2023 8th International Conference on Mathematics and Computers in Sciences and Industry (MCSI) | 979-8-3503-4165-2/23/$31.00 ©2023 IEEE | DOI: 10.1109/MCSI60294.2023.00021

On the Development of a Novel Chatbot Generator Architecture: Design and Assessment Issues

1st Christos Bouras
University of Patras, Computer Engineering & Informatics Department
Patras, Greece
bouras@upatras.gr

2nd Damianos Diasakos
University of Patras, Computer Engineering & Informatics Department
Patras, Greece
up1084632@upnet.gr

3rd Chrysostomos Katsigiannis
University of Patras, Computer Engineering & Informatics Department
Patras, Greece
up1072490@upnet.gr

4th Vasileios Kokkinos
University of Patras, Computer Engineering & Informatics Department
Patras, Greece
kokkinos@cti.gr

5th Apostolos Gkamas
University Ecclesiastical Academy of Athens
Athens, Greece
gkamas@aeavellas.gr

6th Nikos Karacapilidis
University of Patras, Department of Mechanical Engineering
Patras, Greece
karacap@upatras.gr

7th Yannis Charalabidis
University of Aegean, Department of Information & Communication Systems Engineering
Samos, Greece
yannisx@aegean.gr

8th Zoi Lachana
University of Aegean, Department of Information & Communication Systems Engineering,
Samos, Greece
zoi@aegean.gr

9th Charalampos Alexopoulos
University of Aegean, Department of Information & Communication Systems Engineering
Samos, Greece
alexop@aegean.gr

Theodoros Papadopoulos
University of Aegean
Department of Information & Communication Systems Engineering
Samos, Greece
t.papadopoulos@aegean.gr

Georgios Karamanolis
Crowd Policy
Athens, Greece
george@crowdpolicy.com

Michail Psalidas
Crowd Policy
Athens, Greece
michael@crowdpolicy.com

***Abstract*—Chatbots are witnessing remarkable growth in various fields. In modern society, chatbots have evolved into innovative digital entities that have revolutionized the way people interact with technology. These conversational AI agents have gone beyond their initial uses to become an integral part of various industries. One of the most important roles of chatbots is customer service, providing 24/7 support, quick problem resolution, and personalized interactions. Chatbots free up human agents by handling routine queries and tasks, allowing them to focus on complex and specific issues, and optimizing overall efficiency and customer satisfaction. To this end, this paper aims to present the architecture of a novel chatbot generator with improved functionality in terms of quality of communication with end users and level of provided services. In addition, the paper presents the results of two pilot chatbot assessments by users with experience in artificial intelligence and chatbot technology.**

Keywords—artificial intelligence, chatbot, machine learning, natural language processing

I. Introduction

Chatbots are advanced artificial intelligence-based software applications designed to simulate human-like conversations with users. They utilize cutting-edge technologies such as Natural Language Processing (NLP) and Machine Learning (ML). Their major functionality is based on NLP, which allows chatbots to comprehend natural language, recognize user intent, and generate appropriate responses.

Chatbots offer benefits such as quick answers, always-on availability, efficient handling of large numbers of interactions, optimized processes, and improved user experience. However, challenges such as understanding context, sentiment analysis, and avoiding response bias require continued research into more meaningful 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, Sentiment Analysis, Dialog Management, Personalization, Knowledge Base Integration, Context Understanding, and Task Automation.

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.

Currently, there are several chatbot types, which are categorized based on their operating methods and the way they interact with end-users. Current state-of-art includes: Menu-based chatbots, Language-based / Rule-based chatbots, Keyword recognition-based chatbots, Machine Learning chatbots, Voice chatbots, and Hybrid chatbots, [1]. The most advanced type is the Machine Learning chatbots, which utilize ML and Artificial Intelligence (AI) techniques, using the history of specific users to adapt and evolve/improve over time. Machine Learning chatbots are still in the research stage because the answers tend to be unpredictable, in relation to Rule-based chatbots which, even with fewer features, manage to be more reliable, [2]. In addition, the examination and assessment of meta-features for chatbot interaction is important to improve the performance and accuracy of chatbots, [3], [4].

In line with the above, the purpose of this article is to present a chatbot architecture that takes into account the

979-8-3503-4165-2/23/$31.00 ©2023 IEEE
DOI 10.1109/MCSI60294.2023.00021

unique needs and difficulties during chatbot technology implementation. At the heart of the proposed architecture lies the chatbot generator, which allows system administrators to automate the chatbot creation process without requiring extensive programming knowledge or technical expertise.

The work described in this paper was conducted in the context of the PYTHIA project, which is co-financed by the European Union and Greek national funds through the Operational Program "Competitiveness, Entrepreneurship, and Innovation". 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. 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 Section II, we introduce the project requirements that guided the implementation of the proposed architecture. Section III presents the proposed chatbot generator architecture. In Section IV, the performance assessment results of two pilot chatbots are described. Finally, Section V outlines concluding remarks and sketches future work directions.

II. Project Requirements

For the needs of this paper, we formulated specific questionnaires 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). Table I presents the main requirements for each partner, which were complemented by specific needs, such as the requirements for security and privacy.

TABLE I. The main requirements of each partner

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

A. Functional Requirements

Certain requirements greatly impact the performance of chatbots while meeting customer needs, [5]. Some of the most important requirements are:

- **Complex Dialogues:** For effective conversation integration, chatbot software requires NLP capabilities for contextual analysis. Key features include discerning question intent, delivering precise answers, and offering suggestions to confirm or resolve the matter at hand.
- **Flexible Data Interfaces:** A chatbot's flexible data interface allows the system to interact with and process data from a variety of sources and formats. These interfaces facilitate integration with various data repositories and external systems, improving the chatbot's ability to answer user queries.
- **Multi-channel Capability:** It refers to chatbots' ability to interact and communicate with users across various communication channels and platforms, rather than being limited to a single interface.
- **Fast Onboarding:** It 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, it is crucial for a smooth 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:** This 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:** They play a crucial role in understanding user behavior, evaluating performance, and optimizing the user experience.
- **Performance and Protection of Personal Data:** There are two essential aspects when it comes to the operation of chatbots and digital applications, particularly concerning user privacy and quality of experience.

B. Technical Requirements

Web chat with chatbot integration requires important technical requirements to enable smooth communication with backend services and personalized interactions with unique user IDs, ensuring seamless messaging. 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.

III. Platform Architecture

This section describes the proposed chatbot generator architecture that follows the partner-specific, functional as well and technical requirements presented in Section II. 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., [6], [7], [8]), with the necessary addition of elements that automate the chatbot development process.

A. The Pythia Platform Architecture

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

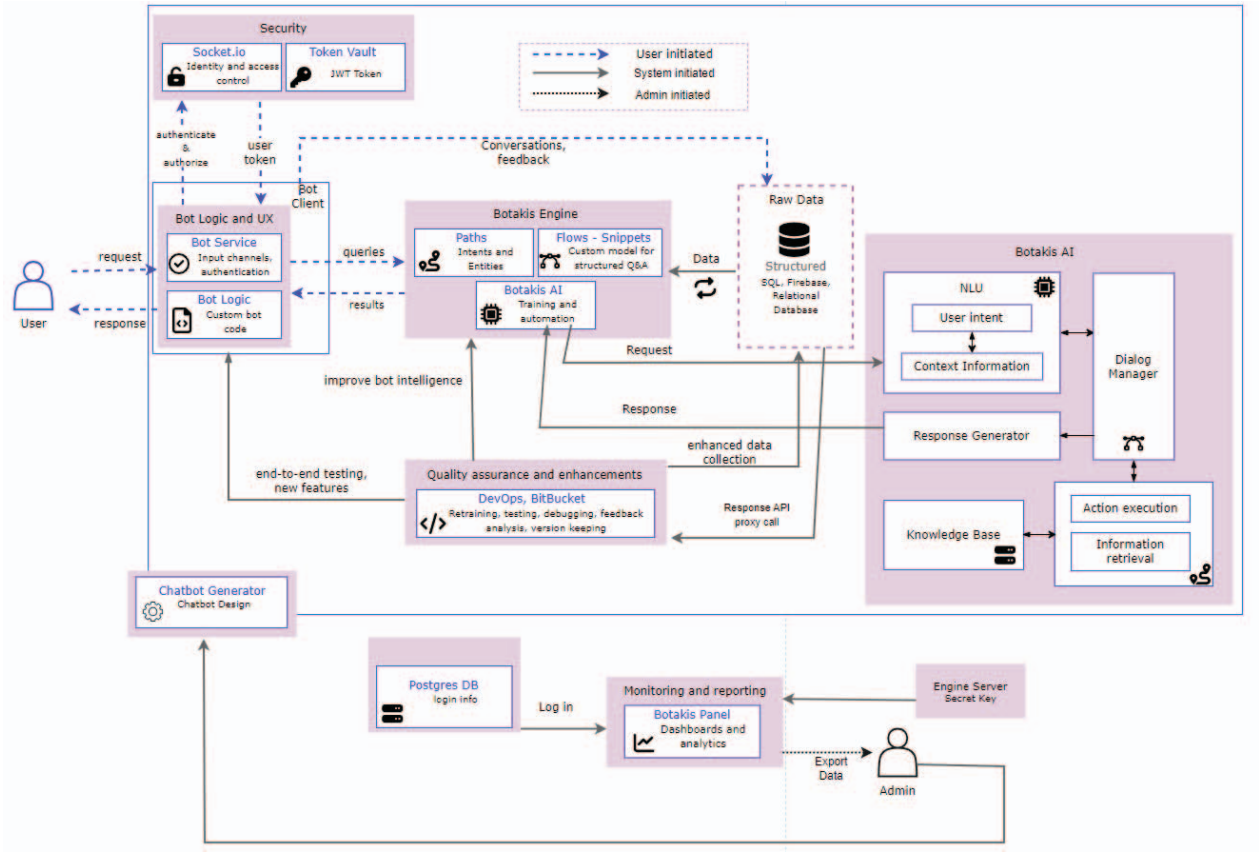

Fig. 1. The proposed architecture.

- The script contains all the HTML of the project while adding all the generated files built 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 the Botakis engine in Figure 1) to return the appropriate content.
- It is worth mentioning that the overall architecture was designed in a way that enables the future utilization of open Large Language Models, (LLMs), as well as the connection to commercial APIs, such as https://openai.com/blog/openai-api.

B. User Message Flow

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

- **Authentication:** Users verify their identity by making a post call to the host URL provided by the chatbot. This step verifies the existence of the user in the database and permissions to access chat history. For first-time visitors, a JSON web token is

generated and stored in the user's browser cookies so that it can be identified later within the chatbot environment.

- **Digital assistant message:** In case the user contacts 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, users can send messages to the chatbot. The chatbot passes these messages to its internal recognition mechanism for processing. This mechanism identifies user intent and entities and determines what the user wants and is interested in. The chatbot will react accordingly. To improve the accuracy of subsequent interactions, content managers train the system to better understand user intent based on the queries generated.
- **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 feedback is important for assessing customer satisfaction. After receiving the final response, the user is asked to rate the chatbot's performance. This feedback helps identify issues related to natural language comprehension and fine-tune chatbot responses for improved precision.

C. Chatbot Generator

The core innovation of this architecture is the Chatbot Generator, which optimizes chatbot development and offers development even with limited programming knowledge. The Chatbot Generator's intuitive interface, coupled with a variety of customizable features, empowers users to effortlessly create custom chatbots tailored to their unique requirements, eliminating the technical barriers that often hinder the development process. Such features include pre-built templates for specific industries that make the process much easier and a user-friendly drag-and-drop interface that allows users to seamlessly design the chatbot's conversation flow and structure, eliminating the need for complex programming. Additionally, the integration of NLP capabilities ensures intelligent responses to user inputs and improves user interactions. Multichannel integration extends the reach and maximizes the Chatbot's impact across platforms such as different websites, messaging apps, and social media platforms. In addition, Chatbot Generators offer analytical tools to monitor performance and user engagement, as well as robust data collection and user management features for personalized user experiences.

D. Natural Language Understanding

Natural language understanding (NLU) can be used in chatbots to effectively understand and interpret user input. NLU technology allows for extracting precise information from end-users' requests and comprehensively understanding the underlying intent, [9]. NLU achieves this through the following three tasks, which improve the interaction with end-users, [10]:

- **Conversation classification** identifies the intent behind user input and links it with different types of actions. Understanding the nature of the interaction is critical to understanding user queries and crafting appropriate responses, [11].
- **Intent classification** focuses on identifying user goals that vary by industry. For example, a hotel reservation may involve booking, canceling, or changing a reservation, whereas a food order may involve submitting, querying, or changing 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 aiming to fully understand the user's request.

E. Dialogue Management

The Dialogue Management (DM) subsystem processes data from other subsystems, manages conversation context, and regulates chatbot behavior. Developing a strong DM strategy is difficult due to the uncertainty about which system behavior will maximize user satisfaction. This discussion focuses on two common DM design challenges: interaction strategy and confirmation strategy, [12].

Interaction strategies define user-driven, system-driven, or mixed-driven conversation control. In user-controlled interactions, the user initiates instructions and the system follows those instructions. In system-controlled interactions, the system controls and makes requests. Mixed initiatives occur when both parties can lead the conversation.

When it comes to confirmation strategies, there are explicit and implicit methods. Explicit confirmation involves asking the user another question to confirm understanding, whereas implicit confirmation incorporates the information received into the answer, [12].

F. Information Retrieval System

Information Retrieval System (IRS) serves two main purposes: providing high-quality results and ensuring rapid information retrieval to increase efficiency. A user interacts with the system through an interface, typically a web browser, connected to the IRS via the HTTP communication protocol. User requests routed through this interface undergo processing using NLU mechanisms which then are converted to an IRS-compatible format.

However, several 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, the user is allowed to select some of the IRS's suggested answers that are more relevant.

G. Backend

The chatbot receives the necessary information from the backend to perform the required task and forwards the message to the dialog management and response generation subsystems, [13]. A rule-based chatbot requires a knowledge base (KB) to store manually created rules, [14]. The chatbot retrieves previous conversations using a relational database (RDB). By considering previous information, the chatbot can communicate more consistently, accurately, and reliably,

[15]. KB development is done by humans and can be time-consuming and difficult. To overcome this difficulty, a new KB was created from the partners' existing KB, [16].

H. Response Generation

After collecting relevant information, the dialog system's next task is to formulate an effective answer and present it optimally. The Response Generation subsystem takes on this responsibility and ensures that the answers are user-friendly. This subsystem includes five processing stages: signal analysis, data interpretation, document design, micro-design, and implementation.

IV. Chatbot Assessment

In the context of chatbot assessment, a workshop was held during the Samos Summit 2023 (Samos, Greece, 26-28 June 2023). This workshop brought together people with experience in artificial intelligence and chatbot technology. During the workshop, particular emphasis was given to the evaluation of two pilot chatbots designed according to the proposed architecture.

Specifically, a questionnaire was created and distributed to experts to serve as a research tool for data collection on various aspects of chatbot performance. The results showed that most of the 15 participants (78.6%) who had the choice to evaluate the chatbot developed either for NBG or GSIS-DG chose to use the latter. The questionnaire concerned the accuracy of user understanding and response (Figure 2), user-friendliness (Figure 3), and response time (Figure 4). The study uses a Likert rating scale of 1 to 5, with higher scores indicating better performance.

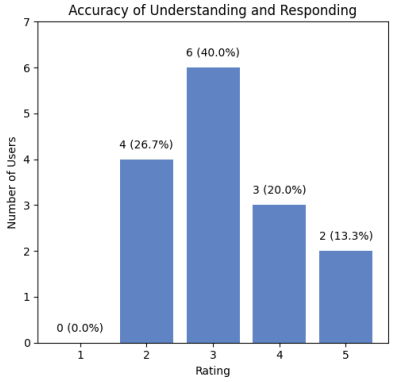

Fig. 2. Chatbot's accuracy of understanding and responding to users.

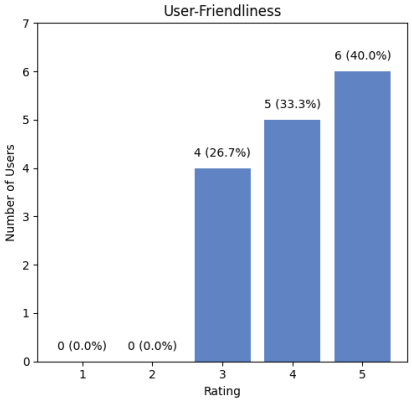

Fig. 3. Chatbot's user-friendliness to users

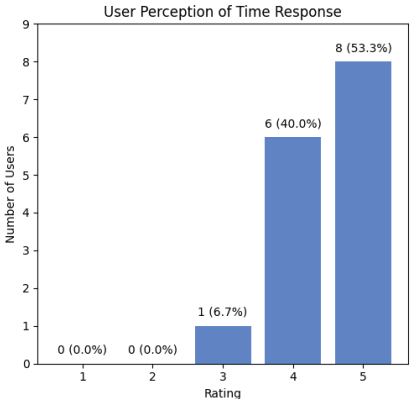

Fig. 4. Users' satisfaction with the chatbot's time response

Overall, user ratings were mostly positive, indicating that the chatbot received positive feedback during the workshop, i.e. most responses were above the average value of 3. Especially for the chatbots' accuracy received four responses below the average, most experts mentioned that their evaluation was mainly targeted at the necessity for enriching the content (e.g., more information, options, etc.). This was somehow expected because the two chatbots are currently in the pilot stage.

V. Conclusions and Future Work

This article described the chatbot generator architecture developed in the context of the PYTHIA project. This architecture was mainly shaped by the requirements and specifications elicited from the National Bank of Greece and the General Secretariat for Information Systems & Digital Governance. These two pilot chatbots follow a rule-based approach and currently support only Greek, however, there are plans to include machine learning functionality and English support. The paper also delved into the main components of the architecture and its operational aspects. Overall, the chatbot generator streamlines the creation and deployment of chatbots, allowing businesses and individuals to efficiently implement chatbot solutions without requiring significant development resources.

The requirements and specifications outlined in this paper are expected to serve as a roadmap for the complete implementation of the foreseen system. Subsequently, the chatbot generator undergoes extensive pilot testing involving multiple end users in different applications to evaluate its overall performance and efficiency. As highlighted in, [17], the evaluation of platform effectiveness and usability has to be done through a set of key performance indicators (KPIs). The purpose of this evaluation is to gauge its ability to provide valuable and relevant information and support relevant transactions in a variety of situations and issues of complexity, ambiguity, and uncertainty. Future development efforts will focus on the incorporation of encryption mechanisms and the safeguarding of critical data, alongside the utilization of cloud infrastructures. This extension will allow the application of the foreseen solution in specific fields such as defense and industry.

Acknowledgments

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] Adamopoulou, E., Moussiades, L: An Overview of Chatbot Technology. Artificial Intelligence Applications and Innovations. AIAI 2020. IFIP Advances in Information and Communication Technology, vol 584. Springer, Cham (2020)

[2] Santhanam, S., Hecking, T., Schreiber, A., Wagner, S: Bots in software engineering: a systematic mapping study. PeerJ Computer Science. 8. e866 (2022)

[3] Oreški. D., Vukovac, D., Hajdin, G: Assessment and Comparison of Meta-features for Educational Chatbots Data and Survey Data. WSEAS Transactions on Advances in Engineering Education, vol. 19, pp. 22-29 (2022)

[4] Do, S., Jeong J: Design and Implementation of RPA Based ChatMES System Architecture for Smart Manufacturing. WSEAS Transactions on Computer Research, vol. 10, pp. 88-92 (2022)

[5] Onlim Website, https://onlim.com/en/top-requirements-for-chatbot-software/ [Accessed on 13/10/2023]

[6] 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 2022 2nd International Conference on Advanced Research in Computing (ICARC), pp. 254-259., Belihuloya, Sri Lanka (2022)

[7] 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)

[8] 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)

[9] Reiter, E.: An architecture for data-to-text systems. In: Proceedings of ENLG-2007 11th European workshop on natural language generation, pp 97–104., Saarbrücken, Germany (2007)

[10] Firdaus, M., Golchha, H., Ekbal, A., Bhattacharyya, P.: A Deep Multi-task Model for Dialogue Act Classification, Intent Detection and Slot Filling. Cognitive Computation 13, 626–645 (2020)

[11] 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. Wiley, Chichester, pp 93–118. (2011)

[12] McTear, M., Callejas, Z., Griol, D.: The Conversational Interface Talking to Smart Devices. Springer International Publishing (2018)

[13] Eleni Adamopoulou, Lefteris Moussiades: Chatbots: History, technology, and applications. Machine Learning with Applications, Volume 2, 100006, (2020)

[14] 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)

[15] Abdul-Kader, S. A., Woods, J.: Survey on Chatbot Design Techniques in Speech Conversation Systems. International Journal of Advanced Computer Science and Applications, 6, pp. 72-80 (2015)

[16] Arsovski, S., Osipyan, H., Oladele, M. I., Cheok, A. D.: Automatic knowledge extraction of any Chatbot from conversation. Expert Systems with Applications, 137 (2019), pp. 343-348 (2019)

[17] Androutsopoulou, A., Karacapilidis, N., Loukis, E., Charalabidis, Y.: Transforming the communication between citizens and government through AI-guided chatbots. Government Information Quarterly, 36(2), pp. 358-367 (2019)