engineering proceedings

MDPI

Proceeding Paper

Intelligent Chatbot System Design, Development, and Deployment for Client Queries: Efficient and Effective Perception and Cognition †

Tlou Sebola *, Michael Ayomoh * and Brain Ndlovu *

Department of Industrial & Systems Engineering, University of Pretoria, Hatfield, Pretoria 0083, South Africa
* Correspondence: u16215347@tuks.co.za (T.S.); michael.ayomoh@up.ac.za (M.A.); bn.ndlovu@up.ac.za (B.N.)
† Presented at the 12th International Electronic Conference on Sensors and Applications, 12–14 November 2025; Available online: https://sciforum.net/event/ECSA-12.

Abstract

The recent synergistic explosion of artificial intelligence and the world of machines, in a bid to make them smarter entities as a result of the fourth industrial revolution, has resulted in the concept of chatbots, which have evolved over the years and gained heightened attention for the sustainability of most human corporations. Organisations are increasingly utilising chatbots to enhance customer engagement through the process of agent-based autonomous sensing, interaction, and enhanced service delivery. The current state of the art in chatbot technology is such that the system lacks the ability to conduct text-sensing in a bid to acquire new information or learn from the external world autonomously. This has limited the current chatbot systems to being system-controlled interactive agents, hence, strongly limiting their functionalities and posing a question on the purported intelligence. In this research, an integrated framework that combines the functionalities and capabilities of a chatbot and machine learning was developed. The integrated system was designed to accept new text queries from the external world and import them into the knowledge base using the SQL (Structured Query Language) syntax and MySQL workbench (version 8.0.44). The search engine and decision-making cluster was built in the Python (version 3.12.7) coding environment with the learning process, solution adaptation, and inference, anchored using a reinforcement machine learning approach. This mode of chatbot operation, with an interactive capacity, is known as the mixed controlled system mode, with a viable human–machine system interaction. The smart chatbot was assessed for efficacy using performance metrics (response time, accuracy) and user experience (usability, satisfaction). The analysis further revealed that several self-governed chatbots deployed in most corporate organisations are system-controlled and significantly constrained, hence lacking the ability to adapt or filter queries beyond their predefined databases when users employ diverse phrasing or alternative terms in their interactions.

Keywords: artificial intelligence; machine learning; chatbot; service sensors; knowledge base; reinforcement learning

check for updates

Academic Editor: Stefano Mariani

Published: 17 November 2025

Citation: Sebola, T.; Ayomoh, M.; Ndlovu, B. Intelligent Chatbot System Design, Development, and Deployment for Client Queries: Efficient and Effective Perception and Cognition. *Eng. Proc.* **2025**, *118*, 57. https://doi.org/10.3390/ECSA-12-26595

Copyright: © 2025 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/).

1. Introduction

Chatbots powered by artificial intelligence (AI) are rapidly being used as strategic sensing tools for the improvement of customer relationship management (CRM) by delivering real-time help, boosting service sensory activities, accessibility, and personalising user experiences. Their integration with customer service frameworks addresses typical

CRM difficulties such as inefficient data handling, inconsistent service delivery, and limited operational hours [1]. One important factor influencing chatbot adoption is user satisfaction. According to [2], chatbot functionalities—such as anthropomorphic traits, ease of use, response accuracy, and interactivity—influence user satisfaction. According to [3], chatbots that exhibit human-like traits are viewed as empathetic as they increase user engagement and satisfaction through service sensors and cognitive intelligence. Privacy issues, however, continue to stand in the way of wider acceptance. It was stated in [4] that users' comfort and contentment may be negatively impacted by perceived risks related to AI systems handling data. This suggests that chatbot design must incorporate service sensors, privacy, and trust.

Due to the current state of the art in chatbot technology, the system is unable to carry out text-sensing to learn from the outside world or gain new information on its own. This has severely restricted the capabilities of the existing chatbot systems and raised doubts about their alleged intelligence by confining them to system-controlled interactive agents. In this research, the goal is to create a more intelligent, adaptive, and context-aware chatbot framework by incorporating machine learning techniques, such as deep learning and reinforcement learning, into a chatbot system to enhance its knowledge base by storing new queries to increase the chatbot's effectiveness and efficiency.

Ref. [1] conceived chatbot performance using three essential quality factors, drawing on information systems theory: System quality encompasses responsiveness, availability, and flexibility. Poor system sensor performance can lower user engagement and make decision-making more difficult. Information Quality: User perception and system confidence are strongly impacted by the accuracy, comprehensiveness, and relevancy of chatbot responses [5]. Service Quality: Characteristics like certainty, empathy, and dependability influence customer satisfaction and intent to use the chatbot again. The triadic effects of system, information, and service quality on satisfaction and loyalty in AI-assisted CRM are highlighted by earlier studies [6], which is consistent with these dimensions. Integration with current databases and systems must be easy for chatbot deployment to be effective. Ref. [7] claimed that implementing a chatbot on well-known platforms, such as WhatsApp, improves its efficacy and reduces user entry barriers [1] through developed context-aware discourse systems using transformer-based models and long short-term memory (LSTM). CRM performance in academic contexts was greatly improved by real-time data processing and service automation made possible through integration with a university database via secure APIs.

Advances in service sensor schemes, machine learning (ML), natural language processing (NLP), and artificial intelligence (AI) have significantly changed the chatbot development industry. Understanding the architectural, functional, and evaluative foundations of conversational agents is essential as they become increasingly integrated into digital ecosystems. Ref. [8] provided a literature review with a thorough examination of previous studies, tracing the development of chatbot technologies, breaking down their essential elements, and assessing the methodologies that determined their effectiveness. It places Cahn's contribution in larger scholarly and technical discussions, emphasising both his groundbreaking research and new developments in conversational AI.

Alan Turing's idea of the imitation game, sometimes called the Turing Test [9], is often cited as the intellectual origin of chatbot systems. "Can machines think?" was the crucial question posed by Turing's theoretical framework. According to [8], ELIZA paved the way for rule-based dialogue systems by identifying keywords and translating them into pre-programmed responses. This paradigm was expanded by later inventions like ALICE [10], which used Artificial Intelligence Markup Language (AIML), and PARRY [11], which imitated a paranoid schizophrenic. These systems included domain-specific response

capabilities and hierarchical rule structures, but they remained essentially deterministic. The emergence of virtual assistants such as Amazon's Alexa and Apple's Siri highlights a move away from symbolic to sub-symbolic AI approaches and is an example of the movement towards systems that use statistical models and real-time web-based knowledge retrieval. In [8], it was stated that evaluation is essential to chatbot creation because it guarantees the system's linguistic integrity, user satisfaction, and functionality. The PARADISE (PARAdigm for DIalogue System Evaluation) framework, which combines subjective and objective measurements, is thoroughly described by [8]. While objective metrics concentrate on task completion, dialogue cost, and system efficiency, subjective measures include user-reported satisfaction on aspects including naturalness, friendliness, and readiness to reuse [12]. A chatbot can sense, comprehend, extract, and interpret meanings from human inputs via service sensors and metrication processes, and natural language processing. The research in [8] went into detail about important NLP tasks like sensing, information extraction, intent categorisation, and Dialogue Act (DA) recognition.

There are three primary types of generating strategies: generative, rule-based, and retrieval-based models. ALICE and other rule-based systems depend on pre-established pairs. On the other hand, retrieval-based algorithms employ sizable collections of human dialogues (such as those found on Reddit and Twitter) and choose relevant answers using similarity scoring methods like TF-IDF and cosine similarity [8]. At the forefront of chatbot technology are generative models. Statistical Machine Translation (SMT) models that approach response creation as a translation task are covered by [8]. Sequence-to-Sequence (Seq2Seq) models that use recurrent neural networks (RNNs) and encoder–decoder architectures are more sophisticated. Based on past discourse, these models are trained to anticipate answer sequences [13]. To maximise long-term conversation coherence and user engagement, reinforcement learning is used to solve issues such as answer blandness and repetition [14]. The knowledge base serves as the foundation for a chatbot's ability to retrieve information. While contemporary systems make use of web-scraped data, annotated dialogue sets, and crowdsourced chat logs, early systems relied on manually created corpora [15]. Tracking user intent, controlling context, and producing responses that are appropriate for the situation are all part of dialogue management techniques. For state management, Ref. [8] emphasises methods like policy-based reinforcement learning (RL), finite state machines, and Partially Observable Markov Decision Processes (POMDPs).

The introduction of deep learning (DL) caused dialogue systems to undergo a paradigm change. Direct answer creation from discourse history was made possible by RNNs, specifically Seq2Seq architectures [16,17]. These models learned representations from data, eliminating the requirement for manually constructed dialogue states. By adding latent variables, further developments like Variational Autoencoders (VAEs) improved generative capacities and allowed systems to provide more contextually appropriate and varied answers [18]. These methods also made it easier to train dialogue models from beginning to end, which helped make contemporary conversational agents more scalable. Although deep learning allowed algorithms to mimic human reactions, it did not ensure user satisfaction or long-term coherence. In response, scholars started using RL to optimise discourse policies according to long-term incentives [19]. Through reinforcement learning (RL), the agent learns to select answers that optimise cumulative user satisfaction by redefining discourse as a sequential decision-making process. A lack of data frequently limits the initial applications of RL to simulations [14]. However, initiatives like the Amazon Alexa Prize created chances for "learning in the wild," wherein real users were used to train and assess systems like MILABOT [18]. The creation of the MILABOT chatbot is an example of how developments in deep learning, statistical modelling, rule-based systems, and RL have come together. Its real-world learning framework and ensemble architecture mark a

major advancement in the creation of scalable, flexible, and captivating dialogue systems. MILABOT illustrates the effectiveness and feasibility of integrating deep reinforcement learning into conversational AI through supervised pretraining, reinforcement learning optimisation, and deployment-based evaluation.

2. Research Design and Approach

This section presents research design and methodology, which integrated reinforcement learning (RL) techniques into the chatbot system. The methodological articulation is premised on the integration of the sub-entities of the designed chatbot system and the training procedure that facilitated the process of text-query sensing intelligence and the corresponding cognitive response carried out effectively and efficiently in a bid to communicate with the querying client. Figure 1 presents the system's architecture for the integrated framework. The architecture is premised on the Design Science Research (DSR) methodology, which utilises the concepts of design, build, and evaluation of artefacts to address challenging real-world issues. The research adapted Doyle et al.'s (2019) multi-phase DSR model.

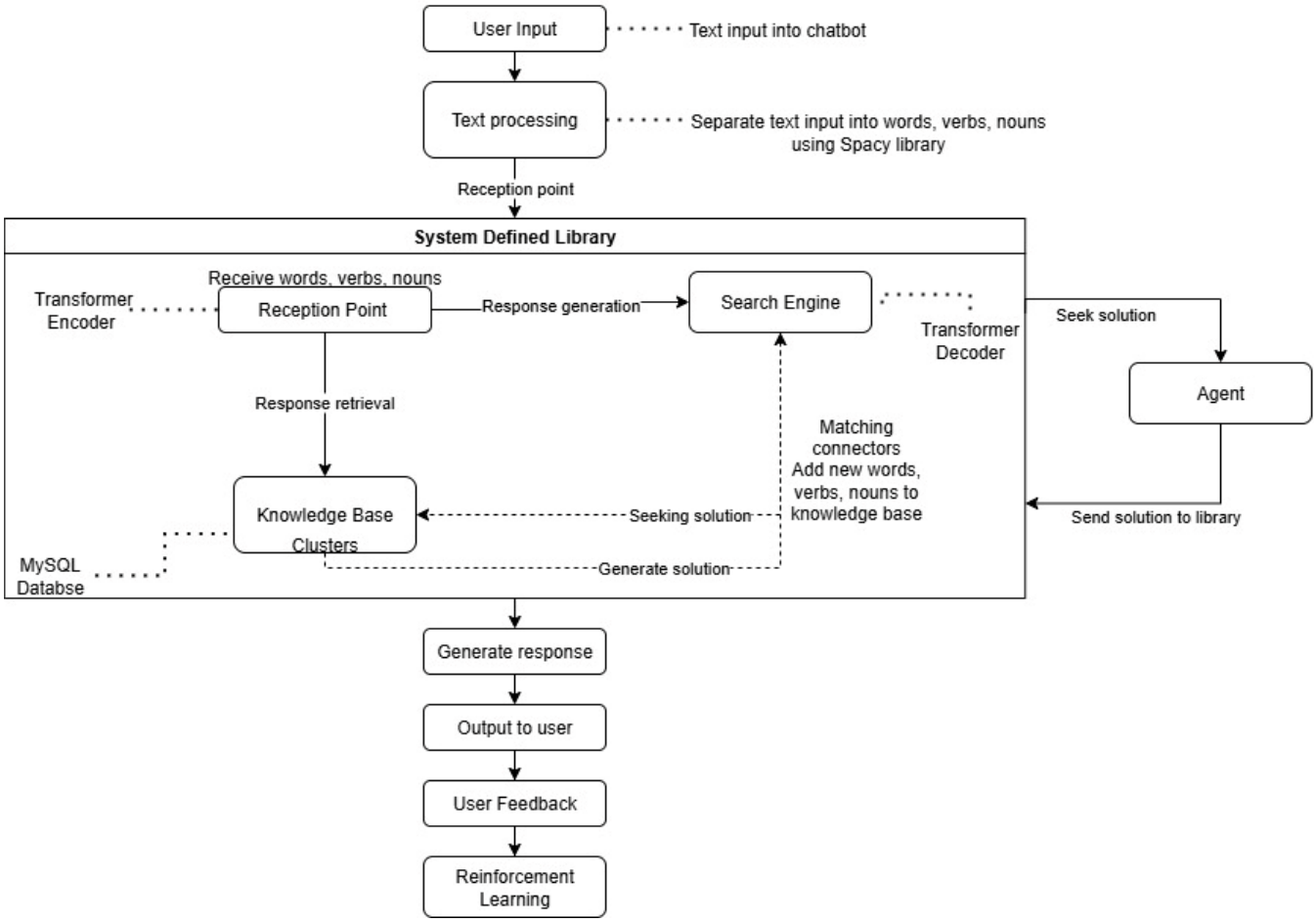


Figure 1. System's architecture.

2.1. System's Architecture

Figure 1 depicts how the system functions from the time a client inputs text into the chatbot until a response is provided. The chatbot's design is limited to accepting text inputs. Using the reinforcement machine learning approach as an anchor, the search engine and decision-making cluster were constructed within the Python coding environment, incorporating sensing, learning, solution adaptation, and inference from necessary machine

learning (ML) and Natural Language Processing (NLP) libraries. MySQL workbench was used to create the database; the database was queried using SQL (Structured Query Language) syntax. Python is highly capable of processing and interpreting data, while MySQL offers an organised framework for storing it.

Key phases of how the system operated are as follows:

- *The User Input Phase*
  The client put a query into the chatbot, such as "is the bank open today?"
  The chatbot processed and interpreted the raw text.
- *Text processing phase*
  The text was standardised by breaking it into words/tokens.
  The NLP spaCy library was used for named entity recognition and identified entities such as "bank". This was to ensure that the input was standard for intent detection and embedding.
- *Transformer encoder phase*
  The system was able to sense the input query and conduct cognition on it by analysing the input to develop understanding. This process can be referred to as Sensors Analyses Gains Understanding (SAGU).
  SBERT transformed sentences into vector spaces and encoded them as dense embedding vectors that were used to conduct semantic search, clustering, and information retrieval. The significance of embeddings lies in their ability to help the chatbot comprehend intent and compare queries to vector-space knowledge base items.
- *Knowledge base phase*
  SBERT embedding was compared to what existed in the database to find a match, and the relevant text was retrieved.
- *Search engine*
  When no match was found in the knowledge base, T5 generated a response. The transformer decoder is the RL policy network, as RL modifies its parameters according to the quality of the generated response.
- *Output to client phase*
  The client received a response that was generated/retrieved.
- The query was sent to a banking agent when the system was unable to respond, and the result was returned to the library before being delivered to the client.
- *Feedback phase*
  The chatbot requested feedback from the client by using thumbs up or down for the conversation. The conversation ended after feedback was provided and/or when the client left the chat loop.
- *Reinforcement Learning phase*
  RL was used for continuous improvement to update its policy and used feedback as its reward signal to ensure that the chatbot achieves better results with time and learns which answers work best for clients.
  Rewards are assessed as follows:
  - Thumbs up is a positive reward.
  - Thumbs down is a negative reward.

  The system-defined library's rules:
- To incorporate these new external text queries into its knowledge base, utilising SQL approaches, the chatbot backend system was able to identify words, nouns, and verbs that were not present in the knowledge base.
- To improve its clusters, the search engine was trained to add newly identified words, nouns, and verbs with a 90% matching rate to its knowledge base via matching connections.

- When the system failed to respond to a client query, it was able to pull in an agent to obtain a response; the agent's response was then saved in a library.

As the sensor layer, the user input records text queries in real time and sends them to the system for additional analysis. After learning, the chatbot used perception to identify intent, context, and semantic similarity using natural language comprehension techniques like SBERT embeddings. Raw text is converted into structured representations that the system can use for reasoning by this perceptive layer. Expanding upon this, cognition was achieved by reinforcement learning algorithms and transformer-based decoders that continuously modified rules in response to user input while producing contextually relevant responses. This allowed the chatbot to develop into an intelligent, client-focused support system by sensing and perceiving client queries as well as cognitively assessing and optimising its responses.

2.2. Transformer Encoder and Decoder

A transformer is a neural network design that converts text into numerical representations called tokens. It is based on the multi-head attention mechanism. After that, each token is converted into a vector by searching for it in a word embedding table. Each layer then contextualises each token with other (unmasked) tokens within the context window using a concurrent multi-head attention approach, increasing the signal for important tokens and decreasing it for less important ones. Figure 2 shows the process flow that involves transformers and illustrates what happens when an intent is found in the knowledge base (KB), as well as when no intent is found, which then triggers the transformer-decoder to generate a response.

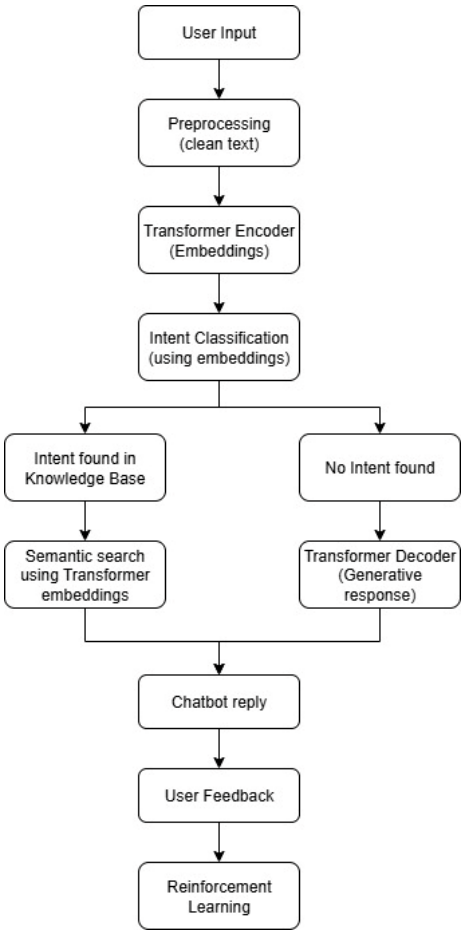


Figure 2. Transformer-enhanced chatbot query handling.

In this study, we used Sentence BERT (SBERT), which is a version of BERT (Bidirectional Encoder Representations from Transformers) that generates dense, meaningful representations of complete sentences. For the transformer decoder, we made use of T5 (text-to-text transformer) to generate a response when information is not retrieved from the knowledge base.

2.3. Reinforcement Learning

The reward function in RL was essential for regulating the learning process and controlling the behaviour of the system. The chatbot had to produce factually accurate, contextually relevant, and user-friendly responses. The RL agent gained knowledge by acting in the environment, changing its present state, and producing a reward. Markov Decision Process (MDP) is as follows:

State (s): User query and conversation history.

Action (a): The decoder's generated response.

Reward (r): Ranks the quality of the generated answer.

Policy (π): The algorithm used by the chatbot to generate responses, parameter θ

$$J(\theta) = E_{\pi\theta} \sum_{t=0}^{T} \gamma^t rt$$

T = dialogue horizon.

γ = discount factor for rewards.

The aim was to determine a policy that maximises the reward.

2.4. Model Development

Several interdependent layers of system architecture, such as natural language comprehension, knowledge base, machine learning, RL model integration, and system orchestration, were required to develop the integrated framework. The process of designing a chatbot system involves several key stages:

Design Phase

During the design phase, the chatbot's goals, functionalities, and user experience are defined. Key aspects include the following:

Sensing the Use Case:

- The chatbot was designed as a live chat assistant for customer service and support via service-centric sensors. The chatbot was modelled for a banking service system.

Design of Conversational Flow:

- Constructing grammatical data structures.
- Formulating answers.

Selecting the required technologies:

- AI-powered chatbot.
- Using NLP libraries and machine learning models.
- Integrating the RL model as a backup plan for the model to address unidentified/new queries.

2.5. Development Phase

During the development stage, the chatbot was constructed utilising NLP libraries, AI models, database management, and a programming language. Key steps include:

Selecting a Development Framework:

- MySQL was used to create and store the database, while SQL syntax was used for queries.
- The chatbot was created using Python as the programming language.
- SpaCy was used for processing text input.
- Transformer encoder and decoder were used to retrieve a response and generate a response, respectively.
- RL was integrated for continuous improvement.

Chatbot training:

- Constant learning through feedback loops and RL techniques.

Integration of Backend:

- Python libraries were used to connect the database to the coding environment.

Frontend and Deployment Channels:

- Python was used to develop the chatbot, and the Python terminal was used for interactions.

2.6. Deployment Phase

The deployment stage made sure the chatbot functions well in real-world settings, and it is then continuously optimised.

- The Python terminal was used for the chatbot interactions.
- Performance tracking and enhancement took place by monitoring the learning phase and documenting database modifications as audit logs.
- Frequent modifications were made to the database in accordance with user interactions and responses.

The DSR methodology offered an organised and iterative way for designing, developing, deploying, and evaluating a practical structure for integrating RL into a chatbot system.

3. Results and Discussion

This section presents the chatbot system's performance results. The database was designed for the banking system, where clients enquire about the bank being open for business or operating for the day. The sample of what exists in the database is shown in Figure 3.

id	Query	Response
1	hello	Hi there! How can I help you?
2	how are you	I'm doing great!
3	goodbye	Goodbye! Have a great day!
4	Are you open today?	Yes, we are open today from 9 AM to 4 PM.
5	Is the bank open right now?	Yes, we are open today from 9 AM to 4 PM.
6	What time do you open?	Yes, we are open today from 9 AM to 4 PM.
7	Is your branch open at the moment?	Yes, we are open today from 9 AM to 4 PM.
8	Can I visit the bank now?	Yes, we are open today from 9 AM to 4 PM.
9	Are you operating right now?	Yes, we are open today from 9 AM to 4 PM.
10	Can I come to the bank right now?	Yes, we are open today from 9 AM to 4 PM.

Figure 3. Preliminary test database library.

A client interacted with the chatbot and enquired if the bank was operating or open on that day. Reinforcement Learning (RL) updates its policy of correctness and stores new queries once the client gives feedback from sensory perception and cognitive response activities. The following two scenarios detail the interactions and how the integrated framework improves the adaptability and accuracy of the chatbot system.

Scenario 1 explains a situation in whereby the chatbot successfully responded to client queries by retrieving answers from the knowledge base.

Scenario 1: Chatbot successfully responded to client queries and received positive feedback.

Client: "Hello"

Chatbot: "Hi there! How can I help?"

Client: "What time do you open?

Chatbot: Yes, we are open today from 9am to 4pm"

Client: "Is the bank not shut?"

Chatbot: "Yes, we are open today from 9am to 4pm"

Chatbot: "Rate the interaction by reacting with a thumbs up or thumbs down"

Client: "Thumbs up reaction"

Client: "Goodbye"

Chatbot: "Goodbye! Have a great day!"

In scenario 2, the client's query is routed to an agent when the system fails to generate a response. The system then learns from the agent's response to improve and expand its knowledge base.

Scenario 2: Chatbot failed to respond to client's query, agent successfully responded to client's query and the overall interaction received positive feedback.

Client: "Hello"

Chatbot: "Hi there! How can I help?"

Client: "What time do you open?

Chatbot: Yes, we are open today from 9am to 4pm"

Client: "Is the branch in Hatfield open?"

Chatbot: "Wait for feedback"

Client: "Hatfield operating hours?"

Chatbot: "You are being routed to an agent"

Agent: "Reviewing your message"

Agent: "Hatfield branch is open from 9am to 4pm"

Agent: "Is there anything else I can help with?"

Client: "No thank you"

Agent: "Routed back to chatbot"

Another day's interaction with a client

Client: "Is the branch in Hatfield open?"

Chatbot: " Yes, Hatfield branch is open from 9am to 4pm"

Chatbot: "Rate the interaction by reacting with a thumbs up or thumbs down"

Client: "Thumbs up reaction"

To continuously expand its knowledge base, the chatbot system was built with an adaptive learning and sensing mechanism that allowed it to learn and store new queries in

real time. An integrated feedback loop was used to achieve this, identifying queries that did not correspond to any pre-existing solutions. By reducing repetition in unanswered queries and increasing accuracy over time, this dynamic learning approach to store new queries ensured the chatbot developed with user interactions.

4. Conclusions

An essential component of the chatbot system's overall effectiveness is its implementation of an adaptive service-sensor scheme and learning mechanism. The knowledge base was regularly improved and changed to reflect real user interactions by allowing the system to receive, via an in-built service-sensor scheme, and store new queries in real time. This feature decreased the frequency of unanswered or repeated requests, ensuring that the chatbot stayed contextually relevant and grew more accurate over time. This self-improving, adaptive architecture showed how reinforcement learning-based conversational chatbots may advance from static, rule-based frameworks to intelligent, autonomous knowledge management systems that can scale and evolve over time. Future research will expand on the knowledge base and user experiences to handle a wider range of client inquiries utilising banking services.

Author Contributions: Conceptualization, M.A.; methodology, T.S.; software, T.S.; validation, T.S. and M.A.; formal analysis, T.S. and M.A.; investigation, T.S.; resources, T.S.; data curation, T.S.; writing—original draft preparation, T.S.; writing—review and editing, T.S. and M.A.; visualisation, T.S. and M.A.; supervision, M.A. and B.N.; project administration, T.S. and M.A. 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: Data are provided in this article.

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

References

1. Sofiyah, F.R.; Dilham, A.; Hutagalung, A.Q.; Yulinda, Y.; Lubis, A.S.; Marpaung, J.L. The chatbot artificial intelligence as the alternative customer services strategic to improve the customer relationship management in real-time responses. *Int. J. Artif. Intell. Res.* **2024**, *6*, 54–72. [CrossRef]
2. Nagarajan, S. The role of interactivity in enhancing chatbot acceptance. *Int. J. Hum.-Comput. Stud.* **2023**, *169*, 102930.
3. Jenneboer, W.; Van Esch, P.; Van den Bossche, A. Anthropomorphism in chatbots: Its effect on user engagement. *J. Retail. Consum. Serv.* **2022**, *65*, 102860.
4. Lee, S.; Kim, J.; Park, Y. Privacy concerns in the use of AI-powered chatbots: The moderating role of trust. *Telemat. Inform.* **2021**, *63*, 101660.
5. Jo, H. System quality and decision-making effectiveness in AI applications. *J. Inf. Syst.* **2022**, *36*, 55–72.
6. Hsu, C.L.; Lin, J.C.C. Exploring service quality, satisfaction and loyalty in AI-enabled services. *Serv. Ind. J.* **2023**, *43*, 23–45.
7. Smutny, P.; Schreiberova, P. Chatbots for learning: A review of educational chatbots. *Educ. Inf. Technol.* **2020**, *25*, 103862.
8. Cahn, J. CHATBOT: Architecture, Design, & Development. Senior Thesis, University of Pennsylvania, Philadelphia, PA, USA, 2017.
9. Turing, A.M. Computing Machinery and Intelligence. *Mind* **1950**, *59*, 433–460. [CrossRef]
10. Wallace, R. *The Elements of AIML Style*; ALICE A.I. Foundation: Menlo Park, CA, USA, 2009.
11. Colby, K.M. Artificial Paranoia: A Computer Simulation of Paranoid Processes. *Artif. Intell.* **1975**, *2*, 1–25. [CrossRef]
12. Walker, M.A.; Litman, D.J.; Kamm, C.A.; Abella, A. PARADISE: A Framework for Evaluating Spoken Dialogue Agents. In Proceedings of the 35th Annual Meeting of the ACL, Madrid, Spain, 7–12 July 1997; pp. 271–280.
13. Cho, K.; van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. *arXiv* **2014**, arXiv:1406.1078.
14. Li, J.; Monroe, W.; Shi, T.; Jean, S.; Ritter, A.; Jurafsky, D. Deep Reinforcement Learning for Dialogue Generation. *arXiv* **2016**, arXiv:1606.01541. [CrossRef]

15. Shawar, B.A.; Atwell, E. Chatbots: Are they Really Useful? *LDV Forum* **2007**, *22*, 29–49.
16. Sutskever, I.; Vinyals, O.; Le, Q.V. Sequence to sequence learning with neural networks. *Adv. Neural Inf. Process. Syst.* **2014**, *27*, 3104–3112.
17. Vinyals, O.; Le, Q.V. A neural conversational model. *arXiv* **2015**, arXiv:1506.05869. [PubMed]
18. Serban, I.V.; Sankar, C.; Germain, M.; Zhang, S.; Lin, Z.; Subramanian, S.; Kim, T.; Pieper, M.; Chandar, S.; Ke, N.R.; et al. A Deep Reinforcement Learning Chatbot. *arXiv* **2017**, arXiv:1709.02349. [CrossRef]
19. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*; MIT Press: Cambridge, MA, USA, 1998.

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.