Towards a RAG-Based WhatsApp Chatbot for Animal Certification Platform Support

Gabriel Vieira Casanova[a], Pedro Bilar Montero[b], Alencar Machado[c] and Vinicius Maran[d]
Laboratory of Ubiquitous, Mobile and Applied Computing (LUMAC), Federal University of Santa Maria, Santa Maria, Brazil

Keywords: Question Answering Systems, Retrieval-Augmented Generation (RAG), WhatsApp, Virtual Assistant, Automated Support, Chatbot, PDSA-RS.

Abstract: Ensuring compliance with animal production certification requirements is often a complex and time-consuming task. This paper presents a domain-specific chatbot designed to assist users in requesting certifications within the PDSA-RS framework. By leveraging Retrieval-Augmented Generation (RAG) and large language models (LLMs), the proposed system retrieves relevant information from specialized documents and generates accurate, context-driven responses to user queries. The chatbot's performance was evaluated on two Brazilian certification platforms, demonstrating its potential to streamline certification requests, reduce errors, and enhance user experience.

1 INTRODUCTION

Chatbots have become an integral part of modern communication, assisting users in tasks such as customer support, information retrieval, and service automation. However, traditional chatbots often face significant limitations, including a lack of flexibility, inability to handle complex queries, and reliance on predefined scripts or rule-based logic. These shortcomings make it challenging for them to provide accurate, context-aware responses in dynamic scenarios (Kucherbaev et al., 2017).

The advent of Large Language Models (LLMs) has addressed many of these challenges by enabling chatbots to understand and generate human-like responses with remarkable fluency and contextual awareness. Yet, despite their capabilities, LLMs have their own limitations. They can struggle with retrieving accurate, up-to-date information, as their training data is static and finite, and may generate plausible but incorrect answers when faced with queries beyond their knowledge (Zhao et al., 2023).

To overcome these limitations, Retrieval-Augmented Generation (RAG) offers a transformative solution. By integrating a retrieval mechanism into LLMs, RAG enables chatbots to access real-time, relevant information from external knowledge sources or documents. This synergy between retrieval and generation allows chatbots to move beyond pre-trained data, delivering more accurate responses, context-aware, and dependable (Zhao et al., 2024).

The Plataforma de Defesa Sanitária Animal do Rio Grande do Sul (PDSA-RS) is a control system designed to support animal health management by organizing and integrating all stages of certification processes for poultry and swine farming in Rio Grande do Sul, Brazil. While the platform is a critical tool for ensuring operational efficiency and compliance, users often encounter frequent questions and challenges when navigating its functionalities. These recurring user questions highlight the need for a support system capable of providing accurate and immediate assistance regarding the platform's use (Descovi et al., 2021). WhatsApp, as one of the most widely used messaging platforms globally, offers an ideal medium for deploying such advanced chatbots. By incorporating a RAG-based chatbot into WhatsApp, users of the PDSA-RS platform can access intelligent, dynamic assistance tailored to their needs. The chatbot is specifically designed to address common user questions about the platform, providing accurate and context-aware responses to streamline user experience and reduce support overhead (Times, 2021).

[a] https://orcid.org/0009-0009-5420-7334
[b] https://orcid.org/0009-0002-9224-7694
[c] https://orcid.org/0000-0003-2462-7353
[d] https://orcid.org/0000-0003-1916-8893

Casanova, G. V., Montero, P. B., Machado, A. and Maran, V.
Towards a RAG-Based WhatsApp Chatbot for Animal Certification Platform Support.
DOI: 10.5220/0013473500003929
In *Proceedings of the 27th International Conference on Enterprise Information Systems (ICEIS 2025) - Volume 1*, pages 1077-1084
ISBN: 978-989-758-749-8; ISSN: 2184-4992

This paper presents the development of a RAG-based Chatbot for WhatsApp, designed to support users of the PDSA-RS platform. Leveraging the combined strengths of LLMs and retrieval systems, the chatbot aims to deliver accurate, contextually aware interactions, enhancing user engagement and improving understanding of the platform's functionalities (Gao et al., 2018).

The paper is structured as follows: Section 2 provides an overview of the key background concepts relevant to the research and the proposed approach. Section 3 introduces the methodology for developing the Chatbot for the Rio Grande do Sul Animal Health Defense Platform (PDSA-RS), specifically designed to provide support through WhatsApp. The simulation of the proposed solution is detailed in Section 4. Finally, Section 5 presents the conclusions of this work.

2 BACKGROUND

2.1 Chatbots

Chatbots are software applications engineered to mimic human conversation, predominantly through text-based interfaces. Their development has seen substantial advancements, transitioning from basic rule-based systems to sophisticated AI-powered conversational agents (Pantano and Pizzi, 2023).

The inception of chatbots dates back to the mid-20th century. A notable milestone was achieved in 1966 with the creation of ELIZA by Joseph Weizenbaum. ELIZA utilized pattern matching and substitution techniques to simulate dialogue, setting a foundational precedent for subsequent advancements in natural language processing (NLP) and conversational AI (Weizenbaum, 1966).

The 2010s witnessed a pivotal transition towards AI-driven chatbots. The emergence of machine learning and NLP technologies significantly enhanced chatbots' ability to comprehend and generate human-like responses. Platforms such as IBM Watson, Microsoft Bot Framework, and Google's Dialogflow provided developers with robust tools to create more intelligent and versatile chatbots (Wolf et al., 2019).

The 2020s have brought further innovations in conversational AI, with chatbots becoming increasingly context-aware and capable of managing intricate queries. Large Language Models (LLMs) have been instrumental in this progress, enabling chatbots to understand and produce text in a manner akin to human communication. Companies are progressively integrating chatbots into diverse applications, ranging from customer service to personal assistants and educational tools (Radford et al., 2019).

The evolution of chatbots, from simple rule-based systems to advanced AI-driven softwares, has been marked by continuous innovation. Driven by advancements in NLP and AI, chatbots have become integral to modern technology. As research progresses, chatbots are set to further transform user experiences across various domains (Adamopoulou and Moussiades, 2020).

2.2 WhatsApp

WhatsApp is a popular messaging app that allows users to send text messages, make voice and video calls, and share media files. It was founded in 2009 by Brian Acton and Jan Koum and was acquired by Facebook in 2014. WhatsApp has become one of the most widely used communication tools globally, with over 2 billion users.

WhatsApp offers a variety of features that make it a versatile communication tool, including text messaging, voice and video calls, media sharing, status updates, and end-to-end encryption. Its widespread adoption and continuous updates make it a reliable choice for both personal and professional communication (Times, 2021).

2.3 Large Language Models

Large Language Models (LLMs) represent a significant advancement in the field of artificial intelligence, particularly in natural language processing. These models have evolved rapidly, driven by innovations in deep learning and the availability of vast amounts of text data (Chang et al., 2024).

The 2010s saw the introduction of transformative models like Word2Vec and GloVe, which focused on word embeddings. The real breakthrough came with the introduction of the Transformer architecture by Vaswani et al. in their 2017 paper "Attention is All You Need." This architecture, based on self-attention mechanisms, allowed for more efficient processing of sequential data, paving the way for larger and more complex language models (Singh, 2024).

In 2018, Google introduced BERT (Bidirectional Encoder Representations from Transformers), which revolutionized the field by demonstrating state-of-the-art performance on a wide range of natural language processing tasks. The early 2020s witnessed a proliferation of LLMs, with models like RoBERTa, T5 (Text-to-Text Transfer Transformer), and others building on BERT's success. Notably, the introduction of models like ChatGPT by OpenAI showcased

the practical applications of LLMs in conversational AI (DATAVERSITY, 2024).

2.4 Retrieval-Augmented Generation

Retrieval-Augmented Generation (RAG) is a technique that combines information retrieval (IR) with Large Language Models (LLMs). This method enhances the capabilities of language models by retrieving relevant documents or pieces of information from an external corpus and using that retrieved information to generate more accurate and contextually relevant responses (Gupta et al., 2024).

Traditional text generation models rely heavily on the data they have been trained on and their internal parameters to generate outputs. However, these models can sometimes lack the ability to access specific, up-to-date, or domain-specific knowledge. RAG overcomes these limitations by incorporating an external information retrieval step into the model pipeline, which consists of two main stages:

- **Retrieval Phase.** The first step involves retrieving relevant documents or pieces of text from a large external corpus, typically using an information retrieval system.
- **Generation Phase.** After retrieving the relevant information, a generative model is employed to process the retrieved documents along with the original input, all in a single prompt, and generate a final response.

The usage of Retrieval-Augmented Generation has a set of advantages:

- **Access to External Knowledge.** RAG leverages external sources like databases, files, or websites to enhance responses.
- **Up-to-date Information.** The retrieval step can provide the latest information, keeping RAG models current, especially for rapidly changing topics.
- **Domain-Specific Information** RAG can be tailored to specific domains, enabling expert-level generation for niche topics.
- **Reduced Model Size.** By relying on external retrieval, RAG models avoid storing all knowledge internally, resulting in a more efficient model.

In the context of enterprise information systems, RAG can be applied to tasks such as automated query answering, document summarization, and knowledge management. For example, RAG could be used to provide real-time, evidence-backed answers to employee queries, summarize complex legal or technical documents, or assist in customer service by retrieving and generating responses from enterprise knowledge bases (Li et al., 2022).

2.5 PDSA-RS

The *Plataforma de Defesa Sanitária Animal do Rio Grande do Sul* (PDSA-RS) is a web platform that offers tools to official veterinarians in animal sanitary control in Rio Grande do Sul state (Brazil). Using advanced technologies, the platform provides real-time data to support decision-making, with features such as epidemiological analysis, disease spread modeling, and the evaluation of control measures. The platform is continuously evolving to address emerging challenges in animal health (Perlin et al., 2023).

PDSA-RS can be accessed via a web platform and a mobile application. The web platform allows for detailed data analysis and visualization, ideal for professionals needing in-depth information on disease trends and control efforts. The mobile app provides convenient, on-the-go access, enabling users to report incidents, view outbreak details, and receive alerts. This dual-access design enhances communication and supports quick responses to disease threats, helping to strengthen animal health defenses in Rio Grande do Sul.

The platform includes a support center that is hosted in WhatsApp to help users use its features effectively. An automated chatbot is being developed to further streamline support, providing immediate help with common inquiries and guidance on using the platform. This feature aims to reduce waiting times and improve user experience by offering quicker access to necessary information and troubleshooting directly through WhatsApp.

3 METHODOLOGY

This section presents the methodology for developing the Chatbot for the Plataforma de Defesa Sanitária Animal do Rio Grande do Sul (PDSA-RS), specifically designed to provide support through WhatsApp. The focus is on establishing a robust architecture that integrates with the existing platform. The approach involves a modular design, ensuring scalability and flexibility in development. The architecture (Figure 1) includes distinct layers responsible for user interaction, data processing, and system management. The development process has followed an agile methodology, allowing for continuous improvement and adaptation based on feedback and testing throughout the implementation stages.

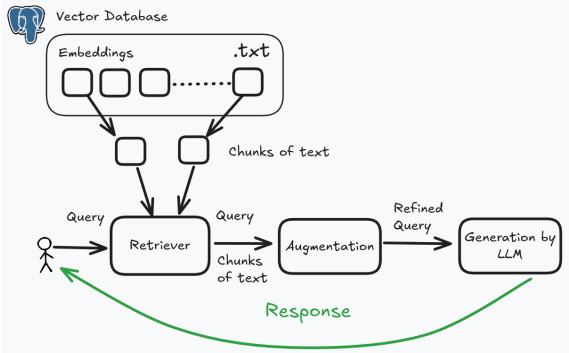

Figure 1: The proposed RAG Architecture.

3.1 Connection Module

This module is responsible for the direct integration with WhatsApp, ensuring the reception and sending of messages. When the server starts, a QR code is displayed in the terminal to pair the connection with a device. After that, the server listens for messages from phone numbers, with no restriction to contacts. All incoming messages are forwarded to the synchronization module for further handling, while all outgoing messages are processed via the processing module.

The technologies employed in the system module are summarized in Table 1. This table provides an overview of the core technologies used for message handling, system communication, and WhatsApp integration, essential to the platform's functionality.

Table 1: Employed Technologies.

Technology	Description
TypeScript	A JavaScript superset that adds static types.
AMQPLib	A library for interacting with RabbitMQ in Node.js.
Node.js	A runtime for server-side development.
NPM	A package manager for dependencies.
WPPConnect	A library for integrating WhatsApp with Node.js.

3.2 Synchronization Module

This module addresses the need to reassemble fragmented messages, a common behavior among WhatsApp users. By ensuring that long messages split across multiple parts are properly synchronized, it maintains message integrity before forwarding to the processing module.

When a message fragment is received, the module checks Redis for an existing entry. If found, it appends the fragment; if not, it creates a new entry. The fragments are queued for processing after a 40-second timer. If a new fragment arrives during this time, the timer resets to 40 seconds. After 40 seconds timer times out, we assume the user has finished his question and the complete message is passed to the next module.

The technologies employed in this system's module are summarized in Table 2. This table provides an overview of the core technologies used for message synchronization.

Table 2: Employed Technologies.

Technology	Description
Java	A popular programming language used for server-side applications.
Spring	A framework for building Java-based server-side applications.
Redis	An open-source in-memory data structure store used for caching

3.3 Processing Module

The Processing Module applies the Retrieval-Augmented Generation (RAG) technique to retrieve and generate a response based on the question, returning it to WhatsApp. Its operation involves receiving the defragmented message, i.e., the complete user question, and performing a similarity search in a vector database. This query returns similar questions with their respective answers, enhancing the context so that the LLM generates a human-like response that aligns with the user's intent. When the response is ready, it is sent to a queue, where the Connection Module listens and forwards it to the respective user on WhatsApp. The sequence of steps that defines the operation of the Processing Module is outlined as follows:

3.3.1 Document Ingestion

This step involves preparing the knowledge base by processing a JSON file containing the most relevant questions and answers from users of PDSA-RS and converting them into vector representations suitable for retrieval. The sub-processes include:

- **Text Extraction.** Extracting specialized query collected relevant text data to a JSON file by consulting team's members a from user support at the help center of PDSA-RS.

- **Partitioning or Chunking Strategy.** Splitting the text content into chunks of size 7500 tokens with an overlap of 100 tokens to preserve contextual continuity. This Chuncking technique is sepcifically denominated as Sliding Window Chunking.
- **Embedding or Vectorization.** Transforming the text chunks into vector representations using the `nomic-embed-text` vectorization model. These embeddings are then stored in the PostgreSQL database integrated with PGVector for efficient retrieval.

3.3.2 Vector Storage

The vector representations generated during the ingestion process are stored in a PGVector-enabled PostgreSQL database, optimized for similarity search.

3.3.3 Retrieval

A similarity search is performed on the vector database using the vectorized representation of the user's query to find the most relevant chunks of text that match the user's question.

3.3.4 Response Generation

With temperature set to zero ensuring precise and deterministic outputs, the retrieved information is used to enhance the context, enabling a language model, `Ollama` with `LLaMA 3.2 7B`, to generate a coherent and human-like response aligned with the user's intent. This configuration minimizes randomness, ensuring alignment with the setup.

3.3.5 Employed Technologies

The technologies employed in this system's module are summarized in Table 3.

Table 3: Employed Technologies.

Component	Description
nomic-embed-text	Large context length text encoder used for vectorizing text chunks.
LLaMA 3.2b	Large language model used for generating human-like responses.
Pip	Package used to manage dependencies.
Python	Programming language.

3.4 Docker Infrastructure

The infrastructure supporting the proposed Chatbot is built on Docker to ensure modularity, scalability, and simplified deployment. The tech-stack is summarized in Table 4.

Table 4: Docker Infrastructure.

Component	Description
RabbitMQ	Message broker for asynchronous communication with a monitoring interface.
Redis	In-memory data store for message fragments, with Redis Insight for monitoring.
PostgreSQL	A relational database for storing structured data.
PgVector	Extension for PostgreSQL enabling vector similarity search.
PgAdmin	Interface for managing PostgreSQL databases.
Ollama Server	A server to get up and running with large language models.
Open WebUI	Web interface for system configuration and monitoring of Ollama Server.

3.5 System Design

The system design of the proposed chatbot provides a high-level view of its architecture, outlining the main modules and their interconnections. It illustrates how these modules collaborate to achieve the chatbot's functionality. The design highlights the flow of data, interactions between modules, and the integration points necessary for the system's operation. This image serves as a blueprint for understanding the system's overall structure and its cohesive operation (see Figure 2).

4 SIMULATION

4.1 Objectives

The goal of the simulation is to test the behavior of the proposed Chatbot in a controlled WhatsApp environment. The simulation aims to emulate a user-chatbot interaction by leveraging two smartphones, where one device acts as the user and the other as the interface for the chatbot connected to the server.

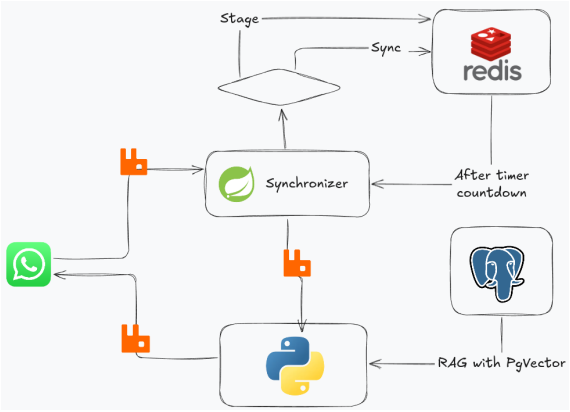

Figure 2: The Complete Chatbot System Design.

4.2 Requirementes

The simulation setup involves the following components and roles:

- **Smartphone 1 (User).** Sends messages (questions or prompts) to the chatbot, emulating a typical PDSA-RS user interaction.
- **Smartphone 2 (Chatbot).** Acts as the interface for the chatbot, paired with a server that processes incoming messages and generates appropriate responses.
- **Server.** Runs the proposed chatbot application, including components for receiving, processing, and responding to messages. A computer is required to run the server, which is responsible for handling all the message processing tasks.
- **Extracted Conversations.** From early 2023 to late 2024, five conversations with PDSA users were extracted to ingest in the vector database recurring questions. Each file contained over 500 lines, highlighting common challenges and needs. This data offers valuable insights to improve PDSA's support and user experience.

4.3 Configuration

The servers for this simulation are executed on a Lenovo 3i Laptop with the following specifications:

- **Processor.** Intel Core i5 10th Gen
- **RAM.** 8 GB DDR4 3200MHz
- **Storage.** 256GB SSD
- **Operating System.** Arch Linux
- **GPU.** NVIDIA GeForce GTX 1650 with 4 GB of dedicated memory

4.4 Conditions

To ensure controlled and repeatable testing behavior, the following conditions are maintained:

- Both smartphones must be connected to a stable network.
- The server is pre-configured with the chatbot application and necessary dependencies.
- A predefined set of test messages (e.g., FAQ-style questions) is used for consistency in testing.

4.5 Experiment

The experiment scenario is divided as follows:

- **Message Transmission.** Smartphone 1 sends a text message to Smartphone 2 via WhatsApp. The message mimics a typical PDSA-RS user inquiry, such as "O que é a PDSA-RS?" (What is PDSA-RS?) or "Como faço para realizar uma nova solicitação de certificado para uma granja de suíno?" (How do I submit a new certificate request for a swine farm?).
- **Processing.** Smartphone 2, paired with the server, receives the message. The server processes the text using the chatbot's architecture, which includes natural language understanding, response generation, and message formatting.
- **Response.** The connection server awaits the message from the Processing server and sends the generated response back to Smartphone 1 on WhatsApp, completing the interaction.

4.6 Results

The expected outcomes of the simulation include:

- Smartphone 2 successfully receives and forwards messages to the server.
- The server correctly processes the input and generates coherent, contextually relevant responses using the RAG technique.
- Smartphone 1 receives accurate responses in real-time, effectively mimicking a natural chatbot-user interaction.

The final and expected output of the real simulation is displayed in Table 5:

5 CONCLUSION

This paper presented the need to automate customer service at PDSA-RS's call center and proposed the

Table 5: The Simulated PDSA user-chatbot interactions (in Portuguese).

Smartphone 1 (User Input)	**Smartphone 2 (Chatbot Response)**
Como posso cadastrar um novo RT na PDSA?	*Para o cadastro como RT, você pode optar por ser independente ou representante de uma empresa. Se você é um RT independente, é necessário solicitar o cadastro diretamente na plataforma PESA pelo e-mail registrogranja@agricultura.rs.gov. Se você está vinculado a uma empresa, o RT representante da empresa tem a possibilidade de cadastrar novos usuários diretamente pelo módulo de gestão. Após o cadastro, o novo usuário receberá automaticamente um e-mail com os detalhes de acesso, incluindo login e senha.*
E Preciso fazer uma nova solicitação de certificado para uma granja de suíno, como faço?	*É necessário que você esteja como RT da granja em questão. Após, é necessário a realização de uma coleta, a inserção de no mínimo dois relatórios trimestrais e o preenchimento do formulário de tuberculina e leptospirose. Tendo todos os dados necessários, basta acessar o item "Solicitar Certificado" no menu lateral da plataforma.*

implementation of a chatbot capable of providing contextualized responses to PDSA, using RAG and LLM techniques. Many technologies were employed in each module of the application to compose the totalitty of the RAG based WhatsApp Chatbot proposal.

The main contributions include the automation of customer service, improving efficiency and availability of the service, and the application of multiple technologies to solve real-world problems.

To further enhance the Chatbot and maximize its impact, it's recommend expanding its functionalities, integrating with other platforms, and conducting live user testing in the real-world scenario of PDSA-RS to evaluate the long-term effects of the Chatbot implementation.

- **Monitoring Dashboard.** Develop a web interface to monitor questions and answers in real time, enabling performance analysis and the identification of areas for improvement.
- **Conversation History Management.** Implement a module to manage and easily access interaction history, assisting in analysis and future chatbot training.
- **Scalable Solutions.** Replace PGVector with more robust tools, such as **Elasticsearch**, for vector queries.
- **Long-Term Impact Evaluation.** Conduct studies to assess the effects of the chatbot on user experience and organizational efficiency over time.
- **Multimodal Agents.** Utilize LLMs that process not only text but also images, documents, and even audio, expanding the chatbot's capabilities.
- **Real Environment Testing.** It is necessary to test the chatbot in a real test environment, with real users from PDSA-RS, as the paper only covers a simulation.
- **Evaluation of Response Effectiveness.** To ensure continuous improvement, it is essential to apply metrics that assess the relevance, precision, and overall quality of the chatbot's responses.

ACKNOWLEDGEMENTS

This research is supported by FUNDESA (project 'Combining Process Mapping and Improvement with BPM and the Application of Data Analytics in the Context of Animal Health Defense and Inspection of Animal Origin Products in the State of RS - UFSM/060496) and FAPERGS, grant n. 24/2551-0001401-2. The research by Vinícius Maran is partially supported by CNPq, grant 306356/2020-1 - DT2.

REFERENCES

Adamopoulou, E. and Moussiades, L. (2020). Chatbots: History, technology, and applications. *Machine Learning with Applications*, 2:100006.

Chang, Y., Wang, X., Wang, J., Wu, Y., Yang, L., Zhu, K., Chen, H., Yi, X., Wang, C., Wang, Y., et al. (2024). A survey on evaluation of large language models.

DATAVERSITY (2024). A brief history of large language models - dataversity.

Descovi, G., Maran, V., Ebling, D., and Machado, A. (2021). Towards a blockchain architecture for animal sanitary control. In *ICEIS (1)*, pages 305–312.

Gao, J., Galley, M., and Li, L. (2018). Neural approaches to conversational ai: Question answering,

task-oriented dialogues and social chatbots. *Foundations and Trends in Information Retrieval*.

Gupta, S., Ranjan, R., and Singh, S. N. (2024). A comprehensive survey of retrieval-augmented generation (rag): Evolution, current landscape and future directions.

Kucherbaev, P., Psyllidis, A., and Bozzon, A. (2017). Chatbots as conversational recommender systems in urban contexts. *arXiv preprint arXiv:1706.10076*.

Li, H., Su, Y., Cai, D., Wang, Y., and Liu, L. (2022). A survey on retrieval-augmented text generation. *arXiv preprint arXiv:2202.01110*.

Pantano, E. and Pizzi, C. (2023). Ai-based chatbots in conversational commerce and their effects on product and price perceptions. *Electronic Markets*.

Perlin, R., Ebling, D., Maran, V., Descovi, G., and Machado, A. (2023). An approach to follow microservices principles in frontend. In *2023 IEEE 17th International Conference on Application of Information and Communication Technologies (AICT)*, pages 1–6. IEEE.

Radford, A. et al. (2019). Language models are unsupervised multitask learners. Technical report, OpenAI GPT-2.

Singh, A. (2024). The rise of large language models(llms). Accessed: 2025-01-19.

Times, T. N. Y. (2021). How chatbots are changing customer service. Accessed: 2023-10-01.

Weizenbaum, J. (1966). Eliza—a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9(1):36–45.

Wolf, T. et al. (2019). Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations (EMNLP 2020)*.

Zhao, P., Zhang, H., Yu, Q., Wang, Z., Geng, Y., Fu, F., Yang, L., Zhang, W., Jiang, J., and Cui, B. (2024). Retrieval-augmented generation for ai-generated content: A survey. *arXiv preprint arXiv:2402.19473*.

Zhao, W. X., Zhou, K., Li, J. Y., Tang, T. Y., Wang, X. L., Hou, Y. P., Min, Y. Q., Zhang, B. C., Zhang, J. J., Dong, Z. C., Du, Y. F., Yang, C., Chen, Y. S., Chen, Z. P., Jiang, J. H., Ren, R. Y., Li, Y. F., Tang, X. Y., Liu, Z. K., Liu, P. Y., Nie, J. Y., and Wen, J. R. (2023). A survey of large language models. *arXiv preprint arXiv:2303.18223*.