

NÁDILA DA SILVA DE AZEVEDO

**A NOVEL METHODOLOGY FOR DEVELOPING TROUBLESHOOTING CHATBOTS
APPLIED TO ATM TECHNICAL MAINTENANCE SUPPORT**

Trabalho de Conclusão de Curso apresentado à banca avaliadora do Curso de Engenharia de Computação, da Escola Superior de Tecnologia, da Universidade do Estado do Amazonas, como pré-requisito para obtenção do título de Engenheiro de Computação.

Orientador(a): Prof. Dr. Carlos Mauricio Serodio Figueiredo

Manaus – Dezembro – 2024

Agradecimentos

Agradeço primeiramente a Deus pela sua graça e direção ao longo desta jornada acadêmica. Expresso minha mais profunda gratidão aos meus pais, Luciane Lima e Raimundo Honório, cujo amor, dedicação e valores transmitidos foram fundamentais para minha formação e para a concretização deste trabalho.

Manifesto meu sincero reconhecimento ao meu orientador, Professor Doutor Carlos Maurício, por sua valiosa orientação acadêmica, disponibilidade e contribuições essenciais que enriqueceram significativamente esta pesquisa. Sou igualmente grato à equipe do Laboratório de Sistemas Embarcados (LSE), em especial ao Dr. Gustavo Aquino, Me. Leandro Okimoto e Me. Hendrio Bragança, mentores que proporcionaram aprendizados inestimáveis, contribuindo não apenas para este trabalho, mas também para meu desenvolvimento acadêmico e profissional.

Estendo meus agradecimentos ao colega Maicon Pantoja, cujo suporte técnico, orientações precisas e momentos de descontração foram vitais durante este percurso. Sua parceria e colaboração foram determinantes para a conclusão bem-sucedida deste trabalho.

A todos que, direta ou indiretamente, contribuíram para esta realização, meu sincero agradecimento.

Ficha Catalográfica

Ficha catalográfica elaborada automaticamente de acordo com os dados fornecidos pelo(a) autor(a).
Sistema Integrado de Bibliotecas da Universidade do Estado do Amazonas.

D278n de Azevedo, Nádila da Silva
A novel methodology for developing troubleshooting chatbots applied to ATM technical maintenance support / Nádila da Silva de Azevedo . Manaus : [s.n], 2023.
23 f.: il., color.; 21,0 cm.

TCC - Graduação em Engenharia de Computação- Universidade do Estado do Amazonas, Manaus, 2023.
Inclui Bibliografia.
Orientador: Figueiredo, Carlos Maurício Serodio.

1. Chatbot. 2. Automated teller machine. 3. Natural language processing. 4. Human computer interaction. 5. Retrieval based chatbot.
I. Figueiredo, Carlos Maurício Serodio (Orient.) II. Universidade do Estado do Amazonas. III. Título

CDU(1997)62:004



FOLHA DE APROVAÇÃO

A Novel Methodology for Developing Troubleshooting Chatbots Applied to ATM Technical Maintenance Support

Nádila da Silva de Azevedo

Trabalho de Conclusão de Curso defendido e aprovado pela banca avaliadora constituída pelos professores:

rof. Carlos Maurício Serodio Figueiredo, D.Sc.
Presidente

Prof. Tiago Eugenio de Melo, D.Sc.
Avaliador

Prof. Leandro Yuiti Silva Okimoto, M.Sc.
Avaliador

Manaus. 19 de julho de 2024.

Resumo

Este trabalho apresenta uma metodologia para o desenvolvimento de chatbots de troubleshooting voltados ao suporte técnico para manutenção de caixas eletrônicos, utilizando a plataforma Google Dialogflow CX. A pesquisa aborda a estruturação de uma base de conhecimento para facilitar a criação de assistentes virtuais, com um protocolo experimental que avalia o desempenho do chatbot em dois cenários: reconhecimento de intenção (IR) e reconhecimento de entidades nomeadas (NER), alcançando precisões de 0,95 e 0,88, respectivamente. A metodologia proposta oferece uma abordagem sistemática para interpretar e processar consultas de usuários, superando técnicas existentes e possibilitando a criação de agentes conversacionais flexíveis e escaláveis. Embora limitada a chatbots baseados em recuperação de informações, a metodologia se destaca como uma solução robusta e adaptável, com potencial aplicação em diferentes setores, contribuindo significativamente para o aprimoramento do suporte técnico e da experiência do usuário.

Palavras-Chave: Chatbot; Framework; Metodologia; Dialogflow; Suporte Técnico; Reconhecimento de intenção; Reconhecimento de entidades.

Abstract

The banking industry has been employing Artificial Intelligence (AI) technologies to enhance the quality of its services. More recently, AI algorithms, such as Natural Language Understanding (NLU), have been integrated into chatbots to improve banking applications. These chatbots are typically designed to cater to customers' needs. However, research in the development of troubleshooting chatbots for technical purposes remains scarce, especially in the banking sector. Although a company possesses a knowledge database, a standard methodology is essential to guide an AI developer in building a chatbot, making the modeling of technical needs into a specialized chatbot a challenging task. This paper presents a novel methodology for developing troubleshooting chatbots. We apply this methodology to create an AI-powered chatbot capable of performing technical ATM maintenance tasks. We propose the TroubleshootingBot, an experimental protocol to obtain data for evaluating the chatbot through two scenarios. The first scenario detects user intent, and the second recognizes desired values in a user's phrase (e.g., three beeps or two beeps). For these scenarios, we achieved accuracies of 0.93 and 0.88, respectively. This work represents a significant advancement in virtual assistants for banking applications and holds potential for other technical problem-solving applications.

Keywords: Machine Teller; Natural Language Processing; human computer interaction; Virtual Assistant; Retrieval based chatbot; Deep Learning; Intent Recognition; Named Entity Recognition; Google Dialogflow CX.

'É difícil pensar em uma grande indústria que não será transformada pela inteligência artificial. Isso inclui saúde, educação, meios de transporte, varejo, comunicações e agricultura.' *Andrew Ng*.

Article

A Novel Methodology for Developing Troubleshooting Chatbots Applied to ATM Technical Maintenance Support

Nádila Azevedo , Gustavo Aquino *, Leonardo Nascimento , Leonardo Camelo , Thiago Figueira , Joel Oliveira , Ingrid Figueiredo , André Printes , Israel Torné  and Carlos Figueiredo 

Embedded Systems Laboratory, State University of Amazonas, Manaus 69050-020, Brazil

* Correspondence: gustavaoqui@gmail.com

Abstract: The banking industry has been employing artificial intelligence (AI) technologies to enhance the quality of its services. More recently, AI algorithms, such as natural language understanding (NLU), have been integrated into chatbots to improve banking applications. These chatbots are typically designed to cater to customers' needs. However, research in the development of troubleshooting chatbots for technical purposes remains scarce, especially in the banking sector. Although a company may possess a knowledge database, a standard methodology is essential to guiding an AI developer in building a chatbot, making the modeling of technical needs into a specialized chatbot a challenging task. This paper presents a novel methodology for developing troubleshooting chatbots. We apply this methodology to create an AI-powered chatbot capable of performing technical ATM maintenance tasks. We propose the TroubleshootingBot, an experimental protocol to obtain data for evaluating the chatbot through two scenarios. The first scenario detects user intent, and the second recognizes desired values in a user's phrase (e.g., three beeps or two beeps). For these scenarios, we achieved accuracies of 0.93 and 0.88, respectively. This work represents a significant advancement in virtual assistants for banking applications and holds potential for other technical problem-solving applications.



Citation: Azevedo, N.; Aquino, G.; Nascimento, L.; Camelo, L.; Figueira, T.; Oliveira, J.; Figueiredo, L.; Printes, A.; Torné, I.; Figueiredo, C.

A Novel Methodology for Developing Troubleshooting Chatbots Applied to ATM Technical Maintenance Support. *Appl. Sci.* **2023**, *13*, 6777. <https://doi.org/10.3390/app13116777>

Academic Editor: Daeho Lee

Received: 25 April 2023

Revised: 25 May 2023

Accepted: 26 May 2023

Published: 2 June 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: chatbot; automated teller machine; natural language processing; human computer interaction; virtual assistant; retrieval based chatbot; deep learning; intent recognition; named entity recognition; Google Dialogflow CX

1. Introduction

Financial institutions play a pivotal role in the economy by offering specialized services that diminish the cost of discovering savings and borrowing opportunities [1–3]. Moreover, these institutions accept deposits from savers and lend to borrowers, thereby promoting liquidity in the economy and facilitating financial activity. Furthermore, financial institutions function as a marketplace for money and assets, allowing capital to be efficiently allocated where it is most needed [1].

Banks represent the predominant type of financial institutions, delivering essential services to society. They are integral to the perpetuation of commerce, offering financial support, facilitating payment transactions, and granting personal credit, thus catalyzing the expansion of both national and international trade [3].

The most recent Brazilian Federation of Banks (Febraban) statistics suggest that 76% of bank drafts occur through automated teller machines (ATMs) in Brazil [4]. An ATM is a device that facilitates financial transactions without the need for direct intervention by a bank agent. It encompasses numerous peripheral devices, including a monitor, keypad, card reader, bill dispenser, printers, and more complex modules. These may include devices employing special colored ink on banknotes in response to ATM attacks [5].

Bank branches may encounter challenges in providing satisfactory service when these machines malfunction, attributable to the time needed to engage a technician and rectify

the issue. Therefore, maintaining ATM availability in bank branches presents a significant challenge in the banking industry, as they play a vital role in providing round-the-clock service to clients. Thus, ensuring that ATMs are consistently in optimal operating conditions to meet customers' needs is of paramount importance [4].

The process of ATM maintenance for banks comprises two distinct levels of support: self-service and assisted service maintenance [6]. Bank agents are typically responsible for "self-service maintenance", the primary level of maintenance, which involves performing fundamental maintenance tasks to ensure the ATM's proper functioning. These tasks include replenishing the cash supply, rebooting the ATM to address minor technical issues, troubleshooting straightforward error messages, and replacing receipt paper. The primary maintenance level aims to tackle routine problems swiftly and efficiently, minimizing downtime and ensuring that the ATM remains operational for customers.

Implementing self-service maintenance can prove arduous for a bank agent. Optimally, the bank operator should endeavor to resolve the issue using fundamental knowledge before escalating the issue to the more costly technical support. Regrettably, many bank agents lack the expertise to rectify basic ATM problems, resulting in bank branches struggling to maintain satisfactory ATM availability. Should the bank operator fail to rectify an ATM issue, technical support typically becomes necessary [6].

The second level, known as "assisted service maintenance", is a technical level responsible for resolving more complex problems, such as hardware replacement. Technical support staff are highly trained and experienced, with the expertise to diagnose and repair any ATM issue [6].

This two-tiered approach to maintenance ensures that customers receive prompt and effective support, regardless of the nature of their problem. It minimizes downtime and enhances overall ATM performance, ensuring customers can access their banking services. However, with the advancement of the technologies embedded in ATMs, the complexity of installing internal equipment such as recycling modules and smart check deposits has increased. The maintenance of these devices has become more complex and operators often fail at self-service maintenance. This necessitates opening a service request for assisted service maintenance to resolve the ATM issue. In many cases, the call is unnecessary, and there is no real need to send a technician to maintain the equipment. Such erroneous calls generate unnecessary delays and cause equipment unavailability for customers [6].

These problems related to the two-tiered approach underscore the need for a more advanced solution to assist bank agents in resolving ATM-related issues. For the first level of assistance, it is crucial to acknowledge all the information needed to address the problem. Data structuring is fundamental to ensuring the bank agent executes the maintenance routine properly. Indeed, Cárcel-Carrasco and Cárcel-Carrasco [7] investigated the impact of organizing and making these records available to all collaborators through qualitative research. The results showed that knowledge management can optimize time and reduce maintenance costs and operating times. Based on these considerations, a pivotal aspect of our study revolves around the meticulous utilization of a meticulously curated dataset. This dataset has been constructed from a comprehensive knowledge database that is commonly utilized in the maintenance process. The dataset reflects the practical challenges and complexities users may encounter by encompassing real-world technical maintenance scenarios. This realism enhances the chatbot's ability to effectively comprehend user queries and provide contextually accurate responses, thereby enhancing the user experience. Furthermore, the dataset plays a crucial role in training the intent recognition and entity recognition models, enhancing their performance. Finally, the dataset ensures greater accuracy and reliability in understanding and interpreting user requests by exposing the models to various technical maintenance-related intents, entities, and associations.

The bank agent is often unaware of the precise issues the ATM could encounter. However, it is possible to describe the abnormalities which shape the conversation's context. Given these circumstances, determining an approach to address this barrier is necessary.

Artificial intelligence (AI) solutions can be employed to improve the efficiency and effectiveness of ATM maintenance processes at the first level, which is handled by non-technical operators [8]. In modern banking, many new digital technologies have been driven by AI at their core. This has led to innovative disruptions of banking channels, such as automated teller machines, online banking, and mobile banking. It has also transformed services such as check imaging, voice recognition, chatbots, and solutions such as AI investment advisors and AI credit selectors [9].

Chatbots' application in this sector is an emerging trend. According to Febraban, chatbots and virtual assistants have become increasingly popular in the banking sector [4]. AI-based chatbot technology is a noteworthy and popular development in the financial industry. Numerous approaches exist to create these systems, one of which is through closed platforms. These platforms offer the advantage of utilizing state-of-the-art technologies, which can reduce development time and offer the flexibility to build complex conversations [10,11]. Moreover, these platforms have allowed us to construct information retrieval chatbots, which do not generate answers at run time but retrieve predefined textual data [12].

Named entity recognition (NER) and intent recognition (IR) are essential components of natural language understanding (NLU) in chatbots [13,14]. Identifying specific entities and user intent enables chatbots to deliver personalized and relevant responses, leading to a better user experience. NER and IR are typically achieved through machine learning algorithms. These algorithms can analyze patterns in language and map them to predefined categories, allowing chatbots to interpret user inputs quickly and accurately.

The Google Dialogflow CX (Customer Experience) framework, a platform utilized for the development of AI-powered chatbots, plays a pivotal role in our research [15]. This framework was deliberately chosen for its robust AI capabilities, notably its incorporation of bidirectional encoder representation from transformers (BERT) for the training of information retrieval (IR) and named entity recognition (NER) models [16]. The purpose of employing this framework lies in its ability to facilitate the modeling and maintenance of virtual agents across various tasks. This includes, but is not limited to, healthcare management and appointment scheduling, thereby enhancing the scope and efficiency of our study [15].

Despite its growing popularity, developing virtual assistants for the bank maintenance process is challenging due to the limited amount of available data. Additionally, translating the complex and two-layered maintenance process into a virtual assistant solution is not straightforward: there is no clear guidance on best practices or an established methodology. Furthermore, the available literature for creating virtual assistants capable of providing advanced instructions for technical problems is scarce.

This scenario presents a challenge for businesses seeking to leverage virtual assistants to improve the efficiency and effectiveness of their maintenance processes. The lack of clear guidance and development methodology makes it arduous to implement troubleshooting chatbots that can provide the level of support required for the complex and technical nature of the bank maintenance process, especially for self-service maintenance. As a result, businesses and organizations face a significant challenge in developing virtual assistants that can meet the needs of their maintenance processes.

In this work, we propose a novel methodology for troubleshooting chatbots and introduce TroubleshootingBot, an experimental protocol to evaluate the chatbot through two scenarios. Using the presented framework, we developed a retrieval-based troubleshooting chatbot to resolve technical issues of the ATM maintenance process. Specifically, we introduce AI into the self-service maintenance, offering a problem expert chatbot for bank agents to solve issues that do not require the replacement of a hardware component. With the expertise of the virtual assistant, bank agents can quickly and accurately solve ATM issues, reducing the need for technical support and improving the availability of ATMs. In addition to enhancing the customer experience, this approach reduces the costs associated with technical support, leading to increased profitability for banks.

Finally, we propose a novel methodology to develop a chatbot based on a complex knowledge database. Specifically, our contributions can be summarized as follows:

1. A novel methodology for modeling a bank knowledge base into a troubleshooting chatbot with the Google Dialogflow CX framework.
2. An experimental protocol, named TroubleshootingBot, to evaluate a chatbot through critical IR and NER tasks.
3. A dataset with technical ATM issues from a bank agency for training both IR and NER models to identify ATM module issues.
4. A modern AI-powered chatbot to guide bank agents in the self-service maintenance process.

The paper is organized as follows: Section 2 introduces the background. Section 3 presents the related works. Section 4 illustrates the proposed virtual assistant framework. Section 5 presents the experimental protocol. Finally, in Section 6, we discuss the results.

2. Background

2.1. Natural Language Understanding

Natural language understanding (NLU) is a subfield of artificial intelligence (AI) that aims to enable machines to comprehend and analyze natural language based on concepts, entities, sentiments, and keywords [17] using machine learning and natural language processing (NLP) techniques. At the core of most chatbot applications lies NLU [18]. Chatbots need to allow the extraction of structured information from natural language inputs with unstructured semantic inputs, such as natural language inputs [19]. Nevertheless, developing an entire NLU system is challenging and expensive; therefore, the standard market practice consists of employing NLU solutions to create chatbots [20,21].

In the context of chatbots, NLU is utilized to analyze the question asked by the user and complete their requests [11]. To achieve this, NLU extracts domain-specific entities and intents from a text. Intents are the intentions or purposes with which the user asks a question. Intent recognition (IR) is a crucial component of NLU that involves identifying the intent behind a user's query [22]. Named entity recognition (NER) is another subtask of NLU that includes identifying and categorizing named entities in text or speech. Named entities are objects, people, locations, organizations, and other entity names in natural language [23,24].

To illustrate this with an example, consider the ATM maintenance process. In this context, the words "on" and "off" are classified as entities representing the status of a red LED on the receipt printer. These entities might carry different implications in different scenarios, highlighting the importance of context in NLU. Now, when a user inputs "The LED is on", the NLU system springs into action. The intent recognition (IR) component identifies the intent of the user's message—in this case, to report the status of the LED. Concurrently, the named entity Recognition (NER) component pinpoints the word "on" as an entity specifying the LED's current status.

By utilizing these entities and intents, an NLU engine can generate a targeted response that can efficiently assist users with their queries [25]. Thus, IR and NER—intent and entity recognition—form the twin pillars of NLU. Their applications are extensive, ranging from information extraction and text understanding to the creation of knowledge bases [24].

2.2. Virtual Assistant Types

When developing chatbots, it is crucial to consider the criteria for classification. The classification standards include the objectives, development techniques, communication style, and knowledge domain [26]. Additionally, some authors Adamopoulou and Moussiades [10] define criteria based on the platforms utilized to build a chatbot. They can be either from closed or open platforms. One of the most popular frameworks from a closed platform for developing conversational agents is Google Dialogflow CX, which offers a range of features for designing, building, and deploying conversational experiences.

Regarding the development method, the authors divide the types of chatbots between rule-based, also known as template-based [27], and AI-based [12,28]. Furthermore, AI-based chatbots can be further categorized into retrieval-based and general-based chatbots [10,12].

Rule-based chatbots, also known as pattern-matching chatbots, are a type of chatbot architecture that utilizes a set of predefined rules and patterns to generate responses to user input [29]. They are simple and easy to design and implement, and perform effectively for structured tasks with a limited range of possible inputs [12], providing prompt and precise responses to user queries and requests; moreover, they do not require abundant training data availability or machine learning algorithms. The predefined rules and patterns that generate responses are typically created by experts and cover a range of possible user inputs and scenarios. When a user enters a query or request, the chatbot matches the input against the predefined rules and patterns to determine the appropriate response. Therefore, this model is not robust against spelling or grammatical errors from the user [10], and it may struggle with more complex tasks that require understanding context, nuances, and natural language, as they are unable to learn from new inputs or adapt to new situations. Rule-based chatbots are best suited for specific and repetitive tasks, such as customer service inquiries or FAQ support that employ a question-answering chatbot [13]. Singh et al. [30] and Vishwakarma [31] present examples of these types of chatbots.

Considering AI-based chatbots, virtual assistants are categorized based on their information processing and response generation techniques. This section discusses two types of chatbots: (i) retrieval-based and (ii) generative models. This information is relevant for selecting the appropriate virtual assistant architecture for a specific task, thus enhancing performance. Retrieval and generative models are the two main categories into which AI can be further classified. AI models are based on machine learning algorithms, allowing them to learn from an existing database of human conversations [12].

Retrieval-based chatbots are the first type of AI-based chatbots, with an architecture that uses a predefined set of responses to generate a response to user input. They typically employ keyword matching and similarity measures to select the best answer from a predefined set. As a result, they only need a small amount of training data or limited use of complicated machine learning algorithms, and they can handle a wide range of user inputs. However, retrieval-based chatbots may need help with more complex tasks that require context, nuance, and natural language understanding. For example, suppose they are still looking for a suitable answer in their predefined set. In that case, they may respond with generic or unhelpful responses.

Retrieval-based chatbots are primarily designed to handle operational problems, particularly those related to troubleshooting, by efficiently retrieving and providing relevant information for customer support and other service-oriented tasks that require a prompt and accurate response [10,28,32,33].

The second type of AI-based chatbot, generative chatbot, utilizes natural language generation (NLG) techniques to generate responses to user inputs. In contrast to rule-based and retrieval-based chatbots, generative chatbots do not rely on predefined responses or rules. Instead, they produce responses based on their understanding of the input and the context. They use machine learning algorithms, such as deep learning, to analyze large amounts of data and learn how to generate human-like responses. They can understand and create natural language queries and handle a variety of inputs, including ambiguous or incomplete entries. Generative chatbots often tackle more complex tasks, such as providing recommendations or advice, engaging in conversation, and entertainment. In Kapočiūtė-Dzikiėnė [34], the applications of these chatbots are illustrated.

In our work, we follow the standard procedure already adopted by bank agencies in their maintenance process. Additionally, we only support a few specified ATM modules because not all modules allow for self-service maintenance. In this way, the most appropriate category of chatbot is retrieval-based using artificial intelligence resources.

2.3. Dialogflow

Currently, Google provides two closed platforms for chatbot development: Dialogflow ES and Dialogflow CX. “CX” stands for Customer Experience, while “ES” stands for Essentials. Although both versions provide tools to design VA, Dialogflow CX has functionalities to construct more complex conversational systems.

Google Dialogflow CX allows the creation, management, and development of conversational applications, such as chatbots and virtual assistants. It is easy to use and enables the construction of complex interactions with the user. Dialogflow CX presents an innovative approach to agent creation using state machine-based design. The tool gives precise and defined control over the conversation, improving the end user experience and simplifying the development process [15]. The following topics describe the components of Dialogflow CX [15]:

- **Agents:** These are responsible for processing the simultaneous conversations with the end user. An agent performs the natural language processing and understanding of the varieties of human language. It is based on a state machine architecture, which allows developers to explicitly and thoroughly control the conversation to create personalized and enhanced conversation experiences for their users. In addition, the agent allows adding context to the conversation, which can help improve the accuracy of the agent’s responses.
- **Flows:** These define conversation topics and the associated conversation paths. Each agent has an initial default flow. A dialog flow represents a sequence of interactions between the user and the virtual assistant. These flows allow you to define a series of questions and answers to guide the conversation with the user and conditions to direct the conversation to different paths according to the user’s responses.
- **Pages:** These group intentions into a more organized and manageable structure. They act as a container for related intents, allowing them to manage and edit intents more efficiently. Pages are beneficial in more complex conversations with many interconnected intentions and dialog flows. They help keep them organized and easy to manage, making it easier to edit and manage the conversation model. Pages also allow for the sharing intentions and dialog flows between multiple virtual assistants or applications, which is especially useful when consistency across numerous user communication channels is desired.
- **Event-Handler:** In Dialogflow, event-handlers are actions that are triggered by the user’s entry or for an event in the backend of the application. It uses an event-handler to give a user response every time a piece of information is not matched. This could be an intent or entity.
- **Intents:** These represent the user’s intent in a conversation. They are defined based on sample input sentences that the user may submit and help Dialogflow understand what the user is trying to achieve or ask. Each intent represents a specific action the user wants to perform, such as making a reservation at a restaurant, getting weather information, or asking for help with a task. Dialogflow uses these intentions to steer the conversation with the user toward the correct answer. In addition, intents can also include variables, known as parameters, that help capture specific information from the user, such as the date or location of a reservation. These variables allow Dialogflow to personalize its response to the user, making the conversation more effective and relevant. Finally, intentions in Dialogflow enable the model to understand what the user is trying to achieve or ask and steer the conversation to the correct response based on the customization of variables.
- **Entities:** These represent specific and relevant information for the conversation, such as names of people, places, dates, and times. They extract relevant information from the user’s input sentences and store them as variables, which helps in customizing the virtual assistant’s responses. There are many types of predefined entities in Dialogflow, such as date, time, number, address, and custom entities; they represent specific information for an application. Additionally, entities can also be utilized to

define patterns of phrases and expected behaviors for different types of information, such as names of people or addresses. This helps Dialogflow understand user input sentences more accurately and efficiently.

- Contexts: These help to maintain the conversation context between the user and the chatbot. They are utilized to remember relevant information and make it available throughout the conversation. For example, if the user is having issues with an ATM, the conversation context can include information about the type of transaction they are trying to perform and the date and time of the issue. This information can be employed to provide a more precise and relevant solution to the user. Contexts are also fundamental for controlling the flow of the conversation and avoiding the virtual assistant from getting stuck on a specific task or question. For instance, if the user changes the subject, the context can be altered to reflect the new topic of the conversation.

2.4. Evaluation of Chatbots

There is no established standard evaluation protocol for chatbots. A brief investigation was conducted to define the evaluation strategies. There are several strategies to measure the performance of chatbots. They can be one of two categories: human evaluation, which involves analyzing user feedback, and automated evaluation, which uses metrics such as precision, recall, and F1-score to evaluate the chatbot’s classification abilities. Even though human evaluation is valuable, it can be expensive and time-consuming [12].

Another metric is the exact match accuracy, which evaluates the IR and NER together [35]. This metric refers to the number of times a chatbot correctly identifies the user’s intent and any relevant entities mentioned in their input as compared to the total count of inputs.

The chatbot background is the IR and NER algorithms that can be evaluated as any other machine learning classifier. The metrics for automated evaluation utilize the frequency of true positives (*TPs*), true negatives (*TNs*), false positives (*FPs*), and false negatives (*FNs*), which are terms used to describe the occurrence of correct (*TP*) and incorrect (*FP*) predictions for a given class, as shown in Table 1.

Table 1. Metrics utilized to evaluate chatbots.

Metric	Equation
Accuracy	$\frac{TP+TN}{TP+TN+FP+FN}$
Recall	$\frac{TP}{TP+FN}$
Precision	$\frac{TP}{TP+FP}$
F1-score	$2 \times \frac{(Precision \times Recall)}{Precision+Recall}$

3. Related Works

This section discusses material related to the development of conversational agents. The interest in developing chatbots has increased due to their ability to provide automation and enhancement of essential tasks in various sectors and industries [36,37]. Moreover, demonstrations of the use of chatbots are noticeable when the target audience needs to be well-informed about procedures to execute a troubleshooting task, which is the objective of this work.

3.1. General Chatbot Applications

One of the most significant contributions to the development of chatbots with NLP features was the Eliza program, created by Weizenbaum in 1966 [38]. The operation of this chatbot involves analyzing keywords within the user’s input and matching them against predefined rules. While this approach made a notable contribution for its time, it had limitations in understanding contextual information. Other programs that became part of the first generation of chatbots include Parry [39] and A.L.I.C.E [40]. In Singh and Thakur [29], yeung

Shum et al. [41], the authors demonstrate the increasing popularity of this technology that has been seen in multiple applications created since then. The rising generation of digital assistants has been applied in distinct scenarios, such as e-commerce, e-learning, healthcare, and the industrial sector, to improve efficiency and enhance user experience.

In the e-commerce industry, brands are adopting chatbots to increase and engage customers, as evidenced in [42,43]. In the research conducted by Khan [44], an e-commerce sales chatbot platform is proposed to provide customer support. The project comprises five distinct modules, each playing a pivotal role in the overall system. Notably, the NLU engine assumes a particularly critical position within this framework. While this work provides a comprehensive overview of the system architecture, it, unfortunately, lacks detailed information regarding the performance evaluation of the NLU engine utilized in the chatbot. A comprehensive evaluation encompassing the entire system would be highly valuable in order to gain deeper insights into its overall effectiveness and capabilities.

By reducing the need for human presence in situations involving the COVID-19 pandemic, as shown by Amiri and Karahanna [45], the assistance capacity of this tool is expressed in various use cases ranging from scheduling vaccines to disseminating information about the coronavirus.

E-learning [46] is a promising area where a chatbot can be applied. Huang et al. [47] extensively reviewed the impact and relevance of integrating chatbots in education, highlighting the technological affordances that make them an attractive tool for the field. Instances of this application can be seen in the industrial sector, where chatbots have been acting as instructors for workers. Casillo et al. [48] proposes an approach regarding the training of new workers carried out by a virtual assistant prototype and reports results demonstrating that this system's advantage is the reduction of the learning process. In this aspect, Colabianchi et al. [49] also presents a point of view related to the role of chatbots as mentors with Popeye, a chatbot that trains new employees in container inspection.

3.2. Troubleshooting Chatbots

The domain of maintenance represents a field where chatbots have been employed. Troubleshooting chatbots are designed to identify and solve technical issues. They aim to provide efficient and effective technical support by identifying and addressing common problems.

Chatbots as technical assistants is a topic explored by Alhassan et al. [50], in which the authors proposed a methodology for an IT chatbot framework to assist common IT problems. Despite the demonstrated merits of incorporating chatbots in the troubleshooting domain, this study lacks empirical data pertaining to the chatbot's efficacy in effectively discerning and comprehending user descriptions to receive the appropriate procedure.

Another application of chatbots regarding the maintenance process can be seen in [51] where the chatbot, Telmi, is responsible for providing customer support for troubleshooting tasks. This work shows us the number of issues resolved by the chatbot and the characteristics of the problems that were not. One of the traits that characterized the issues that the chatbot could not address was the difficulty in identifying the intent resulting in a higher level of FN. This result highlights the importance of creating a well-constructed training dataset to maximize evaluation metrics effectively.

Previous works only presented the accomplished results without explaining the methodology of transforming the knowledge database into a chatbot with artificial intelligence [44,48–51]. Additionally, the works are poor in evaluation metrics. Finally, prior research does not cover the experimental protocol to assess chatbots [23,52]. Therefore, our work aims to bridge these two gaps in the field of artificial intelligence chatbots.

To the best of our understanding of the methodologies for developing chatbots, transmuting an automated teller machine (ATM) maintenance knowledge base into a chatbot remains largely uncharted territory [12,53,54]. This presents numerous opportunities for further exploration and innovation in this sphere [48]. Moreover, existing methodologies have been predominantly tailored for different application contexts [49], resulting

in a noticeable scarcity of direct comparisons in the context of ATM maintenance [50,51]. Consequently, it is imperative to conduct a thorough examination of the available methodologies [48,49] and to evaluate the ramifications of adapting these methodologies within the specific milieu of ATM maintenance [44,52].

The research work of [55] focuses on meticulously modeling a specific process within a business process model and notation (BPMN) framework and subsequently transforming it comprehensively into a functional chatbot. The primary transformation pipeline encompasses several key components. Initially, a graph normalization block is employed to load the BPMN diagram and ensure the structural integrity of the graph. Subsequently, a label processing block gathers the linguistic information inherent to the model. Furthermore, a dialog graph construction block generates finite state automata (FSA) based on the BPMN model, governing conversational transitions. A natural language generator (NLG) produces chatbot responses by generating text based on each node of the BPMN model. Finally, the encoded FSA is integrated into an AIML engine for subsequent processing. While the methodology for chatbot development is commendable, its application to an ATM maintenance knowledge base reveals significant drawbacks.

The current structure of a bank agency knowledge base, primarily comprising text-based procedures and specific information, necessitates better alignment with the node-based representation of the BPMN model. This misalignment poses challenges in accurately mapping content, resulting in incomplete or oversimplified representations within the chatbot. Additionally, the BPMN model's lack of granularity limits its ability to capture intricate troubleshooting steps, diminishing the effectiveness of the generated chatbot. Another noteworthy drawback to consider is the inherent limitation in capturing and preserving critical information from the conversation efficiently. Such content registration holds utmost significance, as it plays a pivotal role in the maintenance process, precisely determining the appropriate execution of procedures based on the key elements discussed within the conversation.

Although our application does not primarily focus on the educational field, it is pertinent to acknowledge that existing methodologies employed in developing educational chatbots incorporate the concept of a virtual assistant. A notable example can be found in the work of [56], where a methodology for constructing a conversational chatbot tutor is proposed. This approach leverages the application of first-order logic, which serves as a formal language specialized in inference and symbolic representation of knowledge, to delineate the chatbot's knowledge base. Nonetheless, the utilization of this methodology for implementing and troubleshooting chatbots within our specific context would present inherent challenges. The inadequacy of first-order logic in comprehensively capturing the intricacies and complexities inherent in natural language poses a significant hurdle. Given that the maintenance process necessitates interactions with users who may articulate their issues or queries differently, first-order logic, although proficient in representing rudimentary relationships and logical operations, may encounter difficulties when faced with more intricate or ambiguous scenarios. Consequently, this limitation could impede the chatbot's ability to handle diverse user queries and effectively provide appropriate responses.

The third analyzed methodology was the work of [57], which demonstrates the integration of an ontology and a knowledge base to develop an efficacious programming assistant chatbot. First, the authors introduce the Rela-Model, an ontology that organizes the knowledge base, precisely delineating the interrelations between programming concepts and the pre-established rules governing inferential processes. Then, to support its functionality, this model is amalgamated with a meticulously structured repertoire of scripts, encompassing five key components: script nomenclature, related content, an array of interrogatives and corresponding responses, and a set of rules meticulously employed for question selection within the script for the chatbot. This systematic procedure forms the bedrock of knowledge base construction for a question-and-answer-oriented chatbot. Furthermore, it is pertinent to note that chatbots reliant upon ontology-based systems often

require users to articulate queries utilizing terminology or expressions that align with the ontology's lexical framework.

3.3. Banking Assistants

Virtual assistants are being increasingly utilized to provide customers with prompt and personalized responses to their queries, thereby enhancing the overall customer experience [58]. Therefore, recent research has focused on developing and improving virtual banking assistants to serve customers better.

Several scholars have proposed and evaluated dialogue systems in the banking industry to improve customer service and satisfaction. Rustamov et al. [59] conducted the development and evaluation of several dialog management pipelines to verify the most suitable one for application in a banking context. The primary objective of this research was to investigate the dialogue manager and NLU components. A comprehensive experiment was conducted involving the utilization of fastText, an open-source library widely employed for text classification, and custom machine learning models trained on a specialized banking dataset. The experimental findings clearly indicated that the custom machine learning models outperformed fastText in terms of accuracy.

Another field of research concerns investigating the impact of virtual banking assistants on customer engagement and satisfaction to comprehend the significant role that chatbots play in the banking industry, as evidenced in [60]. Furthermore, customer acceptance is a crucial factor to consider when implementing chatbots in the banking industry. Alt et al. [61] have identified several key factors that can increase chatbots' appeal to customers and enhance their usage. While their work effectively highlights the substantial presence of virtual banking assistants catering to customer needs, it regrettably overlooks the mention of any applications specifically pertaining to troubleshooting chatbots for agent banks. This conspicuous absence suggests that the domain of this category of chatbots remains largely unexplored, presenting an as-yet uncharted domain, which holds promise for future research and development endeavors.

The applications mentioned above demonstrate the indispensability of chatbots in the banking industry. With their ability to provide personalized and efficient customer service, chatbots are valuable assets in improving customer satisfaction and loyalty. However, our research has found few references to chatbots focused on solving technical problems for ATM maintenance [62]. Therefore, our work seeks to create a chatbot with specialized technical expertise to help bank agents in the task of self-service maintenance.

3.4. Virtual Assistants Using Dialogflow

Dialogflow has been widely applied in various scenarios due to its diverse range of tools that enable developers to construct complex conversational agents. In this section, we report examples of the two versions of Dialogflow: CX and ES. Both platforms have been employed across different contexts.

Dhavan [21] developed a chatbot system with Dialogflow ES that aims to detect the symptoms of a possible heart attack through the user-provided description, which was collected using entity parameters. The user provides inputs regarding factors such as the severity of chest pain or the presence of breathing difficulties, among other pertinent details. After the patient submits the descriptions of their symptoms, the chatbot prompts the user to complete a questionnaire. Subsequently, these data are forwarded in sequence to a support vector machine model, which carries out the prediction.

The evaluation metrics employed were accuracy (0.824), precision (0.843), and recall (0.843). Although the results appear promising, it is imperative to emphasize the criticality of ensuring precise responses for a medical tool to provide accurate responses. Therefore, the metrics results should exceed a threshold of 0.9. Furthermore, the intents and entities utilized to retain the conversation context could also have been evaluated, as the chatbot's ability to deliver appropriate responses to the user is contingent upon the effective performance of these evaluative measures.

The work of Muhammad et al. [63] demonstrates a conversational tool that facilitates English language learning for students, utilizing Dialogflow ES and incorporating an English conversation book as its knowledge base. Additionally, entity extraction techniques were utilized to retain essential information from user responses. Dall’Acqua and Tamburini [64] employed the latest of Dialogflow (Dialogflow CX) to develop a conversational agent capable of delivering standard customer services such as subscription, instruction for download, and discounts section after receiving a user query. To recognize user input, this author used intent recognition and NER to collect relevant information and define the conversational context.

The papers mentioned above focus on the chatbot developed rather than on the methodology for transforming a specific knowledge base into a retrieval-based chatbot using the entities (flows, pages, entities, intents, etc.) available in Google Dialogflow.

4. Proposed Chatbot Development Methodology

The proposed methodology demonstrates the structured organization of a knowledge database for modeling a troubleshooting chatbot with the Dialogflow CX tool. Thus, we provide an overview of the conversational process inside the developed agent to deliver an appropriate response to users.

4.1. Knowledge Base

Bank agencies frequently possess a specialized knowledge database that contains the detailed information and procedures for addressing ATM maintenance tasks. In addition, the knowledge database contains information regarding the ATM’s capabilities, such as biometrics, printers, and monitors. One of the most pivotal bits of information in the knowledge database is the relationship between the different ways of reporting an ATM problem with a specific module. Table 2 illustrates an example of module entries and the respective associations.

Table 2. Examples of training phrases for modeling user intents for each ATM module.

Training Phrases	Label
I’m having problems with biometrics.	Biometry
I can’t type correctly, the keyboard has a problem.	Client Keyboard
The receipt printer is improperly curling the paper.	Receipt Printer
The equipment is having problems reading the card.	card reader
The check scanner is not recognizing checks correctly.	Check Scanner
Problems initializing video monitor.	Video Monitor
Lock problem.	Safe Lock
The ATM is not performing withdrawal operations correctly.	Recycler Module

In the knowledge database, various modules are present, each accompanied by a detailed description of its functionality and a list of phrases that pertain to the module. Furthermore, the knowledge base contains procedures for fixing problems with each module, providing a comprehensive resource for bank agents. Table 3 presents an overview of the knowledge database information for a specific module.

4.2. Dialogflow Modeling

A financial institution already has a mass of data with prior knowledge regarding resolving issues and the steps to follow for each specific ATM module. The information presented in Table 3 summarizes the relevant knowledge for building a chatbot. These processes lead to the manual solution of the problems; therefore, it is possible to translate this knowledge into a tool that can perform this process automatically.

Google Dialogflow CX is a powerful solution for integrating a knowledge database into an intelligent agent. Section 3 describes the main components of Dialogflow.

Table 3. Technical support script example utilized by an ATM operator to solve problems related to the cash dispenser.

Module Info	Details
Module Function	Cash Dispenser
Payment Model	ATM 4534-158, ATM 4534-185, ATM 4534-188, ATM 4534-237, ATM 4534-257, ATM 4534-272
Related Entries	Payer module disabled. Payer module disabling. Error in the payer module. Inoperative payer module. Screwing in the feed module channels. Tightening in the k7 of the payer module. Payer module stopped. Withdrawal unavailable. Rolling banknotes. Does not pay.
Actions	Step 1: Check if you have cash screwed into the Presenter Mouthpiece (Shutter). - If positive, unscrew the bill carefully so as not to damage any of the module’s mechanisms. Step 2: Check for cash wrapped around the Presenter straps. - If positive, unscrew the bill carefully so as not to damage any of the Module’s mechanisms. Step 3: Check if you have cash wrapped around the feeding belts, which are behind the payment module. - If positive, unscrew the bill carefully so as not to damage any of the Module’s mechanisms. Step 4: Check if you have cash screwed into the four power channels (place where the K7s fit). - If positive, unscrew the bill carefully so as not to damage any of the Module’s mechanisms. Step 5: Check if you have cash wrapped inside each of the K7s. - If positive, unscrew the bill carefully so as not to damage any of the Module’s mechanisms. Step 6: Check if the Reject K7 is full. - If positive, physically empty the Rejection K7. Step 7: NOTE: After carrying out all the verification, perform the module substitution in the equipment. Step 8: Open a technical ticket.

For our problem, the number of module tables, such as the one in Table 3, dictates the number of agent flows. The training sentences for the IR and NER models are extracted from the “Related Entries” field. Analyzing these sentences allow us to identify when to apply IR, IR plus NER, or NER alone. Based on the “Actions” it is possible to determine the number of pages required for each module flow. Assuming that there is only a predetermined number of expected responses in the steps of a page, we can utilize payloads (Dialogflow components capable of sending unstructured data such as quick buttons, in this case). Given that an associated image, document, or video is in a given step, we also utilize payloads to send the information.

Some translations of Dialogflow implementation from the knowledge database are complex. For example, provided that the procedures within the “Actions” of the modules repeat themselves, they can be componentized, creating a “custom flow.” In addition, parameters can inform the user’s previous steps in the conversation, thus minimizing unnecessary repetition. Occasionally, some modules may have conflicting “Related Entries”, so we can create a new flow to ask the user questions and direct them to the correct module. The diagram in Figure 1 illustrates how to translate the knowledge database into an intelligent agent.

The chatbot was designed to recognize user intent and execute transitions using AI technologies such as intent recognition (IR) and named entity recognition (NER). The transitions are determined by transitional rules that can leverage IR and NER. In addition, we implemented payloads and event-handlers to improve the conversational flow further and enhance the user experience. For instance, when the available input options are limited, quick reply buttons within the payload can be utilized to guide the user to respond appropriately. Furthermore, we can send images or videos from complex maintenance in-

structions in the payload. Finally, event-handlers deal with phrases where the user’s intent is not recognized by providing expected inputs or asking the user to rephrase their request.

The process was divided into flows and pages using the Dialogflow CX framework for modules in the knowledge base that contain complex, step-based guidelines. This enhances navigation and facilitates access to information for the user.

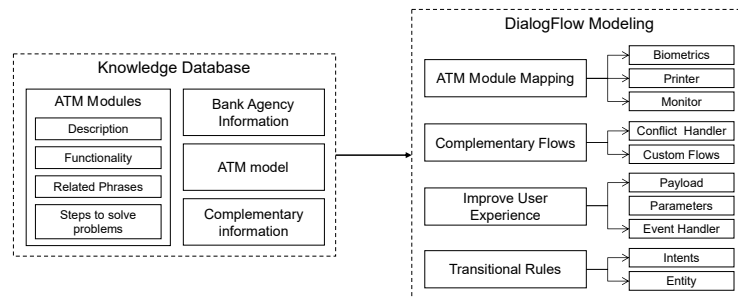


Figure 1. The proposed methodology.

In conclusion, the distinct knowledge base for each ATM provides a wealth of information that can be leveraged to create a virtual assistant. Furthermore, utilizing state-of-the-art AI technologies and deliberate design can achieve a smooth and streamlined user experience, giving bank agents a valuable tool for resolving technical problems without needing technical assistance. Although this solution may not be able to address all ATM problems, this advanced guidance minimizes unnecessary calls.

4.3. Conversational Workflow

After modeling and developing a virtual assistant in Dialogflow CX, a client can converse with the agent, which operates using a knowledge base from a bank agency. The knowledge base could, for example, be built upon expertly curated content such as the technical support script example utilized by an ATM operator to solve problems related to the cash dispenser as illustrated in Table 3. As the user interacts with the virtual assistant, a complex sequence of processes occurs within Dialogflow CX, involving many different components.

These processes enable the agent to comprehend the user’s input, determine the following appropriate action, and respond accurately and contextually. In addition, by harnessing machine learning and natural language processing, the agent can emulate a human-like conversation and provide an engaging and helpful user experience. Figure 2 offers a graphical representation of the various components and processes within Dialogflow CX, illustrating the operational process that leads to an accurate response for the user.

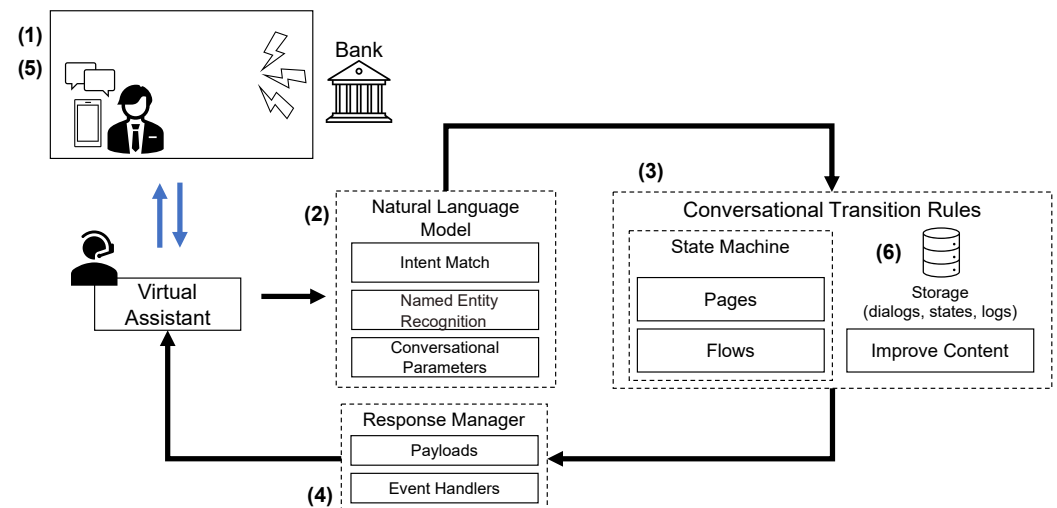


Figure 2. Conversational workflow.

The method involves the following steps:

1. *User initiates the conversation:* The user starts a conversation with the agent by either typing a message or speaking to a voice assistant device. For instance, a customer may voice concerns related to an ATM dispenser, such as “withdrawal unavailable” or “payer module stopped”.
2. *Agent comprehends user input:* The agent employs natural language understanding (NLU) to identify the user’s intent and extract relevant entities from the input provided. For example, recognizing issues related to an ATM dispenser from our knowledge base. If the user’s entry does not match any intent in the system, an event handler called “no-match” is triggered, prompting the user to rephrase the request.
3. *Agent determines the next action:* Using the identified intent and entities, and drawing from its knowledge base (Table 3), the agent leverages its trained machine learning models and conversational transition rules to determine the appropriate next course of action in response to the user’s input. The agent also stores relevant conversational data based on the current state of the dialog.
4. *Agent responds to the user:* The agent formulates an appropriate response to the user’s input, derived from its knowledge base. The response could be structured or unstructured data. In the structured case, the response is conveyed through textual communication, such as steps to resolve the ATM dispenser issue. In contrast, in the unstructured case, it can be conveyed via customized payloads, such as buttons, images, or videos.
5. *User initiates a subsequent interaction:* The user provides another input based on the agent’s response, to which the agent must respond appropriately, considering all previous interactions and information from the knowledge base.
6. *Agent retrieves the pertinent conversation variables:* The agent retrieves the relevant conversation variables stored in step 3, using these alongside the knowledge base to inform subsequent interactions and repeats steps 2 to 5 until the conversation is successfully resolved or ended.

In conclusion, a knowledge base similar to the one illustrated in Table 3 forms an essential part of the conversational workflow, aiding in more accurate and useful interactions between the user and the agent.

5. Experimental Protocol

As mentioned in the related literature outline in Section 3, there is currently no established standard for evaluating chatbots. To accomplish this task, we proposed TroubleshootingBot, a novel experimental protocol whereby it was possible to create a test dataset. It should be noted that the training data is not subject to this protocol, since it comes from the knowledge database. The test datasets were utilized to evaluate the IR and NER models in two pre-established scenarios.

This experiment evaluates the behavior of our solution when users interact with the chatbot. One of our solution’s primary challenges is accurately identifying the ATMs and providing the appropriate procedures related to this problem while extracting all relevant information to give an appropriate response. To better understand the critical points of the conversation, we divided the analysis into two scenarios. In the first scenario, we analyzed intents classification when users described the ATM’s problem. In the second scenario, we evaluated the employment of the NER technique associated with IR at critical conversational flows.

Only automated evaluation metrics were utilized to analyze the system, while human evaluation metrics [12] were not considered. Instead, we employed a confusion matrix and the metrics specified in Section 2.4 to evaluate this scenario. These metrics are particularly useful in assessing the system’s ability to identify user intent and correctly fill the associated slots.

We conducted a preliminary test utilizing phrases derived from the knowledge database. This process is described below.

To perform this experiment, we followed the steps below:

1. We meticulously selected critical modules to focus on to assess the effectiveness of our intent recognition (IR) and named entity recognition (NER) algorithms. We adopted this protocol to evaluate two scenarios: IR detection and NER detection. Specifically, we chose to only evaluate target intents directly related to ATM modules for our IR analysis. For the NER task, we selected entities essential for identifying potential issues with ATMs.
2. Randomly selected phrases from the knowledge database commonly utilized to reference ATM modules were selected. However, to ensure the reliability of our evaluation, we could not utilize these same phrases for testing. Therefore, we entered these phrases into a third-party paraphrasing tool, which generated a variety of completely new phrases with similar contexts. This process provided a set of phrases with unseen variations to evaluate the accuracy of our IR and NER algorithms.
3. To ensure that the generated phrases reflected the intended context, we comprehensively analyzed each phrase to identify any potential errors in interpretations. We then labeled each generated phrase according to its corresponding intent or entity by manually analyzing its underlying meaning and identifying relevant entities.
4. To further validate the reliability of our modules, we manually generated a set of phrases intentionally created to deviate from the expected context for IR. Additionally, for NER, we created training phrases that did not contain any expected entities and labeled them as no-match.
5. By employing this technique, we developed a dataset for our evaluation scenario.

5.1. Datasets

This section presents the datasets utilized for training and evaluating the chatbot. These datasets comprise an intent classification dataset and an entity recognition dataset. To ensure the reliability of our evaluations, we followed the aforementioned experimental protocol for generating evaluation samples. In addition, we utilized the knowledge database to obtain training samples (Table 3).

5.2. Intent Recognition

The intent recognition dataset was created to train a model for IR tasks, which classified issues related to ATM modules, such as biometrics, keyboard, printer, card reader, scanners, monitor, safe locks, and recycler. Only a few intents are critical to our system, particularly those identifying ATM problems. Table 4 presents the number of training and evaluation phrases employed in creating the intent recognition model. Table 2 provides examples of training phrases used to model user intent for each ATM module.

Upon successful comprehension of the ATM-related issue, the chatbot can offer exact assistance to the user by following the requisite procedures appropriate to a particular ATM module. Normally, the user may not know what module is responsible for the issue. However, the user can elucidate the error messages or symptoms that originated the problem.

For the evaluation of no-match events, a total of 14 phrases were manually generated. These phrases were created by devising utterances outside the scope of the known intents. For instance, the phrase “The ATM’s camera is not working” refers to a feature not supported by the ATM model and, therefore, is not covered by our knowledge database. As a result, any intent that cannot be accurately categorized is expected to be mapped to a no-match event-handle.

Although the evaluation sentences were generated by paraphrasing some of the sentences in the knowledge database, it is important to mention that we have more evaluations than training sentences. Therefore, this approach presents a challenging problem for the models.

Table 4. Summary of the number of training and evaluation phrases used for the creation of the IR model.

Intent	Training Phrases	Evaluation Phrases
Biometry	10	36
Client keyboard	9	41
Receipt printer	16	32
Card reader	6	34
Check scanner	5	38
Video monitor	14	29
Safe lock	8	31
Recycler module	15	35
No-match	-	14
Total	83	304

5.3. Named Entity Recognition

The NER dataset employed in our work has been annotated with named entities. The sentences from this dataset contain relevant information that can be treated as entities. These entities were utilized, for example, to specify the current status of an LED (whether on or off) or the object that caused jamming in the ATM.

In a certain user's "Related Entry", a piece of information may be critical to differentiate situations. For example, "ATM is beeping 3 times". The number 3 is an essential value to identify the problem. If the number of beeps were 2, the issue could be different. It is not appropriate to use only the IR model in these cases. Instead, combining an IR model and a NER model is recommended.

Table 5 is the foundation for constructing our NER model. Only a few entities are essential to our system and utilized to build the dataset. These entities' training phrases (or synonyms) were obtained from the knowledge database. To create the evaluation set, we consider the group of entities demonstrated in Table 6, we generated 19 random phrases without any specified entities to verify whether the model would identify any false positive for a given entity.

Table 5. Description of mapped entities utilized to create the NER model. The entities were meticulously categorized into distinct groups that indicated their respective classifications and associations.

Entity Group	Entity	Description	Number of Synonyms
Number of bips	2	Number of beeps emitted by the safe.	2
	3		2
LED status	On	Status of the red LED located on the receipt printer.	7
	Off		6
Paper state	Tangled	Paper (receipt) status after LED status check.	2
	Not tangled		4
Remove delay	Remove delay	Option for the user to remove ATM delay.	3
Screw defect	Paper	Type of crash described in the message.	3
	Banknote		5
	Check		3
Total			37

Table 6. Overview of the entities that were analyzed during the experiments. The table below describes the number of synonyms utilized to train the NER model and the corresponding number of testing phrases utilized for the evaluation process.

Entity Group	Entity	Number of Synonyms	Number of Evaluation Phrases
Led status	On	7	20
	Off	6	37
	paper	3	20
Screw defect	banknote	5	19
	check	3	18
No-match	-	-	29
Total		24	143

6. Results

In this study, we developed and evaluated a chatbot to assist banking agencies in the self-service maintenance domain as described in Section 1. The chatbot was trained using a dataset derived from the knowledge base, containing training phrases that banking agents commonly utilized. It is worth noting that we divided our analysis into two distinct scenarios. The first scenario aimed to examine the information retrieval (IR) process described in Section 5.2. In contrast, the second scenario assessed the performance of NER task in our system, described in Section 5.3, to extract relevant information from a phrase.

6.1. First Scenario—IR Evaluation

This scenario was developed to enable the evaluation of the performance of the IR process executed in our chatbot. Table 7 presents the average result for the automated evaluation metrics (Section 2.4) that were utilized to evaluate the intent’s related ATM defects.

Table 7. Macroaverage metric results.

Accuracy	Precision	Recall	F1-Score
0.95	0.94	0.92	0.93

The results presented above demonstrate that the IR process executed by the chatbot was highly effective in identifying the user’s intent with an accuracy of 0.95, indicating that a significant number of phrases from the testing set were correctly classified. The second metric, precision, with a value of 0.94, indicates that a very low rate of testing phrases was misclassified. This is crucial, as the users rely on the chatbot to correctly classify their input to receive the procedure that may solve the ATM’s issue. If the chatbot misclassifies a user’s intent, it could provide the wrong procedure, which may cause the banking agent to focus on other parts of the ATM unrelated to the given issue. This action may result in a waste of time and resources and a frustrating experience for the user. The recall metric result (0.92) indicates that the IR model missed a low rate of testing phrases. However, this can be a point of improvement, as a higher recall score indicates that the model could correctly identify more of the relevant input.

The figure below illustrates the confusion matrix Figure 3, where the no-match class represents the utterances in the model that lacked confidence in its predictions to meet the threshold for accurate classification. We analyzed the confusion matrix to better understand the chatbot’s performance when the user describes a defect.

The matrix above reveals several noteworthy findings upon closer examination. First, the model accurately classified utterances related to keyboards, card readers, video monitors, and receipt printers without misclassification. However, it encountered challenges in correctly classifying phrases related to the check scanner, with a recall of 0.89 and a high frequency of confusion with the no-match class. This result indicates that predictions for utterances related to this intent were often below the threshold.

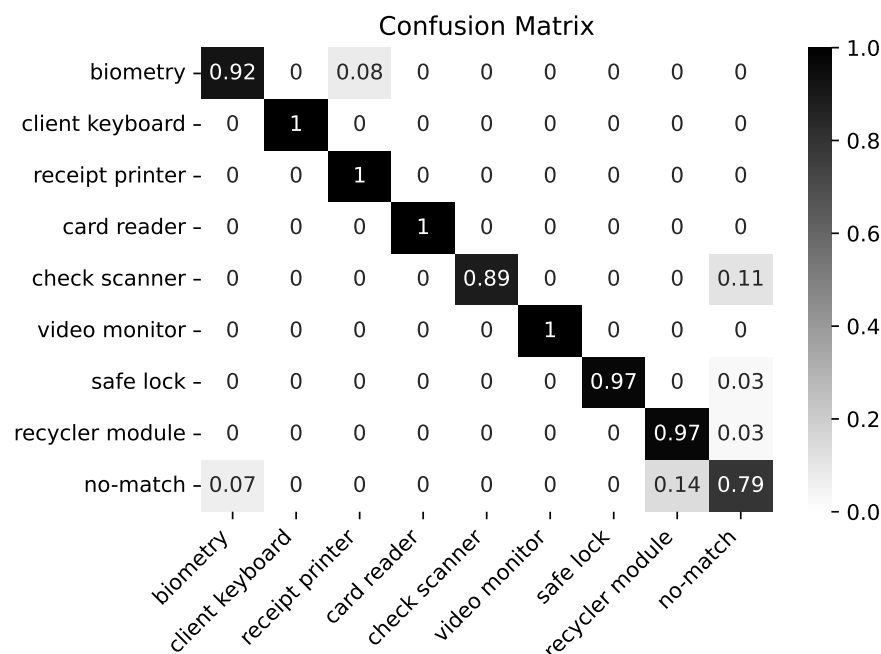


Figure 3. The displayed figure illustrates the relationship between actual and predicted intents regarding the evaluation of the IR system in the first scenario.

In addition, during the analysis, it was discovered that the recycling module intent had the highest level of confusion with the no-match class, suggesting that probabilities associated with utterances in this class were also below the threshold. The overall analysis of the confusion matrix indicates that the intent classification model exhibits commendable performance, characterized by a limited number of false positives and false negatives. Nevertheless, the analysis results also indicate certain aspects that necessitate improvement.

6.2. Second Scenario—NER Evaluation

The extraction of information through NER is crucial in specifying the context of a conversation triggered by an intent. Therefore, the evaluation of the NER approach in this scenario is presented in this section. Table 8 displays the overall performance of the NER system implemented in our model. The two groups of entities analyzed in this study are utilized in various contexts within the conversation.

Table 8. The average results for the second evaluation scenario showcase the performance of the named entity Recognition (NER) model. The figure above provides an overview of the model’s effectiveness in identifying and extracting named entities from utterances.

Accuracy	Precision	Recall	F1-Score
0.94	0.96	0.96	0.95

Based on the analysis of the macroaverage results presented in Table 8, we can draw some inferences regarding the model’s performance in detecting entities when inputting a test phrase. The high accuracy value of 0.94 indicates that the model has a greater likelihood of recognizing the correct entity for each inputted testing phrase. The precision and recall value of 0.96 indicates that the model has a low rate of detecting false positives and false negatives for each entity. The F1-score of 0.95 demonstrates a balanced measure between these two metrics. The average macro results suggest that the model accurately recognizes the relevant information from a phrase with a low error percentage. This outcome is significant because it indicates that this method provides the conversational agent with accurate information extraction capabilities when relevant information from a user’s entry needs to be extracted to define or change the context of the dialog.

6.2.1. LED Status

In this experiment, we analyzed the group of entities, LED status, described in Table 6, that are triggered when the user needs to inform the status of a LED. The entities that were annotated in its training phrases were “On” and “Off”, as specified in Table 6. Table 9 presents the metrics results for these entities.

Table 9. Macroaverage results for the entities.

Accuracy	Precision	Recall	F1-Score
0.88	0.90	0.90	0.90

The aforementioned results suggest that the model performs adequately in the task of NER. The average macro results indicate that for the entities “On” and “Off”, the model could precisely recognize the entities in the text, with a higher rate of 90%, but had a low percentage of predicting FNs, demonstrated by the average recall.

According to the confusion matrix in Figure 4, the percentage of FPs (11%) for entity “On” demonstrates that the level of precision for this entity is significantly lower than the expected level, indicating that the synonyms for this entity present a high variation. For the entity “Off”, the level of precision is even worse, which is indicated by the number of FPs for this entity.

The confusion matrix in Figure 4 shows the NER model performance in recognizing the entities in the testing phrases. The results reveal that the NER model accurately extracted all the entities associated with the “On” entity with a higher recall. Furthermore, the quantity of confusion between the entities is additional relevant information, as some entities belonging to the class “Off” were predicted by the model as being from class “On”. This indicates that the model may face difficulties distinguishing between these two entities. Furthermore, it denotes that it is necessary to introduce more synonyms for the entity that was confounded.

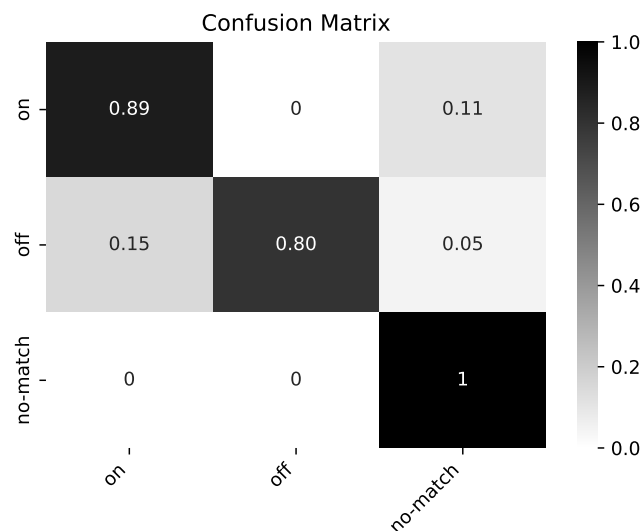


Figure 4. Confusion matrix for the entity extraction of the intent LED.

6.2.2. Screw Defect

This experiment aimed to analyze the NER task performance for the entities belonging to the screw defect entity group, as show in Table 6. The matrix below demonstrates the model’s overall performance in predicting these entities.

The results depicted in Figure 5 demonstrate that the NER model could accurately identify all entities in the testing phrases without any misclassification. Furthermore, these findings suggest that the number of synonyms specified in Table 6 for all entities pertaining

to the screw defect group was optimal, thus enabling the model to extract the entities from the testing phrases efficiently.

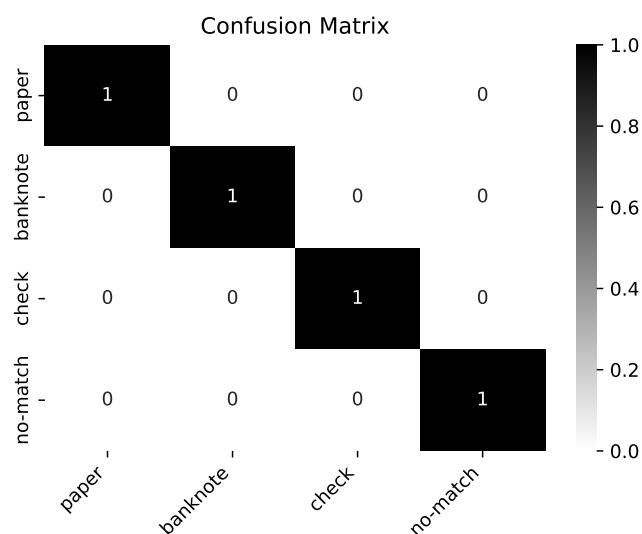


Figure 5. Confusion matrix for screw defect entity extraction.

7. Conclusions

This study introduced an innovative methodology for developing troubleshooting chatbots tailored to provide technical support for ATM maintenance, utilizing the Google Dialogflow CX framework. A critical aspect of our methodology is the detailed demonstration of structuring a knowledge database to facilitate the creation of a virtual assistant. Furthermore, we proposed a distinctive approach to evaluating chatbot performance, focusing on aspects such as intent recognition and named entity recognition.

When compared to other methodologies used in chatbot development [48–51,55–57], our strategy offers a more granular approach toward understanding the nuances of user intent and offers a systematic technique for structuring knowledge databases. This allows for an enhanced interpretation and processing of user inquiries, consequently leading to more accurate responses.

Our approach can be extrapolated to various troubleshooting chatbot applications, specifically those related to the banking sector. This makes our methodology a flexible and scalable solution for designing conversational agents. The knowledge database created during this process can be conveniently updated and expanded, ensuring that the chatbot maintains its relevance and accuracy in light of evolving user needs and technological advancements. Our methodology, therefore, presents a substantial improvement over existing techniques, providing a robust and adaptive solution for the development of troubleshooting chatbots.

It is worth mentioning that this methodology applies solely to retrieval-based chatbots, owing to the requirement of following a sequence of procedures developed by technical engineers and not being indicated, therefore, to generative-based chatbots. Nevertheless, the experimental protocol presented in this study can be utilized to evaluate various types of chatbots, particularly conversational agents that require a detailed description from the user and incorporate IR and NER techniques in their design and implementation.

Using the proposed methodology, we implemented a state-of-the-art chatbot. Our performance analysis showed we achieved 0.95 and 0.88 accuracies for IR and NER, respectively. These results demonstrate our methodology's effectiveness in creating a high-performing chatbot that can deliver a positive user experience.

In summary, this work provides a valuable contribution to the field of chatbot development, highlighting the potential of the Google Dialogflow CX framework to create sophisticated and effective conversational agents. Furthermore, our methodology can be utilized as a

foundation for developing chatbots in various domains and industries, offering a valuable tool for improving customer service, providing support, and enhancing user experiences.

Author Contributions: Conceptualization, N.A., G.A., L.N. and J.O.; methodology, G.A. and J.O.; software, N.A., G.A., L.C. and J.O.; validation, G.A., L.N. and J.O.; formal analysis, G.A. and J.O.; investigation, N.A., J.O. and G.A.; resources, G.A.; data curation, N.A., G.A., J.O. and C.F.; writing—original draft preparation, N.A., G.A., L.N. and J.O.; writing—review and editing, G.A., J.O., I.F., A.P., T.F. and I.T.; visualization, G.A., T.F. and J.O.; supervision, I.F., I.T., C.F. and A.P.; project administration, I.T., I.F. and A.P.; funding acquisition, I.T. and A.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data are unavailable due to privacy restrictions.

Acknowledgments: This research and development project was supported by Diebold Nixdorf, a global leader in providing innovative software and hardware solutions to the financial and retail industries.

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

Abbreviations

The following abbreviations are used in this manuscript:

AI	artificial intelligence
IR	intent recognition
NER	name entity recognition
NLP	natural language processing
NLU	natural language understanding

References

- Korkmaz, S. Impact of bank credits on economic growth and inflation. *J. Appl. Financ. Bank.* **2015**, *5*, 51–69.
- Petkovski, M.; Kjosevski, J. Does banking sector development promote economic growth? An empirical analysis for selected countries in Central and South Eastern Europe. *Econ. Res.-Ekonom. Istraž.* **2014**, *27*, 55–66. [CrossRef]
- Nguyen, P.T. The Impact of Banking Sector Development on Economic Growth: The Case of Vietnam's Transitional Economy. *J. Risk Financ. Manag.* **2022**, *15*, 358. [CrossRef]
- Deloitte; Federação Brasileira de Bancos (FEBRABAN). Pesquisa FEBRABAN de Tecnologia Bancária 2022—Volume 3 Transações Bancárias. 2022. Available online: <https://cmsarquivos.febraban.org.br/Arquivos/documentos/PDF/pesquisa-febraban-2022-vol-3.pdf> (accessed on 26 February 2023).
- Wang, Y.; Zhang, Y.; Sheu, P.C.; Li, X.; Guo, H. The Formal Design Model of an Automatic Teller Machine (ATM). *Int. J. Softw. Sci. Comput. Intell.* **2010**, *2*, 102–131. [CrossRef]
- Diebold Nixdorf. *Self-Service Reloaded: How Industry Leaders Maximize Customer Engagement and Strategic ROI*; Diebold Nixdorf: Green, OH, USA, 2019; Volume 1, pp. 1–42.
- Cárcel-Carrasco, J.; Cárcel-Carrasco, J.A. Analysis for the knowledge management application in maintenance engineering: Perception from maintenance technicians. *Appl. Sci.* **2021**, *11*, 703. [CrossRef]
- Tripathi, S.; Garg, R.; Varshini, K. Role of Artificial Intelligence in the Banking Sector. *Int. J. Res. Publ. Rev. J.* **2022**, *3*, 433–442.
- Dobrescu, E.M.; Dobrescu, E.M. Artificial intelligence (AI)-the technology that shapes the world. *Glob. Econ. Obs.* **2018**, *6*, 71–81.
- Adamopoulou, E.; Moussiades, L. Chatbots: History, technology, and applications. *Mach. Learn. Appl.* **2020**, *2*, 100006. [CrossRef]
- Borah, B.; Pathak, D.; Sarmah, P.; Som, B.; Nandi, S. Survey of textbased chatbot in perspective of recent technologies. In *Computational Intelligence, Communications, and Business Analytics, Proceedings of the Second International Conference, CICBA 2018, Kalyani, India, 27–28 July 2018*; Revised Selected Papers, Part II 2; Springer: Singapore, 2019; pp. 84–96.
- Caldarini, G.; Jaf, S.; McGarry, K. A literature survey of recent advances in chatbots. *Information* **2022**, *13*, 41. [CrossRef]
- Ramesh, K.; Ravishankaran, S.; Joshi, A.; Chandrasekaran, K. A survey of design techniques for conversational agents. In *Information, Communication and Computing Technology, Proceedings of the Second International Conference, ICICCT 2017, New Delhi, India, 13 May 2017*; Revised Selected Papers; Springer: Singapore, 2017; pp. 336–350.
- Nadeau, D.; Sekine, S. A survey of named entity recognition and classification. *Linguisticae Investig.* **2007**, *30*, 3–26. [CrossRef]
- Google LLC. Dialogflow CX Documentation. 2021. Available online: <https://cloud.google.com/dialogflow/cx/docs> (accessed on 26 February 2023).

16. Devlin, J.; Chang, M.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv* **2018**, arXiv:1810.04805.
17. Khurana, D.; Koli, A.; Khatter, K.; Singh, S. Natural language processing: State of the art, current trends and challenges. *Multimed. Tools Appl.* **2023**, *82*, 3713–3744. [[CrossRef](#)] [[PubMed](#)]
18. Abdellatif, A.; Badran, K.; Costa, D.E.; Shihab, E. A Comparison of Natural Language Understanding Platforms for Chatbots in Software Engineering. *IEEE Trans. Softw. Eng.* **2022**, *48*, 3087–3102. [[CrossRef](#)]
19. Braun, D.; Mendez, A.H.; Matthes, F.; Langen, M. Evaluating natural language understanding services for conversational question answering systems. In Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue, Saarbrücken, Germany, 15–17 August 2017; pp. 174–185.
20. Godse, N.A.; Deodhar, S.; Raut, S.; Jagdale, P. Implementation of chatbot for ITSM application Using IBM watson. In Proceedings of the 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), Pune, India, 16–18 August 2018; pp. 1–5.
21. Dhavan, S. Smart Medicare Chatbot Using Dialogflow and Support Vector Machine Algorithm. *Int. J. Res. Appl. Sci. Eng. Technol.* **2021**, *9*, 1848–1860. [[CrossRef](#)]
22. Suhaili, S.M.; Salim, N.; Jambli, M.N. Service chatbots: A systematic review. *Expert Syst. Appl.* **2021**, *184*, 115461. [[CrossRef](#)]
23. Mohit, B. Named entity recognition. In *Natural Language Processing of Semitic Languages*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 221–245.
24. Li, J.; Sun, A.; Han, J.; Li, C. A Survey on Deep Learning for Named Entity Recognition. *IEEE Trans. Knowl. Data Eng.* **2022**, *34*, 50–70. [[CrossRef](#)]
25. Zubani, M.; Sigalini, L.; Serina, I.; Putelli, L.; Gerevini, A.E.; Chiari, M. A performance comparison of different cloud-based natural language understanding services for an Italian e-learning platform. *Future Internet* **2022**, *14*, 62. [[CrossRef](#)]
26. Hussain, S.; Ameri Sianaki, O.; Ababneh, N. A survey on conversational agents/chatbots classification and design techniques. In *Web, Artificial Intelligence and Network Applications, Proceedings of the Workshops of the 33rd International Conference on Advanced Information Networking and Applications (WAINA-2019), Matsue, Japan, 29 March 2019*; Springer: Cham, Switzerland, 2019; pp. 946–956.
27. Luo, B.; Lau, R.Y.; Li, C.; Si, Y.W. A critical review of state-of-the-art chatbot designs and applications. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2022**, *12*, e1434. [[CrossRef](#)]
28. Thorat, S.A.; Jadhav, V. A Review on Implementation Issues of Rule-based Chatbot Systems. *SSRN Electron. J.* **2020**.
29. Singh, S.; Thakur, H.K. Survey of various AI chatbots based on technology used. In Proceedings of the 2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), Noida, India, 4–5 June 2020; pp. 1074–1079.
30. Singh, J.; Joesph, M.H.; Jabbar, K.B.A. Rule-based chatbot for student enquiries. *J. Phys. Conf. Ser.* **2019**, *1228*, 012060. [[CrossRef](#)]
31. Vishwakarma, A. A Review & Comparative Analysis on Various Chatbots Design. *Int. J. Comput. Sci. Mob. Comput.* **2021**, *10*, 72–78. [[CrossRef](#)]
32. Hien, H.T.; Cuong, P.N.; Nam, L.N.H.; Nhung, H.L.T.K.; Thang, L.D. Intelligent assistants in higher-education environments: The FIT-EBot, a chatbot for administrative and learning support. In Proceedings of the 9th International Symposium on Information and Communication Technology, Danang City, Vietnam, 6–7 December 2018; pp. 69–76.
33. Suta, P.; Lan, X.; Wu, B.; Mongkolnam, P.; Chan, J.H. An overview of machine learning in chatbots. *Int. J. Mech. Eng. Robot. Res.* **2020**, *9*, 502–510. [[CrossRef](#)]
34. Kapočiūtė-Dzikienė, J. A Domain-Specific Generative Chatbot Trained from Little Data. *Appl. Sci.* **2020**, *10*, 2221. [[CrossRef](#)]
35. Larson, S.; Leach, K. A Survey of Intent Classification and Slot-Filling Datasets for Task-Oriented Dialog. *arXiv* **2022**, arXiv:2207.13211.
36. Bhoir, S.V.; Patil, S.R.; Mogul, I.Y. *Person-Based Automation with Artificial Intelligence Chatbots: A Driving Force of Industry 4.0*; Elsevier: New York, NY, USA, 2022; pp. 215–244. [[CrossRef](#)]
37. Suhel, S.F.; Shukla, V.K.; Vyas, S.; Mishra, V.P. Conversation to automation in banking through chatbot using artificial machine intelligence language. In Proceedings of the 2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), Noida, India, 4–5 June 2020; pp. 611–618.
38. Weizenbaum, J. ELIZA—A computer program for the study of natural language communication between man and machine. *Commun. ACM* **1966**, *9*, 36–45. [[CrossRef](#)]
39. Huesmann, L.R.; Schank, R.C.; Colby, K.M. Computer Models of Thought and Language. *Am. J. Psychol.* **1974**, *87*, 751–754. [[CrossRef](#)]
40. Wallace, R.S. *The Anatomy of A.L.I.C.E.*; Springer: Dordrecht, The Netherlands, 2009; pp. 181–210. [[CrossRef](#)]
41. yeung Shum, H.; dong He, X.; Li, D. From Eliza to XiaoIce: Challenges and opportunities with social chatbots. *Front. Inf. Technol. Electron. Eng.* **2018**, *19*, 10–26. [[CrossRef](#)]
42. Chung, M.; Ko, E.; Joung, H.; Kim, S.J. Chatbot e-service and customer satisfaction regarding luxury brands. *J. Bus. Res.* **2020**, *117*, 587–595. [[CrossRef](#)]
43. Landim, A.R.D.B.; Pereira, A.M.; Vieira, T.; de B. Costa, E.; Moura, J.A.B.; Wanick, V.; Bazaki, E. Chatbot design approaches for fashion E-commerce: An interdisciplinary review. *Int. J. Fash. Des. Technol. Educ.* **2022**, *15*, 200–210. [[CrossRef](#)]

44. Khan, M.M. Development of an e-commerce sales Chatbot. In Proceedings of the 2020 IEEE 17th International Conference on Smart Communities: Improving Quality of Life Using ICT, IoT and AI (HONET), Charlotte, NC, USA, 14–16 December 2020; pp. 173–176.
45. Amiri, P.; Karahanna, E. Chatbot use cases in the Covid-19 public health response. *J. Am. Med. Inform. Assoc.* **2022**, *29*, 1000–1010. [[CrossRef](#)]
46. Sangrà, A.; Vlachopoulos, D.; Cabrera, N. Building an inclusive definition of e-learning: An approach to the conceptual framework. *Int. Rev. Res. Open Distrib. Learn.* **2012**, *13*, 145–159. [[CrossRef](#)]
47. Huang, W.; Hew, K.F.; Fryer, L.K. Chatbots for language learning—Are they really useful? A systematic review of chatbot-supported language learning. *J. Comput. Assist. Learn.* **2022**, *38*, 237–257. [[CrossRef](#)]
48. Casillo, M.; Colace, F.; Fabbri, L.; Lombardi, M.; Romano, A.; Santaniello, D. Chatbot in industry 4.0: An approach for training new employees. In Proceedings of the 2020 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE), Takamatsu, Japan, 8–11 December 2020; pp. 371–376.
49. Colabianchi, S.; Bernabei, M.; Costantino, F. Chatbot for training and assisting operators in inspecting containers in seaports. *Transp. Res. Procedia* **2022**, *64*, 6–13. [[CrossRef](#)]
50. Alhassan, N.A.; Albarrak, A.S.; Bhatia, S.; Agarwal, P. A Novel Framework for Arabic Dialect Chatbot Using Machine Learning. *Comput. Intell. Neurosci.* **2022**, *2022*, 1844051. [[CrossRef](#)]
51. Følstad, A.; Taylor, C. Conversational repair in chatbots for customer service: The effect of expressing uncertainty and suggesting alternatives. In *Chatbot Research and Design, Proceedings of the Third International Workshop, CONVERSATIONS 2019, Amsterdam, The Netherlands, 19–20 November 2019*; Revised Selected Papers 3; Springer: Cham, Switzerland, 2020; pp. 201–214.
52. Mleczko, K. Chatbot as a Tool for Knowledge Sharing in the Maintenance and Repair Processes. *Multidiscip. Asp. Prod. Eng.* **2021**, *4*, 499–508. [[CrossRef](#)]
53. Lin, C.C.; Huang, A.Y.; Yang, S.J. A review of ai-driven conversational chatbots implementation methodologies and challenges (1999–2022). *Sustainability* **2023**, *15*, 4012. [[CrossRef](#)]
54. Trivedi, A.; Gor, V.; Thakkar, Z. Chatbot generation and integration: A review. *Int. J. Adv. Res. Ideas Innov. Technol.* **2019**, *5*, 1308–1311.
55. López, A.; Sánchez-Ferreres, J.; Carmona, J.; Padró, L. From process models to chatbots. In *Advanced Information Systems Engineering, Proceedings of the 31st International Conference, CAiSE 2019, Rome, Italy, 3–7 June 2019*; Proceedings 31; Springer: Cham, Switzerland, 2019; pp. 383–398.
56. Sánchez-Díaz, X.; Ayala-Bastidas, G.; Fonseca-Ortiz, P.; Garrido, L. A knowledge-based methodology for building a conversational chatbot as an intelligent tutor. In *Advances in Computational Intelligence, Proceedings of the 17th Mexican International Conference on Artificial Intelligence, MICAI 2018, Guadalajara, Mexico, 22–27 October 2018*; Proceedings, Part II 17; Springer: Cham, Switzerland, 2018; pp. 165–175.
57. Nguyen, H.; Tran, T.V.; Pham, X.T.; Huynh, A.T.; Do, N. Ontology-based integration of knowledge base for building an intelligent searching chatbot. *Sens. Mater.* **2021**, *33*, 3101–3123. [[CrossRef](#)]
58. Sarbabidya, S.; Saha, T. Role of chatbot in customer service: A study from the perspectives of the banking industry of Bangladesh. *Int. Rev. Bus. Res. Pap.* **2020**, *16*, 231–248.
59. Rustamov, S.; Bayramova, A.; Alasgarov, E. Development of dialogue management system for banking services. *Appl. Sci.* **2021**, *11*, 995. [[CrossRef](#)]
60. Fares, O.H.; Butt, I.; Lee, S.H.M. Utilization of artificial intelligence in the banking sector: A systematic literature review. *J. Financ. Serv. Mark.* **2022**.
61. Alt, M.A.; Vizeli, I.; Săplăcan, Z. Banking with a Chatbot – A Study on Technology Acceptance. *Stud. Univ. Babeş-Bolyai Econ.* **2021**, *66*, 13–35. [[CrossRef](#)]
62. Wube, H.D.; Esubalew, S.Z.; Weldesellase, F.F.; Debelee, T.G. Text-Based Chatbot in Financial Sector: A Systematic Literature Review. *Data Sci. Financ. Econ.* **2022**, *2*, 232–259. [[CrossRef](#)]
63. Muhammad, A.F.; Susanto, D.; Alimudin, A.; Adila, F.; Assidiqi, M.H.; Nabhan, S. Developing English conversation chatbot using dialogflow. In Proceedings of the 2020 International Electronics Symposium (IES), Surabaya, Indonesia, 29–30 September 2020; pp. 468–475.
64. Dall’Acqua, A.; Tamburini, F. Implementing a Pragmatically Adequate Chatbot in DialogFlow CX. In Proceedings of the CLiC-it, Milan, Italy, 26–28 January 2021.

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