Proc. of the 10th International Conference on Engineering and Emerging Technologies (ICEET)
27-28 December 2024, Dubai, UAE

Development of a chatbot to support an university institutional website

Telmo Sampaio*, Pedro Filipe Oliveira†, Paulo Matos†
*Instituto Politécnico de Bragança, Campus de Santa Apolónia, 5300-253 Bragança, Portugal
a48561@alunos.ipb.pt
†*Research Centre in Digitalization and Intelligent Robotics (CeDRI),*
Laboratório Associado para a Sustentabilidade e Tecnologia em Regiões de Montanha (SusTEC),
Instituto Politécnico de Bragança, Campus de Santa Apolónia, 5300-253 Bragança, Portugal
{poliveira, pmatos}@ipb.pt

2024 International Conference on Engineering and Emerging Technologies (ICEET) | 979-8-3315-3289-5/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICEET65156.2024.10913604

***Abstract*—Currently, the institutional website of the Polytechnic Institute of Bragança (IPB) lacks an interactive system that allows users to obtain information quickly and efficiently. With the increase in the use of online services and the preference of students for instant interactions, there is a growing need for tools that facilitate communication and access to information. This work proposes the development of a chatbot to support the IPB website, with the aim of improving user experience and providing quick and accurate answers to their queries. The main objective of this project is to create an intelligent chatbot capable of understanding and responding to a wide range of questions related to IPB. To achieve this, modern natural language processing and machine learning technologies such as Streamlit, Langchain, and OpenAI are used. The system includes an automated Web Scraper to keep information up-to-date, an architecture based on efficient document segmentation, and an administrative back office for monitoring and management. This development represents a significant advance in how users interact with the IPB website, offering a more efficient and accessible communication channel. The chatbot has the potential to reduce the workload of administrative staff while improving user satisfaction by providing accurate and instant answers to their questions.**

***Index Terms*—chatbot, webscraper, machine-learning, streamlit**

I. Introduction

In the rapidly evolving landscape of higher education, institutions are increasingly recognizing the need to transform their communication strategies to better serve their digital-native student population. Numerous studies and articles have explored the potential benefits of using AI systems and chatbots in the academic field. [1] As traditional communication channels like email and telephone become less favored among younger generations, educational institutions must adapt to meet students where they are - on social media and instant messaging platforms. This shift has created an opportunity for innovative solutions, particularly in the form of chatbots powered by artificial intelligence, which can provide immediate, accurate, and consistent responses to student inquiries. The IPB, recognizing this technological and cultural change, aims to enhance its student support services through the implementation of a dedicated chatbot system. Educational technologies enable distance learning models and provide students with the opportunity to learn at their own pace. [2] This initiative addresses the growing demand for real-time information access, particularly concerning crucial aspects such as registration procedures, enrollment processes, and financial matters including tuition fees and other administrative costs. Using modern educational technologies, this solution not only promises to streamline communication, but also aligns with the broader trend of digital transformation in higher education institutions.

II. State of the art

Artificial intelligence (AI) is steadily integrating into our daily lives through the development and application of intelligent software and hardware. Chatbots serve as a prime example of AI systems and represent one of the most basic yet widely used forms of intelligent human-computer interaction. These programs simulate intelligent responses during conversations, either through text or voice, and are capable of understanding one or more human languages. Chatbots are highly versatile, finding utility in fields such as education, business, and e-commerce. [3]

A. History of Chatbots

The origins of chatbots trace back to the mid-20th century, when computer scientists began exploring the idea of creating machines capable of understanding and responding to human language, facilitating simple conversations. In 1950, Alan Turing—a computer scientist, logician, cryptanalyst, and pioneer in the field—posed a groundbreaking question: Can a computer communicate in a way that is indistinguishable from a human? This led him to propose the Turing test, a practical criterion for evaluating whether a machine can exhibit intelligent behavior comparable to that of a human.

The Turing test involves a human evaluator engaging in natural language conversations with both a machine and a human, without knowing which is which. The evaluator's task is to identify the machine. If the machine succeeds in convincing the evaluator of its humanity, it is considered

979-8-3315-3289-5/24/$31.00 ©2024 IEEE

to have passed the test. This concept has since become a benchmark for assessing artificial intelligence. [4]

1) Eliza: The first chatbot, named ELIZA, was created by Joseph Weizenbaum in 1966. It employed pattern matching and substitution techniques to simulate human conversation. Designed to mimic conversational exchanges, ELIZA operated by processing user input and matching it against a set of pre-programmed responses. It utilized a script designed to simulate interactions with a psychotherapist, creating the illusion of meaningful dialogue. [5]

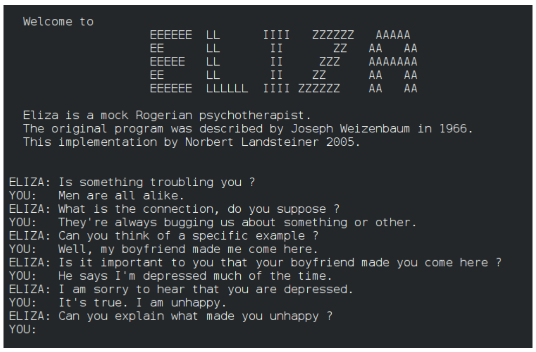

Fig. 1. Example of a conversation with ELIZA

B. Chatbot Types

Chatbots can be classified in various ways depending on their functionalities and purposes of use. Here are some examples:

- Based on Knowledge Domain:
  1) Open Domain Chatbots: They cover a wide variety of topics and are not limited to a specific knowledge domain. [6] Their aim is to imitate human conversation in a more natural and fluid way, answering a variety of questions that can range from everyday issues to more abstract or complex topics.
  2) Closed Domain Chatbots: These are specialized in specific areas of knowledge or sector, such as health, finance, or customer service. They are designed to provide precise answers and solutions within their field of expertise, using a detailed and specific knowledge base. [7]
- Based on the Service Provided:
  1) Interpersonal Chatbots: They facilitate interaction between people and services, acting as intermediaries that help users carry out tasks such as booking flights, restaurants, or scheduling appointments. They are configured to understand and process specific requests, making transactions more convenient and efficient. [8]
  2) Intrapersonal Chatbots: These are designed to operate within messaging platforms, acting as digital companions that offer personal assistance, entertainment, or emotional support. They are customized to interact more intimately and personally with users, offering a conversational experience that can range from functional to recreational, such as counseling, emotional support, or health promotion that requires a sense of human touch. [9] An example of an intrapersonal chatbot is the *Alexa-Cortana* integration, where a user can ask *Alexa* to send an email using *Cortana*, or access resources from one assistant through the other, demonstrating how chatbots from different ecosystems can work together. [10]

C. Use of chatbots in customer service

In this paper, we will focus mainly on the use of customer service chatbots. Companies have increasingly begun to adopt chatbots as a customer service support system to provide a more personalized customer service experience. This technology is attracting attention because it brings great advantages to customer support, such as being able to serve several customers at the same time, having the ability to provide multilingual support, being available 24/7 at no extra cost, relieving humans of repetitive tasks, and being able to respond to a variety of query solving questions without delay. [11]

D. Use of Chatbots in Portugal

Chatbots are increasingly being used as customer support on various company websites, and Portugal is no exception, with several companies betting on this growing technology.

1) NOS: Network Operating System (NOS) has presented a chatbot solution, powered by generative artificial intelligence, to support its customers. Available in the NOS Forum 24 hours a day, the intelligent virtual assistant promises to maintain a natural and dynamic conversation with users. NOS customers can find the intelligent virtual assistant in the NOS Forum, able to answer any question put to it in this space. The chatbot will respond autonomously, based on public information on NOS products and services.

2) CTT: Cargo and Transit Control (CTT) - the Portuguese Post Office has introduced its first chatbot powered by Generative Artificial Intelligence (ChatGPT), aiming to "transform the customer service experience." Named 'Helena,' the chatbot is built on Microsoft Azure OpenAI technology and provides CTT customers with real-time assistance that seamlessly integrates informational and transactional support. Customers can use 'Helena' to check parcel status, find zip codes, learn about required documents for subscribing to savings certificates, inquire about store hours, pay tolls, navigate customs procedures, and access a wide range of other information instantly. [12]

III. Materials and Methods

A. Description of the problem

The central problem addressed in this project is the need for an efficient chatbot to support the IPB's institutional website. With the significant increase in the use of on-line services and students' preference for interactions on social networks and instant messaging, there is a growing demand for tools that enable fast and effective communication. The aim is to develop a chatbot that can answer frequently asked questions, provide useful information, and improve user interaction with the institutional website.

B. Proposed solution

The proposed solution involves creating a chatbot using modern natural language processing (NLP) and Machine Learning (ML) technologies. The chatbot will be able to understand and respond to user queries accurately and efficiently. The implementation will be based on *Streamlit*, *Langchain*, and *OpenAI* technologies, which offer a robust infrastructure for developing personalized chatbots.

C. Tools and Technologies Used

Several tools and technologies will be used to implement the chatbot, such as:

- Programming Languages: Python will be the main language for development due to its popularity and the vast library of support for NLP and ML.
- Frameworks: *Langchain* will be used to manage the chatbot's components, integrating various data sources and language models. Streamlit will be the platform for developing an interactive and accessible user interface.
- Language Models: We will use *OpenAI's* recent *GPT-4o-mini* due to its cost-effectiveness, which will be used to interpret queries and generate responses.
- Storage and Indexing: *Pinecone* will be used to index and store documents, ensuring fast and efficient retrieval of information.

D. Web Scraping

To feed the chatbot, we developed a Web Scraper to collect and convert all relevant content from the IPB website into Markdown files. This process is fully automated, thus facilitating data extraction. The Web Scraper is structured as follows:

- Link extraction: The Scraper iterates through all the Uniform Resource Locators (URLs) present in the IPB domain.
- Downloading and Converting Content: For each link, the Hyper-text Markup Language (HTML) content of the page is downloaded and then converted into Markdown format using the *markdownify* library. In this process, every trace of HTML code is removed, making the information as clean as possible and making it easier for the chatbot to read.
- File management: The files are saved in a folder with the code of the school from which the information was taken in the file name, to make it easier to create the statistics that will be made later.

Figure 2 shows the Web Scraper process:

E. Chatbot development

The chatbot is designed to optimize interaction with users, guarantee accurate and relevant responses, and use previously uploaded and processed information. The system's architecture

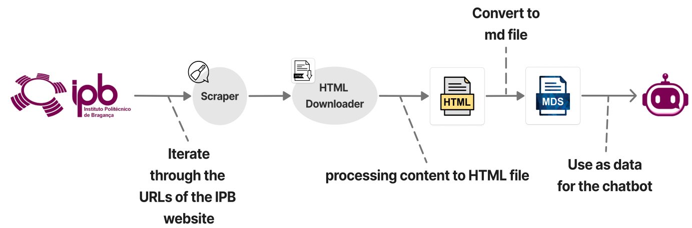

Fig. 2. Web Scraper

is based on efficient document segmentation, intelligent data storage, and an AI chain to generate responses.

The md files are converted into small chunks of text, which, in turn, are transformed into embedding, which are numerical representations of the text that can be used for machine learning tasks. The embeddings are now stored in a vector where they can be accessed later.

Figure 3 illustrates how the information provided to the chatbot is divided up and stored.

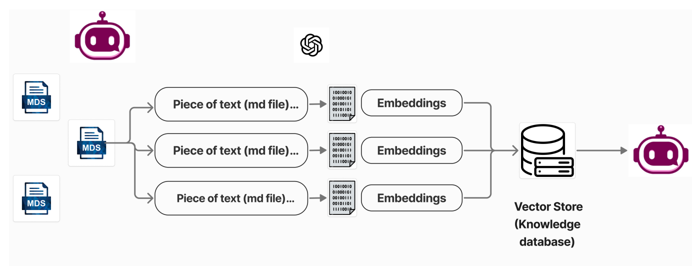

Fig. 3. Architecture of data stored.

When the language model receives a question, it consults the Vector Store, a database specialized in storing and indexing embedded data. Examples of Vector Stores are *Pinecone* and *Chroma*, which allow information to be retrieved based on semantic similarity. During the query, the question is converted into an embedding, and the system searches for the stored texts whose numerical representations have the greatest semantic correspondence with the embedding of the question. The semantic search results are then sorted, returning the best possible answer to the user's question.

Figure 4 shows how the stored information is searched to provide the user with the most appropriate response based on the content of the files.

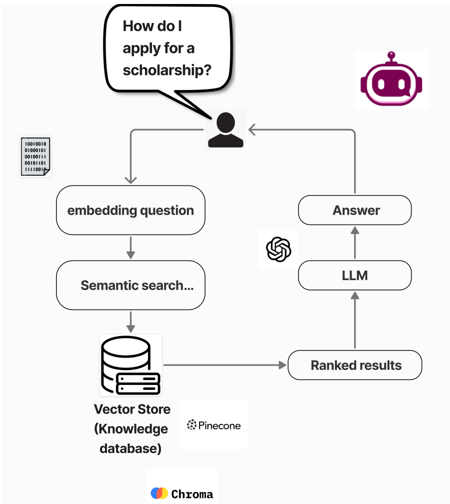

Fig. 4. Architecture of a Question-Answering.

1) Data processing and document processing: The project starts by loading documents from a directory using *LangChain's DirectoryLoader* tool. These documents are then processed and split into smaller parts using the *RecursiveCharacterTextSplitter*. Where this splitting is important for efficient document retrieval and embedding, it allows the smaller parts to be more easily indexed and searched. To improve performance and enhance user experience, a caching mechanism was implemented, which uses *LocalFileStore* to store the embeds generated by *OpenAI's* text-embedding-3-small model. This caching system significantly reduces processing time and operating costs by reusing previously generated embedding"s. The following table shows the comparison between *OpenAI's* text embedding models.

TABLE I
OpenAI TEXT EMBEDDING TEMPLATES.

Model	**Pages (per dollar $)**	**Performance in MTEB EVAL**
text-embedding-3-small	62,500	62.3%
text-embedding-3-large	9,615	64.6%
text-embedding-ada-002	12,500	61.0%

As we can see from the table, the text-embedding-3-large model, despite performing slightly better, is 550 % more expensive than the text-embedding-3-small model. For our project, the small template is sufficient to deal with the simplicity and volume of texts that the chatbot needs to process, making the additional cost of the large template unjustifiable.

2) Storage and retrieval of vectors: The processed documents are incorporated and stored in a vector repository where we first used chroma, but for scalability and performance reasons, we used *Pinecone*, allowing efficient retrieval of similar documents based on user queries. This feature is especially useful when dealing with large volumes of data and when ensuring fast response times.

TABLE II
PROS AND CONS OF *Pinecone* AND *Chroma*.

	Pros	**Cons**
Pinecone	Real-time search Scalability Automatic indexing Python support	Cost Limited consultation
Chroma	Open-source Extensible consultation Community support	Deployment complexity Performance

3) Recovery and Response Chain: The Retrieval and Response Chain is an essential component of the chatbot that combines the ability to retrieve relevant information with the generation of responses in natural language. This approach is particularly effective when dealing with large volumes of documents, as is the case with the chatbot developed. The first step in the chain is to establish a mechanism to retrieve relevant information from previously uploaded and processed documents. To do this, an Large Language Model (LLM) is used to understand the user's intent and formulate a search query in order to retrieve relevant information from the previously saved files. The retrieved documents are combined and processed to provide a coherent and informative response. The language model used to generate the final answer takes into account the full context of the conversation, including the documents retrieved and the history of the conversation with the user. If the user's question does not have enough context, the model generates a complementary question in order to obtain more context. The language model used for both chains was *OpenAI's gpt-4o-mini*. We decided to use this model due to its effectiveness, as it is one of the best performing models (see Figure 5).

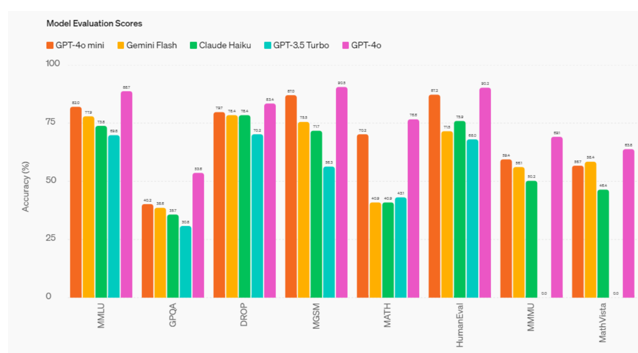

Fig. 5. Performance comparison

The get_response function is responsible for using all the components mentioned above. First, the function checks whether there is a *vector_store* in the session; if not, it initializes it based on the documents loaded previously, creating embeddings to facilitate information retrieval. The function then checks for ambiguity in the processed input, if there is, it generates a clarification question for the user and adds this interaction to the conversation history. If the input is clear, the function creates a retrieval string which uses the conversation history to make a search query, and a Reply string, which responds based on the information present in the retrieved documents. The information and statistics of each interaction are also saved in MongoDB.

4) Interface and Feedback : The chatbot is integrated into *Streamlit*, providing an interactive web interface that allows users to enter questions and receive answers. For a more natural and engaging user experience, a word-by-word visualization function is used, simulating a progressive response from the chatbot. User feedback is collected via *streamlitfeedback*, which allows you to rate the chatbot's responses and add comments. This feedback is recorded and can be used to continuously improve the model over time.

F. Back office development

To complement our chatbot, we have developed a back office, which serves as an administrative interface for viewing statistics and managing the documents used by the chatbot. This back office is a useful tool for monitoring the chatbot's performance and ensuring that it provides up-to-date and relevant information. To implement it, we used Flask for the backend, HTML for the back office interface, JavaScript to create the graphics and MongoDB to store the statistics.

1) Statistics page: One of the back office pages is the Statistics page, designed to provide a clear and detailed overview of user interactions with the chatbot. The page is divided into two parts one showing graphs of the relevant statistics, and the other showing user feedback messages. The graphs allow you to monitor different chatbot performance metrics such as:

- Total interactions: We display the total number of user interactions with the chatbot.
- Tokens Used: We show the total number of tokens processed by the chatbot, which helps us assess the workload, the volume of data processed and the cost of the chatbot.
- Average latency: This graph shows the chatbot's average response time, a crucial metric for ensuring the efficiency and speed of the system.
- User Feedback: This graph shows the feedback count, which lets you know whether users are satisfied with the chatbot's responses.

Figure 6 shows the implemented statistics page. it is also possible to change the time interval of the statistics displayed, making it possible to evaluate their evolution over time.

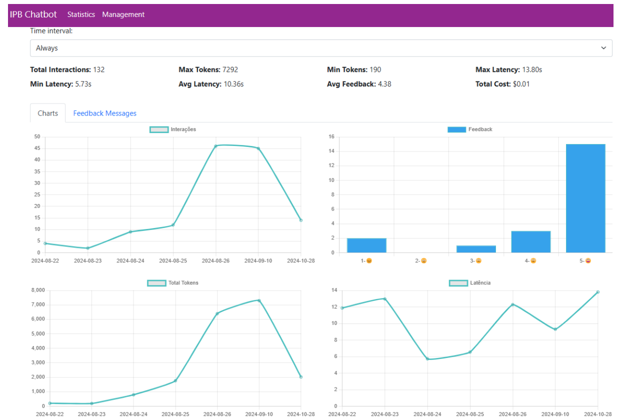

Fig. 6. Statistics page

2) Management page: The back-office management page was developed to make it easier to manage the files used by the chatbot. This page allows administrators to view, update and access the contents of the files that are used to feed the chatbot, ensuring that the information provided to users is always up-to-date and relevant. In the following, we detail the main features of the Management Page:

- Viewing files: In the files section, administrators can see a list of all the files fed to the chatbot. The list of files includes the name of each file and its size. Each file listed on the management page can be opened to view its contents; this feature is useful for quickly checking the contents of files and ensuring that the information is accurate and up-to-date.
- Updating files Another essential feature on the Management Page is the ability to update the data in the files used by the chatbot. To do this, we've implemented a button that, when pressed, will run the Web Scraper implemented earlier, causing the files to be replaced with new ones. This implementation is useful so that when there are information updates on the IPB website, the administrator can manually force an update of the files. After the update, the system displays the date and time of the last update, ensuring that administrators know exactly when the information was last updated.

IV. Results

To evaluate the chatbot's performance, we asked a few more questions, and the accuracy of the results can be seen in the table below:

TABLE III
ACCURACY OF ANSWERS

Question	Accuracy
1	100%
2	100%
3	100%
4	0%
5	100%
6	100%

1) Course director: An important piece of information for students is to know who the course director is. In this test question, we asked the chatbot who the director of the Computer Engineering course is, to which it answered correctly (see Figure 7);
2) Access requirements: A useful piece of information for prospective students is to know the entry requirements for the course they want to study. In this test, we asked the chatbot the entry requirements for the Sports course, to which the chatbot replied with all the relevant information about the requirements.
3) Canteen: We asked the chatbot about the canteen schedules, to which it responded correctly by mentioning the schedules for the canteen on the Bragança campus and also on the Mirandela campus.
4) CTESP: The answers of Higher Professional Technical Courses (CTESP) were incorrect; this specific question was "What are the dates for applying to a Higher Professional Technical Course?". The chatbot response gave incorrect application dates, a significant error, since accurate information on application deadlines is essential for prospective students and must always be precise and up to date.
5) Residences: We asked the chatbot about the residences to which it replied about all the residences available, their capacity, and their amenities.
6) Scholarship: A very important topic for new students is scholarships. In this test, we asked the chatbot about applying for a scholarship which included all deadlines and useful links.

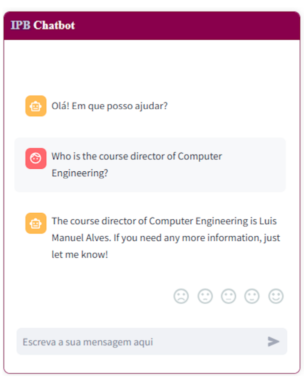

Fig. 7. Course director question.

In general, the chatbot demonstrated 100% accuracy in all other responses, which is a remarkable performance. However, this single error underlines the need for continuous improvement, especially in areas involving critical information.

V. Conclusions

The development of a chatbot to support the IPB website represents a significant step forward in improving the user experience, benefiting students, staff and visitors. Throughout the development process, a number of technical challenges were overcome, including efficiently integrating large volumes of institutional data, ensuring accurate and relevant responses, and creating an intuitive and accessible interface. The implementation of an automated Web Scraper to collect and convert the content of the IPB website into Markdown format demonstrates an effective solution for keeping the chatbot's information up to date. The system's architecture, based on technologies such as *Streamlit*, *Langchain* and *OpenAI*, provided a solid foundation for the development of a chatbot capable of understanding and responding to a wide range of queries. The implementation of an administrative back office, with statistics visualization and document management features, adds a layer of control and monitoring to the system. This tool allows administrators to track the chatbot's performance, analyze user feedback and keep information up to date. This capacity for supervision and continuous adjustment is fundamental to guaranteeing the system's relevance, effectiveness and constant improvement.

Acknowledgment

The authors are grateful to the Foundation for Science and Technology (FCT, Portugal) for financial support through national funds FCT/MCTES (PIDDAC) to CeDRI (UIDB/05757/2020 and UIDP/05757/2020) and SusTEC (LA/P/0007/2021).

References

[1] K. Zhang and A. B. Aslan, "Ai technologies for education: Recent research & future directions," *Computers and Education: Artificial Intelligence*, vol. 2, p. 100025, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2666920X21000199
[2] S. Wollny, J. Schneider, D. Di Mitri, J. Weidlich, M. Rittberger, and H. Drachsler, "Are we there yet?-a systematic literature review on chatbots in education," *Frontiers in artificial intelligence*, vol. 4, p. 654924, 2021.
[3] E. Adamopoulou and L. Moussiades, "An overview of chatbot technology," in *IFIP internatiooverview of chatbotnal conference on artificial intelligence applications and innovations*. Springer, 2020, pp. 373–383.
[4] M. Zemčík, "A brief history of chatbots," *DEStech Transactions on Computer Science and Engineering*, vol. 10, 2019.
[5] M. Gobiet, "The history of chatbots – from eliza to chatgpt," Feb. 2024. [Online]. Available: https://onlim.com/en/the-history-of-chatbots/
[6] D. Adiwardana, M.-T. Luong, D. R. So, J. Hall, N. Fiedel, R. Thoppilan, Z. Yang, A. Kulshreshtha, G. Nemade, Y. Lu *et al.*, "Towards a human-like open-domain chatbot," *arXiv preprint arXiv:2001.09977*, 2020.
[7] K. Nimavat and T. Champaneria, "Chatbots: An overview types, architecture, tools and future possibilities," *Int. J. Sci. Res. Dev*, vol. 5, no. 7, pp. 1019–1024, 2017.
[8] A. Beattie, A. P. Edwards, and C. Edwards, "A bot and a smile: Interpersonal impressions of chatbots and humans using emoji in computer-mediated communication," in *Communicating Artificial Intelligence (AI)*. Routledge, 2020, pp. 41–59.
[9] L. Xu, L. Sanders, K. Li, J. C. Chow *et al.*, "Chatbot for health care and oncology applications using artificial intelligence and machine learning: systematic review," *JMIR cancer*, vol. 7, no. 4, p. e27850, 2021.
[10] A. Shuman, "The official microsoft blog," Aug. 2017. [Online]. Available: https://blogs.microsoft.com/blog/2017/08/30/hey-cortana-open-alexa-microsoft-amazons-first-kind-collaboration/
[11] P. William, G. R. Lanke, V. N. R. Inukollu, P. Singh, A. Shrivastava, and R. Kumar, "Framework for design and implementation of chat support system using natural language processing," in *2023 4th International Conference on Intelligent Engineering and Management (ICIEM)*. IEEE, 2023, pp. 1–7.
[12] "Ctt lançam chatbot com inteligência artificial," 2023. [Online]. Available: https://www.silicon.pt/ctt-lancam-chatbot-com-inteligencia-artificial-95953.html