ChatbotLLM – Training Educational Chatbots on the Materials Uploaded by Teachers

Supun De Silva
School of Science and Technology
Athabasca University
Edmonton, Alberta
sdesilva2@learn.athabascau.ca

Maiga CHANG
Faculty of Science and Technology
Athabasca University
Athabasca, Canada
maiga.chang@gmail.com
*corresponding author

***Abstract*—This paper presents an IEEE Northern Canada Section's Capstone Project Award winning research project, ChatbotLLM, a system to train educational chatbots on the materials uploaded by teachers. Using ChatbotLLM, teachers can create a course, choose the desired configuration options, upload the materials, and start to train an educational chatbot on the uploaded course materials. Once a chatbot is finished training, both teachers and students can test and interact with the chatbot with their browsers as well as access it through any other available client applications. This research aims to overcome shortcomings of educational technologies using generative AI, mainly bias and hallucinations. Currently an evaluation is underway using a university level course at UK to understand and compare user perceptions towards responses provided by ChatbotLLM chatbots and other generative AI chatbots such as ChatGPT.**

Keywords—Natural Language Processing, Extractive Text Summarization, Cosine Similarity, Rephrasing, Web Scraping, Large Language Models, Generative AI, Optical Character Recognition

I. INTRODUCTION

Ilieva and colleagues (2023) identify the lack of interactivity through engagement, real-time feedback, and personalized learning experiences as three main issues of learning technologies used at universities [1]. Educational chatbots using Generative Artificial Intelligence (GenAI) technologies can help overcome these issues. Chatbots are conversational agents that use text or voice interfaces to communicate using natural language with users [2]. They can facilitate real-time feedback, interaction, and personalized learning using natural human-like language [3].

In addition, the presence of a pedagogical conversational agent (PCA), such as a chatbot, can help improve students' learning through three named effects: the Persona effect, the Proteus effect, and the Protégé effect [3]. The Persona effect refers to the phenomena that students perceive their learning experience more positively when a PCA is present [4]. The Proteus effect refers to the phenomena that students are motivated to try to become similar to their PCA in features such as wording their responses, approach to solving a problem, etc. [5]. Lastly, the Protégé effect refers to the greater efforts a student will put to teaching their PCA educational concepts than teaching themselves those same educational concepts [6]. Therefore, there is great interest in educational chatbots using GenAI technology.

However, GenAI is not without its shortcomings such as biases and hallucinations [7]. Bias refers to responses generated by such a chatbot that might favor or disfavor a person or a group [8]. Hallucination refers to when a response from such a chatbot incorporates fake information, facts, situations, etc. [9]. Due to these limitations, care must be utilized when adopting GenAI chatbots.

The research team developed ChatbotLLM, an open-access system to train educational chatbots. Chatbots trained by ChatbotLLM utilize extractive text summarization to respond to user questions based on the uploaded material by the teachers to overcome the shortcomings of educational chatbots using GenAI. Extractive text summarization is a type of summarization that uses the words, sentences, or paragraphs already present in the content to generate a summary [10]. Generative AI will still be utilized; however, to only format the response sent to the user to be more human-like while not changing any of the keywords or meanings of the output generated through extractive text summarization.

The rest of the paper is structured as follows. Section 2 contains a literature review about chatbots and educational chatbots. Section 3 introduces the architecture of the ChatbotLLM system developed by the research team. Section 4 explains the server-side processes of creating a chatbot and interacting with a trained chatbot. Section 5 introduces how a teacher could use ChatbotLLM to train their own educational chatbot. At the end, Section 6 discusses the potential future works this research has.

II. LITERATURE REVIEW

Chatbots are computer programs designed to enable human-computer conversation. They have been in use for over half a century. The first chatbot created was ELIZA, developed by Weizenbaum in 1966, which used pattern matching to emulate a psychotherapist engaging in dialogue with human users [11]. In the mid-1990s, ALICE (Artificial Linguistic Internet Computer Entity) was introduced, leveraging Artificial Intelligence Markup Language (AIML) to parse user inputs and generate appropriate responses based on a predefined set of knowledge records [12]. Another notable chatbot that preceded modern virtual assistants like Alexa and Siri is SmarterChild [13]. These chatbots offered conversational capabilities on messaging platforms and provided users with quick access to information and services. At present, chatbots can be deployed

to messaging applications such as Slack or Discord, created as standalone web applications, such as ChatGPT, or integrated into existing applications such as Duolingo [14].

Educational chatbots have been in use since the early 1970s to help guide learners through educational content and to engage with them in dialog [15]. There are many ways in which educational chatbots can engage with learners in dialog including text-based, voice-based, and non-verbal [16]-[18]. Text-based communication will involve the user and the chatbot communicating through typed text. Voice-based chatbots communicate using spoken voice and is more accessible to older and some people with special needs [19]. Non-verbal communication from chatbots will be in the form of facial expressions or display of emotions to user input [20]. This requires the chatbot to be embodied as a cartoon character or a human.

III. System Architecture

The ChatbotLLM is a web application built using HTML, CSS, JavaScript, PHP, and Python to train educational chatbots for teachers. In addition, it utilizes libraries and application programming interfaces (APIs) such as OpenAI and Google Gemini for output rephrasing, Tesseract for optical character recognition (OCR), Google's Speech Recognition for transcription, Sentence Transformer to capture relationships between words that form a sentence, etc. It is hosted on the cloud and accessible to users at https://chatbot.vipresearch.ca/.

Teachers can use an online dashboard to create a course and to upload the course materials for training chatbots. The uploaded materials are then sent to the server for storage and processing. A chatbot will begin training or be queued to train if there are other chatbots that are currently being trained. After creating a course, teachers can choose that course to upload course materials. A zip file containing multiple files within a single folder can be uploaded or the individual files can be uploaded one at a time. Document formats accepted include DOC(X), PPT(X), XLS(X), MP3, PDF, and TXT. If any unsupported document formats exist within an uploaded zip file, they will be ignored.

Fig. 1 outlines the architecture of the ChatbotLLM system. When a course is created and the course materials are uploaded, the "chatbot creation process" takes actions. The process needs to be done by five modules working together, they are: convert-to-text, extract-urls, crawl-and-extract, enrich, and data-preparation. The convert-to-text module identifies and saves all text from various document types and then extract-urls module discovers any links existed. With those links, the crawl-and-extract module visit the correspondent webpages scraping the texts. At the end the enrich module is house keeping the texts and the data-preparation module helps to prepare the dataset for chatbot training. The important libraries the ChatbotLLM adopts are highlighted in Fig. 1.

Once the chatbot is trained, users can interact with the chatbot. To interact with the chatbot, a model ID (that contains the course ID) and a question will be required. When ChatbotLLM receives these input, Cosine Similarity will be used to find the sentence that is most similar to the user question. ChatbotLLM then asks for the selected model's help to obtain the most similar paragraphs. If no sentence matches the question, then ChatbotLLM will respond saying that it cannot answer this question. Afterwards, the extractive text summarization output based on the most similar paragraphs will be sent to GenAI for rephrasing if the course is not opted out to use GenAI.

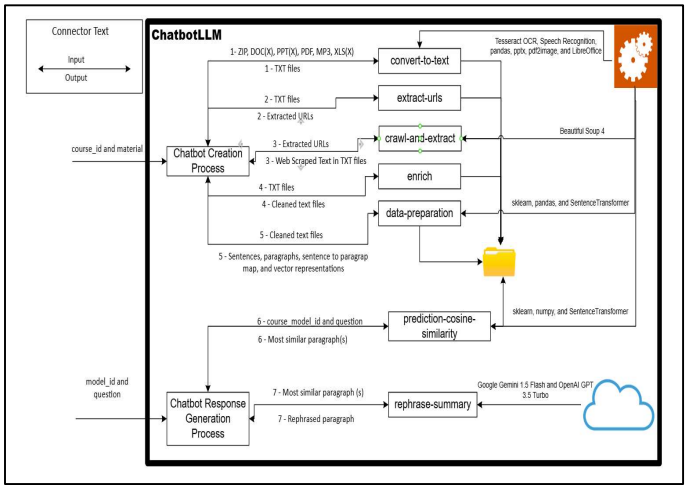

Fig. 1. ChatbotLLM system architecture

IV. Operation

When teachers create a course and upload the course materials, which could be in various formats such as PDF, PPT(X), DOC(X), XLS(X), MP3, or TXT. All PPT(X), DOC(X), and XLS(X) files are converted to PDFs before extracting text so that all files can be handled in a uniform manner. The LibreOffice command line application in headless mode is used to convert these files to PDFs on the server. When considering PDF files, there are three cases that can occur: the PDF file could be image-based, text-based, or a combination of the two. To handle all these types of PDF files uniformly, all PDF files are converted to image-based PDF files and then OCR is performed to extract text using Tesseract. MP3 files are transcribed to text using Google's Speech Recognition library. Any files uploaded as TXT files do not need any processing and can be directly saved to the server.

ChatbotLLM first extracts text from all the uploaded course files as Fig. 2 shows, so that the data is in a consistent format that is easy to work with programmatically.

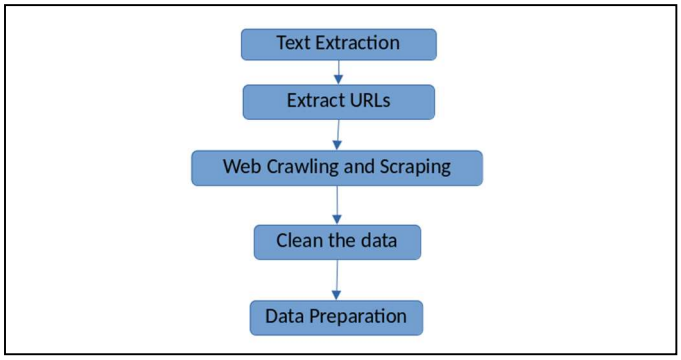

Fig. 2. The entire workflow to prepare the chatbot once an instructor uploads all the course material

Regular Expression (regex) parsing is then used to extract the URLs found in the course materials into text files (i.e., Step 2 in Fig. 2). Those URLs are stored in a separate text file for each course and each URL is scraped to a depth of 1 to obtain text on those pages. The depth refers to the nesting level. The current page has a depth of 0 and pages linked to the current page will have a depth of 1. Therefore, ChatbotLLM scrapes the text from the current page and any pages linked on this page (i.e., Step 3 in Fig. 2). The scraped text is saved in text files with filenames that are MD5 hashes of the URL.

Step 4 is cleaning the data. Cleaning the data involves removing excess whitespace, tabs, and new lines at the beginning and end of each line, ensuring words are separated by only one whitespace, and removing any non printable characters from the text.

The final step is the data preparation step. Data preparation includes extracting the sentences and paragraphs from the corpus, creating a file that maps a sentence to the paragraph it appears in, and creating a file containing the vector representations of each sentence to be used with Cosine Similarity. The sentence transformer model, all-MiniLM-L6-V2, is used to create the vectors of text. Creating the vector representations of sentences is mainly for efficiency so that we can look them up in this file when a question is asked rather than manually recalculating the vector representations repeatedly every time for a question. Storing the vector representations in a file sped up the chatbots response by over 55x for a university level course – calculating the vector representations and providing a response took 672 seconds while using the pre-saved vectors took 12 seconds to provide a response.

The entire chatbot preparation process will keep track of which files it has already processed for each course and will not process those files again. This allows teachers to not have to upload all the files at the same time. In addition, if a new file is uploaded it will be processed in the next minute if there is no queue of other files to be processed on ChatbotLLM.

The Cosine Similarity model does not require any additional training apart from the files created in the data processing step. It requires the sentences files, paragraphs files, the sentence-to-paragraph map, and the vector representations of the text. On the other hand, the BERT and Sentence Transformer models will require time to train. As the name suggests, they will use BERT and Sentence Transformer as the body and a simple neural network that performs text classification as the head to utilize transfer learning. The text classification neural network will train the model to provide a number given a sentence as input. The output number can be mapped to a paragraph.

When either Cosine Similarity model is ready or a chosen neural network model has been trained completely, users like teachers and students can interact with the chatbot (i.e., a trained model). There are several key steps for the chatbot creating the response to a question. The first step is using Cosine Similarity to find the most similar sentence from the pre-saved vectors. Afterwards, the chatbot asks for paragraph from a trained model with the sentence. Due to the lack of punctuation in text extracted from MP3 files as well as PPT(X) and XLS(X) files, ChatbotLLM at this moment considers 100 words to be a paragraph. Therefore, there is no guarantee that a paragraph returned by a trained model will start at a new sentence or end after a complete sentence.

The incomplete sentences of a 100-word paragraph returned by a model will reduce the quality of the response. To prevent this ChatbotLLM uses GenAI such as Google's Gemini 1.5 Flash (by default) and OpenAI's GPT 3.5 Turbo to help rephrasing the paragraph, if GenAI rephrasing is not opted out for the course. ChatbotLLM uses prompt to ensure that the meaning and the key terms of the paragraph will not be changed by the GenAI while rephrasing. The rephrased paragraph will be human-like content shown to the user as the chatbot's response. Such response example can be seen in Section IV.

V. Using ChatbotLLM as an Educator

Fig. 3 shows the ChatbotLLM[1] website. The Management Dashboard button allows teachers to create a course, configure the course and chatbot options, and upload the course materials for training educational chatbots. A course can have more than one chatbot trained on the uploaded materials with different settings. The Training Progress Dashboard button allows teachers and students to interact with a model once it has finished training. The time taken to train a chatbot will depend on the amount of course materials. It may take a couple of hours or a day or two for a university level course. Therefore, an instructor can upload course materials and start the chatbot creation in the weekend before the semester starts.

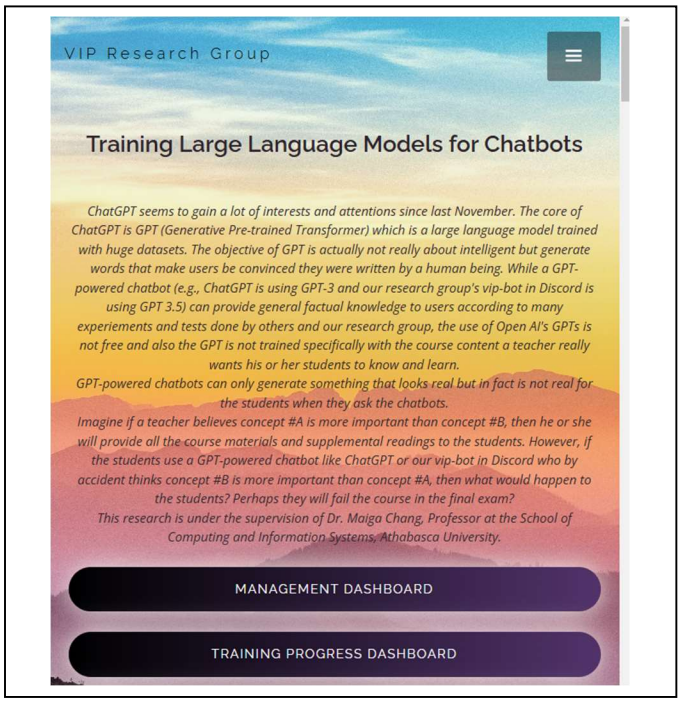

Fig. 3. ChatbotLLM Research Website

The first step in creating an educational chatbot for teachers would be to upload the course materials. The Management Dashboard button will be used to create a course, configure a

[1] https://chatbot.vipresearch.ca

course and chatbot options, and upload the course materials. The teachers need to use "Add Course" panel on the dashboard (see Fig. 4) to create a course and the form to upload course material. To create a course, a course id and course name are required. All the other options can be left as they have default options provided for good results. However, teachers can tweak and experiment with different options to see how they affect the output from a trained chatbot.

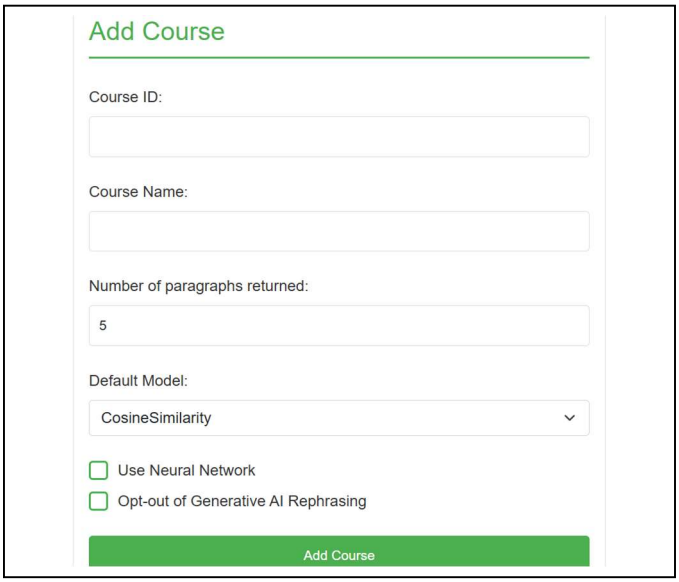

Fig. 4. The panel allows teachers to create a course.

The option "number of paragraphs returned" specifies how many paragraphs will be returned by the chatbot when there are many paragraphs that sufficiently answers the question asked by the student. The default value is set to 5. The "default model" option allows the teachers to specify which model will be used by client applications when they are using ChatbotLLM service, such as the ChatbotLLM's built-in Training Progress Dashboard[2], Discordbot VIP-Bot [2], and the Visualized Editing Environment [3]. Currently the Cosine Similarity model is automatically created and chosen by default. More models could be added for teachers to choose from and experiment with.

The final two options ask teachers whether they would like to use a neural network and whether they would like to opt out of using GenAI to rephrase the paragraphs a chatbot identified for a question. At present, ChatbotLLM supports the neural network model training on the smaller size of course materials, therefore, by default "the use of neural network" option is unchecked and Cosine Similarity model will be automatically created for any course created. When the option is checked, then in addition to the Cosine Similarity model a default neural network model will be also trained.

The final option deals with using GenAI rephrasing to rephrase the output generated (i.e., identified paragraph for extractive text summarization) to be more human-like. By default, the output generated will be rephrased by sending it to a GenAI model such as Google's Gemini 1.5 Flash or OpenAI's GPT 3.5 Turbo to rephrase it to a single coherent paragraph for returning to the students. If teachers choose to opt it out, then no rephrasing will be performed and the returned paragraphs will be presented to the students as is. Afterwards, the teachers can scroll down to the "Upload Course Material" panel (see Fig. 6). They can select the course ID they just created from the dropdown list and upload a zip file containing all the course material or upload individual files one-by-one. Fig. 5 shows the dropdown list with course IDs a teacher can choose from after creating their course. After this step is complete, ChatbotLLM will start the process of training a chatbot on the uploaded materials.

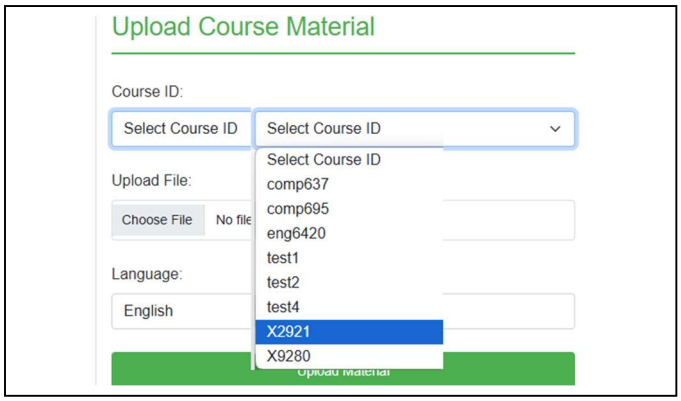

Fig. 5. The panel allows teachers to upload course material for the created course.

Giving a couple of hours to a day or two for the chatbot to be trained, depending on the amount of course material, the teachers can visit the Training Progress Dashboard to see if a chatbot for their course is available (see Fig. 6). A model that is finished training can be selected from the dropdown list and the teacher/student can ask a question from the model by typing it in the textbox provided and pressing the submit button. The model will respond with the answer below.

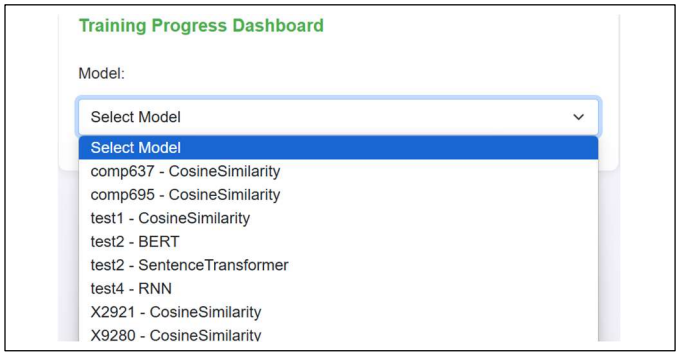

Fig. 6. The training progress dashboard to interact with trained chatbots

If a chatbot is not yet trained, then the teachers cannot see the model from the dropdown list and can check back at a later time. If a chatbot is trained, then the teachers can select the chatbot and ask a question in the textbox provided. Fig. 7 shows the chatbot (i.e., the trained Cosine Similarity model on the X9280 course at an UK university) responds to a question, "What is the self determination theory?". In this manner,

[2] https://chatbot.vipresearch.ca/training_progress_dashboard.php

[3] https://vp.vipresearch.ca/

teachers can create their own educational chatbot and test its performance before incorporating it with available client applications for easy use by their students.

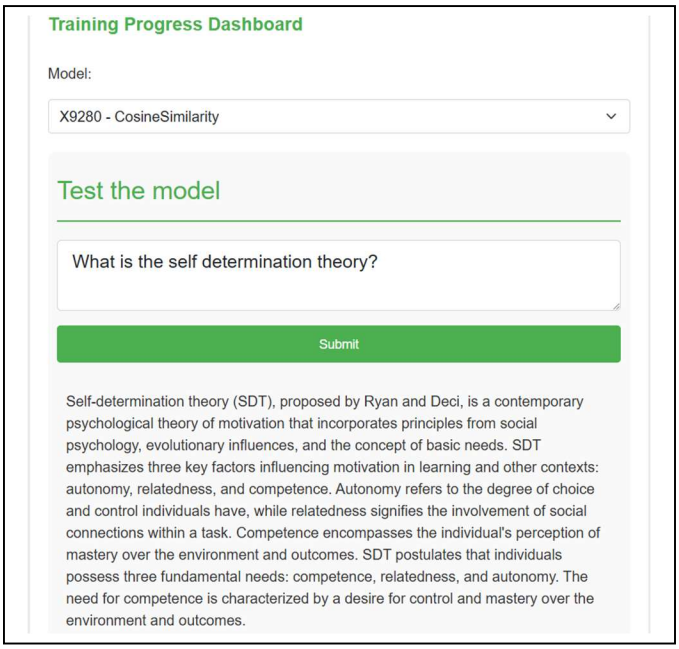

Fig. 7. Interacting with a trained model using the Training Progress Dashboard

VI. Future Work

Currently an evaluation is underway at an UK University using the ChatbotLLM system for their "Understanding Education in the 21st century" course. The goal of this evaluation is to compare user perception between a chatbot trained with the proposed ChatbotLLM and the OpenAI's GPT 3.5 turbo model.

Furthermore, ChatbotLLM plans to incorporate retrieval augmented generation (RAG) to generate better summaries. With this change, the user's question, the identified most similar paragraphs, and a prompt will be sent to a GenAI such as OpenAI's GPT or Google's Gemini to generate a summary which will be returned to the user.

There are some limitations that will be addressed in the future. One such limitation is that text extracted from MP3 files through transcription lack any punctuation. The effect of this is that if GenAI rephrasing is not used, the response may not start at the beginning of a sentence or end following a period. Another shortcoming at present occurs in the URL extraction phase. Since any XLS(X), DOC(X), and PPT(X) files are converted to PDF files for text extraction and then all PDF files are converted to image-based PDF files to handle all the text extraction uniformly, some of the extracted URLs will be malformed and a 404 status will be reached when attempting to scrape the text from them.

Acknowledgment

The research team wants to thank and acknowledge the support by: NSERC Discovery Grant, NSERC Discovery Development Grants and Athabasca University-Alberta Innovate (AU-AI) Summer Research Studentship. Without their support, this research would not have been possible.

References

[1] G. Ilieva, T. Yankova, S. Klisarova-Belcheva, A. Dimitrov, M. Bratkov, and D. Angelov, "Effects of generative chatbots in higher education," Information, vol. 14, p. 492, 2023, doi: 10.3390/info14090492

[2] S. A. Nikou, A. Guliya, S. Van Verma and M. Chang. (2024). "A Generative Artificial Intelligence empowered chatbot: System usability and student teachers' experience," 20th International Conference on Intelligent Tutoring Systems (pp. 330-340), Thessaloniki, Greece, June 10-13, 2024. https://doi.org/10.1007/978-3-031-63028-6_27

[3] D. Pérez-Marín, "A review of the practical applications of pedagogic conversational agents to be used in school and university classrooms," Digital, vol. 1, no. 1, pp. 18–33, 2021. doi: 10.3390/digital1010002

[4] J. Lester, S. Converse, S. Kahler, S. Barlow, B. Stone, and R. Bhogal, "The persona effect: Affective impact of animated pedagogical agents," in Proc. SIGCHI Conf. Human Factors in Computing Systems, Atlanta, GA, USA, Mar. 1997, New York, NY, USA: ACM, 1997

[5] N. Yee and J. Bailenson, "The Proteus effect: The effect of transformed self-representation on behavior," Hum. Commun. Res., vol. 33, pp. 271–290, 2007

[6] C. Chase, D. Chin, M. Oppezzo, and D. Schwartz, "Teachable agents and the Protégé effect: Increasing the effort towards learning," J. Sci. Educ. Technol., vol. 18, pp. 334–352, 2009

[7] F. F. Nah, R. Zheng, J. Cai, K. Siau, and L. Chen, "Generative AI and ChatGPT: Applications, challenges, and AI-human collaboration," Journal of Information Technology Case and Application Research, vol. 25, no. 3, pp. 277–304, 2023, doi: 10.1080/15228053.2023.2233814

[8] E. Ntoutsi, P. Fafalios, U. Gadiraju, V. Iosifidis, W. Nejdl, M. E. Vidal, et al., "Bias in data-driven artificial intelligence systems−An introductory survey," Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, vol. 10, no. 3, pp. 1-14, 2020, doi: 10.1002/widm.1356

[9] Z. Ji, N. Lee, R. Frieske, T. Yu, D. Su, Y. Xu, et al., "Survey of hallucination in natural language generation," ACM Computing Surveys, vol. 55, no. 12, pp. 1–38, 2023, doi: 10.1145/3571730

[10] A. P. W. Widyassari, S. R. Rustad, G. F. Shidik, E. Noersasongko, A. Syukur, A. Affandy, and D. R. M. Setiadi, "Review of automatic text summarization techniques & methods," J. King Saud Univ. - Comput. Inf. Sci., vol. 34, no. 4, pp. 1029–1046, 2022. doi: 10.1016/j.jksuci.2020.05.006

[11] J. Weizenbaum, "Eliza–a computer program for the study of natural language communication between man and machine," *Communications of the ACM*, vol. 9, no. 1, pp. 36–45, 1966

[12] B. AbuShawar and E. Atwell, "Alice chatbot: Trials and outputs," *Computación y Sistemas*, vol. 19, no. 4, pp. 625–632, 2015

[13] O. Chukhno, N. Chukhno, K. E. Samouylov, and S. Shorgin, "A chatbot as an environment for carrying out the group decision making process," in *ITTMM (Selected Papers)*, pp. 15–25, 2019

[14] L. T. Car, D. A. Dhinagaran, B. M. Kyaw, T. Kowatsch, S. Joty, Y.-L. Theng, and R. Atun, "Conversational agents in health care: scoping review and conceptual analysis," *Journal of Medical Internet Research*, vol. 22, no. 8, p. e17158, 2020

[15] D. Laurillard, *Rethinking university teaching: A conversational framework for the effective use of learning technologies*, Routledge, 2013

[16] P. Wik and A. Hjalmarsson, "Embodied conversational agents in computer assisted language learning," *Speech Communication*, vol. 51, no. 10, pp. 1024–1037, 2009

[17] S. Chaudhuri, R. Kumar, I. K. Howley, and C. P. Rosé, "Engaging collaborative learners with helping agents," in *AIED*, pp. 365–372, 2009

[18] Z. Ruttkay and C. Pelachaud, *From browstotrust: Evaluating embodied conversational agents*, vol. 7, Berlin: Springer Science & Business Media, 2006

[19] R. N. Brewer, L. Findlater, J. Kaye, W. Lasecki, C. Munteanu, and A. Weber, "Accessible voice interfaces," in *Companion of the 2018 ACM Conference on Computer Supported Cooperative Work and Social Computing*, pp. 441–446, 2018

[20] A. Serenko, N. Bontis, and B. Detlor, "End-user adoption of animated interface agents in everyday work applications," *Behaviour & Information Technology*, vol. 26, no. 2, pp. 119–132, 2007

AUTHORS' BACKGROUND

Your Name	Title*	Research Field	Personal website
Supun DE SILVA	BScIS Graduate	Artificial Intelligence; Natural Language Processing	https://www.linkedin.com/in/supun-de-silva/
Maiga CHANG	Associate Dean, Research & Innovation Full Professor	Artificial Intelligence; Natural Language Processing; Intelligent Tutoring Systems; Intelligent Agent and Chatbot Technology; Game-based Learning, Training and Assessment; Learning Behaviour Analysis; Learning Analytics and Academic Analytics; Health Informatics; Data Mining; Computational Intelligence; Evolutionary Computation; Museum Education; Mobile Learning and Ubiquitous Learning; Healthcare Technology, etc.	https://maiga.athabascau.ca

*This form helps us to understand your paper better, the form itself will not be published.

*Title can be chosen from: master student, Phd candidate, assistant professor, lecture, senior lecture, associate professor, full professor