and interpret human language. The schematic of the new model illustrates how these modules work together to provide a high-quality user experience (Figure 1).

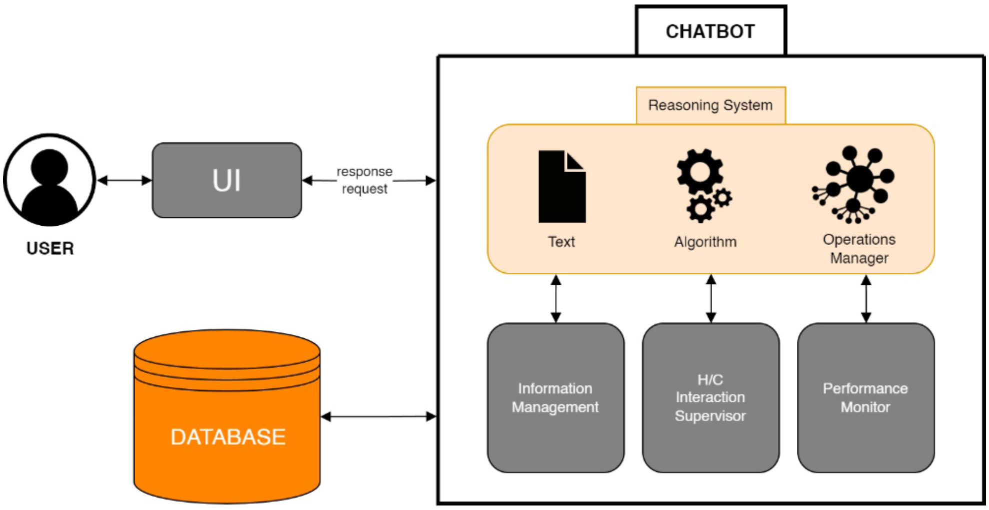


Figure 1. The principle of the hybrid e-learning AI-driven chatbot model.

The user-driven process begins and concludes with the user's interaction with the system. Facilitated through a user interface (UI), the user's responses or requests are transmitted to the UI, initiating a chain of interactions. The UI interfaces with a database housing educational materials, linking seamlessly to a chatbot equipped with a comprehensive set of elements. This includes a reasoning system with three key components: the "Text" component handling input data preprocessing, the "Algorithm" serving as the computational engine for deriving conclusions, and the "Operations Manager" orchestrating tasks and managing overall system performance. Additionally, the chatbot incorporates information management to organize received and produced information, an H/C interaction supervisor ensuring appropriate responses, and a performance monitor to assess and enhance the chatbot's functionality. Together, these elements form an integrated and responsive system, combining user interaction, educational content, and advanced reasoning capabilities.

Once the generative pre-training phase has been completed, in which the model acquires basic linguistic understanding, it is adapted to specific tasks through fine-tuning. In this approach, the model is precisely adapted to smaller data sets tailored to the respective tasks. In this way, a dual objective is achieved: the general linguistic understanding acquired during pre-training is maintained and, at the same time, the model's ability to perform specific tasks is emphasized.

3.2. Model Architecture

The model architecture based on the transformer framework represents an exceptionally sophisticated integration of key components that form the basis for performing complex natural language processing (NLP) and text classification tasks. These mechanisms allow the model to dynamically evaluate the meaning of different words in a sentence, leading to a deeper understanding of the context and subtle relationships between words, linking key components that are explained in more detail below.

Once the generative pre-training phase has been completed, in which the model acquires a basic linguistic understanding, it is adapted to specific tasks through fine-tuning. In this approach, the model is precisely adapted to smaller data sets tailored to the respective tasks.

3.2.1. Text and Position Embedding

This section deals with the embedding of words and positional information in the model. It goes beyond the concepts of traditional word embeddings; it transforms words into vectors while considering positional aspects of word order. Through this integration, the model is able to position words precisely in context, increasing the depth of its language understanding. The model receives input in the form of a sequence of words or tokens representing textual data. Each token undergoes a transformation process into a vector that includes both textual and positional embedding.

Sophisticated word embedding techniques include advanced methods to represent words as dense vectors in a high-dimensional space. The main goal of these methods is to encapsulate the semantic and contextual relationships between words in a deeper and more efficient way. Within the proposed hybrid model developed for the management of educational materials, the integration of sophisticated word embedding techniques plays a central role in significantly improving the model's capabilities in understanding and processing textual data.

The use of sophisticated word embedding techniques in the proposed hybrid model proves to be indispensable and serves as a cornerstone for the creation of semantically enriched representations derived from the input data. Through this embedding, the model is able to recognize the subtleties of language to promote a comprehensive understanding of educational materials. The use of these advanced techniques is essential for the model's skillful handling of complicated textual data. This includes recognizing grammatical structures and extracting relevant information from educational materials. By incorporating sophisticated word embedding methods, the proposed hybrid model experiences a remarkable improvement in its capabilities, ensuring a more nuanced and accurate interpretation of educational content. These methods contribute significantly to the model's capabilities and enable it to navigate through complex linguistic constructs. This enables a deeper understanding of educational materials and ultimately increases the overall effectiveness of the educational materials management system.

Mathematically, this can be expressed as follows:

Let input represent a sequence of words or tokens $\{w_i\}$ that represent textual data. Vector representations are defined for each word w_i as *Embed*(w_i), where *Embed*(w_i) represents the embedding of the word. Let *PositionEmbedding*(p_i) be the vector representation of position p_i. The integration of words and position information can be described as follows:

$$Embedfinal(w_i) = LayerNorm(W_q \cdot Embed(w_i) + W_k \cdot PositionEmbedding(pos_i)) \quad (1)$$

where W_q and W_k are weight matrices for query (q) and key (k), and p_i stands for the position of the word w_i. *Embedfinal* is the resulting embedding for the word w_i. *Embedfinal* is the resulting embedding for the word w_i. Each embedding *Embedfinal*(w_i) undergoes a transformation process into a vector that includes both textual and positional embedding.

In matrix form this can be expressed as follows:

$$Embed_{final}(w_i) = LayerNorm\left(\begin{bmatrix} W_q & W_k \end{bmatrix} \cdot \begin{bmatrix} Embed(w_i) \\ PositionEmbeding(pos_i) \end{bmatrix}\right) \quad (2)$$

3.2.2. Masked Multi-Self Attention

This sophisticated mechanism ensures that the model focuses on the relevant parts of the input text for each token and uses masking to avoid the influence of future information on the current token. In addition, the model can adapt its information processing to the current context, which increases the accuracy of the text analysis.

In mathematical terms, for attention in the *i*-th step and *h*-th attention head:

$$Attention_{hi} = Softmax\left(\frac{W_{ghi}x(EmbedFinal + PositionEmbedding)}{\sqrt{dk}}\right) \cdot (W_{vhi} \cdot (EmbedFinal + PositionEmbedding)) \quad (3)$$

where *dk* is the dimension of the key, and *Softmax* is the function that calculates the probabilities.

In matrix form, this can be written as:

$$Attention_{hi} = Softmax\left(\frac{W_{ghi} \cdot \begin{bmatrix} EmbedFinal \\ Position\ Embedding \end{bmatrix}}{\sqrt{dk}}\right) \cdot \left(W_{vhi}\begin{bmatrix} EmbedFinal \\ PositionEmbedding \end{bmatrix}\right) \quad (4)$$

The outputs from all transformations are combined and a linear transformation is added:

$$Combined_h = Concat(Attention_{h1}, Attention_{h2}, \ldots, Attention_{hi}) \quad (5)$$

in matrix form as:

$$Combined_h = Concat\left(\begin{bmatrix} Attention_{h1} \\ Attention_{h2} \\ \vdots \\ Attention_{hi} \end{bmatrix}\right) \quad (6)$$

Furthermore, the following is:

$$FinalEmbedding_h = LayerNorm(U_{hi} \cdot Combined_h + \gamma_{hi} \cdot Combined_h + \beta_{hi}) \quad (7)$$

in matrix form:

$$FinalEmbedding_h = LayerNorm\left(\begin{bmatrix} U_{hi} & \gamma_{hi} & \beta_{hi} \end{bmatrix} \cdot Combined_h\right) \quad (8)$$

where *Concat* denotes the vector concatenation operation, and γ_{hi} and β_{hi} are the layer normalization parameters for each attention head.

Also, let X be the input matrix $L \cdots d$, where L represents the length of the text (number of tokens) and d is the dimension of the embedding vector. Weight matrices are defined for queries (W_Q), keys (W_K), and values (W_V) with the dimensions $d \cdots d_k$, where d_k is the dimension of the key:

$$Q = X \cdot W_Q,\ K = X \cdot W_K,\ V = X \cdot W_V \tag{9}$$

in matrix form is expressed as:

$$\begin{bmatrix} Q \\ K \\ V \end{bmatrix} = \begin{bmatrix} X \cdot W_Q \\ X \cdot W_K \\ X \cdot W_V \end{bmatrix} \tag{10}$$

This formulation illustrates the use of the attention mechanism and ensures that the model is able to consider relevant information while considering the current context and avoiding the inclusion of future data during processing.

And the attention function for each token i is defined as:

$$Attention_i = Softmax\left(\frac{Q_i(K_i)^T}{\sqrt{dk}}\right) \cdot V_i \tag{11}$$

where Q_i, K_i, and V_i are the rows of the matrices Q, K, and V, respectively.

The *Softmax* function generates a probability distribution, and scaling with $\sqrt{dk}$ helps to stabilize the training.

3.2.3. Layer Normalization

Layer normalization, used to maintain stability and accelerate model training, represents a pivotal component of performance optimization. This layer normalization technique allows the model to process data consistently and efficiently and to strike a balance between stability and speed.

Mathematically, it can be expressed that for each sample l, a normalization is performed over all dimensions d:

$$N(X) = \frac{X - \mu}{\sigma} \bigodot \gamma + \beta \tag{12}$$

where μ is the vector of mean values across dimensions, $\mu_i = \frac{1}{d}\sum_j^d x_{ij}$ for i = 1,. . ., d, σ is the vector of standard deviations across the dimensions, $\sigma_i = \sqrt{\frac{1}{d}\sum_{j=1}^d \left(x_{ij} - \mu_i\right)^2}$ for i = 1,. . ., d, γ is the scaling vector (learnable parameter), and β is the displacement vector (learnable parameter).

This normalization ensures that the mean values across each dimension are close to zero and the standard deviations are close to one, which facilitates model training by stabilizing the data distribution.

3.2.4. Feed Forward

This layer performs additional transformations on the output of the attention layer and forms more complex features. The Feed-Forward Transformation can be expressed mathematically as follows:

$$FF(X) = \sigma(W_{ff}\delta\left(W_{feed}X + b_{feed}\right) + b_{ff} \tag{13}$$

expressed in matrix form:

$$FF(X) = \sigma\left(\begin{bmatrix} w_{11} & w_{21} & \cdots & w_{1n} \\ w_{12} & w_{22} & \cdots & w_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{m1} & w_{m2} & \cdots & w_{mn} \end{bmatrix} \cdot \delta\left(\begin{bmatrix} w_{feed11} & w_{feed21} & \cdots & w_{feed1n} \\ w_{feed12} & w_{feed22} & \cdots & w_{feed2n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{feedm1} & w_{feedm2} & \cdots & w_{feedmn} \end{bmatrix} X + \begin{bmatrix} b_{feed1} \\ b_{feed2} \\ \vdots \\ b_{feedm} \end{bmatrix}\right) + \begin{bmatrix} b_{ff1} \\ b_{ff2} \\ \vdots \\ b_{ffn} \end{bmatrix}\right) \tag{14}$$

The reapplication of Layer Normalization after this process ensures additional layer normalization and preserves the stability that is crucial for sustained and successful information processing following additional complex transformations.

$$LN(Y) = \frac{\gamma Y}{\sigma^2 + \epsilon} + \beta \tag{15}$$

where Y is the output from the Feed-Forward Layer, γ and β are normalization parameters, μ and σ are mean values and standard deviation, and ϵ is a small constant for stability.

In matrix form this can be expressed as follows:

$$\begin{bmatrix} ln(Y_1) \\ ln(Y_2) \\ \vdots \\ ln(Y_n) \end{bmatrix} = [A] \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix} + [B] \tag{16}$$

where $A = \frac{\gamma}{\sigma^2+\epsilon}$ and $B = \beta$.

3.2.5. Text Prediction

This component represents the core of the model and enables the prediction of the next words in the text, which is crucial for generative pre-training tasks where the model acquires fundamental language understanding.

The text prediction component of the model can be described mathematically by additionally including several latent variables and layers. Let X and Y be vectors that represent the input context of the text and the distribution of the probability of the next word. First, a latent vector Z is introduced, which represents complex features that the model learns from the context:

$$Z = ReLU(W_{hid1} \cdot X + b_{hid1}) \tag{17}$$

where W_{hid1} is the weight matrix for the first hidden layer, b_{hid1} is the bias vector for the first hidden layer, and $ReLU$ is the activation function.

Matrix-wise:

$$Z = \begin{bmatrix} Z_1 \\ Z_2 \\ \vdots \\ Z_n \end{bmatrix}, W_{hid1} = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1n} \\ w_{21} & w_{22} & \cdots & w_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{m1} & w_{m2} & \cdots & w_{mn} \end{bmatrix}, X = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_n \end{bmatrix}, b_{hid1} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix} \tag{18}$$

Then, a second latent vector H is introduced using the information from Z:

$$H = ReLU(W_{hid2} \cdot Z + b_{hid2}) \tag{19}$$

where W_{hid2} and b_{hid2} are the weight matrix and the bias vector for the second hidden layer.

Expressed in matrix notation:

$$H = \begin{bmatrix} H_1 \\ H_2 \\ \vdots \\ H_n \end{bmatrix}, W_{hid2} = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1n} \\ w_{21} & w_{22} & \cdots & w_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{m1} & w_{m,} & \cdots & w_{mn} \end{bmatrix}, X = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_n \end{bmatrix}, b_{hid2} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix} \tag{20}$$

Finally, H is used to predict the distribution Y:

$$Y = Softmax(W_{out} \cdot H + b_{out}) \tag{21}$$

in matrix form:

$$Y = Softmax\left(\begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1,n} \\ w_{12} & w_{22} & \cdots & w_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{m1} & w_{m2} & \cdots & w_{mn} \end{bmatrix} \cdot \begin{bmatrix} H_1 \\ H_2 \\ \vdots \\ H_n \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix} \right) \tag{22}$$

This architecture enables the model to learn hierarchical and complex features from the context of the text, adaptability through multiple hidden layers, and precise prediction of the next word.

3.2.6. Text Classifier

This component is used to classify texts into specific categories, allowing the model to adapt to specific fine-tuning tasks. This modularity makes the architecture highly adaptable to various language tasks responding to the requirements for precise analysis and text generation based on a rich language context. Overall, the integration of key components makes the proposed architecture an exceptionally robust foundation for tackling natural language processing challenges.

Mathematically, it can be expressed as follows: let x_i be the input vectors of text and y_i the corresponding categories. The model parameters include the weights W_{TC} and the bias b_{TC} for the classification layer. Then, the expression can be formulated using the *Softmax* function:

$$P(y_i|x_i) = \frac{e^{W_{TC} \cdot x_i + b_{TC}}}{\sum_j e^{W_{TC} \cdot x_j + b_{TC}}} \tag{23}$$

in matrix form written as:

$$(y_i|x_i) = \frac{exp\left(\begin{bmatrix} W_{TC} & b_{TC} \end{bmatrix} \cdot \begin{bmatrix} x_{i1} \\ x_{i2} \\ \vdots \\ x_{in} \\ 1 \end{bmatrix} \right)}{\sum_j exp\left(\begin{bmatrix} W_{TC} & b_{TC} \end{bmatrix} \cdot \begin{bmatrix} x_{j1} \\ x_{j2} \\ \vdots \\ x_{jn} \\ 1 \end{bmatrix} \right)} \tag{24}$$

where $P(y_i \mid x_i)$ represents the probability that the input text x_i belongs to the category y_i. For fine-tuning, it is possible to adjust the weights W_{TC} to adapt the model to specific classification tasks.

The proposed architecture enables the model to integrate information about the text and its context and to apply attention mechanisms and deep transformations to achieve high precision in natural language processing tasks and text classification. Moreover, it provides a robust foundation for applying the model in a variety of language-related tasks addressing the need for precise analysis and text generation based on extensive linguistic context (Figure 2).

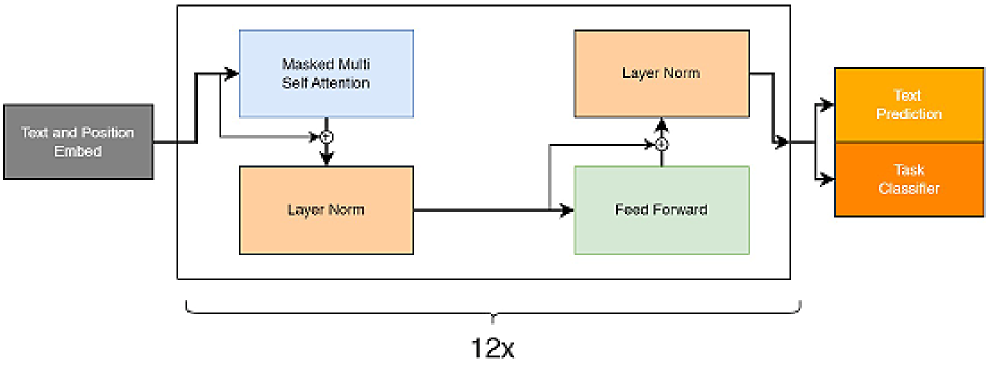


Figure 2. Transformer architecture.

The complexity of an algorithm does not necessarily have a proportional effect on the computing resources required. It is often assumed that the complexity of an algorithm reflects its efficiency and the way in which the resource requirements grow as the input size increases. However, it is important to understand that algorithms can be complex but at the same time fast and efficient in performing their tasks. In the context of educational materials, which typically involves the processing of text and PDF files, this algorithm can provide comprehensive and efficient processing, regardless of the amount of data, without the need for significant material investment. It is emphasized that the efficiency of an algorithm does not always depend on its complexity. The decisive factor is the algorithm's ability to process certain types of data and tasks effectively. Therefore, algorithms can be tailored to deliver optimal results with minimal computational effort.

3.3. Model Functional Principle

This hybrid text processing model provides sophisticated analysis and generation of responses. The model processes complex textual data containing different information, writing styles, and contextual nuances. Through meticulous tokenization, it transforms the text into a comprehensible form for further analysis. Text processing includes lemmatization, stop word removal, and keyword extraction. In addition, the model utilizes word vector representation techniques to create semantically rich representations of the input data.

Advanced forms of syntactic tagging, such as complex neural networks, are used to identify and label grammatical relationships between words. Through in-depth syntax analysis, the model recognizes grammatical structures, including subjects, objects, and other elements, enabling a better understanding of structural and semantic relationships in the text. By applying pattern recognition algorithms to extract relevant information from complex sentences, the model also uses its own database for an additional contextual layer.

Through decision-making model ensembles, the model evaluates multi-criteria factors to select the optimal response. These factors include semantic relevance, grammatical correctness, and the history of previous responses. The result is a high-quality, contextually coherent response generated by the model adaptable to various scenarios and contexts (Figure 3).

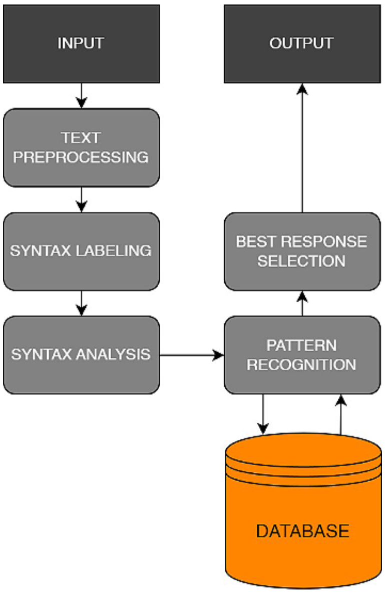


Figure 3. Model operating principle.

The initial phase of text preprocessing involves transforming raw text data into a format that can be analyzed. This includes measures such as lowercasing and eliminating punctuation. After this step, the sentence structures are examined through syntax labeling and syntax analysis, in which the grammatical roles of each word are determined. The next step is pattern recognition, in which the model identifies recurring phrases or common themes in the text. Using these patterns, the model selects the most appropriate response based on the input it receives. To formulate its responses, the model relies on a database that contains a wealth of information. Finally, the output represents the model's final response, which is generated through a meticulous analysis of the input and interacting with the database.

The integration of this approach not only promotes a deep understanding of language, but also efficiently incorporates various aspects of linguistic analysis, providing responses of exceptional precision and contextual relevance. Contemporary methods of natural language processing form the basic structure of this model, which focuses on the subtleties of linguistic dynamics.

3.4. Modeling Tools

Modeling was performed with the Python programming language using packages that provide a variety of tools and algorithms for functions such as data preprocessing, feature extraction, model training, and evaluation. Tkinter was used as the standard Python package for the creation of graphical user interfaces (GUIs). In addition, the Python Imaging Library (PIL) package facilitated the opening, manipulation and saving of various image formats, including PPM, PNG, JPEG, GIF, TIFF, and BMP. LangChain is a framework designed to simplify the creation of applications with LLMs. Serving as a framework for language model integration, LangChain's applications largely overlap with those of language models in general and include document analysis, summaries, chatbots and code analysis. PyPDF2 is a package for splitting, merging, cropping, and transforming PDF files. It can also add custom data, viewing options and passwords to PDF files and extract text and metadata from PDFs.

The OS module in Python provides functions for interacting with the operating system. The OS module provides a portable way to use operating system dependent functions and provides operations for working with file and directory paths, reading environment variables and other operating system related tasks. The Difflib module provides classes and functions for comparing sequences. For example, it can be used to compare files and generate information about differences between files in various formats, including HTML, as well as contextual and uniform differences. The web browser module provides a simple interface for launching web browsers. While it is primarily intended for opening web pages, it can also be used to open local files, including PDF files, with the appropriate application on the system.

4. API Implementation

An API (application programming interface) is a set of defined rules and specifications that enable communication between software applications. It allows applications to access functions and data from other applications, services, or platforms. The OpenAI API is used in the program code, as the following line shows (Figure 4).

```
os.environ['OPENAI_API_KEY'] = 'API_key'
```

Figure 4. Calling GPT-4 through API.

Through the API, GPT-4, OpenAI's most advanced system, is invoked. The model is based on transformer neural network technology, eliminating the need for training for each specific task, making it ideal for personal data applications.

User interface

The user interface is created using the Tkinter package. The basic components include:

- Main window: Initialized with root = tk.Tk(). This is the main window of the application in which other elements are located.
- Query Input Field: A text field in which the user can enter queries or keywords. Initialized with input_entry = tk.Entry(root, width = 50).
- Buttons: There are several buttons in the user interface, including one for processing queries (send_button), searching for keywords (keyword_button), searching for images (image_search_button) and starting a new conversation (new_conversation_button).
- Text Field for Results: The results of queries are displayed in this area. It is initialized with result_text = tk.Text(root, wrap = tk.WORD, width = 100, height = 20, state = tk.DISABLED).

The user interface is designed to be as simple as possible in order to demonstrate its functionality. In case of further development and implementation, it is possible to customize the user interface to new esthetic trends to make it more intuitive and better.

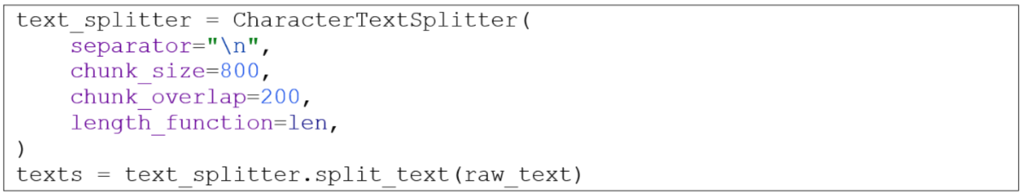
Database and Data Processing

The program is based on the local storage of data, especially PDFs and images. These data are stored in directories defined by the variables pdf_folder and image folder of particular interest is how the PDF content is processed. The variable pdf_folder refers to a database with PDF files of educational materials that the program uses to generate responses. PDF files undergo a text extraction process in which each page of each file is transformed into a searchable textual format, accumulating content in a variable called 'raw_text' (Figure 5).

```
def open_pdf(event, pdf_path):
    webbrowser.open(pdf_path)
```

Figure 5. Loading PDF files.

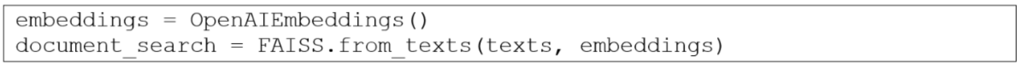
To enable an efficient search, the text is split into parts using CharacterTextSplitter. The text is segmented into parts with a size of 800 characters and an overlap of 200 characters, as defined in the line (Figure 6):



```
text_splitter = CharacterTextSplitter(
    separator="\n",
    chunk_size=800,
    chunk_overlap=200,
    length_function=len,
)
texts = text_splitter.split_text(raw_text)
```

Figure 6. Text segmentation.

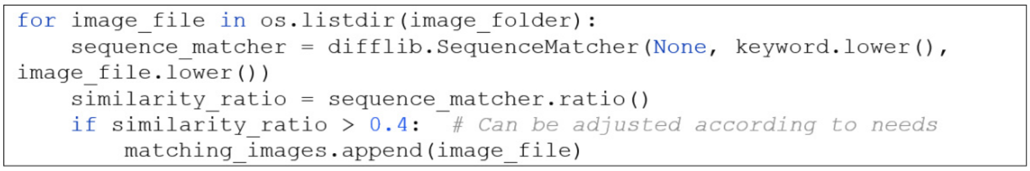
In addition to text splitting, OpenAIEmbeddings are used for contextual searches, which convert each text segment into a vector form or embedding. These vector shapes are then indexed with the FAISS tool, which enables a quick search for similar text segments. This can be seen in this line (Figure 7):



```
embeddings = OpenAIEmbeddings()
document_search = FAISS.from_texts(texts, embeddings)
```

Figure 7. Document search.

Furthermore, to search for textual data, there is functionality for searching images with the function search_image(). This function uses the 'difflib.SequenceMatcher', which searches for images based on the similarity of names and compares the similarity between two sequences and returns a similarity ratio. If the similarity_ratio is higher than a set threshold (in this case 0.4), the image is considered similar to the keyword and displayed to the user (Figure 8). The proposed hybrid model is designed to be adaptable and flexible by changing the educational materials in the database and changing the specificity of the output by changing the code. Any educational field that can represent knowledge in the form of text will have no problems with this model. The value 0.4 can be adjusted as required. The larger the number, the more similar the output is to the input i.e., expressions are extracted from the database that are more similar to the word entered. If you increase the "similarity number", the range of information that can be extracted becomes smaller. At the same time, precision increases, which is beneficial for retrieving specific data from materials. However, creativity is affected as more information is transformed into an idea and a new whole.



```
for image_file in os.listdir(image_folder):
    sequence_matcher = difflib.SequenceMatcher(None, keyword.lower(),
image_file.lower())
    similarity_ratio = sequence_matcher.ratio()
    if similarity_ratio > 0.4:  # Can be adjusted according to needs
        matching_images.append(image_file)
```

Figure 8. Calculation of similarity ratios.

This model is an excellent solution for educational materials in the social sciences, as the data are often in the form of text, which fits perfectly with the capabilities of natural language processing (NLP) models. The STEM field often involves math and engineering tasks where it is necessary to apply concepts such as differential equations, recursion, and other mathematical models. Such tasks require more than just an understanding of the linguistic context. Nevertheless, this model is also used in STEM fields as it facilitates the understanding of theory and enables relevant information to be found quickly. This enables a better understanding of complex concepts in STEM through the language model and contributes to a more efficient approach and interpretation of materials.

In this version of the program, each image must be entered into the database individually with the corresponding name of the subject on the image, but there are packages that can use machine learning to recognize images directly from files and extract information about their content. For better user interaction, the program uses the process_query() function, which uses OpenAI to generate context-based responses based on the context of the conversation and the documents found by the search. Additionally, there is also a search_keyword function that allows users to search for keywords directly in PDF files. If the keyword is found, the file name is displayed to the user as an active link, which they can click on to open and read the file in a viewer. In addition to the local directory structure, which is used for simplicity, there is the option of scaling to cloud databases for increased security and performance requirements.

The final operating principle of the hybrid model is illustrated in Figure 9.

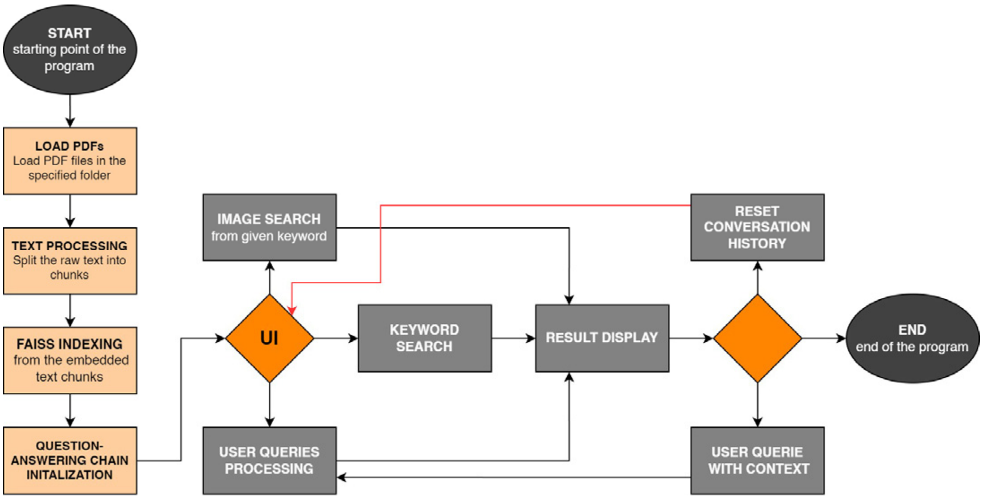

START
starting point of the program
LOAD PDFs
Load PDF files in the specified folder
TEXT PROCESSING
Split the raw text into chunks
FAISS INDEXING
from the embedded text chunks
QUESTION-ANSWERING CHAIN INITALIZATION
IMAGE SEARCH
from given keyword
UI
USER QUERIES PROCESSING
KEYWORD SEARCH
RESULT DISPLAY
RESET CONVERSATION HISTORY
USER QUERIE WITH CONTEXT
END
end of the program

Figure 9. Final operating principle of the hybrid AI-driven e-learning model.

The process commences with loading PDF files into a designated folder, followed by the raw text from these files into manageable chunks, considering the potential token overflow in documents. A FAISS index is created from the embedded text chunks, enabling efficient similarity searches for large-scale data. Initiating the question-answering chain involves setting up algorithms and data structures to respond to user queries. Image searches are based on specified keywords and user interaction takes place via the user interface (UI). User queries undergo a processing principle that involves comprehending the user's inquiry, matching it with relevant database information and formulating a response. Keyword searches are performed, and the obtained results are displayed to the user as a response to the query or as a result of the keyword searches. The user has the option to reset the conversation history, enabling a fresh start without being influenced by previous interactions, ending with the completion of the program.

5. Model Testing and Evaluation

Various teaching materials were used to demonstrate the program. The decision to focus on national materials in the database was deliberate, particularly with regard to a controlled and targeted initial test phase. For a broader and more comprehensive evaluation, however, it is necessary to extend the scope to include international materials. Expanding on this, it could be stated that during the testing phase, a diverse set of random materials was intentionally chosen to assess the model's adaptability and performance. It is important to mention that the model, which is based on the OpenAI API, is internationally applicable. This is due to the fact that the large language model (LLM) accessed through the API has undergone extensive training in different languages, rendering it versatile and proficient in dealing with linguistic nuances on a global scale.

In line with the research perspective, the material processing and output generation of the model are language-independent. The API enables seamless interaction with the language model, which has been trained for a variety of languages. This inherent multilingualism ensures that the model can effectively process and respond to queries in different linguistic contexts.

In summary, while the initial database was focused on national materials for testing purposes, the model itself is designed to be internationally applicable, demonstrating its versatility and adaptability across different languages and content sources.

5.1. Model Testing

The program, with a user interface created with the Tkinter package, contains a field for entering the user's query and four buttons (Figure 10).

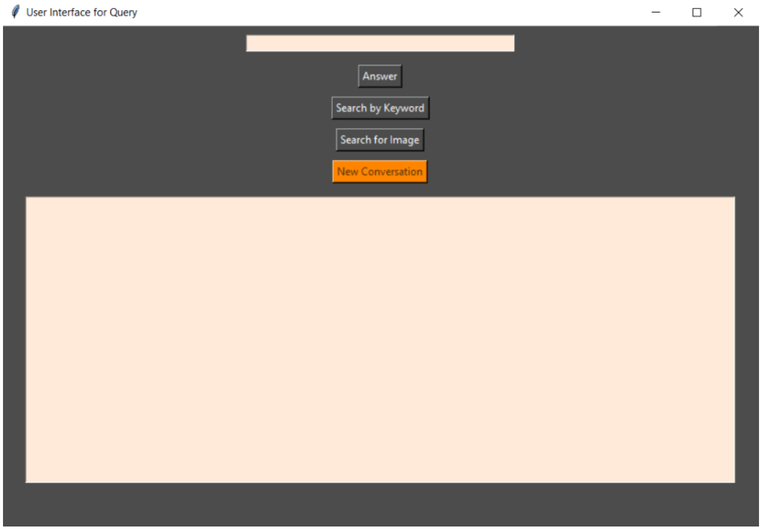
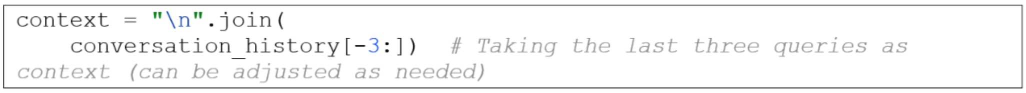


Figure 10. Program interface.

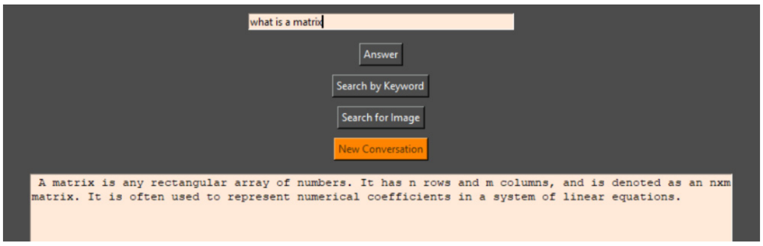
By clicking the 'answer' button or the 'RETURN' key on the keyboard, the program searches the local database and uses the implemented GPT model to generate a response to the query. The program is not case-sensitive or syntax-sensitive but recognizes similar words. After clicking the button, the entered query is cleared for easier subsequent input. Based on the line of code (Figure 11):



```
context = "\n".join(
    conversation_history[-3:])  # Taking the last three queries as
context (can be adjusted as needed)
```

Figure 11. Context and history of the conversation.

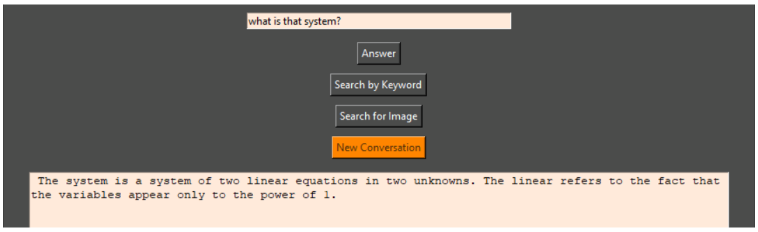
The program retains the context of the conversation, and the user can ask a question that relates to the previously answered question. The conversation depth in the code is set to 3 (the program remembers the context of the last 3 responses) but can be increased at the expense of response speed. If GPT cannot generate a meaningful answer or there is no information in the database, it returns 'I do not know' To delete the saved conversation that the program uses as context for its next response, the 'New Conversation' button is used (Figures 12–14).



what is a matrix

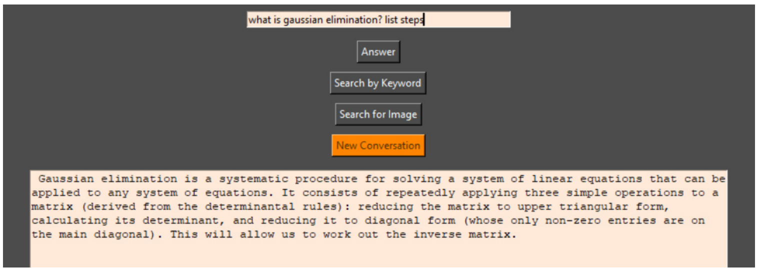
```
 A matrix is any rectangular array of numbers. It has n rows and m columns, and is denoted as an nxm
matrix. It is often used to represent numerical coefficients in a system of linear equations.
```

Figure 12. Answer to the question "What is a matrix?", disregarding syntax.



```
 The system is a system of two linear equations in two unknowns. The linear refers to the fact that
the variables appear only to the power of 1.
```

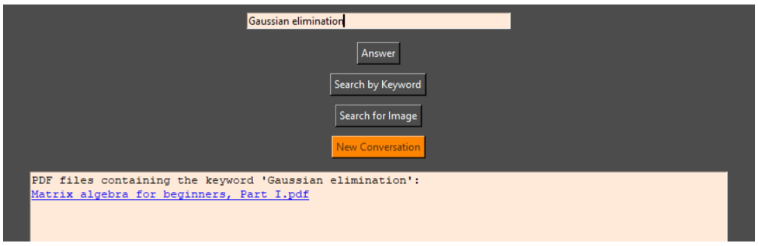
Figure 13. Continuation of the question from the previous inquiry.



```
 Gaussian elimination is a systematic procedure for solving a system of linear equations that can be
applied to any system of equations. It consists of repeatedly applying three simple operations to a
matrix (derived from the determinantal rules): reducing the matrix to upper triangular form,
calculating its determinant, and reducing it to diagonal form (whose only non-zero entries are on
the main diagonal). This will allow us to work out the inverse matrix.
```

Figure 14. Example of a query and response from educational materials.

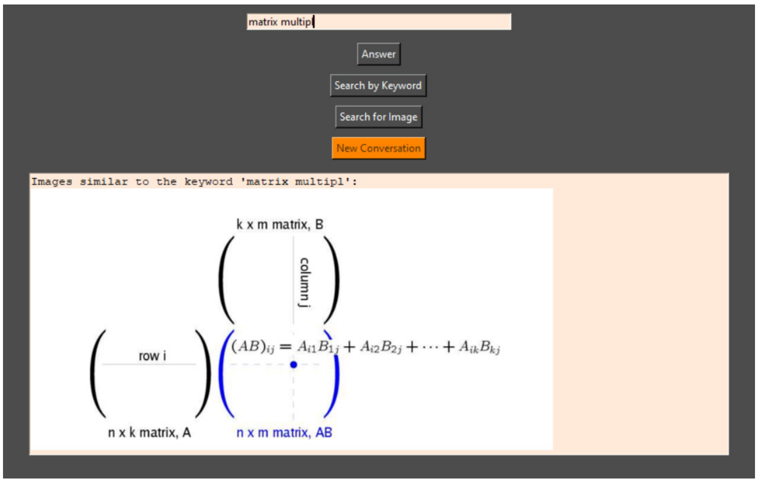
The 'Search Keyword' option searches for all similar words that match the user's input and lists the PDF files that contain them. The name of each file is only listed once, even if the same word appears several times. After the listing, the user can click on the PDF file, which then opens in the browser and is ready for reading (Figure 15).



```
PDF files containing the keyword 'Gaussian elimination':
Matrix algebra for beginners, Part I.pdf
```

Figure 15. Listing PDF documents that contain the word "Gaussian elimination".

The 'Search Image' option lists all images with the same or a similar name based on user input (Figure 16).



```
Images similar to the keyword 'matrix multipl':
```

Figure 16. Listing images containing the words "matrix multipl(ication)" or similar.

This study deliberately focuses on the essential functionalities of the program to search for educational material and emphasizes the simplicity and accessibility of this information. It is important to note that the focus was not on an in-depth examination of the user interface design (UI) or user experience (UX) aspects. Instead, the main focus was on the functionality of the algorithm and how it effectively performs its tasks. The primary user interfaces used in the study are purely for demonstrating functionality, while future development should necessarily focus on creating a user-friendly interface to enable wider use by educators and students, as emphasized in the study.

Although not explicitly mentioned in the paper, there is a need to raise awareness of the importance of ethical considerations, including data protection, information security, and potential bias in educational materials or the model itself. The bias of the model results from the bias in the data on which the model is trained. Since the OpenAI model API is used, pre-trained parameters (weights) configured during training are imported. The database with the training materials is not the database on which the model was trained. It is the database from which the model reads the data in a previously learned way, only with different information. In the developed application, a special emphasis is placed on security. Emphasis is placed on regularly updating all packages and libraries to avoid potential security vulnerabilities. The need to implement cryptography to ensure the inviolability of information is emphasized and the importance of security measures within the system itself is highlighted. These sections of the paper clearly show how seriously security issues related to the developed application are being addressed. It is important to point out that this is a local application that draws on materials that the user enters into the local database. This local nature of the application provides users with an additional layer of control over their own data, reducing the risk of unauthorized access or misuse. All of these measures have been carefully thought through to ensure responsible and ethical use of the system in the context of education.

5.2. Model Evaluation

This hybrid model test focuses on basic performance evaluation metrics, including *precision*, *recall*, and *F*1 *score*. These key indicators provide a deeper insight into the precision, recall and overall efficiency of the model in the context of analyzing responses, searching for keywords and generally its capabilities. By analyzing these metrics in detail, it is intended to provide a comprehensive picture of the model's performance to enable a relevant and informative assessment of its effectiveness.

Precision represents the ratio between the true positive predictions and all positive predictions of the model. This metric is valuable for assessing how accurate the model's responses are among all selected model predictions. High precision indicates that the model tends to produce correct results when it claims to know something.

Precision is defined as the ratio of true positives to the sum of true positives and false positives, expressed mathematically as:

$$Precision = \frac{True\ positives}{True\ positives + False\ positives} \tag{25}$$

This formula quantifies the accuracy of the model's predictions concerning positive instances.

Recall measures the ratio of true positive predictions to all true positive instances in the dataset. This metric highlights the model's ability to identify all relevant information, regardless of the total number of positives. A high *recall* indicates that the model tends to capture the majority of relevant responses.

Recall, also known as sensitivity or true positive rate, is defined as the ratio of true positives to the sum of true positives and false negatives, expressed mathematically as:

$$Recall = \frac{True\ positives}{True\ positives + False\ negatives} \tag{26}$$

This formula measures the ability of the model to correctly identify all relevant instances of a positive class.

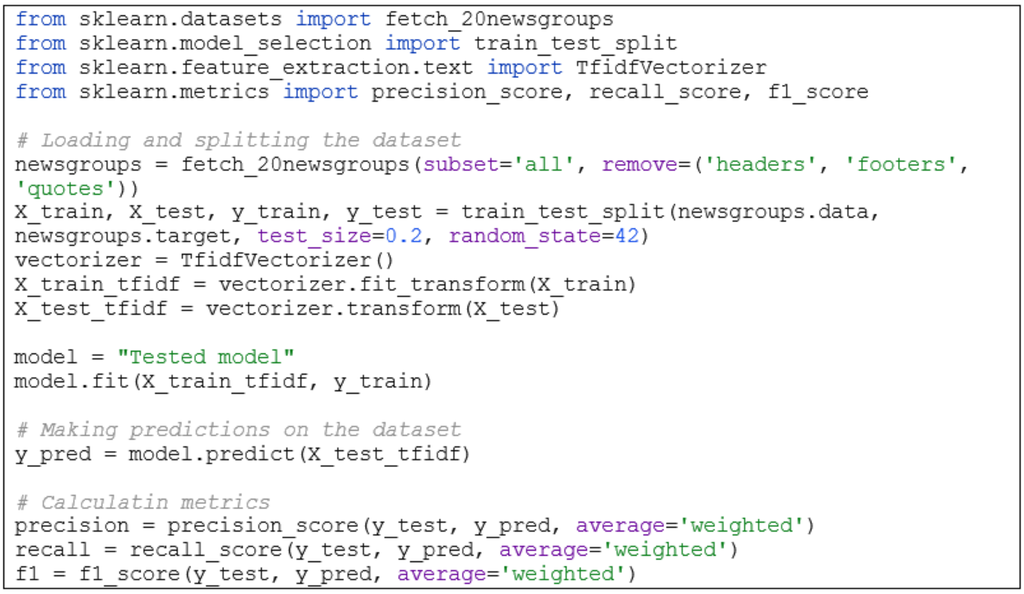
The *F*1 *score* is the harmonic mean between precision and recall. This metric provides a balanced assessment of the model's performance, which is particularly useful when precision and recall are in conflict. The *F*1 *score* is often chosen to measure the performance of a model faced with an unbalanced dataset, as it incorporates both metrics into one value.

The *F*1 *score*, which is the harmonic mean of *precision* and *recall*, is expressed as follows:

$$F1\ score = 2 \cdot \frac{Precision\ \cdot\ Recall}{Precision + Recall} \tag{27}$$

This formula combines precision and recall into a single metric, providing a balanced measure of a model's performance, especially in scenarios where false positives and false negatives are of varying importance.

The analysis of these three key indicators provides a detailed insight into the ability of our model to provide highly accurate, comprehensive, and relevant answers, which is crucial for its application in the context of educational materials. The hybrid model was assessed using the Skicit-learn Python dataset, i.e., 20 Newsgroups dataset, a well-known, public collection of documents from different newsgroups (Figure 17). This dataset was selected for evaluation purposes due to its representation and coverage of various topics. By utilizing this benchmark dataset, the performance of the model should be thoroughly evaluated in a real-world scenario involving multi-class text classification. This evaluation provides valuable insights into the model's capacity to accurately categorize documents across diverse subjects, as well as its ability to handle complex, multi-topic text data with effectiveness and resilience. When evaluating the performance of the model across different categories, not only the accuracy of the predictions is considered but also their distribution across the various classes. Therefore, an additional layer of precision and reliability is incorporated in a code, as when faced with uncertainty, it transparently responds by indicating that it does not possess the necessary information.



```
from sklearn.datasets import fetch_20newsgroups
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics import precision_score, recall_score, f1_score

# Loading and splitting the dataset
newsgroups = fetch_20newsgroups(subset='all', remove=('headers', 'footers',
'quotes'))
X_train, X_test, y_train, y_test = train_test_split(newsgroups.data,
newsgroups.target, test_size=0.2, random_state=42)
vectorizer = TfidfVectorizer()
X_train_tfidf = vectorizer.fit_transform(X_train)
X_test_tfidf = vectorizer.transform(X_test)

model = "Tested model"
model.fit(X_train_tfidf, y_train)

# Making predictions on the dataset
y_pred = model.predict(X_test_tfidf)

# Calculatin metrics
precision = precision_score(y_test, y_pred, average='weighted')
recall = recall_score(y_test, y_pred, average='weighted')
f1 = f1_score(y_test, y_pred, average='weighted')
```

Figure 17. Model evaluation Python code.

The analysis of these three key indicators provides a detailed insight into the ability of our model to provide highly accurate, comprehensive, and relevant answers, which is crucial for its application in the context of educational materials (Table 1).

Table 1. Model performance metrics.

Relation Type/Topic	*Precision*	*Recall*	*F1 Score*	Msg Count
alt.atheism	0.976	0.919	0.947	799
comp.graphics	0.587	0.600	0.596	973
comp.os.ms-windows.misc	0.865	0.733	0.794	985
comp.sys.ibm.pc.hardware	0.933	0.882	0.907	982
comp.sys.ibm.mac.hardware	0.825	0.827	0.826	961
comp.windows.x	0.733	0.762	0.8747	980
sci.electronics	0.596	0.508	0.548	981
misc.forsale	0.750	0.730	0.740	972
rec.autos	0.840	0.932	0.884	990
rec.motorcycles	0.713	0.535	0.611	994
rec.sport.baseball	0.845	0.868	0.856	994
rec.sport.hockey	0.833	0.798	0.815	999
sci.crypt	0.658	0.585	0.619	991
Weighted average	0.781	0.745	0.760	~969

Table 1 shows the evaluation results of the hybrid model for various relationship types and topics. For example, in the category "alt.atheism", the precision is 0.976, which means that 97.6% of the predictions labeled as "alt.atheism" actually belonged to this category. The recall is 0.919, which means that the model identified 91.9% of the actual "alt.atheism" instances. The F1 value is 0.947 and suggests that the model achieves a high level of accuracy and completeness in its predictions for the given category (94.7%). Msg count (message count) indicates the total number of instances in each category. The high average precision of 0.781 indicates that the model tends to correctly identify true positive examples for the given categories. A high average recall of 0.745 indicates that the model is able to identify a larger proportion of true positive examples. In addition, high average *F*1 *score* values of 0.768 indicate a good balance between precision and recall for the given categories.

These values reflect the overall performance of the model, considering the number of instances in each category. The weighted averages provide a balanced overview of the effectiveness of the model in the different areas, taking into account the unbalanced distribution of the data. Based on these results, the model can be assessed as robust, with room for improvement in certain categories to achieve even better performance.

6. Practical Implementation, Research Limitations and Future Work

Delving into the potential of the hybrid model for centralized database access in education, it becomes evident that the practical implementation, real usability, and potential challenges of using such a system in educational settings are of central importance. The practical implementation of the hybrid model involves the seamless integration of advanced technologies, including the transformer framework and the LLM/chatbot API, to efficiently process user requests and retrieve relevant information from a comprehensive database of educational materials. This implementation requires the development of a robust infrastructure capable of handling large volumes of user requests simultaneously. In addition, the model's decision-making model, which evaluates multi-criteria factors to select the optimal response, must be carefully implemented to ensure precision and adaptability to diverse educational content.

The practicality of the hybrid model is characterized by its ability to provide educators and students with a powerful tool for accessing and managing educational resources. Using advanced natural language processing techniques, the model provides precise and accurate responses to user queries, thereby enhancing the overall user experience. Its adaptability to diverse educational content ensures that it can handle a wide range of queries and requests, making it a valuable asset in modern educational settings. Moreover, the model's potential integration into e-learning platforms underlines its potential to enrich and enhance the interactive learning experience for students.

The use of the hybrid model in educational settings may present certain challenges, including the need for comprehensive training and support for educators and students to utilize the system effectively. Ensuring security and protection of sensitive data, such as API keys embedded in the source code, is paramount to safeguard against potential vulnerabilities. The scalability of the model to accommodate varying user loads and seamless integration with existing education platforms can be a challenge to implement. The adaptability of the model to diverse educational content requires careful consideration of the semantic and contextual nuances in the educational materials to ensure accurate responses.

The current model faces challenges when working with extremely large datasets due to memory limitations. Problems may arise when processing different educational materials of varying complexity and quality and when considering different formats such as text, images, PDFs, or videos. The current model only supports PDF formats, so it is crucial to develop adaptable techniques for processing and extracting information from different sources. The differences in the quality, precision, and relevance of information in teaching materials are a challenge that requires the attention of educators and students, as the code itself has difficulty controlling these parameters. On the other hand, the program has no problems working with different languages as it relies on an API that supports a wide range of languages.

Although the model supports working with different languages through the API, it is important to note that there may be potential difficulties in dealing with linguistic nuances and specificities that may vary in different cultural contexts.

The inclusion of academic journals can greatly enrich the model by providing access to the latest research in various disciplines. Adding relevant information to the database itself will further improve the results of the model. It is important to emphasize that the future expansion of the database depends on the users and their willingness, considering the local nature of the application. If the model were to go online, the dynamics would change significantly.

In order to improve the functionality of the model, the possibility of integration with external tools for recognizing language nuances is being considered, as well as improving the mechanisms for recognizing and processing different cultural contexts in teaching materials. As the user base grows, the speed of answering queries is a challenge, but the algorithms used are optimized to work with larger data sets. In the event of a possible slowdown of the program due to a large database, the implementation of caching techniques, i.e., the temporary storage of data to speed up access to this data in the future, is being considered.

It is important to emphasize that given the local nature of the application, there is no need to deal with scalability issues for the user. To ensure fast and efficient query processing, additional mechanisms are being developed to optimize the algorithms and possible improvements to the model itself are being explored to ensure a high level of performance. User feedback plays a crucial role in improving the model and fixing potential issues. As this is a local application, feedback needs to be collected through other channels, such as a website or a program mailing list.

Consideration is also being given to introducing additional methods of collecting user experience, including analyzing user interactions with the application to better understand user preferences and needs. The integration of new technologies can significantly improve the capabilities of the hybrid model, allowing it to better understand and process complex queries and provide personalized responses. Multimodal learning can extend the model to work with different types of data, including images without embedded text, audio, and video recordings. Implementing adaptive learning systems and voice search capabilities would further enrich the model's functionalities, enable personalized recommendations, and improve the overall user experience.

Improving the school system by integrating new technologies is an essential part of adapting education to the digital age. While applications like this are already a step in this direction, there are many innovative technologies that can be considered for further integration and improvement of the learning experience. Adaptive learning systems supported by machine learning algorithms open the doors for a personalized learning approach. These systems can analyze students' behavior and adapt the teaching material to their needs to create a unique learning experience for each individual. AI-powered personalized learning platforms provide tailored learning pathways and support students' individual needs. Mobile technologies enable access to educational resources at any time and from any place. Data and learning analytics provide valuable insights into student progress, allowing educators to adapt their approach and make informed decisions. By using cloud-based education tools, school systems can achieve collaborative learning and efficient storage of resources in the cloud. Together, these technologies can transform education and prepare students for the demands of today's digital society. Integrating these technologies is a key strategy to improve school systems and provide a relevant, interactive learning experience.

7. Conclusions

In an age when access to information is often scattered and uneven, centralization and customization become key components of an effective educational process. The implementation of artificial intelligence in education, especially through the use of algorithms such as OpenAI, opens up new horizons for improving the learning process. The developed application successfully addresses the challenge of decentralized educational materials by providing users with a central source of information that is simultaneously fast, reasonably accurate, and interactive. With its ability to provide answers from multiple sources, perform keyword-based searches and present visual content through images, this application becomes a valuable tool for anyone engaged in the learning process.

The integration of an API for an existing large language model (LLM)/chatbot plays a central role and has a profound impact on the overall functionality in the context of the study. The use of the transformer framework demonstrates a commitment to state-of-the-art natural language processing and ensures that the model exceeds the requirements of the various educational content.

Given current technological advances, applications such as these are likely to be essential in the future of education. Their potential lies not only in increasing learning efficiency, but also in potentially changing the approach to education itself. Considering the rapid advancement of artificial intelligence, the idea that future generations will have access to even more advanced and customized learning tools is a promising prospect.

Author Contributions: Conceptualization, D.B.; methodology, D.B., M.Š., D.J. and J.Ž.G.; software, D.B. and M.Š.; validation, D.B., M.Š. and J.Ž.G.; formal analysis, D.B., M.Š., D.J. and J.Ž.G.; investigation, D.B. and M.Š.; resources, D.B., M.Š. and D.J.; writing—original draft preparation, D.B. and M.Š.; writing—review and editing, D.J. and J.Ž.G.; visualization, M.Š.; supervision, D.B.; project administration, D.B.; funding acquisition, D.B. and J.Ž.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Financial Support of University of Zagreb "Optimization and personalization of multimedia content using artificial intelligence".

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Lee, G.; Hartmann, V.; Park, J.; Papailiopoulos, D.; Lee, K. Prompted LLMa ad Chatbot Modules for Long Open-domain Conversation. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics, ACL 2023, Toronto, ON, Canada, 9–14 July 2023. [CrossRef]
2. Mills, A.; Bali, M.; Eaton, L. How do we respond to generative AI in education? Open educational practices give us a framework for an ongoing process. *J. Appl. Learn. Teach.* **2023**, *6*, 16–30. [CrossRef]
3. Villanueva, D.P.P.; Aguilar-Alonso, I. A Chatbot as a Support for Educational Institutions. In Proceedings of the 62nd International Scientific Conference on Information Technology and Management Science of Riga Technical University, ITMS 2021, Riga, Latvia, 14–15 October 2021. [CrossRef]
4. Farah, J.C.; Ingram, S.; Spaenlehauer, B.; Lasne, F.K.-L.; Gillet, D. Prompting Large Language Models to Power Educational Chatbots. In Proceedings of the 22nd International Conference on Web-Based Learning, ICWL 2023, Sydney, Australia, 26–28 November 2023. [CrossRef]
5. Rahman, M.M.; Watanobe, Y. ChatGPT for Education and Research: Opportunities, Threats, and Strategies. *Appl. Sci.* **2023**, *13*, 5783. [CrossRef]
6. Vandamme, F.; Kaczmarski, P. Chatgpt: A Tool towards an Education Revolution? *Sci. Paedagog. Exp.* **2023**, *60*, 95–134. [CrossRef]
7. Lakshmi, G.; Brindha, S.; Revanya Devi, M.; Divya, J.; Shobhanali, N. AI-powered Digital Classroom. In Proceedings of the 2022 International Conference on Communication, Computing, and Internet of Things, IC3IoT 2022, Chennai, India, 10–11 March 2022. [CrossRef]
8. Sonderegger, S. How Generative Language Models can Enhance Interactive Learning With Social Robots. In Proceedings of the 19th International Conference on Cognition and Exploratory Learning in the Digital Age, CELDA 2022, Lisbon, Portugal, 8–10 November 2022.
9. El Azhari, K.; Hilal, I.; Daoudi, N.; Ajhoun, R.; Belgas, I. An Evolutive Knowledge Base for "AskBot" Toward Inclusive and Smart Learning-based NLP Techniques. *Int. J. Adv. Comput. Sci. Appl.* **2023**, *14*, 413–422. [CrossRef]
10. de Barcelos Silva, A.; Gomes, M.M.; da Costa, C.A.; da Rosa Righi, R.; Barbosa, J.L.V.; Pessin, G.; de Doncker, G.; Federizzi, G. Intelligent personal assistants: A systematic literature review. *Expert Syst. Appl.* **2020**, *147*, 113193. [CrossRef]
11. Floridi, L.; Chiriatti, M. GPT-3: Its Nature, Scope, Limits, and Consequences. *Minds Mach.* **2020**, *30*, 681–694. [CrossRef]
12. Okonkwo, C.W.; Ade-Ibijola, A. Chatbots applications in education: A systematic review. *Comput. Educ. Artif. Intell.* **2021**, *2*, 100033. [CrossRef]
13. Ismail, F.; Tan, E.; Rudolph, J.; Crawford, J.; Tan, S. Artificial intelligence in higher education. A protocol paper for a systematic literature review. *J. Appl. Learn. Teach.* **2023**, *6*, 56–63. [CrossRef]
14. Kasneci, E.; Sessler, K.; Küchemann, S.; Bannert, M.; Dementieva, D.; Fischer, F.; Gasser, U.; Groh, G.; Günnemann, S.; Hüllermeier, E.; et al. ChatGPT for good? On opportunities and challenges of large language models for education. *Learn. Individ. Differ.* **2023**, *103*, 102274. [CrossRef]
15. Niam Ibna Riza, A.; Hidayah, I.; Santosa, P.I. Use of Chatbots in E-Learning Context: A Systematic Review. In Proceedings of the 2023 IEEE World AI IoT Congress, AIIoT 2023, Virtual, 7–10 June 2023. [CrossRef]
16. Ganesan, M.; Deepika, C.; Harievashini, B.; Krithikha, A.S.; Lokhratchana, B. A Survey on Chatbot Using Artificial Intelligence. In Proceedings of the 2020 International Conference on System, Computation, Automation and Networking, ICSCAN 2020, Pondicherry, India, 3–4 July 2020. [CrossRef]
17. Lee, L.-K.; Fung, Y.-C.; Pun, Y.-W.; Wong, K.-K.; Yu, M.T.-Y.; Wu, N.-I. Using a Multiplatform Chatbot as an Online Tutor in a University Course. In Proceedings of the 2020 International Symposium on Educational Technology, ISET 2020, Bangkok, Thailand, 24–27 August 2020. [CrossRef]
18. Ha, M.; Nehm, R.H. Measuring Scientific Understanding Across International Samples: The Promise of Machine Translation and NLP-Based Machine Learning Technologies. In *Advancing Natural Language Processing in Educational Assessment*, 1st ed.; Yaneva, V., von Davier, M., Eds.; Routledge: New York, NY, USA, 2023; Volume 1, pp. 200–216.
19. Ji, Z. A Multi-modal Seq2seq Chatbot Framework. In Proceedings of the 2021 International Conference on Wireless Communications, Networking and Applications, Berlin, Germany, 17–19 December 2021; Lecture Notes in Electrical Engineering; Volume 942, pp. 225–233. [CrossRef]
20. Nguyen, H.N.; Teerakanok, S.; Inomata, A.; Uehara, T. The comparison of word embedding techniques in RNNS for vulnerability detection. In Proceedings of the 7th International Conference on Information Systems Security and Privacy, ICISSP 2021, Virtual, 11–13 February 2021. [CrossRef]
21. Hu, Z.; Dong, Y.; Wang, K.; Chang, K.-W.; Sun, Y. GPT-GNN: Generative Pre-Training of Graph Neural Networks. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2020, Virtual, 23–27 August 2020. [CrossRef]
22. Lauriola, I.; Lavelli, A.; Aiolli, F. An introduction to Deep Learning in Natural Language Processing: Model, techniques, and tools. *Neurocomputing* **2022**, *420*, 443–456. [CrossRef]
23. Bird, J.J.; Ekárt, A.; Faria, D.R. Chatbot Interaction with Artificial Intelligence: Human data augmentation with T5 and ensemble for text classification. *J. Ambient. Intell. Humaniz. Comput.* **2023**, *14*, 3129–3144. [CrossRef]

24. Radford, A.; Narasimhan, K.; Salimans, T.; Sutskever, I. Improving Language Understanding by Generative Pre-Training. Available online: https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf (accessed on 29 August 2023).
25. Nithyanandam, S.D.; Kasinathan, S.; Radhakrishnan, D.; Jebapandian, J. NLP for chatbot applications: Tools and techniques Used for Chatbot Applications, NLP Techniques for Chatbot, Implementation. In *Deep Natural Language Processing and AI Applications for Industry 5.0*, 1st ed.; Tanwar, P., Saxena, A., Priya, C., Eds.; IGI Global: Hershey, PA, USA, 2021; Volume 1, pp. 142–168. [CrossRef]
26. Abdilah, H.Z.; Partino; Madjid, A. Enhancing Student Well-being through AI Chat GPT in the Smart Education University Learning Environment: A Preliminary Review of Research Literature. In Proceedings of the 1st International Conference on Environment and Smart Society, ICEnSO 2023, Yogyakarta, Indonesia, 9–10 August 2023. [CrossRef]
27. Clarizia, F.; Colace, F.; Lombardi, M.; Pascale, F.; Santaniello, D. Chatbot: An Education Support System for Student. In Proceedings of the 2018 International Symposium on Cyberspace Safety and Security, Amalfi, Italy, 29–31 October 2018. [CrossRef]
28. Hiremath, G.; Hajare, A.; Bhosale, P.; Nanaware, R. Chatbot for Education System. *Int. J. Adv. Res. Ideas Innov. Technol.* **2020**, *4*, 37–43.
29. Park, Y.; Shin, Y. A Block-Based Interactive Programming Environment for Large-Scale Machine Learning Education. *Appl. Sci.* **2022**, *12*, 13008. [CrossRef]
30. Cain, W. Prompting Change: Exploring Prompt Engineering in Large Language Model AI and Its Potential to Transform Education. *TeachTrends* **2023**, *68*, 47–57. [CrossRef]
31. Laato, S.; Morschheuser, B.; Hamari, J.; Björne, J. AI-Assisted Learning with ChatGPT and Large Language Models: Implications for Higher Education. In Proceedings of the 2023 IEEE International Conference on Advanced Learning Technologies (ICALT), Orem, UT, USA, 10–13 July 2023. [CrossRef]
32. Jeon, J.; Lee, S.Y. Large language models in education: A focus on the complementary relationship between human educators and ChatGPT. *Educ. Inf. Technol.* **2023**, *28*, 15873–15892. [CrossRef]
33. Einarsson, H.; Lund, S.H.; Jónsdóttir, A.H. Application of ChatGPT for automated problem reframing across academic domains. *Comput. Educ. Artif. Intell.* **2024**, *6*, 100194. [CrossRef]
34. Li, X.; Henriksson, A.; Duneld, M.; Nouri, J.; Wu, Y. Evaluating Embeddings from Pre-Trained Language Models and Knowledge Graphs for Educational Content Recommendation. *Future Internet* **2024**, *16*, 12. [CrossRef]
35. de Fine Licht, K. Integrating Large Language Models into Higher Education: Guidelines for Effective Implementation. *Comput. Sci. Math. Forum* **2023**, *8*, 65. [CrossRef]
36. Birenbaum, M. The Chatbots' Challenge to Education: Disruption or Destruction? *Educ. Sci.* **2023**, *13*, 711. [CrossRef]
37. Sajja, R.; Sermet, Y.; Cwiertny, D.; Demir, I. Platform-independent and curriculum-oriented intelligent assistant for higher education. *Int. J. Educ. Technol. High. Educ.* **2023**, *20*, 42. [CrossRef]
38. Gao, Y.; Nuchged, B.; Li, Y.; Peng, L. An Investigation of Applying Large Language Models to Spoken Language Learning. *Appl. Sci.* **2024**, *14*, 224. [CrossRef]
39. Syed Mustapha, S.M.F.D. Predictive Analysis of Students' Learning Performance Using Data Mining Techniques: A Comparative Study of Feature Selection Methods. *Appl. Syst. Innov.* **2023**, *6*, 86. [CrossRef]
40. Oliveira, P.F.; Matos, P. Introducing a Chatbot to the Web Portal of a Higher Education Institution to Enhance Student Interaction. *Eng. Proc.* **2023**, *56*, 128. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.