

Research Article

Framework for Educational Domain-Based Multichatbot Communication System

Zojan Memon,¹ Hamideh Aghian ,² Muhammad Shahzad Sarfraz,³
Akhtar Hussain Jalbani ,⁴ Rozita Jamili Oskouei ,² Khuda Bux Jalbani ,⁵
and Ghulam Hussain Jalbani ⁴

¹Department of Information Technology, University of Sufism and Modern Sciences, Bhit Shah 70140, Pakistan

²Department of Computer Science and Information Technology, Mahdisha Branch, Islamic Azad University, Mahdisha, Iran

³Department of Computer Science, National University of Computer and Emerging Sciences, Chiniot-Faisalabad Campus, Chiniot, Islamabad, Pakistan

⁴Department of Information Technology, Quaid-E-Awam University of Engineering, Science and Technology, Nawabshah 67450, Pakistan

⁵Riphah Institute of Systems Engineering, Riphah International University, Islamabad 44000, Pakistan

Correspondence should be addressed to Rozita Jamili Oskouei; rozita2020j@gmail.com

Received 22 January 2021; Revised 24 February 2021; Accepted 19 April 2021; Published 6 May 2021

Academic Editor: Shah Nazir

Copyright © 2021 Zojan Memon et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Education is an area where innovation moves slowly. In this study, we will propose a framework with a novel approach that will support the development of a multi-interactive chatbot's system for an educational area using AIML 2.0. The system will facilitate the students for their learning towards an outcome-based education domain. The proposed framework will be composed of a user module which consists of user and user interface, chat agents module which will respond to the user query, chatbot KB which will act as the brain for the chatbot system, and socket system for establishing the communication link. Finally, the proposed system will be evaluated using a confusion matrix. The multichatbot communication system will support text-based dialogues on a limited set of questions related to education. However, the system will be implemented in java. The outcomes of this research will be useful for the education sector where these intelligent systems will help the students in schools, universities, and other training scenarios.

1. Introduction

AI deals with the development of intelligent systems which can think and act as humans. There are various tasks that intelligent systems do, such as learning, planning, and speech recognition, while research related to AI is extremely specialized as well as technical.

Chatbot is one of the types of intelligent systems which communicate with the user like a human through voice commands or text or both. These chatbot systems are capable of generating an appropriate response or an appropriate action based on the user input. The quality of such kind of systems can be determined through the significance of their output which is being chosen by the chatbot in response. Basically, chatbots were intentionally designed to

replicate the communication smartly with single or multiple users, while NLP and ML are the important aspects of a chatbot system [1].

Chatbots have changed the way we think and live because chatbots had the capability of being present and ready to provide help by performing tasks while conducting conversation anytime and anywhere [2]. These intelligent chatbot systems are being used in many areas such as banking, customer support, security, health, and education.

As the increasing utilization of technology has been changing the way students learn and understand the information as the chatbots have provided a personalized learning environment to the students [3]. In education, the chatbots can be used to teach the students by conducting a lecture in a series of messages in order to make it look like a

standardized chat conversation. These chatbots help to improve communication and productivity, minimize ambiguity, enhance the learning process, and assess the level of understanding of the student from interactions [4].

Various chatbot systems for education purposes exist but there are some situations where a single chatbot system is not sufficient to atomize various operations or handle various domain requests [5]. However, the use of AI technology to formulate most of the operations of an autonomous organization can be difficult for a single chatbot to handle. Therefore, there is a need of a platform which supports multichatbot communication to facilitate learner in a group learning environment where they can be able to learn through interactions between human and chatbot and between the chatbots, where not only the single chatbot agent but also multiple intelligent chat agents work together as a team to help learners towards their educational problems. The purpose of developing such kind of systems is to reduce the agent's complexity for various domains and to observe the advantages of multiagents over a single chat agent in an interactive learning environment.

Therefore, in this research paper, we have proposed a framework which is an effective, simple, easy, and low-cost approach for designing text-based multi-interactive chatbots for education.

The proposed framework will be composed of major three modules: (1) user module, (2) agents module, and (3) module socket system. The user module will consist of the user interface through which the user will interact with the system, the agents module will consist of two chatbots that will respond to the user query, and the third module socket system will help the chat agents and users to communicate.

This paper is split up into six major parts: in Section 2, we will explore existing chatbots; Section 3 will demonstrate the framework along with its implementation, tools, and techniques that will be used to accomplish the research work; Section 4 will illustrate the results and discussion which will describe how do multichat agents simulate conversation in a quiz style on Outcome Based Education; Section 5 will explain the assessment of our multiagent chatbot communication framework which will demonstrate the accuracy of the system, and Section 6 will bring a conclusion.

2. Related Work

Various tutoring systems were built so as to provide an easy, quick, and all the time available learning environment to the learners.

Hilles and Naser [6] developed a tutoring system that teaches the fundamentals of the database known as MDB by using the ITSB authoring tool. The MDB teaches by conducting a series of lectures. To model the domain knowledge, the system had utilized constrained-based modeling and was capable of observing the actions of the students and learning capabilities.

Al-Hanjori et al. [7] designed and developed a computer-based coaching program that applies AI which helps students to learn computer networks. The system was developed using the ITSB tool. The system was able to provide educational

content by presenting intelligently based on the information level, knowledge of the subject, preferred level of details, and assessment level of the student.

Jia [8] had developed a CSIEC English tutoring system which consists of a chatbot that chats with English learners anytime and anywhere. According to the user input, context, domain information, and common-sense information and with logic, the response was being generated by the bot. NLML forms were used to express the entire knowledge.

Danforth et al. [9] developed an intelligent medical tutoring system that simulates actual patients reliably; the system allowed the students to interact with patients in a native language to get an appropriate record of patient, symptoms, and so forth; and the system makes up diagnoses along with an appropriate treatment based upon the disease where these students can perform their professional practices in a simulated environment.

Holotescu [10] developed an educational MOOCBuddy chatbot which provides a personalized educational environment with MOOCs. The purpose of the chatbot was to assist the users by discovering news regarding MOOCs. The tutors were also able to incorporate MOOCs in their courses.

Dutta [11] proposed a web-based intelligent tutoring chatbot tool that was developed to assist high school students in order to learn their general knowledge subjects. Different chatbot platforms were evaluated: "api.ai, wit.ai, Luis.ai, and Pandorabots." On the basis of the evaluation results, Dialogflow.com (Api.ai) was selected to develop the chatbot. The intelligent chatbot was capable of engaging in small talks with the learners. Mahdi et al. [12] developed an information security tutoring system with the help of the "ITSB" tool. It helps the new learning students to become a good professionals in the area of security.

Hiremath et al. [13] developed a chatbot system for education which provides responses to user queries regarding education. To make the system scalable and highly interactive, user-friendly chatbots were using its local database as well as web database to provide responses. The techniques that were used by the chatbot system were ML, NLP, data processing algorithms, and pattern matching to increase the performance.

Kumar and Rose [14] proposed Basilica architecture for developing conversational agents that support collaborative learning where several learners interact with agents. The architecture carried out object-situated. The authors have created three explicit conversational specialists that were being created utilizing this design.

Graesser et al. [15] developed a conversational intelligent auto tutoring system that was holding diverse initiative conversational dialogue. The system helps college-level students to study regarding computer literacy. The system enhances the learning process by presenting complex contents to the learner which is being answered by students in English.

Holland et al. [16] developed J-LATTE, an intelligent constrained-based tutoring system that was teaching a subset of the java programming language. The system was composed of two modes, concept mode and code mode with concept mode the students designing the programs instead

of specifying contents of statements, whereas, with the coding mode, the students complete the code.

Shawar and Atwell [17] introduced machine learning methods for an Arabic chatbot, where the user input is being accepted in Arabic and responses are extracted from Quran, a java program that reads text from the corpus and converts it into A. I.M.L format used by the ALICE.

Doshi et al. [18] developed an intelligent Android chatbot application that has used program-O which is an AIML interpreter for the generation of the responses of users' input. The system works on text and voice mode and the response generation process is carried out in two phases, (1) preparation of pattern matching and (2) pattern matching behavior, and the chat system can answer the questions which are already trained in its dataset.

Alencar and Netto [19] developed TUCUMA, an intelligent virtual agent that carries out the tracking of students in the virtual environment. The system acts as distance learning and monitoring actions of the students and eliminates uncertainties through conversation while the system is composed of multiagents where these agents were responsible for producing gestures and monitoring of students' activities.

3. Proposed Framework

The proposed framework is composed of several components which will help to achieve the desired results as shown in Figure 1.

3.1. User Module. The user module consists of the user and user interface through which they will interact with the system.

3.2. Chat Agents Module. The agent module of the proposed framework is based on two major components: (1) agent manager and (2) chat agent A and chat agent B.

The agent manager is responsible for managing the overall conversational process between user and chat agents. When a user enters the query in its chat window and clicks the send button the agent manager is responsible to broadcast the user query to chat agent A and chat agent B, once the response for a given query is being generated by the chat agents and sent back to the agent manager which then makes it visible for the user.

3.3. Chat Agent A and Agent B. Chat agents A and B are the chatbots which act as tutors that will respond to the user query related to the trained educational domain or we can say that these agents produce an optimal response according to the user query from their KB. However, these agents are domain experts consisting of KB which acts as the brain of the chatbots.

3.4. Chatbots Knowledge Base AIML. Basically, AIML is an XML specification for programming chatbots like ALICE

using program ab. AIML supports the development of high functioning chatbot system but it all depends upon how suspiciously we have mapped out the logic and conversational flow for our bot.

According to the Pandorabots report, over 300,000 chatbots have been developed using this platform. Pandorabots implement an AIML scripting language for the chatbots' development. However, the language powers the development of the most complex chatbots including multiaward winning, that is, Mitsuku chatbot [20]. AIML chatbots consist of AIML files.

AIML chatbots consist of the AIML files or we can say AIML KB is composed of AIML files which contain various AIML tags as mentioned in Figure 2. We can add the knowledge or train the agents by creating new files along with this AIML-based chatbot system extracting the responses from there.

The fundamental entity of knowledge inside AIML files is called categories where every category contains the user input which is a pattern and an output which is the response or template under which a given pattern is being matched and an optional context.

3.5. AIML Interpreter. Various AIML interpreters are there for different platforms, for example, Program O is an AIML interpreter for Android, Program ab is an AIML interpreter for Java, and so forth. AIML interpreters are capable of performing preprocessing functions in order to expand abbreviations, eliminate misspellings, and so forth. Before an AIML code is processed for the response generation, two important operations are being performed by the AIML interpreter program. These are (1) deperiodization and (2) normalization.

3.6. Graph Master. The Graph Master algorithm is being used by many AIML interpreters. Graph Master enables getting the most accurate or optimal time and provides control of stored memory.

In AIML, all categories are stored in a tree type, which is managed by an object called Graph Master as shown in Figure 3 while the Graph Master stores the AIML patterns along a path where each category is uniquely identified, which is from the root node "r" to a terminal node "t" where the AIML template is being stored.

3.7. AIML Pattern Matching. Basically, AIML employs the model of "pattern matching" or "pattern recognition"; AIML uses CBR case based reasoning where these cases are the categories the AIML finds the best match for a given pattern although the algorithm that the AIML implements for finding the best match for a given input is "k nearest neighbor classification" (KNN) as the model of learning within AIML is supervised learning [23].

Basically, in pattern recognition, the KNN algorithm calculates the distance between the test data and the given input where the input consists of k closest training

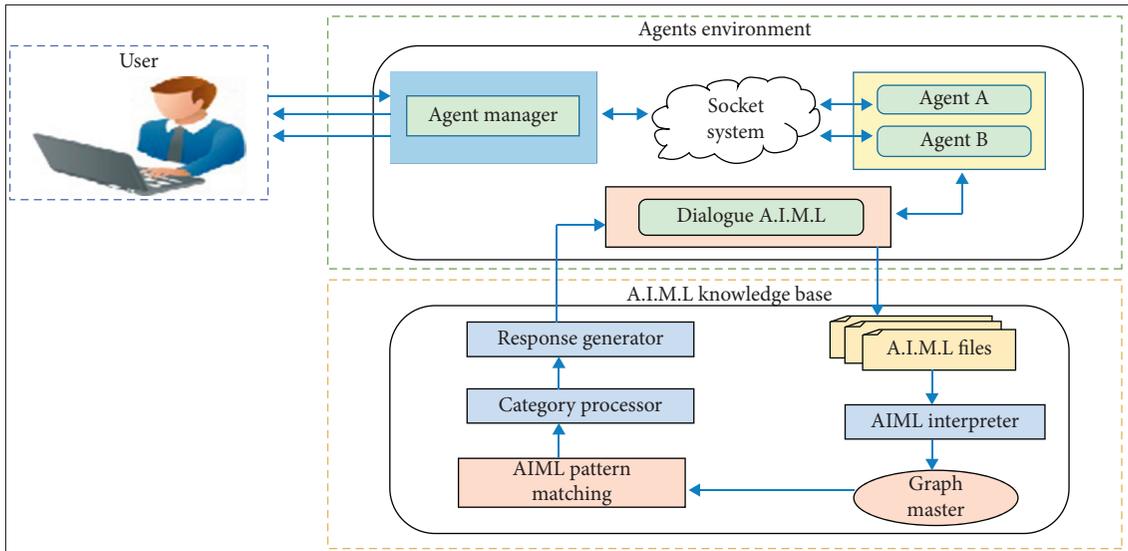


FIGURE 1: Multiagent chatbot communication framework.

<aiml>	Defines the beginning and end of an AIML document.
<category>	Defines the unit of knowledge in Alicebot's knowledge base.
<pattern>	Defines the pattern to match what a user may input to an Alicebot.
<template>	Defines the response of an Alicebot to user's input.
<star>	Used to match wildcard * character (s) in the <pattern> Tag.
<srai>	Multipurpose tag used to call/match the other categories.
<random>	Used <random> to get random responses.
	Used to represent multiple responses.
<set>	Used to set value in an AIML variable.
<get>	Used to get the value stored in an AIML variable.
<that>	Used in AIML to respond based on the context.
<topic>	Used in AIML to store a context so that later conversation can be done based on that context.
<think>	Used in AIML to store a variable without notifying the user.
<condition>	Similar to switch statements in a programming language. It helps ALICE to respond to matching the input.

FIGURE 2: Basic AIML tags [21].

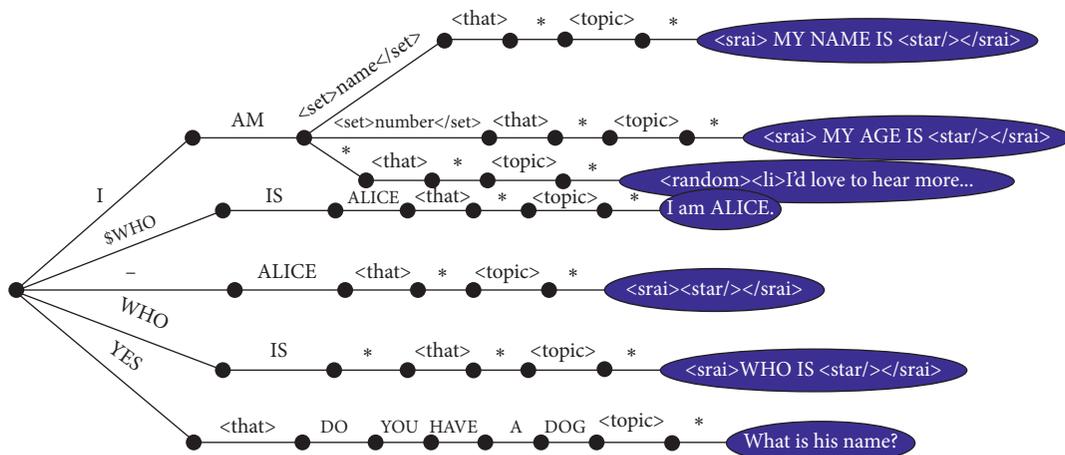


FIGURE 3: The working draft of an AIML Graph Master [22].

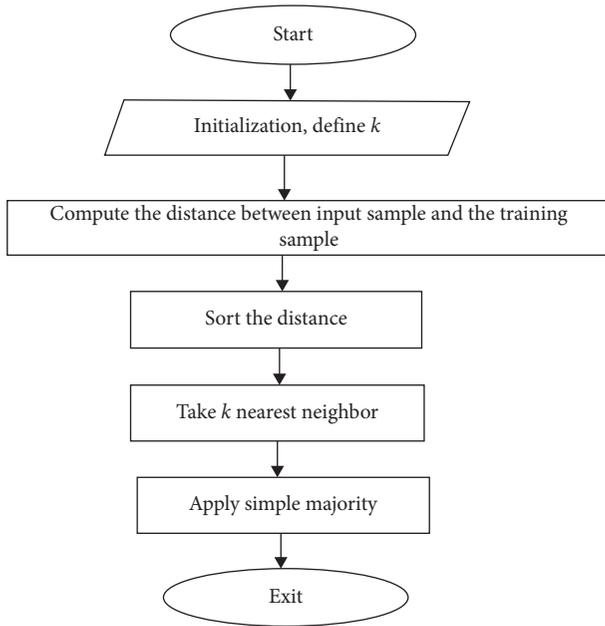


FIGURE 4: KNN classification algorithm [24].

examples and finds the closest match as shown in Figure 4.

3.8. Category Processor. Once the closest match is found for a given pattern, then the AIML category processor process categories within AIML files.

3.9. Response Generator. It reads the TEMPLATE and takes an appropriate action. The processing of dialog response which is being produced by a TEMPLATE diverges from being as simple as replicating the contents to the response [25].

3.10. Socket System. The client and server socket system establishes the Transmission Control Protocol (TCP) communication link among those Chat Agents in order to make them able to send and receive responses on a given port.

4. Results and Discussion

In this paper, we have presented an experimental study which analyzes how does multichat agents' communication will take place. Figure 5 shows the overall system flow chart; the system work flow chart is composed of various levels. The first step of the research flow chart is to collect the domain data in order to train our chatbot system once we collect the data for our desired domain; then, we prepare a possible list of questions with their corresponding responses.

After collecting domain information and preparation of questions and responses, the next level of the flow chart is to prepare an appropriate KB for our chatbot agents. Once the

data training is completed, then, in the next level, we establish the communication link among those chat Agents in order to make them capable of sending and receiving information.

The next level of the flow chart is to generate the responses these chat agents use their KB and generate an appropriate response for a given input query. The last level of the flow chart is that we have performed testing on various data inputs so that to measure the accuracy and efficiency of our chat agents.

4.1. Data Description. The data for this study have been collected from various sources such as books, articles, websites, and OBE experts while the total training data sets are 500 categories as shown in Table 1.

The proposed system provides learners an interactive learning environment with multichatbot tutoring agents for solving their uncertainties.

The multichatbot system consists of the window with the text area reflecting the conversation and a textbox to introduce new requests so that the learners should be able to maintain the conversation with the proposed system to get better accurate results from this system.

Figure 6 shows the typical interface of a system where text-based conversation among the user, agent manager, and chat agent 1 and chat agent 2 and then accurate responses have been provided from chat agent 1 and chat agent 2 as shown in Figures 7 and 8.

5. System Evaluation

The multichatbot systems have been evaluated by performing testing on a range of inputs. Testing sample size, we have considered the "last 100 user messages to our chatbot system" using logs being marinated by the proposed system and the accuracy and other factors are being judged based on the responses generated by the chatbot system for the given user inputs. However, the system is being evaluated using the confusion matrix, which is a performance measurement tool for ML. Basically, it is a type of table which contains 4 diverse groups of predicted and actual values, and it is quite appropriate in this case, as the user's input will be classified and matched with the nearest matching category [26, 27].

In this scenario, the confusion matrix provides a way to reason about accuracy for our chatbot system. Let us classify the TP, TN, FP, and FN in terms of chatbot analogy. Correct mapping refers to the user's input been matched with the expected, or with proper category.

True positive which refers to a user input being matched to an expected category in the approved manner

True negative which refers to the user query not matched with an expected category as no such category is defined instead of generating nothing, it generates the default response

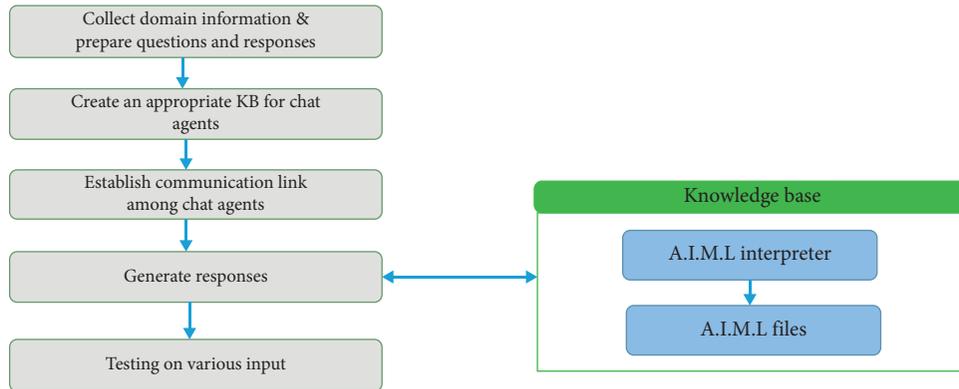


FIGURE 5: System development flow chart.

TABLE 1: Data description.

Total Categories	Total file	Total nodes	Total singletons	Total leaves	Total Branches
500	350 KB	2654	1954	453	2653

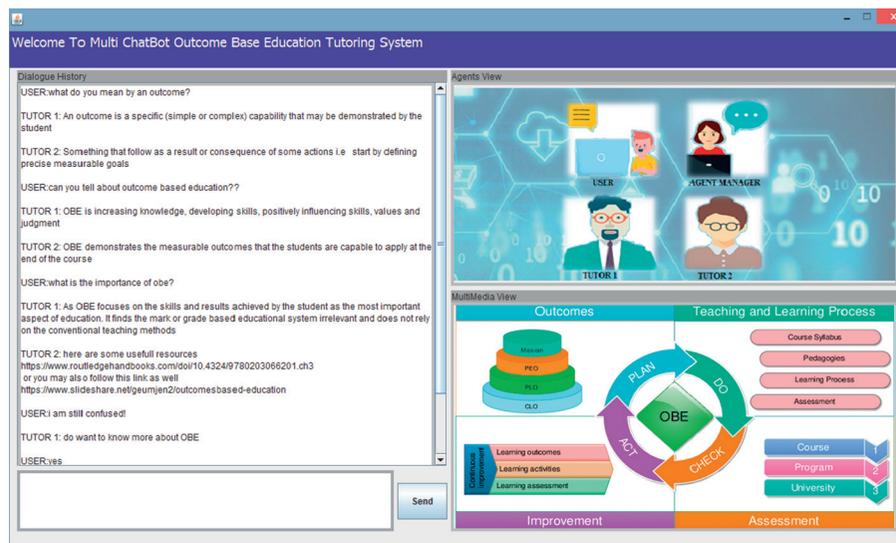


FIGURE 6: Conversational flow among chat agents and users.

False positive which refers to a user query matched either with another category or with the default category because we did not yet handle that phrase

False negative which refers to the user query not matched with the expected category as we have already defined a category to handle that phrase which means it does not work properly on that input

Figure 9 shows the total TP, TN, FP, and FN after the analyzing last 100 messages to our system.

Finally putting the given values for TP, TN, FP, and FN in a confusion matrix as shown in Figure 10, the results we got are as shown in Figure 11 while Figure 12 shows the overall accuracy of the responses generated by these chatbots.

5.1. System Security. The system utilizes the Java secure sockets layer (SSL) socket class in order to provide a secure communication link among the chat agents. However, SSL provides security using protocols such as the SSL or IETF Transport Layer Security (TLS) protocols.

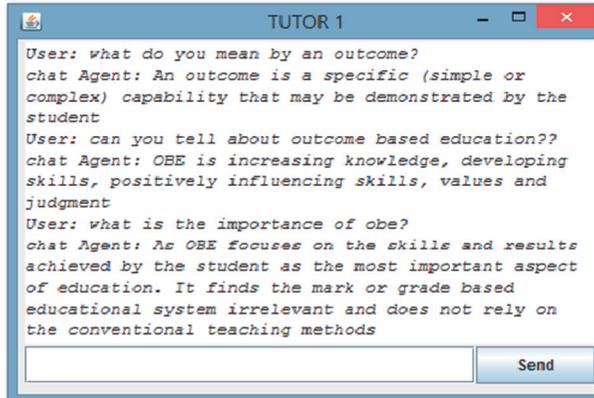


FIGURE 7: Response generation by Tutor 1.

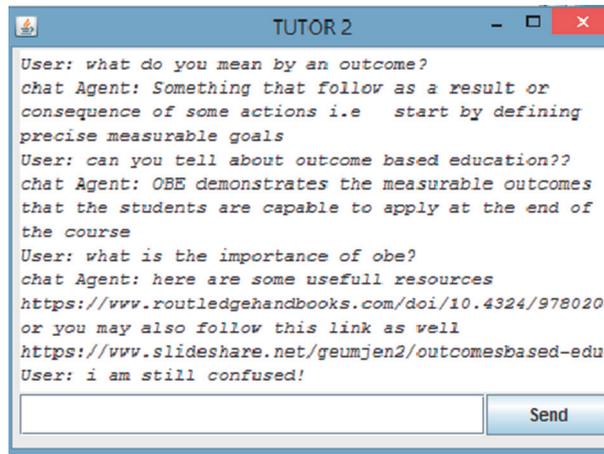


FIGURE 8: Response generation by Tutor 2.

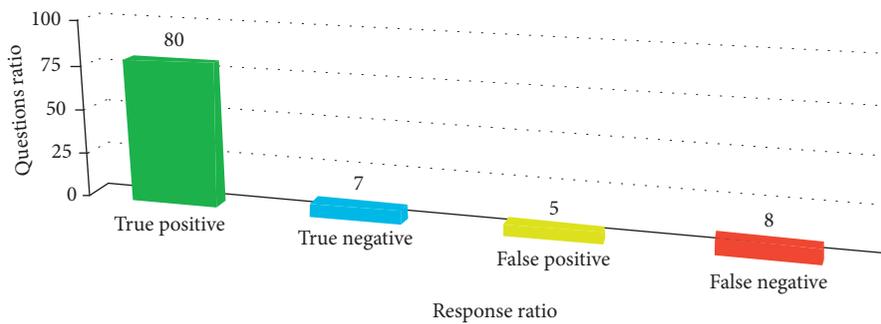


FIGURE 9: The total proportion of TP, TN, FP, and FN.

	True positive	True negative
Predicted positive	80	7
Predicted negative	5	8

FIGURE 10: The values which are inserted in the tool.

Measure	Value	Derivations
Sensitivity	0.9412	$TPR = TP / (TP + FN)$
Specificity	0.5333	$SPC = TN / (FP + TN)$
Precision	0.9195	$PPV = TP / (TP + FP)$
Negative predictive value	0.6154	$NPV = TN / (TN + FN)$
False positive rate	0.4667	$FPR = FP / (FP + TN)$
False discovery rate	0.0805	$FDR = FP / (FP + TP)$
False negative rate	0.0588	$FNR = FN / (FN + TP)$
Accuracy	0.8800	$ACC = (TP + TN) / (P + N)$
F1 score	0.9302	$F1 = 2TP / (2TP + FP + FN)$
Matthews correlation coefficient	0.5038	$TP * TN - FP * FN / \sqrt{((TP + FP) * (TP + FN) * (TN + FP) * (TN + FN))}$

FIGURE 11: The results generated by the confusion Matrix tool [28].

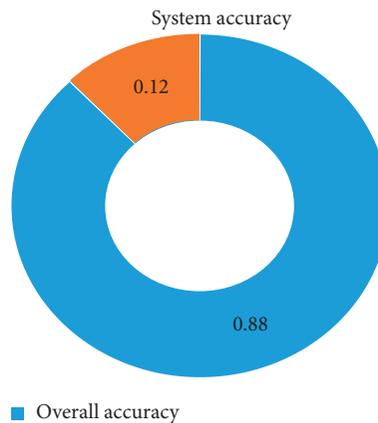


FIGURE 12: The overall accuracy of the system.

6. Conclusion

In this paper, we have proposed the framework with a novel approach that supports the development of a multi-interactive chatbot's system for an educational area using A.I.M.L 2.0. The system will facilitate the students for their learning towards an outcome-based education domain. The proposed system composed of user module consists of user and user interface, chat agents module which will respond to the user query, chatbot KB acting as the brain for the chatbot system, and socket system for establishing the communication link. The system has been evaluated by analyzing the last 100 user messages to our chatbot system. Using the confusion matrix, we found promising results with an overall accuracy of about 88%. By and large, chatbots can be utilized to give essential lectures. The goal is that chatbots can fill in as virtual tutors and that in the process they adjust to the capabilities of the students. However, it has been implemented in java. The

outcomes of this research will be useful for the education sector where these intelligent systems will help the students in schools, universities, and other training scenarios. The emergence of artificial intelligence applications, such as text-based virtual assistants (chatbots) especially in education, is a very new field. These types of systems can be useful in helping teachers and students to solve their educational problems and routine tasks efficiently. This article basically is the foundation for designing and developing the chatbots for educational systems and may provide justifiable results when applying in a particular situation in future implementations [29].

7. Future Work

In this study, these multichatbots systems can respond to text-based dialogues only but, in the future, these chat agents will interpret voice commands as well as images.

Furthermore, we can expand the scope of these chatbots for various domains as well. We can implement this framework in a mixed real environment.

Data Availability

The authors have used their own data. Data can be obtained through contacting jalbaniakhtar@gmail.com.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- [1] A. Schlesinger, K. P. O'Hara, and A. S. Taylor, "Let's talk about race: identity, chatbots, and AI," in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pp. 1–14, Montreal, QC, Canada, 2018 April.
- [2] B. A. Shawar and E. Atwell, "Chatbots: are they really useful?" *LDV Forum*, vol. 22, no. 1, pp. 29–49, 2007.
- [3] A. Kerlyl, P. Hall, and S. Bull, "Bringing chatbots into education: towards natural language negotiation of open learner models," in *Proceedings of International Conference on Innovative Techniques and Applications of Artificial Intelligence*, pp. 179–192, Springer, London, UK, 2006 December.
- [4] A. A. Georgescu, "Chatbots for education—trends, benefits and challenges," in *Proceedings of the Conference proceedings of eLearning and Software for Education (eLSE)*, vol. 2, no. 14, pp. 195–200, "Carol I" National Defence University Publishing House, Bucharest, Romania, 2018.
- [5] Z. Peng and X. Ma, "A survey on construction and enhancement methods in service chatbots design," *CCF Transactions on Pervasive Computing and Interaction*, vol. 1, no. 3, pp. 204–223, 2019.
- [6] M. M. Hilles and S. S. A. Naser, "Knowledge-based intelligent tutoring system for teaching mongo database," *European Academic Research*, vol. 4, no. 10, pp. 8783–8794, 2017.
- [7] M. M. Al-Hanjori, M. Z. Shaath, and S. S. A. Naser, "Learning computer networks using intelligent tutoring system," *International Journal of Advanced Research and Development*, vol. 2, no. 1, 2017.
- [8] J. Jia, "CSIEC: a computer assisted English learning chatbot based on textual knowledge and reasoning," *Knowledge-Based Systems*, vol. 22, no. 4, pp. 249–255, 2009.
- [9] D. R. Danforth, M. Procter, R. Chen, M. Johnson, and R. Heller, "Development of virtual patient simulations for medical education," *Journal For Virtual Worlds Research*, vol. 2, no. 2, 2009.
- [10] C. Holotescu, *MOOCBuddy: a Chatbot for Personalized Learning with MOOCs*, RoCHI, Pune, India, 2016.
- [11] D. Dutta, *Developing an Intelligent Chat-Bot Tool to Assist High School Students for Learning General Knowledge Subjects*, Georgia Institute of Technology, Atlanta, Georgia, 2017.
- [12] A. O. Mahdi, M. I. Alhabbash, and S. S. A. Naser, "An intelligent tutoring system for teaching advanced topics in information security," *World Wide Journal of Multidisciplinary Research Arch and Development*, vol. 2, no. 12, pp. 1–9, 2016.
- [13] G. Hiremath, A. Hajare, P. Bhosale, R. Nanaware, and K. S. Wagh, "Chatbot for education system," 2018.
- [14] R. Kumar and C. P. Rose, "Architecture for building conversational agents that support collaborative learning," *IEEE Transactions on Learning Technologies*, vol. 4, no. 1, pp. 21–34, 2011.
- [15] A. C. Graesser, K. VanLehn, C. P. Rosé, P. W. Jordan, and D. Harter, "Intelligent tutoring systems with conversational dialogue," *AI Magazine*, vol. 22, no. 4, p. 39, 2001.
- [16] J. Holland, A. Mitrovic, and B. Martin, *J-LATTE: A Constraint-Based Tutor for Java*, University of Canterbury, Christchurch, New Zealand, 2009.
- [17] A. Shawar and E. S. Atwell, "An Arabic chatbot giving answers from the Qur'an," in *Proceedings of TALN04: XI Conference sur le Traitement Automatique des Langues Naturelles*, vol. 2, pp. 197–202, Fez, Morocco, April 2004.
- [18] S. V. Doshi, S. B. Pawar, A. G. Shelar, and S. S. Kulkarni, "Artificial intelligence chatbot in android system using open source program-O," *Artificial Intelligence*, vol. 6, no. 4, 2017.
- [19] M. Alencar and J. F. Netto, "Tutor collaborator using multi-agent system," in *International Conference on Collaboration Technologies*, pp. 153–159, Springer, Berlin, Heidelberg, 2014 September.
- [20] A. Khanna, *Pandorabots Chatbot Hosting Platform*, SARANG Bot, Ratnagiri, Maharashtra, India, 2015.
- [21] <https://medium.com/@pemagr/aiml-tutorial-a8802830f2bf%20> (Last Accessed on 20-10-2019).
- [22] <https://gist.github.com/onlurking/f6431e672cfa202c09a7c7cf92ac8a8b> (Last Accessed on 21-4-2019).
- [23] S. Zhang, X. Li, M. Zong, X. Zhu, and R. Wang, "Efficient knn classification with different numbers of nearest neighbors," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 5, pp. 1774–1785, 2017.
- [24] <https://www.slideshare.net/15koolneha/knearest-neighbor-classifier-74464857%20> (Last Accessed on 21-4-2019).
- [25] R. Jindal, R. Kumar, R. Sahajpal, S. Sofat, and S. Singh, "Implementing a natural language conversational interface for Indian language computing," *IETE Technical Review*, vol. 21, no. 4, pp. 243–250, 2004.
- [26] <https://towardsdatascience.com/understanding-confusion-matrix%20a9ad42cdf62?gi=6702c33e3ed6> (Last Accessed on 20-3-2019).
- [27] <https://miningbusinessdata.com/how-do-you-measure-your-dialogflow-bots%20accuracy> (Last Accessed on 21-3-2019).
- [28] <http://onlineconfusionmatrix.com> (Last Accessed on 22-3-2019).
- [29] R. Chocarro, M. Cortiñas, and G. Marcos-Matás, "Teachers' attitudes towards chatbots in education: a technology acceptance model approach considering the effect of social language, bot proactiveness, and users' characteristics," *Educational Studies*, vol. 40, pp. 1–19, 2021.