A Chatbot Generator for Improved Digital Governance

Christos Bouras[1(✉)], Damianos Diasakos[1], Chrysostomos Katsigiannis[1], Vasileios Kokkinos[1], Apostolos Gkamas[2], Nikos Karacapilidis[3], Yannis Charalabidis[4], Zoi Lachana[4], Charalampos Alexopoulos[4], Theodoros Papadopoulos[4], Georgios Karamanolis[5], and Michail Psalidas[5]

[1] Computer Engineering and Informatics Department, University of Patras, 26504 Patras, Greece
{bouras,kokkinos}@upatras.gr, {up1084632,up1072490}@upnet.gr
[2] University Ecclesiastical Academy of Athens, 14561 Athens, Greece
gkamas@aeavellas.gr
[3] Department of Mechanical Engineering, University of Patras, 26504 Patras, Greece
karacap@upatras.gr
[4] Department of Information and Communication Systems Engineering, University of Aegean, 83200 Samos, Greece
{yannisx,zoi,alexop,t.papadopoulos}@aegean.gr
[5] Crowd Policy, 18345 Athens, Greece
{george,michael}@crowdpolicy.com

Abstract. Chatbots, the pioneering conversational artificial intelligence (AI) agents, have experienced remarkable growth and integration in various domains. In modern societies, chatbots have emerged as transformative digital entities, revolutionizing the way humans interact with technology. These conversational AI agents have transcended their initial applications to become integral parts of various industries and daily life. One of the most prominent roles of chatbots is in customer service, where they offer round-the-clock assistance, swift issue resolution, and personalized interactions. By handling routine queries and tasks, chatbots free up human agents to focus on complex and specialized issues, thus optimizing overall efficiency and customer satisfaction. To this end, this paper aims to present and describe the architecture of a novel chatbot generator with improved functionality in terms of quality of communication with end users and level of provided services, with a specialized infrastructure understanding the Greek language. The chatbot generator was developed in the framework of a research project and will be pilot tested by two end-users, the National Bank of Greece (NBG) and the General Secretariat for Information Systems & Digital Governance (GSIS-DG).

Keywords: Artificial Intelligence · Natural Language Processing · Machine Learning · Virtual Assistants

M. Papadaki et al. (Eds.): EMCIS 2023, LNBIP 501, pp. 123–134, 2024.
https://doi.org/10.1007/978-3-031-56478-9_9

1 Introduction

Chatbots are sophisticated artificial intelligence-driven software applications designed to simulate human-like conversations with users. These conversational agents utilize cutting-edge technologies such as Natural Language Processing (NLP) and Machine Learning (ML) to understand and interpret user input, enabling seamless interactions through text or voice commands. At the heart of their functionality lies NLP, which allows chatbots to comprehend natural language, identify user intent, and generate appropriate responses [1]. Through ML, chatbots continuously learn from user interactions, improving their accuracy and efficiency over time.

Generally speaking, chatbots offer a wide range of benefits, including efficient and instantaneous responses to queries, round-the-clock availability, and the ability to handle large volumes of interactions concurrently. By automating routine tasks and providing instant assistance, they optimize processes, save time, and enhance user experiences. While they excel in various applications, the development of chatbots is not without challenges. Issues like context comprehension, sentiment analysis, and avoiding biases in responses remain areas of active research to ensure more meaningful and contextually relevant interactions.

Except from NLP and ML, chatbots leverage an additional array of cutting-edge technologies and functions to offer intelligent and dynamic conversational experiences, including:

- Intent Recognition: By employing sophisticated ML models, chatbots can accurately discern user intent from their queries. This allows chatbots to understand user requests, directing them to the most appropriate responses or actions.
- Sentiment Analysis: To enhance user experience, chatbots can analyze user sentiment and emotions in the input text. This analysis helps chatbots respond with empathy or appropriately address negative feedback, leading to more meaningful interactions.
- Dialog Management: Chatbots employ dialog management systems to maintain coherent and context-aware conversations with users. They can retain information from previous interactions, ensuring smoother and more natural exchanges.
- Personalization: Through ML-driven algorithms, chatbots can tailor responses and recommendations based on user preferences, history, and behavior. Personalization enhances user engagement and satisfaction, creating a more seamless experience.
- Knowledge Base Integration: Chatbots can integrate with knowledge bases and databases, providing access to vast amounts of information in real-time. This integration allows chatbots to deliver accurate and up-to-date responses to user queries.
- Context Understanding: Improved context comprehension empowers chatbots to maintain more coherent and natural conversations. They can remember past interactions, follow-up on ongoing tasks, and adapt responses based on the conversation's context.
- Task Automation: Chatbots excel at automating routine tasks, such as appointment scheduling, order tracking, or information retrieval. By automating these processes, chatbots save time for users and optimize operational efficiency.

As technology continues to evolve, chatbots will likely incorporate more advanced functionalities, enabling them to further enhance user experiences and expand their applications across various industries. With responsible development and ethical considerations, chatbots have the potential to reshape how we interact with technology and provide valuable assistance in our daily lives.

Based on the above, the goal of this paper is to present a chatbot architecture that is targeted at efficiency-oriented digital governance. The proposed architecture takes into account the particular needs and difficulties that organizations encounter when implementing chatbot technology. In the heart of the proposed architecture lies the chatbot generator that enables system administrators to automate the process of creating chatbots without requiring extensive coding knowledge or technical expertise. The paper is part of the PYTHIA project which is co-financed by the European Union and Greek national funds through the Operational Program "Competitiveness, Entrepreneurship and Innovation", under the call "RESEARCH - CREATE - INNOVATE (2nd Cycle)".

The PYTHIA project invests in the technology of Chatbots, AI agents that support natural language conversations and exploit next-generation internet technologies. The primary objective of the project is to significantly improve the functionality of chatbots, in terms of quality and service in communication for end users, by developing a specialized infrastructure for understanding the Greek language. The development of this infrastructure is based on the use of NLP technologies, combined with ML techniques and the integration of Argumentation, Logic, and Structured Dialogue models [2]. The subsystems resulting from this research implement a complete system/platform that supports a new service model called Bots-as-a-Service.

The remainder of the paper is structured as follows: In Sect. 2, we introduce the project requirements that guided the implementation of the proposed architecture. Section 3 presents in detail the proposed chatbot generator architecture. Finally, Sect. 4 outlines concluding remarks and sketches future work directions.

2 Project Requirements

For the needs of this paper, we formulated specific questionnaires, in order to derive the necessary information about the functionalities that the cooperating institutions needed, i.e., the National Bank of Greece (NBG) and the General Secretariat for Information Systems & Digital Governance (GSIS-DG). Based on the partners' response, Table 1 presents the main requirements for each partner, which were complemented by specific needs, such as the requirements for security and privacy.

Table 1. Main requirements of each partner.

Operator	Requirements
NBG	Utilization of Chatbot in the form of Q&A for the circular system "Athena" -NBG Intranet Portal, for use by the employees of the organization
GSIS-DG	Use of the Chatbot to serve citizens on the different platforms/services of the institution

2.1 Functional Requirements

Chatbots have automation options and features that can significantly speed up users' processes and direct them appropriately to the information they need. However, not all chatbot software solutions are equally effective. There are specific requirements that have a substantial impact on chatbot performance while meeting customer needs [3, 4]. A list of the most important requirements of chatbot software is presented below:

- Complex Dialogues: To effectively engage in conversations, chatbot software should possess NLP capabilities that allow it to analyze conversational context. Essential functions of this chatbot include discerning question intent, delivering precise answers, and offering suggestions to confirm or resolve the matter at hand. Proactively seeking information and asking clarifying questions, the best chatbots possess advanced conversational capabilities, ensuring a non-linear conversation.
- Flexible Data Interfaces: Flexible data interfaces in chatbots refer to the capability of chatbot systems to interact with and process data from various sources and formats in a versatile manner. These interfaces allow chatbots to integrate with diverse data repositories and external systems, making them more powerful and adaptable in handling user queries and providing relevant information.
- Multi-channel Capability: Multi-channel capabilities in chatbots refer to their ability to interact and communicate with users across various communication channels and platforms. Instead of being limited to a single interface, chatbots with multi-channel capabilities can engage with users through multiple touchpoints, providing a seamless and consistent user experience across different channels. Multi-channel capabilities could be considered as a non-functional requirement but it is worth mentioning.
- Fast Onboarding: Fast onboarding refers to the process of rapidly integrating users into a system, service, or application with minimal effort and time required on the user's part. In the context of chatbots or digital applications, fast onboarding is crucial for ensuring a smooth and efficient user experience from the very beginning.
- Easy Handling: Chatbots' or digital applications' easy handling is crucial for providing a positive user experience and encouraging user engagement.
- Ongoing Optimization: It involves regularly monitoring the chatbot's interactions, analyzing data, and making iterative adjustments to ensure that the chatbot remains effective, relevant, and aligned with user needs and preferences.
- Analytics & Reporting: In chatbots analytics and reporting play a crucial role in understanding user behavior, evaluating performance, and optimizing the user experience.

- Performance and Protection of Personal Data: Performance and protection of personal data are two essential aspects when it comes to the operation of chatbots and digital applications, particularly concerning user privacy and the quality of user experience.

Regarding the issue of performance in the context of chatbots, it refers to the effectiveness, efficiency, and reliability of the chatbot in delivering accurate and timely responses to user queries and tasks. A high-performing chatbot should be able to understand user intent, provide relevant information, and complete tasks efficiently, resulting in a smooth and satisfactory user experience. Continuous monitoring, optimization, and regular updates are necessary to maintain and enhance the performance of chatbots over time.

Another important requirement is the protection of personal data. As chatbots interact with users and gather information, there is a need to ensure the protection and privacy of users' personal data. Personal data includes any information that can identify an individual, such as names, contact details, or financial information. Chatbot developers must implement robust security measures and data encryption to safeguard this sensitive information from unauthorized access, breaches, or misuse. Compliance with relevant data protection regulations, such as the General Data Protection Regulation (GDPR) in the European Union, is crucial to ensure that personal data is handled ethically and responsibly.

2.2 Technical Requirements

Successful web chat integrations supported by chatbots depend on meeting important technical requirements. The main goal is to establish smooth communication with the chatbot's backend service and generate a unique user ID for personalized interaction while enabling seamless messaging between the user and the chatbot. Such technical requirements include:

- Communication with the Chatbot Backend Service: The main technical requirement to run a web chat (text window) is to be able to communicate with the chatbot's backend service. This communication is required to send messages from users to the chatbot (or vice versa) and to create a unique user ID for each user communicating with the chatbot.
- Communication for loading the required HTML: To integrate the chatbot into a website, it is essential to include a specific line of code within the <head> section of the site's HTML. By including the necessary code, the chatbot widget is created and seamlessly integrated into the website, enabling users to access and interact with the chatbot functionality.
- System Requirements: The system requirements for the application include an Ubuntu 20.04 (LTS) Server or a more recent version. The application is designed to be compatible with this operating system to ensure optimal performance and functionality.

3 Platform Architecture

This section describes the proposed chatbot generator architecture that follows the partner-specific, the functional and the technical requirements presented in Sect. 2. The main components of the architecture are also described. It is worth mentioning, that the conceptual design of the architecture was based on similar ideas put forth in the literature (e.g., [5–7]), with the necessary addition of elements that automate the chatbot development process.

3.1 The PYTHIA Platform Architecture

Figure 1 illustrates in detail the platform architecture proposed in the context of the PYTHIA project.

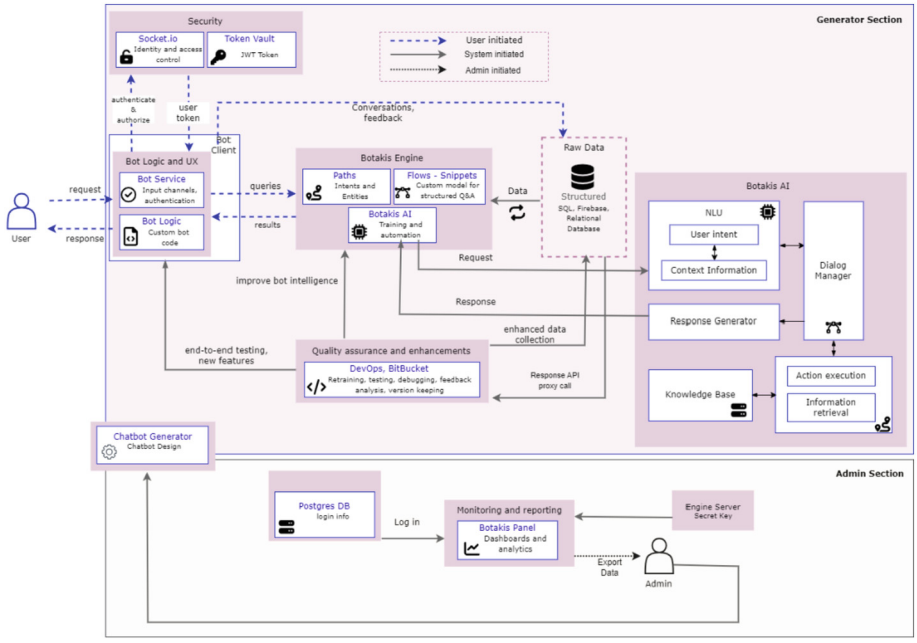

Fig. 1. The proposed architecture.

At the end-user side, native webchat can be added to the website with the use of a script in the HTML head section. The native script includes:

- The project's HTML (Hyper Text Language).
- The JavaScript that creates the build for the Webchat functions.
- The Vendor JavaScript file which includes the NPM Modules that have been used to build the project.
- The generated CSS (Cascading Styles Sheets) that contains all the styling of the Webchat.

- Additional CSS file which includes any styling changes that may occur.

The procedure to interact with the native webchat includes:

- User login to the service's website or the digital assistant's page.
- Through the script on the page, a call (GET) is made to the page to return the native webchat and it is displayed on the site.
- The script contains all the HTML of the project while adding all the generated files build to it.
- Upon the user choosing to initiate a chat via native webchat, a cookie is created and stored on the page which includes a JavaScript Object Notation (JSON) Web Token which is used to achieve user uniqueness. If previously created and not deleted, it is recognized by native webchat and returns the history of the user's conversation with the digital assistant.
- After the page call, native webchat contacts the digital assistant's engine (mentioned as Botakis engine in Fig. 1) to return the appropriate content.

User Message Flow

A series of steps are performed after the user interacts with the chatbot. These steps include:

- Authentication: Users start with authenticating their identity by making a post call to the designed host URL of the chatbot to verify that the user exists in the database and is eligible to access his/her chat history with the chatbot. In case the user is visiting the system for the first time, a new JSON web token will be created which will be stored in the user's browser cookies as well to identify the user the next time he/she enters the chatbot environment again.
- Digital assistant message: In case the user is contacting the digital assistant chatbot for the first time, a welcome message is sent by the engine. This message is not repeated the next time the same user enters the chatbot.
- User message: After authentication, the user will be able to send messages to the chatbot. The chatbot reads the message and internally forwards it to the recognition mechanism. This step captures intents (what the user wants to do) and entities (what the user is interested in). Chatbots understand user intent and then generate a response based on that outcome. For best results while using the chatbot and generated queries, the content is trained by content managers to identify the user's intent more accurately in subsequent messages.
- Response: At this point, the chatbot determines the best answer and sends it to the user. If the chatbot returns a fallback or suggestion message, it means it was unable to respond appropriately.
- Recording: When a user message is received or a response is sent, all chat actions should be logged anonymously in a log database along with performance metrics and common errors.
- Feedback: Gathering customer feedback and satisfaction scores is another important technique. The user is prompted to give the process a rating after the chatbot's final response. Feedback aids in the resolution of issues with natural language comprehension and enhances the precision of chatbot responses.

In Sects. 3.2–3.7 we will introduce and elaborate the main components of the proposed Chatbot architecture. These include the Chatbot Generator, Natural-Language Understanding, Dialogue Management, Information Retrieval System, Backend and Response Generation.

3.2 Chatbot Generator

The main novelty of the proposed architecture is the "chatbot generator". The chatbot generator is a tool that automates the process of creating chatbots without requiring extensive coding knowledge or technical expertise. It allows users to build custom chatbots for various purposes, such as customer service, lead generation, or informational support, using a user-friendly interface and predefined templates.

The chatbot generator offers a range of features and functionalities that users can customize to tailor the chatbot to their specific needs. These features include:

- Pre-built Templates: Chatbot generator provides pre-built templates for different industries or use cases, making it easier for users to get started quickly.
- Drag-and-Drop Interface: Users can design the conversation flow and structure of the chatbot by using a simple drag-and-drop interface, without writing complex code.
- Natural Language Processing (NLP): The chatbot generator includes NLP capabilities, enabling chatbots to understand and respond to user inputs more intelligently. A combination of statistical and rule-based NLP techniques is used to improve the conversational capabilities of chatbots. The chatbot builder understands the syntactic structure of the user's input using a rule-based grammar, allowing it to recognize patterns and extract meaningful elements from the user.
- Multi-Channel Integration: Chatbot generators allow the chatbot to be integrated with various communication channels, such as websites, messaging apps, or social media platforms.
- Analytics and Reporting: The chatbot generator offers built-in analytics and reporting features to track chatbot performance and user interactions.
- Data Collection and User Management: The Chatbot generator can collect user data and manage user profiles to provide personalized experiences.

3.3 Natural-Language Understanding

Natural Language Understanding (NLU) is a crucial component of a chatbot due to enabling it to comprehend and interpret user input effectively. By leveraging NLU technology, the system can accurately extract appropriate information from user queries and create a comprehensive representation of their underlying meaning [8]. NLU accomplishes this through three fundamental tasks: dialogue act classification, intent classification, and information extraction [9]. These processes collectively empower the system to better engage with users and provide more relevant and contextually appropriate responses. In detail:

- Classification of dialog operations involve identifying the purpose of user input, or more precisely, connecting user input with different types of dialog operations. This input can be categorized either as a question, remark, suggestion, or other types of

interactive acts. Knowing the interactivity being performed is important to better understand user requests and determine appropriate responses [10].

- Intent classification identifies the user's primary goals. Intent varies largely by industry. For example, industry inquiries such as food orders, hotel reservations, and weather forecasts. An agent's intent in the hotel booking industry may be to book, cancel, or modify a reservation, and an agent's intent in the grocery ordering industry may be similar, to submit, query, or modify an order.
- Extracting information by survey is the final step in NLU. The chatbot further extracts the necessary details and combines them with interaction actions and intents to fully understand the user's request.

To fully understand user input in the pilot version, a combination of statistical and neural methods was used. Statistical methods such as probabilistic modeling have been used for tasks such as conversational behavior classification, where the system determines the intent of a user's input by associating it with various types of conversational activities. Different dialogues such as requests for information, comments or suggestions. On the other hand, to classify user intent, neural methods, such as deep learning algorithms, have been used.

3.4 Dialogue Management

The Dialogue Management (DM) subsystem processes information from other subsystems, controls and updates the context of the conversation, and regulates chatbot behavior. Designing a robust DM strategy is challenging because it is difficult to predict which system behaviors will lead to high user satisfaction. Here, we discuss two common DM design problems: interaction strategy and confirmation strategy selection [11].

An interaction strategy determines who controls the conversation. Conversations can be user-driven, system-driven, or mixed-driven. In a user-driven interaction, the user takes the initiative, and the system merely executes the user's requests and instructions. In a system-driven dialogue, the system assumes control and merely complies with its demands. Both the user and the system can take over if there is mixed initiative in the interaction. [11].

As far as the confirmation strategies are concerned, the are two types of strategies: explicit confirmation and implicit confirmation. When using the explicit confirmation strategy, the system confirms understanding by asking the user another question. In the implicit acknowledgment strategy, the system includes some of the received information in the response [11].

For the pilot testing purposes, the mixed-driven interaction strategy and the explicit confirmation strategy have been used. However, we plan to extend the functionality by developing the remaining strategies and allowing admins to select their preferred strategy.

3.5 Information Retrieval System

The Information Retrieval System (IRS) has two main purposes. The first one is to achieve quality and effective results. The second one is about how quickly the requested

information is retrieved; it is about efficiency. In order to send a request to the system, the user uses an interface, for example a browser, connected to the IRS via the HTTP communication protocol. Specifically, user requests through the interface are processed using NLU mechanisms and converted into a format suitable for IRS.

However, the following problems are often observed: Some of the answers returned by the IRS are not very relevant to the query. For this reason, relevance feedback techniques were developed and used to improve the quality of results. Using this method gives the user the opportunity to select some of the IRS's suggested answers that are more relevant than others. Therefore, the system can redefine the answer based on user selection.

3.6 Backend

The chatbot gets the information it needs from the backend to perform the required task and forwards the message to the dialog management and response generation subsystems [12]. A rule-based chatbot requires a knowledge base (KB) to store manually created rules [13]. The chatbot retrieves previous conversations using a relational database (RDB). By considering previous information, the chatbot can communicate more consistently, accurately, and reliably [14]. KB development is done by humans and can be time consuming and difficult. To overcome this difficulty, we have automatically created a new KB from the partners' existing KB [15].

3.7 Response Generation

Once the appropriate information is obtained, the next step for the dialog system is to determine what the response will be and how to best present it. The Response Generation subsystem is responsible for generating responses in a format that the user can understand. The specific subsystem includes five processing steps: signal analysis, data interpretation, document design, micro-design, and implementation.

4 Conclusions

This paper presented in detail the chatbot generator architecture developed in the framework of the PYTHIA project. The overall architecture was based on the requirements and specifications defined by two of the project partners, i.e., the National Bank of Greece and the General Secretariat for Information Systems & Digital Governance. The main components of the architecture were presented, and their functionality was analyzed. Overall, the chatbot generator simplifies the process of creating and deploying chatbots, making it accessible to a broader audience and enabling businesses and individuals to implement chatbot solutions efficiently without significant development resources.

The requirements and specifications described in this paper will guide the full implementation of the overall system. Then, the chatbot generator will undergo a thorough pilot testing process with multiple end-users across diverse applications, and its overall performance and efficiency will be evaluated. As stressed in [16], the application of the foreseen platform must be evaluated through a set of Key Performance Indicators

(KPIs), aiming to measure its usefulness and ease of use. The focus of the evaluation would be to assess, for various types of contexts and associated questions (of varying complexity, ambiguity and uncertainty), to what extent it can provide useful and relevant information as well as support for conducting relevant transactions.

Acknowledgements. This research has been co-financed by the European Union and Greek national funds through the Operational Program "Competitiveness, Entrepreneurship and Innovation", under the call "RESEARCH - CREATE - INNOVATE (2nd Cycle)" (project id: T2EDK-01921).

References

1. Doshi, J.: Chatbot user interface for customer relationship management using NLP models. In: International Conference on Artificial Intelligence and Machine Vision (AIMV), pp. 1–4, Gandhinagar, India (2021)
2. Karacapilidis, N., Papadias, D.: A Computational approach for argumentative discourse in multi-agent decision making environments. AI Commun. J. **11**(1), 21–33 (1998)
3. Online Website. https://onlim.com/en/top-requirements-for-chatbot-software/
4. Lau, L., Yang-Turner, F., Karacapilidis, N.: Requirements for big data analytics supporting decision making: a sensemaking perspective. In: Karacapilidis, N. (eds.) Mastering Data-Intensive Collaboration and Decision Making. Studies in Big Data, LNCS, vol. 5, pp. 49–70. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-02612-1_3
5. Avishka, W., Kuhaneswaran, B., Gunasinghe, H.: A novel conceptual chatbot architecture for the Sinhala language – a case study on food ordering scenario. In: Proceedings of the 2022 2nd International Conference on Advanced Research in Computing (ICARC), pp. 254–259. Belihuloya, Sri Lanka (2022)
6. Maher, S.K., Bhable, S.G., Lahase, A.R., Nimbhore, S.S.: AI and deep learning-driven chatbots: a comprehensive analysis and application trends. In: Proceedings of 2022 6th International Conference on Intelligent Computing and Control Systems (ICICCS), pp. 994–998, Madurai, India (2022)
7. Van Thin, D., Quang, T.M., Toan, P.M., Thien, V.M., Hung, L.M., Quan, T.T.: A human-like interactive chatbot framework for Vietnamese banking domain. In: Proceedings of 2022 9th NAFOSTED Conference on Information and Computer Science (NICS), pp. 41–46, Vietnam (2022)
8. Reiter, E.: An architecture for data-to-text systems. In: Proceedings of the ENLG-2007 11th European Workshop on Natural Language Generation, pp. 97–104. Saarbrücken, Germany (2007)
9. Firdaus, M., Golchha, H., Ekbal, A., Bhattacharyya, P.: A deep multi-task model for dialogue act classification, intent detection and slot filling. Cogn. Comput. **13**, 626–645 (2020)
10. Tur, G., Deng, L.: Intent determination and spoken utterance classification. In: Tur, G., de Mori R. (eds.) Spoken Language Understanding: Systems for Extracting Semantic Information from Speech, pp. 93–118. Wiley, Chichester (2011)
11. McTear, M., Callejas, Z., Griol, D.: The Conversational Interface Talking to Smart Devices. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-32967-3
12. Adamopoulou, E., Moussiades, L.: Chatbots: history, technology, and applications. Mach. Learn. Appl. **2**, 100006 (2020)
13. Khanna A., Pandey B., Vashishta K., Kalia K., Bhale P., Das T.: A study of today's A.I. through Chatbot and rediscovery of machine intelligence. Int. J. u- e-Serv. Sci. Technol. **8**, 277–284 (2015)

14. Abdul-Kader, S.A., Woods, J.: Survey on Chatbot design techniques in speech conversation systems. Int. J. Adv. Comput. Sci. Appl. **6**, 72–80 (2015)
15. Arsovski, S., Osipyan, H., Oladele, M.I., Cheok, A.D.: Automatic knowledge extraction of any Chatbot from conversation. Expert Syst. Appl. **137**, 343–348 (2019)
16. Androutsopoulou, A., Karacapilidis, N., Loukis, E., Charalabidis, Y.: Transforming the communication between citizens and government through AI-guided chatbots. Gov. Inf. Q. **36**(2), 358–367 (2019)