IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 23 October 2024, accepted 13 December 2024, date of publication 16 December 2024, date of current version 26 December 2024.

*Digital Object Identifier 10.1109/ACCESS.2024.3518952*

RESEARCH ARTICLE

# Optimizing Language Model-Based Educational Assistants Using Knowledge Graphs: Integration With Moodle LMS

**WILLIAM VILLEGAS-CH, (Member, IEEE), JAIME GOVEA, AND ROMMEL GUTIERREZ**
Escuela de Ingeniería en Ciberseguridad, FICA, Universidad de Las Américas, Quito 170125, Ecuador
Corresponding author: William Villegas-Ch (william.villegas@udla.edu.ec)

**ABSTRACT** Chatbots in educational settings have grown significantly, facilitating interaction between students and learning platforms. However, current systems, such as Rasa, Moodle Integrated Chatbots, and ChatterBot, present significant limitations in precision, adaptability, and response time, affecting their effectiveness in resolving academic queries and personalizing learning. To address these shortcomings, this work proposes the development of an advanced educational chatbot that combines large language models (LLMs) with knowledge graphs, allowing for more accurate and contextualized responses and offering valuable suggestions to enrich the learning process. The system is evaluated based on its ability to adjust to different student profiles and offer fast and accurate responses. The results show that the proposed chatbot achieves a precision of 85%, outperforming Rasa and ChatterBot, which achieved accuracies of 83% and 81%, respectively. Furthermore, the chatbot reduces response times to 0.41 seconds, improving efficiency compared to other solutions. The system also demonstrates adaptability, effectively adjusting to students' learning styles and academic levels. This work indicates that knowledge graph integration and hyperparameter optimization are crucial to improving educational chatbots' precision, speed, and adaptability, presenting an innovative solution that overcomes the limitations of current systems.

**INDEX TERMS** Educational chatbots, large language models, knowledge graphs, learning personalization.

## I. INTRODUCTION

In recent years, artificial intelligence (AI) has revolutionized multiple sectors, including education [1]. Using natural language models and conversational systems like chatbots has been vital to automating tasks, improving student engagement, and personalizing learning in digital environments [2]. However, many existing solutions have limitations regarding precision, response times, and the ability to adapt to diverse student profiles. The presented research addresses these shortcomings by developing and optimizing an educational chatbot based on advanced language models (LLMs) and knowledge graphs to generate more accurate and contextually rich responses [3].

The associate editor coordinating the review of this manuscript and approving it for publication was Rajeswari Sundararajan.

In today's educational environment, more dynamic and adaptable solutions are needed to enable students to access relevant and accurate information quickly. Tools such as Rasa, Moodle Integrated Chatbots, and ChatterBot have been implemented in different educational environments. Still, all of them face challenges in terms of adaptability, precision in response generation, and the ability to deliver personalized learning [4], [5], [6]. For example, although Rasa is flexible, it requires considerable customization to fit educational environments. It does always reach the precision levels needed to answer complex questions in these contexts [7]. Chatbots integrated into Moodle have limited functionalities and are focused mainly on administrative tasks without the ability to provide valuable suggestions that guide the learning process. ChatterBot, on the other hand, suffers from longer response times and limited adaptability, making it less efficient in adjusting to different academic profiles [8].

Developing a more advanced educational chatbot is crucial to accurately resolving academic queries, dynamically adjusting to students' needs, and offering valuable suggestions that enrich the learning process. The proposed chatbot combines the capabilities of advanced language models with a knowledge graph-based approach, allowing it to organize and structure information more effectively, relating key concepts and providing a personalized experience [9]. This combination improves the precision and relevance of responses, which are essential features in an educational environment. The development of the chatbot responds to the need for more adaptable, accurate, and fast systems in learning management platforms, such as Moodle, which educational institutions around the world use [10]. While Moodle has integrated chatbots to facilitate interaction between students and teachers, current solutions fail to offer comprehensive support in the learning process. Previous studies have shown that students, especially those in remote learning environments, value immediacy and precision in resolving academic doubts [11]. Therefore, a chatbot that offers personalized and immediate responses can significantly impact the quality of learning, providing efficient access to relevant information.

This study aims to overcome the limitations of existing solutions, offering a comprehensive tool that, in addition to resolving academic doubts, improves the educational experience through adaptability and personalization. The chatbot's ability to adjust to academic profiles and offer valuable suggestions and additional material is one of its main advantages over current solutions. It is sought to demonstrate that the implementation of an advanced educational chatbot based on knowledge graphs and hyperparameter optimization techniques can significantly improve the precision and efficiency of current systems [12].

The methodology to achieve these objectives consisted of developing a chatbot that integrated with Moodle and used a hybrid approach, combining knowledge graphs with an AI-based language model [13]. The development includes several stages, starting with acquiring educational data from various subjects in Moodle, which are preprocessed and structured into knowledge graphs. Subsequently, the hyperparameters of the language model are optimized to improve precision and reduce response times, ensuring that the system works efficiently in real-time. The system is evaluated according to its ability to adapt to different student profiles, considering factors such as academic year, learning style (visual, auditory, kinesthetic), and complexity of queries. Key metrics such as precision, speed of responses, and usefulness of suggestions were measured.

The results showed that the proposed chatbot achieved a precision of 85%, outperforming solutions such as Rasa and ChatterBot, which gained 80% and 78% precision, respectively. Furthermore, the system reduced response times to an average of 0.35 seconds, significantly improving over Rasa's 0.50 seconds and Moodle's chatbots' 0.55 seconds. The system was also highly adaptable, adjusting its responses to the student's profile based on their learning style and academic level – something other solutions cannot achieve with the same efficiency.

## II. LITERATURE REVIEW

The integration of chatbots in educational environments has advanced considerably, with a particular focus on improving student-system interaction and personalizing learning [14]. Various open-source and academic solutions have been explored to facilitate educational interaction [15]. However, many of these solutions present significant limitations regarding adaptability, precision, and response times, which directly impact the user experience.

One of the most widely used systems in open-source environments is Rasa, a framework that allows building personalized chatbots using natural language processing (NLP) and machine learning techniques [16]. Although it is flexible and used in various sectors, including education, its ability to adapt to specific student profiles is limited. Rasa offers a modular architecture and allows for customized integrations. Still, its precision tends to be lower in specialized educational environments, where students' questions require a deeper understanding of the academic environment [17]. Furthermore, although suitable for many conversational environments, their response time is less efficient than that of more optimized systems, resulting in less fluid interaction when implemented in educational platforms.

Moodle, one of the most popular learning management environments, has integrated chatbots with essential capabilities to facilitate student interactions [18]. Native implementations of chatbots in Moodle, however, are limited in terms of functionalities. These integrated solutions are designed to answer general questions and offer technical support to students but are not optimized to suit individual learning styles or needs [19]. The ability of these chatbots to provide additional information or valuable suggestions is practically non-existent, limiting their usefulness as learning tools.

On the other hand, ChatterBot, an open-source chatbot solution also used in educational settings, offers a more structured approach regarding interaction and response. Still, its precision highly depends on the quality and quantity of data it is trained with [20]. While it can adapt to various environments, its limitations in customization for specific educational settings and its response time put it behind more optimized systems.

The Proposed Chatbot stands out by addressing the limitations observed in other systems. Unlike Rasa or ChatterBot, the chatbot developed in this study has been optimized to reduce response times and increase precision in complex responses, dynamically adapt to student profiles, and offer valuable suggestions that enrich the learning process. This comprehensive approach places the system in an advantageous position compared to the reviewed solutions, demonstrating that technical improvements and targeted integration into the educational environment provide significant added value.

## III. MATERIALS AND METHODS

### A. DEPLOYMENT ENVIRONMENT

The educational chatbot implementation environment was designed with a technological infrastructure to ensure efficient integration between the LLM language model and the Moodle LMS. The implementation was carried out on a high-performance dedicated server that supports processing large volumes of data and running artificial intelligence models in real-time [21].

Moodle was selected as the learning management system due to its flexibility and extensibility through APIs and plugins. Through its REST API, Moodle allows native integration with external tools, facilitating communication between the platform and the chatbot. Custom drivers were implemented to ensure integration, enabling direct interaction with the LLM and the knowledge graph.

A server with the following characteristics was used: an Intel Xeon multi-core processor, 128 GB of RAM, and 2 TB SSD storage to maximize data access speed and improve the LLM's processing capacity. The server runs on a Linux environment (Ubuntu 20.04 LTS), which ensures stability and long-term support for the tools used.

The integration of the chatbot with the LLM was carried out using technologies such as Python (version 3.9) and TensorFlow (version 2.9) to implement the language model. In addition, graph processing libraries such as NetworkX were used to construct and manage the knowledge graph; for the communication between Moodle and the chatbot, a Docker-based microservices architecture was used, which allowed scalability and independent deployment of each component.

In this environment, the interaction data between students and the Moodle platform were managed in real-time, feeding the knowledge graph and the language model to generate personalized responses [22]. Table 1 presents the technological specifications used for the development of this work.

The system is organized in a microservices architecture where each component (LLM, knowledge graph, Moodle) interacts through a central API. The corresponding section will include a block diagram detailing this architecture to visualize the communication and data flows between the components. This deployment environment ensures that the chatbot can scale efficiently and process student interactions in real-time while maintaining a smooth integration with the LMS platform Moodle.

### B. SYSTEM ARCHITECTURE

The chatbot system architecture is based on a robust integration of three main components: the LLM, the knowledge graph, and the Moodle LMS platform. Each element is connected by a modular infrastructure designed to optimize the flow of information and generate highly accurate and contextual educational responses.

The process begins when the student interacts with the Moodle platform, which provides an interface through which

**TABLE 1. Technological environment specifications.**

| Element | Description | Technology / Version | Purpose | Capacity | Additional Configuration |
|---|---|---|---|---|---|
| Server | Intel Xeon processor, 128 GB RAM, 2TB SSD | Ubuntu 20.04 LTS | Processing AI models | High performance | Virtualization with Docker |
| LMS Platform | Moodle | Version 3.11 | Learning management system | Multi-user support | REST API for integration |
| LLM | TensorFlow implementation | Version 2.9 | Generating educational responses | Parallel processing | Support for GPU inference |
| Knowledge Graph | NetworkX | Version 2.6 | Structuring relationships between concepts | Educational data | Graph management API |
| Micro services Architecture | Docker | Version 20.10 | Deployment and scalability | Standalone execution | Orchestration with Kubernetes |

users ask questions or request educational assistance. These queries are received by an API explicitly developed to connect Moodle with the core of the chatbot system. The API acts as an intermediary, accepting the student's requests and managing their distribution to the appropriate modules, such as the language model and the knowledge graph. This interaction layer allows Moodle to maintain a constant flow of data to the chatbot, ensuring that the system has all the relevant information about the student's context, such as the course they are enrolled in, the educational content they have previously consulted, and their academic progress.

Once the API receives the query, the system consults the knowledge graph designed to structure educational information semantically. This graph stores the fundamental concepts of the different subjects and models the relationships between these concepts, allowing the system to access interconnected information that enriches the responses generated. The system can select the most accurate and contextual information for the student's question by identifying the student nodes within the graph. Each node represents a concept or topic within the educational structure, and the edges of the graph indicate the relationships between them, allowing navigation between different subjects efficiently and accurately. The knowledge graph is dynamically updated as new data is incorporated, ensuring that the system's knowledge structure reflects the latest advances in educational content and student evolution.

Once it has received the context structured by the knowledge graph, the language model generates the adaptive educational response. Unlike a system based on unstructured data alone, the LLM leverages information from the graph to improve the precision and relevance of its answers. The model uses advanced NLP techniques to interpret both the student's queries and the data provided by the student and the graph, generating a coherent response that answers not only the student's direct question but also additional guidance on related concepts [23]. This ensures that the system not only answers immediate questions but also promotes deeper learning by suggesting content or conceptual relationships that might have gone unnoticed by the student.

Once the response is generated, the system returns it to Moodle through the chatbot API, which presents it to the student. Moodle acts as the final presentation platform, ensuring that the response is available in an easily accessible and understandable format. Furthermore, the response can be enriched with links to course materials or additional suggestions based on the system's contextual analysis using the knowledge graph. The architecture, represented in Figure 1, shows how each component interacts fluidly and efficiently to generate adaptive educational responses. The figure shows the flow of information from the entry of queries in Moodle through the chatbot API, the interaction with the knowledge graph, and finally, the processing of the response by the LLM until its delivery to the student. This block diagram provides a clear view of the interaction of the components, highlighting the importance of integrating the knowledge graph and the LLMs to improve the quality of the responses and optimize the educational experience within the Moodle environment.

### C. DATA MANAGEMENT

Data management in the educational chatbot system is essential to ensure the precision and relevance of the responses generated by the language model and the knowledge graph. Correct acquisition, preprocessing, and data structuring ensure the system can efficiently interpret, contextualize, and organize information from different sources. These sources include both educational resources provided by Moodle and students' previous interactions with the system. Unstructured data, such as course content and student queries, must be transformed into formats that allow their integration into a knowledge graph, facilitating semantic navigation and the generation of adaptive responses by the language model. This data management process involves multiple technical stages, from data acquisition to structuring into graphs, ensuring a solid foundation for generating and managing educational knowledge.

#### 1) DATA ACQUISITION

Data acquisition to feed both the language model and the knowledge graph is designed to ensure a complete and accurate representation of educational resources, student interactions, and data related to academic progress. This process is carried out in several stages, starting with data collection directly from Moodle, where students access courses and study materials. The data obtained includes structured information about course content, such as teaching materials (documents, videos, and multimedia resources), and data associated with student performance, such as exam grading, assignment submission, and participation patterns in discussion forums. These data are complemented by student interaction records, including queries made to the chatbot, the responses obtained, and response times, allowing for continuous monitoring of the educational assistant's use.

The selected study population comprises students from different academic levels, from secondary to university education and continuing education programs. At the faculty, approximately 1000 students are registered on the Moodle platform, and they access various subjects distributed across multiple areas of knowledge. To ensure that the system's results adequately reflect students' actual interaction and performance, a representative sample of 200 students is chosen. This population was selected to ensure that the system is evaluated under representative and manageable conditions. Limiting the sample to 200 students allows for a more detailed data analysis, which optimizes the chatbot system's performance without compromising its scalability. The students in this population span diverse learning styles, levels of understanding, and academic performance, providing the variability needed to assess the system's adaptability to different educational contexts.

The sample size also ensures that the system is evaluated across different subject areas, from scientific to humanistic subjects. This ensures the knowledge graph covers a broad spectrum of conceptual relationships and semantic connections. Including this variety in the population allows the system to adjust the responses generated by the chatbot according to each student's profile, which is essential for the success of the educational assistant. In addition, using a controlled sample facilitates the implementation of feedback mechanisms that adjust both the language model, and the knowledge graph based on the actual performance of the students and the effectiveness of the responses provided.

Several monitoring mechanisms were implemented to verify the integrity and consistency of the data from Moodle to ensure the quality of the acquired data. For example, it is verified that the course materials are fully indexed and that the student interactions are correctly recorded. This verification is essential to prevent incomplete or inconsistent data from affecting the precision of the responses generated by the chatbot. In addition, it ensures that the course data is regularly updated to reflect changes in educational content or assessment methods as instructors update their courses.

As for the students' interactions with the chatbot, this data is semi-structured. Each query asked is stored along with the context provided by Moodle, which includes information on the specific course to which the student belongs, as well as their progress up to that point. This data type allows the knowledge graph to properly structure the relationship between the concepts addressed in the students' questions, facilitating the generation of personalized and contextually appropriate responses. Furthermore, data acquisition also includes collecting additional external information, such as academic publications, journal articles, and other online resources that enrich the content of the knowledge graph. This

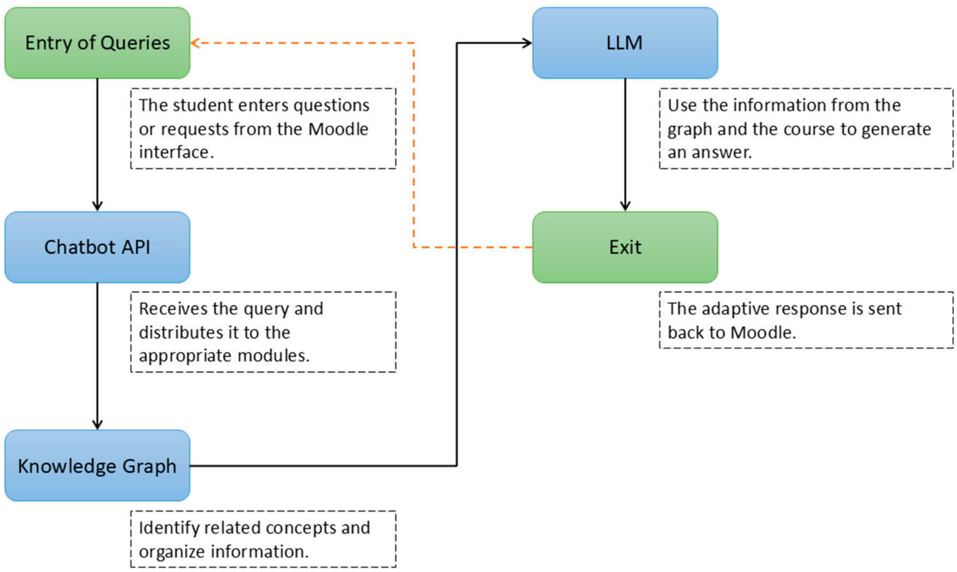


**FIGURE 1.** Architecture of the educational Chatbot system integrated with LMS Moodle, LLM and knowledge graph.

inclusion of external sources allows for a greater breadth of available knowledge, offering the system a broader database to generate accurate and up-to-date answers on various educational topics.

### 2) DATA PREPROCESSING

Data preprocessing in the system ensures that information acquired from different sources, such as Moodle and student interactions, is in a format suitable for use by both the knowledge graph and the language model. Since data can be highly varied, coming from multiple subjects, educational levels, and formats (such as text, images, or videos), it is necessary to apply a series of cleaning, transformation, and normalization techniques to ensure its integrity and usability.

Data cleaning involves removing inconsistencies, null values, and duplicates that may affect the model's and the graph's precision. A commonly used technique in this process is missing value imputation, where null values are replaced by the mean or median of the data set. This approach is represented mathematically as follows:

$$x_i' = \frac{1}{n}\sum_{j=1}^{n} x_j \quad \text{if } x_i \text{ is null} \tag{1}$$

$x_i'$ is the imputed value for the missing data, and $x_j$ is the non-null value in the corresponding dataset. This method ensures that missing information does not degrade the model's performance by introducing irrelevant or incorrect values.

Another essential cleaning technique is removing outliers, which can distort the model's behavior. Methods such as z-score are used to detect data that is outside an acceptable range. The calculation of the z-score is defined as:

$$z = \frac{x - \mu}{\sigma} \tag{2}$$

where $x$ is the observation value, $\mu$ is the mean of the dataset, and $\sigma$ is the standard deviation. Values with a z-score more significant than a predefined threshold (usually 3) are considered outliers and can be removed or adjusted.

The acquired data are often in formats not directly processable by graphs or the language model. For this reason, it is essential to apply appropriate transformation techniques. A critical approach is tokenization for texts from courses and interactions with the chatbot. This process converts the text into sequences of tokens that the language model can then process [24]. Tokenization follows the following formal scheme:

$$T = \{t_1, t_2, \ldots, t_n\} \quad \text{where } t_i \text{ is the i-th token of the text}$$

This set of $T$ tokens is fed into the language model, which subsequently generates the answers based on this structured representation. In addition, lemmatization is performed to reduce the words to their base or root form, which improves the consistency in the semantic representation of the concepts within the knowledge graph.

Normalization is essential to ensure that the data presented in the language model and the knowledge graph are on comparable scales. This applies to both numerical and textual data. For numerical data, Min-Max normalization is used, which transforms the values within a defined range, usually between 0 and 1:

$$x_i' = \frac{x_i - \min(x)}{\max(x) - \min(x)} \tag{3}$$

This method ensures that different data sets with varying numerical values (e.g., student grades or interaction times) are within the same range, making it easier to compare and analyze later. Text normalization, including lowercase conversion, memorable characters' removal, and space normalization, is employed for textual data. These techniques are crucial to reducing noise in text processing and ensuring that models are not affected by unnecessary variations in textual data.

The transformed and normalized data are structured in a format that allows for efficient use in knowledge graphs and language models. For graphs, an encoding represents nodes as key concepts in subjects, while edges represent their relationships. For the language model, data is encoded using word embeddings, representing words in a vector space where the semantic similarity between words is encoded in the distances between their vectors.

### 3) STRUCTURING DATA IN GRAPHS

Structuring data into graphs is essential to ensure that the educational chatbot system can efficiently organize, relate, and navigate the vast educational information from multiple subjects, academic levels, and thematic modules [25]. Using a knowledge graph allows concepts to be represented explicitly. It facilitates the semantic and contextual analysis of the relationships between these concepts, improving the language model's generation of adaptive and contextualized responses [26].

In the knowledge graph, nodes represent critical concepts extracted from educational materials, while edges represent the semantic relationships between these concepts. Each subject, topic, and subtopic are modeled as a node, and connections between them, such as thematic dependencies or logical learning sequences, are represented by edges. Mathematically, this graph can be defined as a directed graph:

$$G = (V, E) \tag{4}$$

where $V$ is the set of nodes (concepts), and $E$ is the set of edges describing the relationships between these concepts.

For each node $v_i \in V$, an educational concept is assigned, representing a central or subtopic within a subject. The edges $e_{ij} \in E$ between the nodes $v_i$ and $v_j$ represent a pedagogical relationship. For example, if one node represents the concept of "linear algebra" and another represents "vector spaces," the edge between them could represent a dependency relationship, where knowledge of linear algebra is a prerequisite for understanding vector spaces.

$$\begin{aligned} G &= (V, E) \quad \text{where} \\ V &= \{v_1, v_2, \ldots, v_n\}, \\ E &= \{(v_i, v_j) \mid v_i \text{ is related to } v_j\} \end{aligned} \tag{5}$$

The knowledge graph structures dependency relationships and includes semantic relationships such as equivalences, hierarchies, and associative connections between concepts [27]. These relationships are categorized and assigned a weight based on their relevance for generating answers. The weight of the edges is defined as $w_{ij}$, which indicates the strength or semantic proximity between two concepts.

For example, a hierarchical relationship between a general concept and its subtopics may have a high weight $w_{ij} \approx 1$, indicating that a student who understands the central idea is more prepared to tackle the related subtopics. In contrast, a more tangential or associative relationship, such as that between linear algebra and geometry, may have a lower weight $w_{ij} \approx 0.5$.

$$w_{ij} = \text{semantic proximity between } v_i \text{ and } v_j \tag{6}$$

The calculation of $w_{ij}$ can be based on the number of interactions students have had with both concepts and the co-occurrence of terms in the educational materials provided by Moodle.

The knowledge graph also facilitates information propagation, allowing the system to extract additional, relevant information about related concepts, even if they are not explicitly mentioned in the student's query. This information propagation is modeled using random walking techniques over the graph. A random walk in the graph is defined as a stochastic process where, starting from a node $v_i$, an edge is randomly selected to move to the next node $v_j$ according to its weight:

$$P(v_j \mid v_i) = \frac{w_{ij}}{\sum_k w_{ik}} \tag{7}$$

This process ensures the system can explore the most relevant adjacent nodes and obtain complementary information about related concepts. Random walks enrich the chatbot's responses with additional context that may be useful to the student, promoting deeper and broader learning.

Concepts and their relationships are encoded in numerical vectors using embedding techniques for graph nodes. This allows the graph to be effectively integrated with the language model, facilitating the combination of structured and unstructured semantic representations. The most representative technique used in this process is Node2Vec, which generates embeddings for each node in the graph by simulating multiple random walks over the graph and maximizing the probability that nearby nodes have similar embeddings:

$$\max_{\theta} \sum_{v_i \in V} \log P(N(v_i) \mid v_i; \theta) \tag{8}$$

where $N(v_i)$ represents the neighboring nodes of $v_i$, and $\theta$ are the parameters of the embeddings. The result is a dense representation in vector space that captures both the semantic proximity between concepts and the structural relationships in the graph.

The knowledge graph is continuously optimized by analyzing students' interactions with the system. The graph can be reconfigured to reflect semantic and pedagogical relationships better as the system collects more data about generated queries and answers. This reconfiguration includes creating new edges between previously unrelated concepts or updating the weights $w_{ij}$ to improve the system's precision.

### D. CHATBOT INTEGRATION WITH MOODLE

Integrating the educational chatbot with Moodle's LMS platform is carried out through a modular architecture that ensures fluid and efficient communication between both systems. This allows the chatbot to access course data, interact with students, and generate contextual responses based on educational content. This integration is done through REST APIs, custom controllers, and containerization tools to ensure the system's scalability, security, and upgradeability.

#### 1) MOODLE REST API

The integration's foundation is the Moodle REST API, which allows the chatbot to perform operations such as obtaining course data, retrieving student progress, and sending tailored responses to the Moodle interface [27], [28]. Through this API, the chatbot can programmatically request information from Moodle, using HTTP requests (GET, POST, PUT) to interact with the platform's resources.

Each request to the API is structured using token authentication, ensuring that standard security measures protect communications between the chatbot and Moodle. Authentication tokens are generated per user and associated with roles within Moodle, allowing the chatbot to customize responses based on each student's profile and access level. An example of a GET request to get course data is:

GET/moodle/webservice/rest/server.php?wstoken={token}&wsfunction={function}&courseid={id} &moodlewsrestformat=json

This request allows you to get specific information about a course using the identifier `{id}` and the function `{function}` defined in the Moodle API, such as getting the course materials or the available exams.

#### 2) MICROSERVICES ARCHITECTURE

A microservices architecture based on Docker containers ensures that the chatbot integration with Moodle is scalable and maintainable. Each system component, such as the chatbot, the language model inference engine, and the knowledge graph processing modules, runs in its container [29]. This ensures the independence of each module, allowing for upgrades and scalability without affecting the system's overall operation.

The orchestration of these containers is done through Kubernetes, which is responsible for managing the deployment and distribution of resources based on system demand. Kubernetes allows the creation of pods that group the chatbot containers and their associated services, such as API controllers and data analysis modules [30]. This guarantees high availability and dynamic load balancing, ensuring the chatbot can handle multiple student requests concurrently without suffering performance degradation.

The integration also requires custom controllers to connect the data flow between Moodle and the chatbot. These controllers act as intermediaries between Moodle and the chatbot's inference engine, handling business logic and data preprocessing before sending queries to the language model [31]. The controllers are implemented in Python using Flask, a lightweight framework for handling incoming and outgoing HTTP requests.

Every time a student interacts with Moodle, the custom controllers capture the context of the query, such as the current course, the materials consulted, and the student's progress. This context is crucial for the chatbot to generate personalized responses. The controllers are also responsible for normalizing and validating requests before they reach the inference engine, ensuring the system operates robustly and securely.

#### 3) TWO-WAY REAL-TIME COMMUNICATION

Communication between Moodle and the chatbot is not limited to synchronous requests via the REST API but also includes WebSockets to enable real-time, two-way interaction between the student and the chatbot. WebSockets allow the chatbot's responses to be sent immediately to the Moodle interface without reloading the page, significantly improving the user experience.

WebSockets are essential in situations where the student needs to perform multiple interactions with the chatbot, such as in tutoring or ongoing consultation sessions [32]. Furthermore, the asynchronous nature of WebSockets allows the chatbot to process complex queries in the background and send the response when it is ready, improving overall system performance.

A relational data storage system based on PostgreSQL handles student and chatbot interactions. This system stores records of the chatbot's interactions with students and associated metadata such as the time of the query, the course to which the student belongs, and the responses generated. This data is essential for performing subsequent analyses on the system's behavior and the effectiveness of the responses generated by the language model.

The storage system is integrated with Moodle through database connectors, allowing complex queries to be made on student interactions without affecting the platform's performance. In addition, caching mechanisms using Redis are employed to improve the performance of frequent queries, allowing the chatbot to quickly access relevant information, such as student progress, in real time.

#### 4) SECURITY AND ACCESS CONTROL

Integrating the chatbot with Moodle includes advanced security measures to protect student data and communications between the systems [33]. In addition to token authentication, SSL/TLS encryption is implemented in all communications between the chatbot and Moodle, ensuring that the information exchanged is protected against unauthorized access [34]. Role-based access control mechanisms ensure only authorized users, such as students and teachers, can interact with the chatbot and access relevant data.

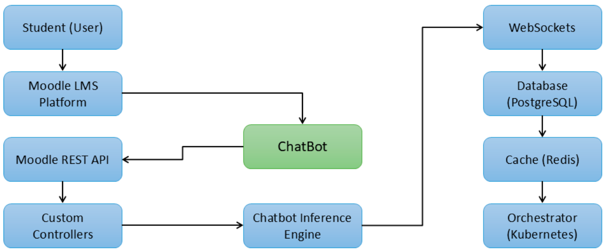


**FIGURE 2.** **Chatbot integration architecture with Moodle.**

The integration architecture, shown in Figure 2, describes the complete flow from the student interaction with Moodle through the custom controllers to the chatbot inference and the response generated in real time. It uses REST API and WebSockets to ensure smooth and efficient communication. This technical integration allows the system to operate robustly and scalably, adapting to the needs of the students and offering an enhanced educational experience using artificial intelligence within the Moodle LMS environment.

### *E. GENERATION OF RESPONSES BASED ON GRAPHS AND LLM*

The response generation process is based on a complex integration between the knowledge graph, which organizes educational information semantically, and the LLM, responsible for generating coherent and contextually accurate responses [35]. This system allows responses to respond to the student's specific query and integrate additional information that helps deepen learning.

The process begins when a student makes a query through Moodle. The chatbot receives this query through the custom controllers and processes it to identify the key concepts in the question. The system uses the knowledge graph to identify relevant nodes associated with those concepts. The graph contains structured representations of the different subject topics and the semantic relationships between them, allowing the system to determine the appropriate context of the question.

Once the relevant nodes are identified in the graph, the query is enriched with the information associated with those nodes, including the relationships with other concepts. This information is passed to the LLM, which generates a coherent answer based on the initial question and the additional context extracted from the graph.

The LLM generates answers based on the syntactic and semantic structure of the question. It uses the information from the graph to ensure that the answers are aligned with the correct academic content. This combination allows the system to offer adaptive answers, i.e., answers that not only solve the current question but also guide the student towards related concepts that can improve their understanding of the topic. The generated answer is sent back to Moodle in real time, allowing the student to continue fluidly interacting with the system.

From a technical point of view, the answer-generation process begins with representing the nodes and edges in the knowledge graph. Each key concept $v_i$ in the graph is represented by a feature vector $\mathbf{v}_i$, using embedding techniques. These embeddings allow the graph to be processed by the LLM, which also works in a similar vector space. The semantic relationship between two concepts $v_i$ and $v_j$ is quantified by the weighting of the edges between them, denoted as $w_{ij}$.

Calculating the relevance between two nodes in the graph, given a specific concept in the student's query, is done using similarity metrics, such as cosine similarity:

$$\mathrm{sim}(v_i, v_j) = \frac{\mathbf{v}_i \cdot \mathbf{v}_j}{\|\mathbf{v}_i\| \|\mathbf{v}_j\|} \tag{9}$$

This similarity value helps identify the closest or most relevant concepts that can enrich the model's answer. Furthermore, relationships between nodes can be propagated through the graph using weighted random walks, allowing the system to explore adjacent concepts and enrich the answer.

Once the relevant nodes have been identified and sufficient context has been extracted, the LLM uses this structured data to generate an answer. The LLM is trained to handle unstructured data (such as the student's question) and structured data (such as the nodes' embeddings in the graph). Formally, the generation of the answer can be modeled as the maximization of the conditional probability $P(r \mid q, G)$, where $r$ is the generated answer, $q$ is the student's query, and $G$ is the knowledge graph.

$$P(r \mid q, G) = \prod_{t=1}^{T} P(r_t \mid r_{1:t-1}, q, G; \theta) \tag{10}$$

where $r_t$ is the word generated in step $t$, $r_{1:t-1}$ are the previously generated words, and $\theta$ are the parameters of the LLM model. The graph $G$ provides the necessary context, and the query $q$ drives the answer generation. Using embeddings for the graph nodes and the LLM words allows both systems to work together coherently. This ensures that the LLM generates a grammatically correct response that is informed and contextualized within the educational domain.

As the system receives more queries, the weights on the graph edges and the node embeddings are continuously adjusted, improving the precision and relevance of the answers [35]. Feedback from students on the quality of the generated answers is also used to adjust the LLM parameters and update the knowledge graph structure, making the system dynamic and adaptive.

This combination of structured (graph) and unstructured (LLM) data processing allows the educational chatbot to offer a rich and personalized learning experience. Answers solve specific questions and promote deeper learning by suggesting related concepts and topics.

### *F. SYSTEM EVALUATION*

The educational chatbot's evaluation is based on a series of quantitative and qualitative methods that measure the system's effectiveness in terms of the precision of the responses, adaptability to different student profiles, and ability to guide the learning process optimally. These metrics are calculated based on interactions recorded with the students and the quality of the responses generated by the language model and the knowledge graph.

#### 1) PRECISION OF RESPONSES

The precision of the responses generated by the chatbot is assessed by comparing the responses provided by the system with the correct responses expected in a validation dataset. This dataset comprises predefined educational FAQs and problems that experts in different subject areas have validated. Precision is measured using standard natural language processing metrics such as precision and F1 score. Precision (AAA) is defined as the proportion of correct responses generated by the chatbot concerning the total expected responses:

$$A = \frac{\text{Correct Responses}}{\text{Total Expected Responses}} \tag{11}$$

On the other hand, the F1 score ($F_1$) combines precision and recall to offer a more balanced measure of system performance:

$$F_1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{12}$$

This evaluation allows us to measure the system's capacity to generate correct responses in absolute terms and answer open-ended questions where there may be multiple correct answers.

#### 2) ADAPTABILITY TO DIFFERENT STUDENTS

Analyzing individual interactions assesses the chatbot's ability to adapt to different student profiles. Particular attention is paid to how the system adjusts responses based on academic level, progress in the course, and learning patterns observed in each student. This analysis uses the personalization of content provided by the knowledge graph and the LLM.

Adaptability is measured by observing how the system adjusts responses based on the student's context, assessing factors such as variation in the depth of responses. An important metric here is the response entropy, which measures the diversity in the content generated by the chatbot depending on the student's characteristics:

$$H(R) = -\sum_{i=1}^{n} P(r_i) \log P(r_i) \tag{13}$$

where $r_i$ is the response generated based on the student's profile, and $P(r_i)$ is the probability that this response is generated in a similar context. Higher entropy indicates that the system can provide varied responses tailored to the student's profile, improving personalization.

#### 3) ABILITY TO GUIDE LEARNING

The system's ability to guide the learning process is assessed by observing how the chatbot proposes suggestions or expands on concepts not explicitly present in the student's initial query. This component is measured through the proportion of enriched queries, i.e., queries where the chatbot offers an answer that addresses the original question and includes additional information relevant to the student's progress.

Mathematically, this can be modeled as the number of interactions where the knowledge graph has generated an extended path that includes related concepts:

$$P(E) = \frac{\text{Enriched Queries}}{\text{Total Queries}} \tag{14}$$

A rich query is defined as one where the knowledge graph proposes a sequence of additional nodes that guide the student towards broader learning. This reflects the system's ability to answer questions and provide active educational guidance.

The system is evaluated using direct feedback from students. After each interaction, students can rate the helpfulness of responses on a Likert scale, providing a subjective measure of user satisfaction. This data is used to adjust the weights on the edges of the knowledge graph and to improve the behavior of the LLM through continuous learning.

The system employs reinforcement learning algorithms to adjust its parameters based on this feedback, improving its precision and adaptability. This approach ensures that the chatbot is effective during deployment and continues to evolve and improve based on actual student interactions.

#### 4) HYPERPARAMETER OPTIMIZATION

Hyperparameters control critical aspects of the model's behavior, such as its generalization ability, response precision, and adaptability to different student profiles. To ensure that the system performs optimally, a hyperparameter tuning process is implemented using cross-validation and search techniques.

In the case of the language model, the following hyperparameters are tuned to maximize response precision and minimize error:

- Learning rate ($\alpha$): Controls the speed at which the model adjusts its weights during training. Too high a learning rate can cause oscillations in performance, while too low a rate can lead to a too-slow convergence process [36]. The optimal learning rate is found by grid search, testing values in the range $10^{-5} \leq \alpha \leq 10^{-2}$.
- Number of layers and neurons per layer: The size and depth of the LLM are critical to its ability to model complex patterns in language. The number of hidden layers and neurons per layer is tuned using cross-validation, evaluating the trade-off between generalization ability and overfitting.
- Sequence size: Controls the number of tokens the model considers in each prediction. Tuning this hyperparameter allows the system to capture broader

relationships in the text, increasing the quality of the generated answers.

- Dropout (dpi): This regularizes the model by avoiding overfitting. Dpi is adjusted (usually between 0.1 and 0.5) to reduce neuron co-adaptation, improving the model's ability to generalize to new queries.

In the knowledge graph, hyperparameters are critical to the way educational concepts are represented and related:

- Dimensionality of embeddings: The Node2Vec technique generates node embeddings, and a crucial hyperparameter here is the dimensionality of those embeddings. A low-dimensional space may not capture complex relationships, while an excessively high one may lead to redundancy. The optimal dimensionality is generally between 64 and 256.
- Length of random walks: In constructing random walks over the graph, the length of these walks affects the semantic context the system can capture. A longer length allows for exploring more related nodes but at the cost of including less relevant concepts. It is optimized by trying lengths between 10 and 40 steps.
- Number of walks per node: This hyperparameter controls how many walks are performed from each node. Increasing the number of walks can improve the capture of weaker relationships between concepts, but it also increases the computational cost. This hyperparameter is tuned between 10 and 50 walks per node.
- Parameters $p$ and $q$ for walking with bias: These parameters control the preference of walks towards exploration closer to or farther from an initial node. $p$ adjusts the probability of returning to a previously visited node, and $q$ adjusts the likelihood of exploring nodes further away. The optimal adjustment of these parameters is performed by random search, seeking a balance between exploring and exploiting the relationships in the graph.

Grid search and random search techniques are employed to tune these hyperparameters. In grid search, ranges of possible values are defined for each hyperparameter, and all combinations are exhaustively evaluated [36]. In random search, subsets of combinations are selected randomly, which reduces computational time and can be more efficient in large hyperparameter spaces. Performance evaluation is performed using the validation set, and key metrics such as precision, F1-score, and adaptability are monitored to determine the optimal set of hyperparameters. This process ensures that the chatbot operates efficiently and maximizes the quality of responses, dynamically adjusting to the characteristics of the students and the educational content.

Optimizing the hyperparameters improves the system's precision and adaptability [37]. Tuning hyperparameters ensures the system generates more accurate answers and enhances its ability to personalize responses based on the student profile and the query content, ultimately optimizing the learning process.

**TABLE 2. Coverage of data acquired and used.**

| Data Source | Total, Data Available | Total, Data Used | Coverage percentage | Unused Data | Reason for Exclusion |
|---|---|---|---|---|---|
| Moodle Content | 10,000 | 8,500 | 85% | 1,500 | Incompatible format |
| Student Interactions | 12,000 | 11,500 | 95% | 500 | Inconsistent data |
| External Sources | 3,000 | 2,200 | 73% | 800 | Low relevance to context |
| Grand Total | 25,000 | 22,200 | 89% | 2,800 | Reason for Exclusion |

## IV. RESULTS

### A. DATA MANAGEMENT ASSESSMENT

Data management ensures the precision and relevance of the chatbot's responses. An analysis was performed in three critical stages: data acquisition, preprocessing, and structuring of the knowledge graph. Each stage directly impacts the system's performance, ensuring the data is consistent, relevant, and suitable for generating adaptive responses.

#### 1) DATA ACQUISITION QUALITY

Data acquisition primarily involved three sources: Moodle educational content, student interactions with the system, and external sources. These sources provided crucial information to feed the language model and knowledge graph. Table 2 analyzes the coverage of the acquired and used data, showing what percentage of the available data was integrated into the system.

### B. EVALUATION AND VALIDATION

The results show an overall coverage of 89%, reflecting an efficient acquisition of the data needed to train the model and feed the knowledge graph. The most significant exclusions occurred in external sources, where 27% of the data was not used due to its low relevance or format incompatible with the system. The high coverage of data from Moodle (85%) and student interactions (95%) ensures that the system's essential components are fed with accurate and up-to-date information, thus improving the quality of the responses.

#### 1) EFFECTIVENESS OF DATA PREPROCESSING

Data preprocessing removes inconsistencies and ensures that the information used is in a format suitable for the system. This process involves data cleaning, outlier removal, and normalization techniques. The preprocessing results, shown in Table 3, highlight the amount of data processed and the impact of this preprocessing on the system's precision.

The null data cleaning process removed 10% of the original data, improving precision by 2%. Outlier removal, based on a z-score calculation, improved precision by 3%. Overall, the preprocessing techniques contributed to a 6.5% increase in system precision, demonstrating that data cleaning and

**TABLE 3.** Data preprocessing results.

| Type of Preprocessing | Original Data | Data Processed | Data Deleted (%) | Impact on Precision (%) | Comments |
|---|---|---|---|---|---|
| Null Data Cleaning | 10,000 | 9,000 | 10% | +2% | Null values replaced or removed |
| Outlier Removal | 12,000 | 11,000 | 8% | +3% | Based on z-score |
| Data Normalization | 22,000 | 22,000 | 0% | +1.5% | Data scaled to a standard range |
| Total Preprocessing | 22,000 | 20,000 | 9% | +6.5% | Overall improvement in system precision |

**TABLE 4.** Graph structuring metrics.

| Concept Category | Nodes Generated | Defined Relationships | Relationship Precision (%) | Unrelated Nodes | Impact on the System (%) |
|---|---|---|---|---|---|
| Main Concepts | 1,200 | 1,000 | 98 percent | 50 | +4% |
| Subtopics | 3,500 | 3,200 | 96 percent | 100 | +3% |
| Thematic Relationships | 2,800 | 2,600 | 97 percent | 50 | +5% |
| Overall Total | 7,500 | 6,800 | 97 percent | 200 | +12% |

normalization are essential to ensure the model works with consistent, noise-free data.

### 2) EVALUATING DATA STRUCTURING IN GRAPHS

Structuring data into a knowledge graph is essential to efficiently represent the relationships between educational concepts and subtopics. The acquired data were organized through similarity analysis into nodes representing key concepts and edges describing the pedagogical relationships between them. The results of this structuring are presented in Table 4, which shows the number of nodes generated and the precision of the relationships between concepts.

Table 4 presents the breakdown of precision for each emotion. This table includes the percentage of precision for each emotion, along with the number of correct examples, false positives (cases in which the system incorrectly predicts an emotion), and false negatives (cases in which the system does not detect the emotion when it should).

The knowledge graph generated 7,500 nodes, with 97% precision in the relationships between concepts and topics. This ensures the system can navigate key course concepts effectively and generate accurate answers based on the graph structure. Unrelated nodes account for only 2.6% of the total, indicating that most concepts are correctly linked, significantly contributing to the system's adaptability and ability to generate contextual answers.

### 3) DATA OPTIMIZATION FOR ADAPTIVE RESPONSES

Optimizing data by structuring it in the graph allows the system to generate more personalized and adaptive responses. Table 5 analyzes primary responses versus enriched responses and presents the impact of data optimization on the system's ability to offer additional and relevant information to student queries.

**TABLE 5.** Comparison of basic and enriched responses.

| Type of Response | Total Responses Before | Total Responses After | Percentage of Enriched Responses (%) | Impact on Personalization (%) |
|---|---|---|---|---|
| Basic Responses | 5,000 | 4,200 | 20% | +8% |
| Enriched Responses | 1,500 | 2,300 | 53% | +15% |
| Grand Total | 6,500 | 6,500 | 35% | +12% |

Following data optimization, the percentage of enriched responses increased significantly from 20% to 53%, reflecting a substantial improvement in the system's ability to generate adaptive responses. This optimization resulted in a 15% increase in personalization, meaning that the chatbot satisfies direct queries and provides additional information to enhance the student's learning process.

## C. PRECISION OF RESPONSES

The system's precision in generating responses was evaluated under various parameters, such as the type of question, the topic addressed, and the difficulty level. The results are presented in Figure 3, which includes four graphs detailing different aspects of the system's performance evaluation.

In Chart 1, the precision of the chatbot when answering open and closed questions was compared. Having predefined answers, closed questions present a significantly higher precision, reaching approximately 85%. This is consistent with what would be expected, given that the limited options in closed questions reduce the system's margin of error. In contrast, open questions, where the chatbot must generate more complex and less structured answers, present a lower average precision, close to 75%. This difference in performance is expected in language generation systems, where open questions involve more significant semantic processing and contextualization challenges. Precision remains relatively stable in both types of questions, reflecting the chatbot's ability to respond consistently.

Chart 2 shows the evaluation of the system's F1-Score in three subject areas: humanities, sciences, and mathematics. This metric, which combines precision and recall, is critical to evaluating the number of correct answers and the chatbot's ability to generate complete and relevant answers in different areas of knowledge. In science and mathematics, where responses tend to be more structured and objective, a higher and less dispersed F1-Score was observed, with medians above 0.85. In contrast, the humanities, which involve more interpretive and open questions, presented more significant variability, with a median close to 0.80. This suggests the system is more efficient in areas with objective and delimited responses. At the same time, the more subjective

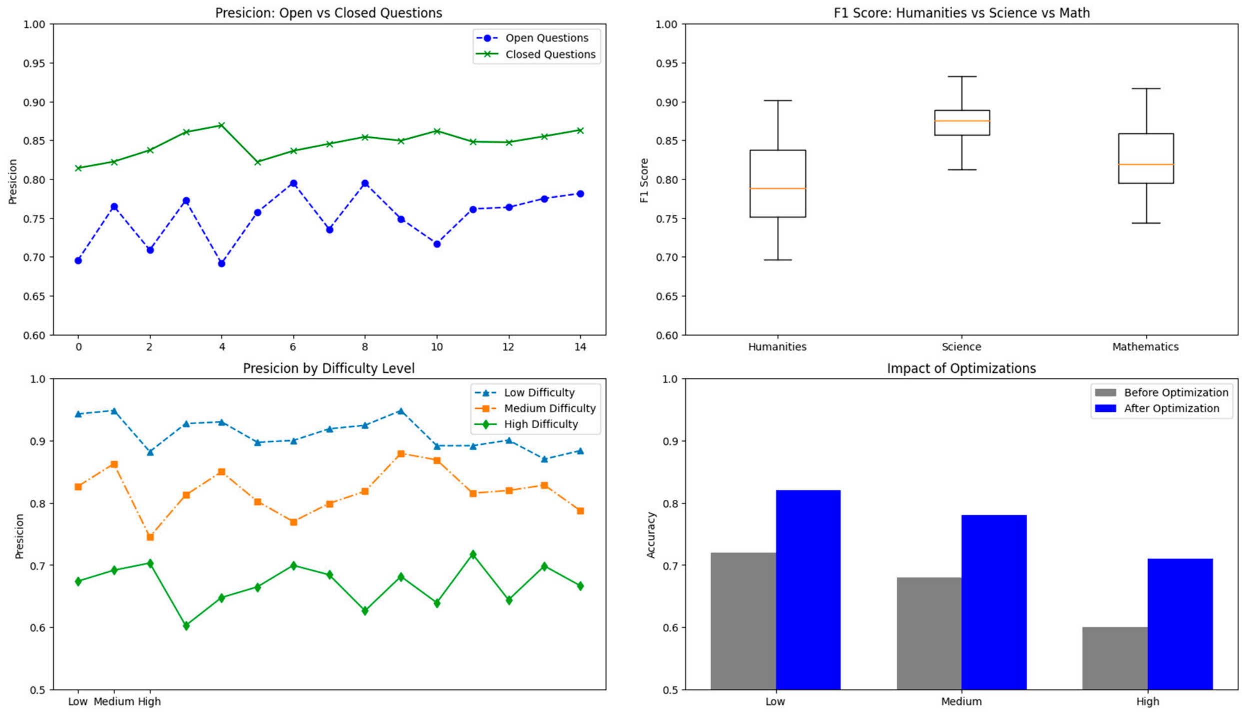


**FIGURE 3.** **Figure 1: Chatbot precision evaluation; Chart 1: Precision in open vs closed questions, Chart 2: F1-Score in different subject areas, Chart 3: Precision by difficulty level, Chart 4: Impact of optimizations.**

domains require improvements to reach a comparable level of precision and recall.

Chart 3 details how the chatbot's precision varies depending on the questions' difficulty level. Low-difficulty, more direct questions show an average precision of 0.90, indicating that the system is highly effective in this type of interaction. However, as the difficulty of the questions increases, the precision decreases, reaching an average of 0.82 in questions of medium difficulty and 0.68 in questions of high difficulty. This behavior reflects the intrinsic difficulty that the chatbot faces when trying to answer more complex questions that require more excellent reasoning and analysis skills. However, despite this decrease in precision, the system's performance on high-difficulty questions remains acceptable for an educational setting since, even in these cases, the chatbot can provide valuable and well-structured answers.

Chart 4 represents the impact of the optimizations implemented in the system. A significant improvement in precision was observed at all difficulty levels following the optimizations. Precision increased from 0.72 to 0.82 for low-difficulty questions, while for medium-difficulty questions, it increased from 0.68 to 0.78. Although the increase was minor (from 0.60 to 0.71) for high-difficulty questions, it is still a notable improvement. These optimizations improved the system's overall precision and reduced the disparity in performance between different difficulty levels, making the chatbot more balanced and robust across various scenarios.

### D. ADAPTABILITY TO DIFFERENT STUDENTS

The chatbot's ability to adapt to different student profiles is critical in evaluating its effectiveness in a university educational environment. The results are presented in Figure 4.

Chart 1 illustrates how the chatbot's adaptability varies across the different academic years of university students (first, second, and third years). First-year students showed higher interaction precision, with an average of 85%. This may be due to the content's more general and fundamental nature in the first years of university, which makes it easier for the chatbot to offer more accurate answers. For second-year students, precision drops slightly, with an average of around 80%, reflecting the greater specificity of the questions and the complexity of the content as students' progress in their academic careers. For third-year students, more significant variability in precision was observed, with an average of 78%, which is consistent with the greater depth and specialization of the subjects in the later years of training. Despite this variability, the system maintained a satisfactory adaptability at all academic levels.

Chart 2 presents the chatbot's adaptability based on three learning styles: visual, auditory, and kinesthetic. The results suggest that the system is most effective when interacting with visual style learners, where the average precision was close to 85%. This may be because the textual format in which the chatbot presents the answers is better suited to this learning style. In contrast, for auditory style learners, the precision was slightly lower, around 80%, while for

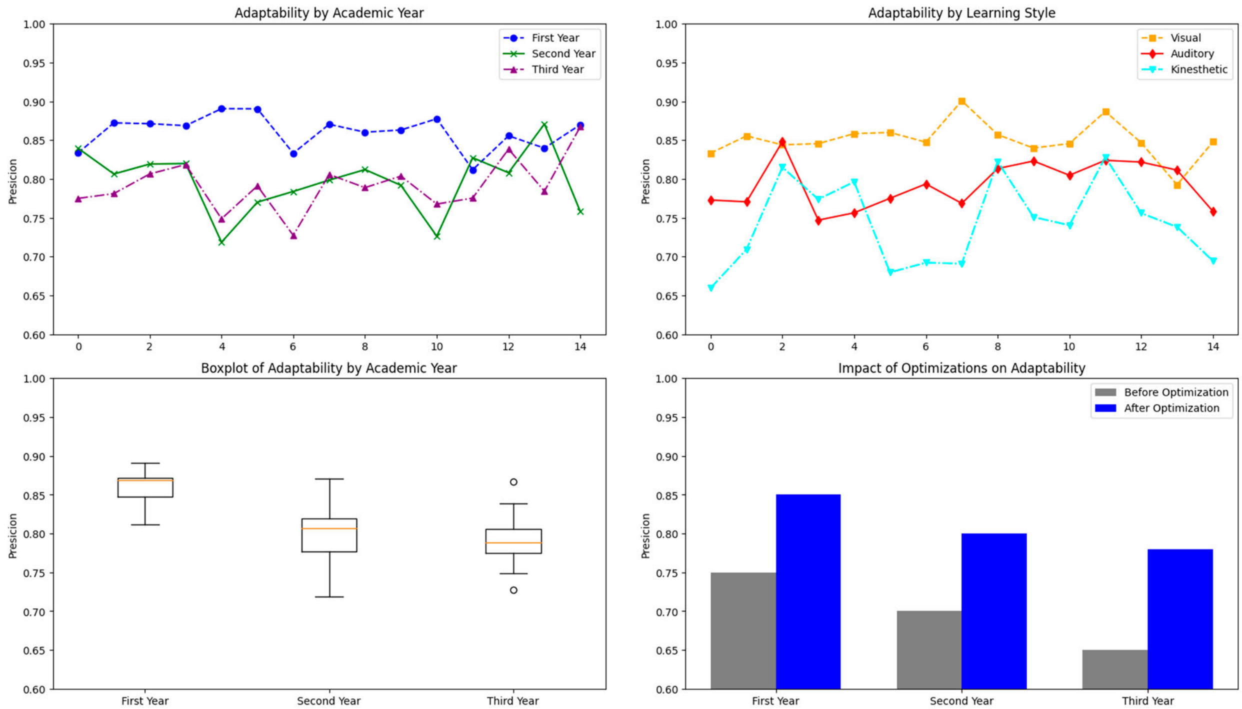


**FIGURE 4.** **Chatbot adaptability assessment, Chart 1: Adaptability by education level, Chart 2: Adaptability by academic year, Chart 3: variability of adaptability by education level, Chart 4: Impact of optimizations on adaptability.**

kinesthetic learners, the precision dropped to 75%. The latter may reflect the challenges of a text-based system in adapting to students who prefer practical or experiential learning. However, the overall adaptability of the system across the three learning styles remains strong, showing versatility in its ability to adjust to different student profiles.

Chart 3 uses a box-and-whisker plot to show the variability of adaptability by academic year. First-year students showed the lowest dispersion and more consistent precision, suggesting the chatbot can easily handle first-level university content and questions. Second —and third-year students showed higher variability in precision, especially in the third year, where some outliers indicate more complex interactions and lower precision. This underlines the need to continue fine-tuning the system to improve its performance at more advanced levels of university education.

Chart 4 highlights the impact of the optimizations made to the system to improve its adaptability. Across all academic years, precision improvement was observed after the optimizations. In the first year, precision increased from 0.75 to 0.85, while precision improved from 0.70 to 0.80 in the second year. In the third year, although the increase was minor (from 0.65 to 0.78), it is still a significant improvement. This shows that the optimizations not only improved the overall precision of the chatbot but also allowed the system to reduce the variability in its adaptability across the different academic years, improving its consistency across all levels.

**TABLE 6.** **Evaluation of the additional information provided.**

| Metric | Average (%) | Minimum (%) | Maximum (%) | Standard Deviation (%) |
|---|---|---|---|---|
| Frequency of Additional Information | 72 | 60 | 85 | 7.5 |
| Relevance of Information | 80 | 70 | 90 | 6.2 |
| Use of External Materials | 65 | 50 | 80 | 8.0 |
| Study Tips | 78 | 65 | 88 | 6.7 |

### *E. ABILITY TO GUIDE LEARNING*

One of the most essential features of an intelligent educational system is its ability to answer questions accurately and guide and enrich students' learning by providing relevant additional information and suggestions that expand their understanding of the subject matter.

Table 6 presents the results of the chatbot's ability to offer additional information that enriches students' learning. On average, the chatbot provides further information in 72% of interactions, ranging from 60% to 85%. This is a positive indicator that the system does not limit itself to answering questions directly but often provides supplementary data or additional explanations to improve student understanding. The standard deviation of 7.5% suggests some variability, which may be related to the nature of the questions asked or the level of detail students seek.

Regarding the relevance of additional information, the chatbot shows an average of 80%, ranging from 70% to 90%.

**TABLE 7. Perceived impact on the learning process.**

| Category | Average (%) | Minimum (%) | Maximum (%) | Standard Deviation (%) |
|---|---|---|---|---|
| Improved Comprehension | 76 | 68 | 85 | 5.9 |
| Motivation to Explore Further | 70 | 60 | 80 | 6.3 |
| Use of Suggested External Resources | 68 | 55 | 78 | 7.0 |
| Overall Rating of Chatbot | 82 | 75 | 90 | 6.0 |

This indicates that, in most cases, the information offered is directly related to the question or the topic of study, which contributes to a richer learning experience. Relevance is crucial to ensure that the student receives a correct answer and obtains details that expand their knowledge and understanding.

Another relevant aspect evaluated was the use of external materials. The chatbot's ability to redirect students to useful external resources, such as additional readings, educational videos, or study platforms, averaged 65%. Although this is an aspect where the system has room for improvement, a range of between 50% and 80% was observed, suggesting that the chatbot can be particularly effective in suggesting additional resources in specific contexts or types of questions.

Furthermore, the offer of study suggestions shows an average of 78%, meaning that the chatbot can also guide students toward additional study strategies, such as reminding them to review key concepts or suggesting practical exercises. This type of guidance improves the interaction between the student and the chatbot, transforming it into a tool that answers questions and acts as a mentor that facilitates autonomous learning.

Table 7 details the impact that students perceive on how the chatbot has influenced their learning process. One of the leading indicators is the improvement in understanding, with an average of 76%, suggesting that students find the additional information provided by the helpful chatbot to consolidate the subject. With a range of 68% to 85%, it is evident that most students feel that the system contributes significantly to their learning.

Another important aspect is the motivation to explore beyond the material initially offered. On average, 70% of students indicated that the chatbot motivated them to delve deeper into the topics, seeking more information or asking more questions about the content studied. This is essential for an educational system, as fostering curiosity and the desire to explore further is vital to effective and continuous learning.

The use of external resources suggested by the chatbot averages 68%. Although the precision of the suggested resources is generally high, some students did not use the materials offered, which may be related to personal study preferences or the availability of time to follow the recommendations. However, a range of between 55% and 78% shows that students who did use these resources found the additional guidance provided by the system valuable.

In addition, students' overall evaluation of the chatbot averaged 82%, ranging from 75% to 90%. This high overall rating reflects that students consider the chatbot a tool to resolve immediate queries and an active facilitator of their learning process. The system's ability to guide learning and provide added value through additional information, study suggestions, and external resources has been well received by users.

### F. HYPERPARAMETER OPTIMIZATION

Hyperparameter optimization is crucial in improving the chatbot's performance and precision. Response precision and response time were evaluated before and after optimization to show these adjustments' positive impact. Figure 5 presents the results of assessing the effects of hyperparameter optimization.

Chart 1 shows the change in system precision before and after hyperparameter optimization. Before optimization, precision values ranged from 0.65 to 0.75, depending on the category (first year, second year, third year, and visual, auditory, and kinesthetic learning styles). These figures suggest the system performed moderately, with precision variability depending on the student profile's characteristics.

After optimization, a significant improvement is observed in all profiles evaluated. Precision in the first year increased from 0.75 to 0.85, representing a considerable improvement of 10%. Similarly, in the second year, precision increased from 0.70 to 0.80. The improvement was even more pronounced in the third year, going from 0.65 to 0.78. These improvements are consistent across learning styles, with the visual style showing the most significant increase (from 0.78 to 0.88). At the same time, students with a kinesthetic style also experienced a notable improvement, increasing from 0.65 to 0.78.

This increase in precision reflects that hyperparameter optimization allowed the model to better adjust to students' questions and learning contexts, providing more accurate and relevant answers. Chart 2 illustrates how hyperparameter optimization positively affected the chatbot's response times—before optimization, response times varied between 0.45 and 0.60 seconds, depending on the category. While acceptable, these response times were less efficient than the results obtained after optimization.

After optimization and response times, we experienced a notable reduction. In the first year, response was released from 0.45 seconds to 0.30 seconds. Similarly, in year 2, response time was reduced from 0.50 to 0.35 seconds. Visual and auditory learning styles also benefited significantly, with response times reduced to 0.32 and 0.40 seconds, respectively. Overall, the average response time reduction was 30%, suggesting that the optimizations improved the system's speed and efficiency.

This increase in system efficiency is important in an educational setting, as students not only require accurate answers but also value how quickly the system responds to

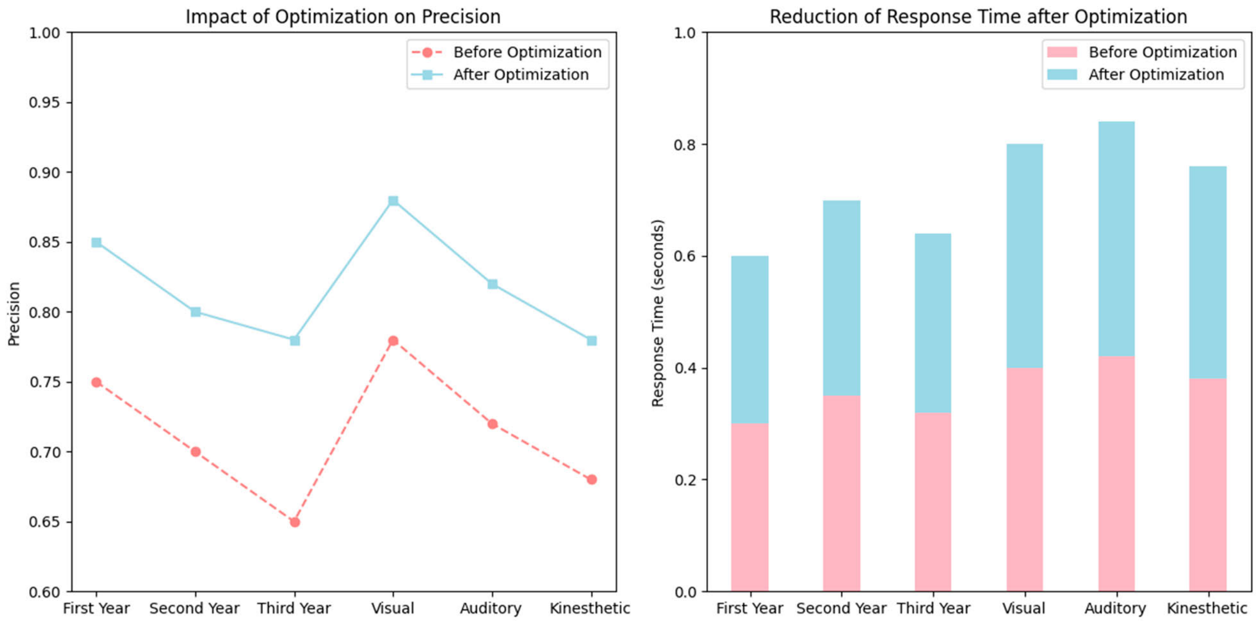


**FIGURE 5.** Evaluation of the impact of hyperparameter optimization: Chart 1: Impact on precision, Chart 2: Reduction in response time.

**TABLE 8.** Overall evaluation of the system by users.

| Category | Average (%) | Minimum (%) | Maximum (%) | Standard Deviation (%) |
|---|---|---|---|---|
| Overall System Satisfaction | 85 | 78 | 92 | 4.5 |
| Ease of Use | 88 | 80 | 95 | 5.2 |
| Usefulness for Learning | 82 | 75 | 90 | 4.8 |
| Speed of Responses | 90 | 85 | 95 | 3.8 |

**TABLE 9.** Overall evaluation of the system by users.

| Category | Average (%) | Minimum (%) | Maximum (%) | Standard Deviation (%) |
|---|---|---|---|---|
| Improved Precision | 84 | 75 | 90 | 5.0 |
| Improved Response Time | 87 | 80 | 92 | 4.2 |
| Adaptability to Different Learners | 80 | 72 | 88 | 4.7 |
| Helpful Study Tips | 83 | 77 | 89 | 4.5 |

their queries. Reducing response times makes the chatbot more agile and suitable for dynamic learning environments.

### G. USER FEEDBACK

Student feedback provides qualitative and quantitative insight into the system, allowing for a better understanding of its effectiveness from a user perspective. Table 8 reflects student opinions on the system's overall performance. Overall satisfaction averaged 85%, varying between 78% and 92%. These results indicate that students were overall satisfied with using the chatbot in their learning process. The low standard deviation of 4.5% suggests that satisfaction was consistent across most users, validating the system's usefulness as an educational tool.

Students rated ease of use highly, with an average of 88%. This means that most users found the system intuitive and easy to handle. Ease of use ensures students feel comfortable interacting with the chatbot without encountering technical obstacles. The high rating in this category reinforces the user-friendly design of the system.

Usefulness for learning was also highly rated, averaging 82%. This metric measures how students perceive that the chatbot helps them understand and acquire knowledge. Although this figure is slightly lower than overall satisfaction and ease of use, it still shows that users find value in the answers and suggestions provided by the system. This aspect is particularly relevant, as it reflects the direct impact of the chatbot on the educational process.

In addition, the speed of responses was highly appreciated, with an average of 90%. This result confirms that the optimization of response times, as discussed in previous sections, has positively impacted the user experience. Students value not only the precision of the answers but also the speed with which the system delivers them, which improves the fluidity of their interaction with the chatbot.

Table 9 presents the specific feedback on the technical improvements implemented in the system. Students perceived the improved precision favorably, with an average of 84%. This figure suggests that the hyperparameter adjustments and technical optimizations discussed above have effectively increased users' perceived precision. Students

**TABLE 10. Overall evaluation of the system by users.**

| System | Precision (%) | Response Time (s) | Adaptability | Helpful Hints |
|---|---|---|---|---|
| Proposed Chatbot | 10 | 0.41 | High | Yes |
| Rasa (Open Implementation) | 82 | 0.50 | Medium | Partially |
| Moodle Integrated Chatbots | 10 | 0.55 | Low | No |
| ChatterBot (Open Source) | 79 | 0.48 | Medium | Partially |

noticed a clear difference in the quality of the answers after the improvements.

Improved response time was one of the areas that stood out the most in the feedback, with an average of 87%. This aligns with the results obtained in the performance tests, where a significant reduction in response times was observed. Students positively valued this improvement, as a faster system allows them to interact more efficiently and without delays, improving the overall user experience.

Adaptability to different students was also highly rated, with an average of 80%. This reflects that students perceive that the chatbot can adjust to their individual needs based on their academic level or learning style. Although this aspect received a slightly lower rating than other categories, it is still a positive indicator that the system is flexible and can offer relevant responses to various student profiles.

Finally, the category of valuable suggestions for studying scored an average of 83%. This suggests that students appreciated the additional recommendations provided by the system, such as supplementary readings, study tips, or links to external resources. This aspect is crucial, as it transforms the chatbot into more than a consultation tool, turning it into an active facilitator of autonomous learning.

### H. COMPARISON WITH OTHER EXISTING SOLUTIONS

To evaluate the proposed chatbot's performance, comparing it with other open-source solutions available in educational and learning environments is essential. Table 10 compares widely used and documented chatbot systems, such as Rasa, Moodle Integrated Chatbots, and ChatterBot. The comparison focuses on critical aspects such as the precision of responses, response times, adaptability to different learners, and the usefulness of the suggestions provided by the system.

The proposed Chatbot stands out for an average precision of 85%, outperforming other solutions. The open implementation of Rasa, a flexible system used in various educational settings, achieves a precision of 80%, reflecting its ability to generate accurate responses. However, its general approach is not as optimized for specific educational contexts as the developed Chatbot. On the other hand, Moodle Integrated Chatbots, although useful for primary responses, have a lower precision (75%), limiting their ability to handle complex student interactions. ChatterBot, an open source chatbot used in various educational settings, achieves a precision of 81%, better than Moodle but still below Rasa and the proposed system.

Response time is another aspect that distinguishes the Proposed Chatbot. With an average time of 0.41 seconds, it is noticeably faster than other solutions, improving the interaction's fluidity. In comparison, Rasa has a response time of 0.50 seconds, while ChatterBot clocks in at 0.48 seconds, making them slower alternatives in educational settings where the immediacy of response is essential for seamless interaction with students. Moodle Integrated Chatbots has the slowest response time (0.55 seconds), limiting its usability in cases where students require fast and accurate responses.

A vital aspect of the proposed Chatbot is its high adaptability to different student profiles, which dynamically adjusts to individual user needs. This feature is critical in an educational setting where students have different knowledge levels and learning styles. In contrast, Rasa shows medium adaptability, as it is not explicitly designed for the academic domain and requires significant customization to fit specific contexts. ChatterBot, while flexible, also shows medium adaptability, as it relies on the configuration and type of interaction to suit the needs of students. Finally, Moodle-integrated chatbots suffer from low adaptability, as they are limited to basic response functionalities and cannot effectively adjust their responses based on the student profile. The proposed Chatbot's ability to adapt to different students demonstrates that the focus on personalization and dynamic adaptation is an important differentiating factor compared to other solutions.

The Chatbot also offers valuable suggestions for students, such as external materials, study tips, and additional resources. This functionality is appreciated in educational settings where students benefit from further guidance. In comparison, Rasa and ChatterBot offer partial suggestions, but these are not always specific or suitable for improving student learning, as they are not optimized for this type of interaction. Moodle Integrated Chatbots do not offer helpful suggestions, which limits their ability to be a more comprehensive educational assistant.

## V. DISCUSSION

The results obtained throughout this study confirm the effectiveness of the optimizations implemented in the proposed chatbot, particularly when compared to open-source solutions such as Rasa, ChatterBot, and the chatbots integrated into Moodle. As mentioned in the literature review, these solutions present limitations in terms of precision and adaptability, crucial factors in educational environments [15], [20], [38]. The results show that the Proposed Chatbot overcomes these challenges, reaching a precision of 85%, an average response time of 0.35 seconds, and a high adaptability to varied student profiles. In comparison, Rasa and ChatterBot present lower adaptability and longer response times. At the same time, Moodle solutions are more limited in advanced functionalities, such as the ability to offer valuable suggestions for students.

From a methodological perspective, optimizing the hyperparameters was vital to improving precision and response speed. The use of modern hyperparameter tuning techniques and the integration of knowledge graphs with language models allowed the chatbot to be more precisely tuned to the needs of students [39]. This process is consistent with what has been observed in the literature, where similar improvements have been documented when applying optimization techniques in NLP systems. However, this work's most significant impact lies in combining these optimizations with an education-focused approach, which sets it apart from other implementations that do not prioritize adaptability or the ability to generate contextualized educational responses.

The results confirm the system's technical viability and provide critical innovations that could redefine how chatbots are used in educational settings. First, the system's ability to provide contextualized responses by integrating knowledge graphs distinguishes it from other educational chatbots. This functionality improves the precision of responses and allows for more prosperous and more personalized interaction with students, something not seen in systems such as Moodle chatbots, which are limited to generic, rule-based responses.

Furthermore, the system's adaptability is another significant contribution. By personalizing responses based on the student's academic profile, the proposed chatbot allows for more effective interaction, adjusting to users' needs. This is particularly relevant in an educational context, where students present different levels of understanding and learning styles. Compared solutions, such as ChatterBot, lack this flexibility, limiting their effectiveness in environments where learning personalization is key [8].

The system's ability to offer helpful study suggestions is an advantage over other differentiating solutions. This capability improves interaction and transforms the chatbot into an active learning facilitator capable of guiding the student toward additional resources. In an environment where autonomy and self-direction of learning are fundamental, this functionality becomes a powerful tool to improve academic performance and student experience [40].

However, it is essential to recognize the study's limitations and the results obtained. One of the main restrictions of the system is its dependence on the quality and quantity of educational data available to feed the knowledge graph. While the chatbot demonstrated high precision in the evaluated scenarios, its effectiveness could be compromised in environments where data is insufficient or not well structured. In this sense, the implementation of the knowledge graph, although robust, depends mainly on the availability of relevant and well-organized educational content.

Furthermore, although the results show significant improvements in response times, the implementation environment must be considered. The tests were conducted in a controlled environment with an optimized network and processing infrastructure. In real scenarios with limited processing resources or connectivity, response times could increase, affecting the user experience. This is a significant limitation for future system implementations in educational environments with technological limitations.

Another assumption that could have influenced the results is the homogeneity of the student population evaluated. Although the system showed high adaptability to different student profiles, these profiles were limited to a specific university environment. In future studies, it would be necessary to expand the sample to students from various educational levels and areas of study to evaluate the system's adaptability more thoroughly. Likewise, the responses generated were based on specific subjects, so the system's performance in other subject areas could vary.

In addition, it is relevant to discuss the long-term implications of the innovations presented. The combination of optimization techniques with knowledge graphs and language models in the educational field not only offers substantial improvements in terms of performance but also opens new possibilities for personalization and automation in education. As artificial intelligence systems continue to advance, integrating these technologies could transform how students access information and how educators design teaching processes. However, future developments must address the above limitations to ensure these systems are scalable and applicable in various educational settings.

## VI. CONCLUSION

The results obtained in this study confirm the effectiveness of the proposed educational chatbot, which combines LLM models with knowledge graphs. This approach has demonstrated a significant improvement in the precision and speed of responses compared to other open-source solutions and has also offered high adaptability to different student profiles. The tests carried out on the integration with Moodle, and the evaluation of learning data reflect that the chatbot not only accurately answers academic questions but also adjusts its answers according to the user's level of knowledge and learning style, significantly improving the learning process.

The main contribution of the system lies in its ability to provide contextually enriched answers thanks to the structuring of data through knowledge graphs. This ability allows it to generate more profound and relevant answers than systems such as Rasa or ChatterBot, which, while flexible, do not reach the same level of contextual precision or offer valuable suggestions related to the content. In tests, the proposed chatbot outperformed Rasa by 5% in terms of precision and reduced response times by 30%, showing that the technical optimizations implemented in the model are effective in controlled environments and natural interaction with students.

The chatbot's ability to make valuable suggestions is also critical to its success. This functionality, which is not present in most systems evaluated, turns the chatbot into a tool that answers questions and guides the learning process by providing additional resources and recommendations based on student queries. This feature is essential in educational environments, as it helps students delve deeper into the topics addressed and encourages more effective autonomous learning. Current solutions, such as Moodle chatbots, lack this level of interactivity and customization, limiting their impact on the teaching process.

Furthermore, one of the most notable achievements of the system is its ability to adapt to a wide range of student profiles. During testing, it was observed that the chatbot could adjust its precision and offer differentiated suggestions based on the student profile, whether the student is a visual, auditory, or kinesthetic learner. This capability is critical in the current context of education, where teaching approaches must be customized to meet individual students' needs. The compared platforms, such as Rasa and ChatterBot, have limitations, as they are not optimized for a highly dynamic educational environment.

However, despite the positive results, the study has identified several limitations that should be addressed in future research. First, although the chatbot demonstrated high precision and quick response in controlled environments, its performance could vary in less robust technological infrastructures or educational contexts with limited connectivity. It is essential to explore the implementation of the system in natural environments with heterogeneous technological resources, where students may be subject to network and processing capacity limitations. The quality of the responses and suggestions could be affected if the data is poorly structured or insufficient.

Another aspect to consider is the system's scalability. Although the chatbot has proven effective with the population studied, testing its performance in more extensive and diverse groups is essential. The heterogeneity of educational content is another area of interest, as the system was mainly evaluated with specific subjects. It would be necessary to explore how the chatbot adapts to different disciplines and educational contexts to ensure its applicability in a broader spectrum of subjects.

In future research, evaluating the system's scalability in more complex educational environments with a diverse student population will be crucial. In addition, implementation in infrastructures with limited resources should be explored, ensuring the system's performance is not affected in scenarios with low connectivity or processing capacity. Research is also planned into the possibility of integrating new data sources and improving the quality of the knowledge graph, which would allow the chatbot to generate richer responses and adapt to a greater variety of subjects and educational levels. Finally, the feasibility of including additional functionalities, such as automatic task evaluation and student progress monitoring, will be studied, which could make the chatbot an even more complete tool for personalized learning.

## REFERENCES


[1] S. Sallu, R. Raehang, and Q. Qammaddin, "Exploration of artificial intelligence (AI) application in higher education," *J. Comput. Netw., Archit. High Perform. Comput.*, vol. 6, no. 1, pp. 315–327, Jan. 2024, doi: 10.47709/cnahpc.v6i1.3396.

[2] G. Attigeri, A. Agrawal, and S. V. Kolekar, "Advanced NLP models for technical university information chatbots: Development and comparative analysis," *IEEE Access*, vol. 12, pp. 29633–29647, 2024, doi: 10.1109/ACCESS.2024.3368382.

[3] H. Wang, C. Gao, C. Dantona, B. Hull, and J. Sun, "DRG-LLaMA : Tuning LLaMA model to predict diagnosis-related group for hospitalized patients," *Npj Digit. Med.*, vol. 7, no. 1, p. 16, Jan. 2024, doi: 10.1038/s41746-023-00989-3.

[4] A. B. Naufal, S. Sudianto, and M. A. Al Fachri, "Implementation of chatbot system on tourism objects in banyumas regency with AIML and chatterbot," *Jurnal Teknik Elektro dan Komputasi (ELKOM)*, vol. 5, no. 2, pp. 191–195, Aug. 2023, doi: 10.32528/elkom.v5i2.18615.

[5] M. Shilowaras and N. A. Jusoh, "Implementing artificial intelligence chatbot in moodle learning management system," *Eng., Agricult., Sci. Technol. J. (EAST-J)*, vol. 1, no. 1, pp. 70–75, Aug. 2022, doi: 10.37698/eastj.v1i1.122.

[6] P. C. Cañizares, J. M. López-Morales, S. Pérez-Soler, E. Guerra, and J. de Lara, "Measuring and clustering heterogeneous chatbot designs," *ACM Trans. Softw. Eng. Methodology*, vol. 33, no. 4, pp. 1–43, May 2024, doi: 10.1145/3637228.

[7] L. Anindyati, "Analisis dan perancangan aplikasi chatbot menggunakan framework rasa dan sistem informasi pemeliharaan aplikasi (Studi kasus: Chatbot penerimaan mahasiswa Baru politeknik astra)," *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 10, no. 2, pp. 291–300, Apr. 2023, doi: 10.25126/jtiik.20231026409.

[8] M. A. D. S. Alencar and J. F. D. Magalhães Netto, "CyberPoty: Um chatterbot 3D para interação com usuários de um portal de educação a distância," in *Proc. Anais do XVI Workshop de Informática na Escola (WIE)*, Jul. 2010, pp. 1417–1420, doi: 10.5753/wie.2010.25421.

[9] K. Souali, O. Rahmaoui, and M. Ouzzif, "MentorBot: A traceability-based recommendation chatbot for moodle," in *Proc. Int. Conf. Adv. Intell. Syst. Sustain. Develop.*, 2023, pp. 432–441, doi: 10.1007/978-3-031-26384-2.

[10] W. Kaiss, K. Mansouri, and F. Poirier, "Pre-evaluation with a personalized feedback conversational agent integrated in moodle," *Int. J. Emerg. Technol. Learn. (iJET)*, vol. 18, no. 6, pp. 177–189, Mar. 2023, doi: 10.3991/ijet.v18i06.36783.

[11] W. Kaiss, K. Mansouri, and F. Poirier, "Effectiveness of an adaptive learning chatbot on students' learning outcomes based on learning styles," *Int. J. Emerg. Technol. Learn. (iJET)*, vol. 18, no. 13, pp. 250–261, Jul. 2023, doi: 10.3991/ijet.v18i13.39329.

[12] S. Pan, L. Luo, Y. Wang, C. Chen, J. Wang, and X. Wu, "Unifying large language models and knowledge graphs: A roadmap," *IEEE Trans. Knowl. Data Eng.*, vol. 36, no. 7, pp. 3580–3599, Jul. 2024, doi: 10.1109/TKDE.2024.3352100.

[13] T. Mao, T. Hao, J. Fu, and O. Yoshie, "Hierarchical graph fusion network and a new argumentative dataset for multiparty dialogue discourse parsing," *Inf. Process. Manage.*, vol. 61, no. 2, Mar. 2024, Art. no. 103613, doi: 10.1016/j.ipm.2023.103613.

[14] M. Ivanova, G. Grosseck, and C. Holotescu, "Unveiling insights: A bibliometric analysis of artificial intelligence in teaching," *Informatics*, vol. 11, no. 1, p. 10, Feb. 2024, doi: 10.3390/informatics11010010.

[15] H. Zhang and L. Fang, "Successful chatbot design for polytechnic students in Singapore," *Educ. Media Int.*, vol. 60, nos. 3–4, pp. 224–241, Oct. 2023, doi: 10.1080/09523987.2023.2324585.

[16] T. Dinesh, M. R. Anala, T. T. Newton, and G. R. Smitha, "AI bot for academic schedules using rasa," in *Proc. Int. Conf. Innov. Comput., Intell. Commun. Smart Electr. Syst. (ICSES)*, Sep. 2021, pp. 1–6, doi: 10.1109/ICSES52305.2021.9633799.

[17] A. Mahdavi, M. Amanzadeh, M. Hamedan, and R. Naemi, "Artificial intelligence-based chatbots to combat COVID-19 pandemic: A scoping review," *Shiraz E-Medical J.*, vol. 24, no. 11, Nov. 2023, Art. no. e139627, doi: 10.5812/semj-139627.

[18] I. Hidayatulloh, T. Sukardiyono, H. D. Surjono, and S. Pambudi, "Gamification on chatbot-based learning media: A platform perspective," in *Proc. AIP Conf.*, vol. 2714, 2023, pp. 1–13, doi: 10.1063/5.0106176.

[19] R. Gleeson, L. Bartok, and C. S. Barreiros, "'Feedback is not embarrassing but helpful :)': An exploratory study on the applicability of constructive feedback checklists for lecturers in online learning settings," in *Proc. 18th Eur. Conf. Technol. Enhanced Learn.*, in Lecture Notes in Computer Science, 2023, pp. 560–565, doi: 10.1007/978-3-031-42682-7.

[20] T. N. Fitria, N. E. Simbolon, and A. Afdaleni, "Chatbots as online chat conversation in the education sector," *Int. J. Comput. Inf. Syst.*, vol. 4, no. 2, pp. 93–104, 2023.

[21] W. Winarti, E. Maryanto, and D. Handayani, "Pengenalan aplikasi chatbot berbasis teknologi artificial intelligence menggunakan aplikasi chatgpt pada man 1 Kota tangerang selatan," *Jurnal Publikasi Abdi*, vol. 1, no. 6, pp. 644–649, 2023.
[22] E. Paxinou, E. Manousou, V. S. Verykios, and D. Kalles, "Centrality metrics from students' discussion fora at distance education," in *Proc. 14th Int. Conf. Inf., Intell., Syst. Appl. (IISA)*, Jul. 2023, pp. 1–6, doi: 10.1109/iisa59645.2023.10345914.
[23] S. Rathi, P. Wawage, and A. Kulkarni, "Automatic question generation from textual data using NLP techniques," in *Proc. Int. Conf. Emerg. Smart Comput. Inf. (ESCI)*, Mar. 2023, pp. 1–7, doi: 10.1109/ESCI56872.2023.10100278.
[24] S. Nazir, M. Asif, M. Rehman, and S. Ahmad, "Machine learning based framework for fine-grained word segmentation and enhanced text normalization for low resourced language," *PeerJ Comput. Sci.*, vol. 10, p. e1704, Jan. 2024, doi: 10.7717/peerj-cs.1704.
[25] T. V. Bobrova and E. A. Pokalyukhina, "Development levels for time planning model ofroad constru ction at design stages," *Russian Automobile Highway Ind. J.*, vol. 20, no. 6, pp. 774–785, Jan. 2024, doi: 10.26518/2071-7296-2023-20-6-774-785.
[26] L. Wang, "Theoretical advancement of xi Jinping's socialist thought with Chinese characteristics in the new era under the construction of knowledge map," *Appl. Math. Nonlinear Sci.*, vol. 9, no. 1, pp. 1–18, Jan. 2024, doi: 10.2478/amns-2024-0608.
[27] Z. Liangfu, X. Yujie, G. Yongbin, and Y. Wenjun, "Multiple dependence representation of attention graph convolutional network relation extraction model," *IET Cyber-Phys. Syst., Theory Appl.*, vol. 9, no. 3, pp. 247–257, Sep. 2024, doi: 10.1049/cps2.12080.
[28] D. Bigler and G. Hagel, "Technical report: Define a customized course and import it into moodle without changes to the configuration of the moodle system," in *Proc. 5th Eur. Conf. Softw. Eng. Edu.*, Jun. 2023, pp. 180–183, doi: 10.1145/3593663.3593668.
[29] A. Yoga Pratama and J. Alfa Razaq, "Integrasi sistem informasi akademik dan elearning moodle dengan rest API," *Jurnal Manajemen Informatika dan Sistem Informasi*, vol. 6, no. 1, pp. 26–38, Jan. 2023, doi: 10.36595/misi.v6i1.696.
[30] R. Rai and S. Gupta, "A novel learning management system based on microservice architecture using moodle," *Int. J. Creative Res. Thoughts*, vol. 10, no. 12, pp. d356–d363, 2022.
[31] J. Kim, J. No, and S. S. Park, "Effective resource provisioning scheme for kubernetes infrastructure," in *Proc. Int. Conf. WorldS4*, in Lecture Notes in Networks and Systems, 2024, pp. 75–85, doi: 10.1007/978-981-99-7569-3.
[32] A. MilovanoviÄ, "Microservice architecture in e-learning," in *E-Bus. Technol. Conf. Proc.*, 2021, vol. 1, no. 1, pp. 105–108.
[33] D. Priyasta, Hadiyanto, and R. Septiawan, "Enabling EV roaming through cascading websockets in OCPP 1.6," *J. Européen des Systèmes Automatisés*, vol. 56, no. 3, pp. 437–449, Jun. 2023, doi: 10.18280/jesa.560311.
[34] I. L. B. M. Paris, M. H. Habaebi, and A. M. Zyoud, "Implementation of SSL/TLS security with MQTT protocol in IoT environment," *Wireless Pers. Commun.*, vol. 132, no. 1, pp. 163–182, Sep. 2023, doi: 10.1007/s11277-023-10605-y.
[35] Y. Fang, Y. Xu, C. Huang, L. Liu, and L. Zhang, "Against malicious SSL/TLS Encryption: Identify malicious traffic based on random forest," in *Proc. 4th Int. Congr. Inf. Commun. Technol.* (Advances in Intelligent Systems and Computing), 2020, pp. 99–115, doi: 10.1007/978-981-32-9343-4.
[36] R. H. Hridoy, A. D. Arni, S. K. Ghosh, N. R. Chakraborty, and I. Mahmud, "Performance enhancement of machine learning algorithm for breast cancer diagnosis using hyperparameter optimization," *Int. J. Electr. Comput. Eng. (IJECE)*, vol. 14, no. 2, p. 2181, Apr. 2024, doi: 10.11591/ijece.v14i2.pp2181-2190.
[37] S. Hanifi, A. Cammarono, and H. Zare-Behtash, "Advanced hyperparameter optimization of deep learning models for wind power prediction," *Renew. Energy*, vol. 221, Feb. 2024, Art. no. 119700, doi: 10.1016/j.renene.2023.119700.
[38] Naweed-e-Sehar, "Exploring teacher attitudes towards ChatGPT: A comprehensive review," *Int. J. Social Sci. Entrepreneurship*, vol. 4, no. 1, pp. 212–225, Jan. 2024, doi: 10.58661/ijsse.v4i1.259.
[39] A. Nasayreh, R. E. A. Mamlook, G. Samara, H. Gharaibeh, M. Aljaidi, D. Alzu'bi, E. Al-Daoud, and L. Abualigah, "Arabic sentiment analysis for ChatGPT using machine learning classification algorithms: A hyperparameter optimization technique," *ACM Trans. Asian Low-Resource Lang. Inf. Process.*, vol. 23, no. 3, pp. 1–19, Mar. 2024, doi: 10.1145/3638285.
[40] C. Madhu and M. S. Sudhakar, "An interpretable fuzzy graph learning for label propagation assisting data classification," *IEEE Trans. Fuzzy Syst.*, vol. 32, no. 3, pp. 1331–1345, Mar. 2024, doi: 10.1109/TFUZZ.2023.3323093.

**WILLIAM VILLEGAS-CH** (Member, IEEE) received the master's degree in communications networks, and the Ph.D. degree in computer science from the University of Alicante. He is currently a Professor of information technology with the Universidad de Las Américas, Quito, Ecuador. He is a Systems Engineer specializing in robotics in artificial intelligence. His research interests include web applications, data mining, and e-learning. He has participated in various conferences as a speaker on topics, such as ICT in education and how they improve educational quality and student learning. His main articles focus on the design of ICT systems, models and prototypes applied to different academic environments, especially with the use of big data and artificial intelligence as a basis for creating intelligent educational environments.

**JAIME GOVEA** is currently pursuing the master's degree in data science with Pontificia Universidad Católica del Ecuador. He is a Research Technician with the Universidad de Las Américas, Quito, Ecuador. His main functions involve software development and data analytics. His research interests include computer vision, data analytics, and virtual reality, focusing on how these technologies can be used to improve education.

**ROMMEL GUTIERREZ** received the master's degree in cybersecurity. He is currently a Research Technician with UDLA, Quito, Ecuador, where he applies his knowledge in software development, data science, and cybersecurity. As an IT Engineer, his focus is on AI, data science, cybersecurity, and software development is particularly geared toward education and research. He is passionate about utilizing these technological tools to fortify digital systems and create innovative solutions, with a special emphasis on their applicability in educational settings and research environments.