Journal of Applied Intelligent System (e-ISSN : 2502-9401 | p-ISSN : 2503-0493)
Vol. 9 No. 2, 2024, pp. 184 – 191

Artificial Intelligence Chatbot for Customer Service in E-Commerce Using Telegram Based on Node.js

Fuaida Nabyla[1], Nurul Mega Saraswati[2], Arif Nursetyo[3], Rito Cipta Sigitta Hariyono[4]
[1,2]Department of Informatics, University of Peradaban, Jalan Raya Pagojengan KM. 3 Paguyangan, Brebes, Central Java, Indonesia, 52276, (0289) 432032
[3,4]Department of Informatics, Bhamada Slawi University, Jalan Cut Nyak Dien, Kalisapu, Central Java, Indonesia 52416, (0283) 6197570
*Email: nabilafuaida@gmail.com[1], nurul.mega.s@gmail.com[2], setyonurarif@gmail.com[*3], rintocipta13@gmail.com[4]*

Ariani Dwi Septi[5]
[5]Department of Informatics, Bhamada Slawi University, Jalan Cut Nyak Dien, Kalisapu, Central Java, Indonesia 52416, (0283) 6197570
Email: arianidsepti01@gmail.com[5]

ABSTRACT - *Currently, the traditional market is increasingly being supplanted by numerous online markets. The fierce competition in the online market necessitates excellent service from sellers to buyers. As a result, many online stores now offer round-the-clock service, which can be financially burdensome if handled manually. Chatbots offer a promising solution by automating the online shopping process, thereby reducing costs and enhancing customer service. To address the need for accurate and prompt responses, this study proposes an intelligent chatbot system built on Artificial Intelligence Markup Language (AIML), specifically tailored to function as an e-commerce assistant. Integrated seamlessly into the Telegram application, the chatbot efficiently processes user input questions through three essential stages: parsing, pattern matching, and data crawling, all powered by AIML technology. Within the AIML process, user requests are systematically categorized into three primary domains: general questions, calculations, and stock checks. Notably, the calculation category encapsulates both order and payment processes. The effectiveness of the proposed method is substantiated by the results of 200 trials, demonstrating its adeptness in accurately addressing all user inquiries. This system is still limited to raw words, preventing it from systematically developing datasets. It is hoped that further research will improve the development of this chatbot and enable systematic data parsing.*

1. INTRODUCTION

Currently, there's a rising demand for online markets among both customers and business individuals, thus fueling the upward economic growth trajectory of these platforms [1]. fierce competition and a global customer base necessitate top-notch services, including round-the-clock availability. Providing 24-hour service entails additional expenses, as it necessitates continuous customer support to deliver services effectively. Implementing a chatbot can serve as a cost-effective remedy for reducing customer service expenses, as it enables the delivery of automated assistance [2]. A chatbot is a computer program designed to engage in intelligent conversations with users. Creating a chatbot requires advanced programming skills and a deep knowledge base, as crafting accurate and natural conversational phrases for an effective chatbot is a demanding endeavor. Thus, a substantial database is necessary to offer satisfactory responses to every interaction [3].

Essentially, a Chatbot is employed to handle simple conversations by addressing user inquiries according to predefined questions stored in a knowledge base [4]. Chatbots are currently undergoing advancements to enhance their efficiency, speed, and accuracy in providing responses. Within the chatbot system, there are multiple processes in place, including data parsing, pattern matching, and data crawling. Some studies choose not to employ the entire process in order to expedite computing. As indicated in research [5], the chatbot system utilizes only two processes, parsing and data crawling. Out of 1500 questions assessed, approximately 1200 questions can be accurately answered, resulting in an overall accuracy rate of around 80%.

Another study only proposed pattern matching and data crawling which resulted in 95% accuracy having a range of response times varying between 7.5 seconds and 48 seconds [6]. For an effective Chatbot application, employing grammar-based data parsing is essential. This approach aids in comprehending the user's intended sentence by describing phrases tailored to the complexity of the grammar utilized [7]. Multiple stages of parsing are conducted through various methods, including case folding, tokenizing, filtering, and stemming. However, numerous issues arise during the pattern-matching process due to the lack of standardization in various rules, stemming from improper data parsing processes [8]. Establishing an effective pattern-matching process involves ensuring that the pattern applied can systematically extract relevant information for analyzing the text [5] this can be accomplished by filtering out irrelevant data and selectively choosing during the parsing phase [9].

Data crawling serves as the ultimate step utilized to find data within the database that corresponds to the results obtained from pattern matching [4]. AIML, an approach conceptualized by Dr. Richard Wallace and Alicebot, can be integrated into the data crawling process to minimize errors encountered in both stages. AIML represents a progression of the Artificial Linguistic Internet Computer Entity (ALICE) method. It emphasizes the necessity for users to consistently maintain focus on the topic of chat conversations [3]. AIML enhances its performance in generating responses to questions and input by leveraging a conversation knowledge base, thus overcoming the limitations of ALICE. This refinement allows AIML to achieve simplicity, efficiency, and lightweight configuration, thereby facilitating smoother user conversations [10]. In this study, a Chatbot system will be proposed to streamline the customer service process in e-commerce platforms.

Conversely, Mhatre et al. [11] utilized AIML and pattern-matching techniques within machine learning to develop an interactive web-based bot aimed at facilitating information exchange through email conversations. Their methodology entails assessing AIML files with the aid of language processing algorithms, with a specific emphasis on pattern matching. Through experiments conducted using Donna Interactive Bot, out of around 700 messages, the response time varied between 7.5 seconds and 48 seconds.The responses were categorized as follows: 5% poor response, 35% excellent response, and 60% responses similar to the request. In assessing the system's efficiency to respond within a specified timeframe, it achieves an overall efficiency rate of 70%. However, a notable limitation arises when the input load increases, leading to a decrease in the system's efficiency. This limitation is primarily attributed to the system's prolonged response time to Client demands.

Pradana et al. [10] utilized AIML and ELIZA techniques similar to those employed in SamBot, integrating a module for maintaining knowledge based on logs. The performance of the system was assessed through the Turing Test or the Loebner Prize chatbot worldwide, resulting in a score of 132 points and 26.4 points, respectively. During the study, researchers received 100 questions from online users, of which 30 questions were considered unsuitable for a response. According to this research, the chatbot's responses are confined to the knowledge possessed by the botmaster. Consequently, the chatbot may provide inaccurate responses to inputs and generate random answers, potentially disrupting users' interactions with the chatbot.

Tufai et al. [12] utilized the A.L.I.C.E algorithm for simulating conversations and employed AIML for generating text-based responses. They applied pattern-matching techniques to analyze messages against predefined questions to generate appropriate responses. However, this research does not specify the response time as the knowledge base is extremely limited. Consequently, the search for specific information may result in increased conversation traffic, potentially leading to a communication gap between users and the chatbot.

2. METHOD

2.1 Artificial Intelligence Markup Language (AIML)

AIML shares similarities with XML in that it comprises a collection of patterns and responses utilized by the chatbot to formulate replies to each sentence provided in diverse programming languages. This characteristic streamlines the chatbot creation process, emphasizing the preparation of AIML documents. Typically, the process of reading text in AIML involves two stages, namely:

1. The first stage entails reading the dialogue text provided by the user and converting it into a vector representation.
2. The second stage involves employing the converter module. During this phase, the pre-processed text is sent to the converter, where it is understood that the first round serves as a pattern while the second round functions as a template. Throughout this process, all punctuation is removed from the pattern, and it is converted into letters. During this stage, a context vector is employed to represent the user input, facilitating the generation of a reply vector, which serves as the corresponding response output. This highlights the importance of data crawling within the database [4].

For further insights into the AIML architecture, refer to Figure 1.

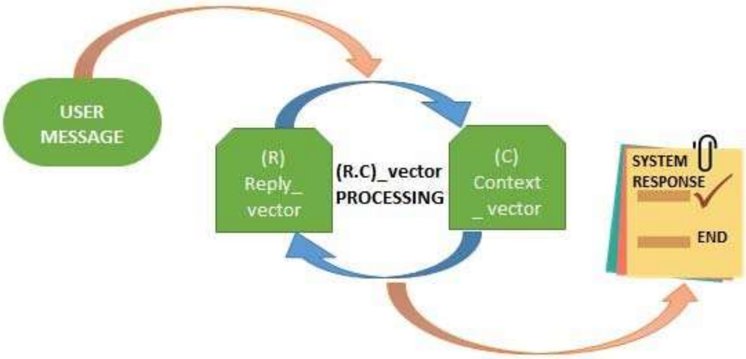

Figure 1. AIML Architecture

2.2 Parsing Data and Pattern Matching

Data parsing is conducted as a procedure to verify if a query string or syntax conforms to the specified query syntax rules. This enables the handling of user inquiries by posing questions from a predefined collection stored in the repository of knowledge. In the development of a chatbot, it becomes imperative to devise a method for parsing the sequence of inputs provided by the user. These parsed inputs are then utilized in the subsequent stage, namely semantic analysis. There are 4 stages involved in the data parsing process, which are as follows:

1. Case Folding, In this phase, all letters within the document are converted to lowercase. This guarantees that exclusively letters ranging from 'a' to 'z' are recognized when users input text into the system. Standardizing the text to lowercase helps in identifying queries accurately within the database, as it eliminates the distinction between uppercase and lowercase letters [4].
2. Tokenizer, this stage involves segmenting the input string into individual words or tokens. It cuts the string into tokens based on spaces or other defined characters, effectively splitting a set of characters in a text into individual words. This process helps distinguish individual words and facilitates further analysis of the input text.
3. Filtering, In this stage, essential words are extracted from the results obtained from the tokenization process. Less important words are removed, and only significant words are retained and stored in the database for further processing. This ensures that only relevant information is considered during subsequent stages of analysis.
4. Stemming, In this stage, the root of each filtered word is identified to determine its index. This process is essential as a document cannot be directly recognized. Stemming involves returning various forms of words to the same representation, thereby reducing the number of different indices for a query. For more details on data parsing and pattern-matching architecture, refer to Figure 2.

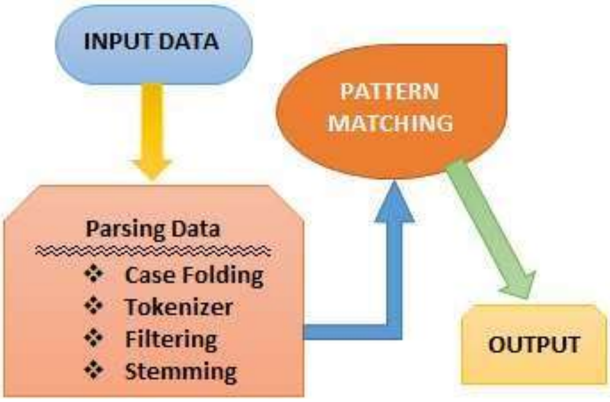

Figure 2. Data Parsing and Pattern Matching

Efficient data parsing enables the pattern-matching process to analyze relevant text effectively by employing sequentially applied patterns to extract useful information. This involves eliminating irrelevant details through selective filtering [13], thereby facilitating a streamlined and efficient pattern-matching process. This technique is widely utilized in many Chatbots and is commonly employed in question-and-answer systems, depending on the type of matching required. It can accommodate various forms of queries, including natural language questions, simple statements, or semantic questions.

2.3 System Architecture for Chatbot

The chatbot system created in this study is tailored for applications within e-commerce settings. The proposed chatbot is capable of handling general conversations related to products, remaining stock availability, order inquiries, and payment processes within the e-commerce platform. Figure 3 illustrates the stages of application development conducted in this study.

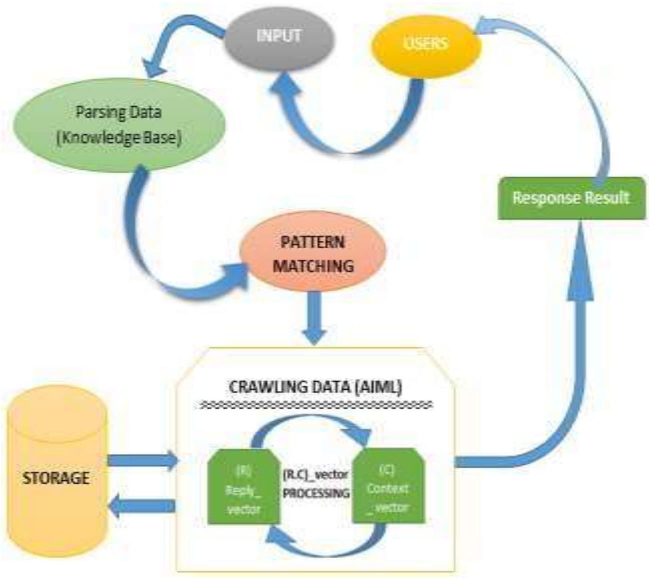

Figure 3. The system architecture of the chatbot

1. Input

   The input data to the system comprises user queries, which are analyzed to ascertain the conversation's direction and formulate an appropriate response. Every input data is attributed an identifier comprising a connection and timestamp. This identification system aids in distinguishing questions originating from various users and prevents the simultaneous processing of multiple questions from the same user.
2. Parsing Data

   The second stage involves determining whether the formulated query adheres to the query syntax rules in the chatbot's knowledge base. This is achieved by dissecting the series of queries submitted by the user. This sequential process is employed to extract valuable information, thereby enabling smooth and efficient operation.
3. Pattern Matching

   During the third stage, pattern matching takes place. Here, the question pattern is categorized into three categories of messages: general inquiries, stock availability checks, and purchasing and payment processes. These categories are further segmented into the calculation procedure.
4. Crawling Data (AIML)

   During this phase, the results of the pattern-matching process are integrated into the context vector to facilitate data crawling within the chatbot's knowledge base. The process of data crawling is stratified based on the classification outcomes derived from pattern matching. Figure 4 delineates the pseudocode that discerns the process of crawling data for general questions, calculation questions, and stock inquiries. Furthermore, the retrieved data is stored as a reply vector, subsequently presented as a response and answer by the chatbot.

```
Start
Function (msg.match)
Input message
Stock = ChatID
Set chatId
Intervalid = match [1]
Set respository = 'message' + match [1]
Pirnt(chatId, respository)
End
```

5. Storage

   Each conversation initiated by a user is assigned a connection identification variable and a timestamp. These identifiers are utilized as aids during the storage procedure. The server records the latest timestamp for each connection in the table. This ensures accurate tracking of the latest interactions. The server maintains a global variable that enables it to manage the deletion of old timestamps. This functionality is crucial for handling user inquiries when posing queries from the predefined set of questions established in the chatbot's knowledge base.
6. Response Result

   During this phase, the outcomes of the chatbot system's response are exhibited on the user interface. Additionally, a filtering process is conducted on the outcomes to manage multiple answers from various categories of inquiries effectively.

3. RESULTS AND DISCUSSION

3.1. Research Results

The chatbot system functions as a means for placing service orders, managing transactions, performing stock computations, and handling common queries. It operates through the Telegram app. To assess its performance, we employ two platforms: a Windows 10 Pro 64-bit server equipped with an i3 processor and 4 GB RAM, and an Android KitKat user device with a Dual Core processor and 512 MB RAM. To assess the performance of the chatbot system, 300 user questions were posed to measure the response time. It's important to note that the testing process was not conducted simultaneously, and there were intermittent pauses that may have affected internet bandwidth. Since this system operates on a client-server model reliant on internet connectivity, the rate of the connection significantly influences the response time. In this research, the internet connection utilized had a download speed of 1.10 Mbps with an upload speed of 2.26 Mbps. The performance of the chatbot system on Telegram is outlined in the figure below.

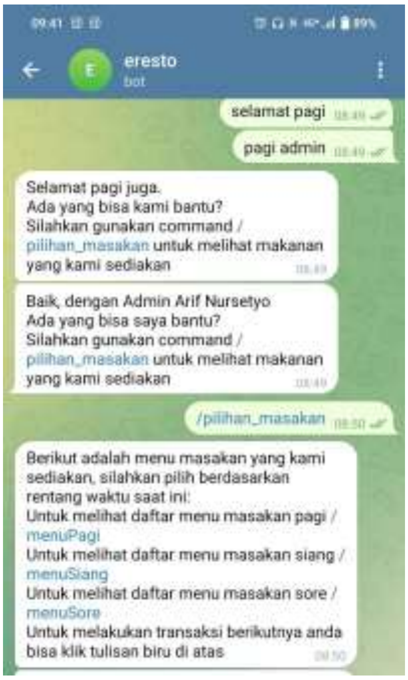

Figure 4. Performance of Chatbot on Telegram

3.2 Research Discussion

The research has effectively created an intelligent chatbot system using AIML technology within the Telegram app to offer E-commerce support. User queries are subjected to three primary stages: parsing, pattern matching, and data retrieval. Within this framework, parsing serves to isolate the essential user request by removing irrelevant elements according to syntax and grammar guidelines. Following that, the pattern-matching phase is carried out to determine the best-suited pattern for addressing the user's query. In this study, the data crawling operation utilizes AIML, categorizing queries into three types: general inquiries, calculations, and stock checks. The specific case examined involves a chatbot tailored for food sales in Indonesia. Analysis of 200 trials reveals that all bot responses were precise, with an average response time of 1.5 seconds.

It is evident that the conversations between users and the chatbot are highly accurate. The chatbot system effectively responds to user questions, providing relevant and precise answers. Additionally, it successfully performs stock, order, and payment calculations. Chatbots are capable of providing users with clear instructions. The average response time observed was 3.4 seconds, with the longest response taking about 7 seconds. This response time falls within the optimal range, which extends from 4.5 seconds to 79.4 seconds. This response speed is notably superior to that reported in research [13], which necessitates a response time varying between 7.5 and 48 seconds. Additionally, the accuracy of responses achieved in this study is exemplary, with a 100% accuracy rate [14]. This outcome surpasses the findings of research [15], which yielded an 88% response accuracy.

4. CONCLUSION

This study has attained a response accuracy rate of 100% through testing with accurate and formal language. Future research aims to enable the system to identify and rectify user typing errors. Additionally, offering automatic recommendations or suggestions for user inputs will be crucial to augmenting the accuracy and fluidity of conversations. This system is still limited to raw words, preventing it from systematically developing datasets. It is hoped that further research will improve the development of this chatbot and enable systematic data parsing.

REFERENCES

[1] A. Kamarudin, "Tata Kelola Teknologi Indonesia," https://andi.ddns.net/, Gorontalo, 2020.

[2] M. Skjuvea, A. Følstad, K. I. Fostervold dan P. B. Brandtzaeg, "A longitudinal study of human–chatbot relationships," *Elsevier Ltd.,* vol. 168, no. 102903, pp. 1-14, 2022.

[3] A. Janssen, D. R. Cardona, J. Passlick dan M. H. Breitner, "How to Make chatbots productive – A user- oriented implementation framework," *Elsevier Ltd.,* vol. 168, no. 102921, pp. 1-22, 2022.

[4] A. Nursetyo, D. R. I. M. Setiadi dan E. R. Subhiyakto, "Smart Chatbot System for E-Commerce Assitance based on AIML," Yogyakarta, 2018.

[5] R. Sutoyoa, A. Chowandaa, A. Kurniatia dan R. Wongsoa, "Designing an Emotionally Realistic Chatbot Framework to Enhance. Its Believability with AIML and Information States," *4th International Conference on Computer Science and Computational Intelligence (ICCSCI),* vol. 157, pp. 621-628, 2019.

[6] S. A. Abdul-Kader dan D. J. Woods, "Survey on Chatbot Design Techniques in Speech Conversation Systems," *(IJACSA) International Journal of Advanced Computer Science and Applications,* vol. 6, no. 7, pp. 72-80, 2015.

[7] S. Geethanjali dan B. A. M. J., "Towards Building a Competent Chatbot – An Analogy of Development Framework, Design Techniques and Intelligence," *International Journal of Innovative Research in Science, Engineering and Technology,* vol. 6, no. 11, pp. 554-562, September 2017.

[8] L. Li, Z. Zhou, N. Bai, T. Wang, K.-H. Xue, H. Sun, Q. He, W. Cheng dan X. Miao, "Naive Bayes classifier based on memristor nonlinear conductance," *ELSEVIER Ltd.,* vol. 129, no. 105574, pp. 1-14, 2022.

[9] B. S. Bani dan A. P. Singh, "College Enquiry Chatbot Using A.L.I.C.E," *International Journal of New Technology and Research (IJNTR),* vol. 3, no. 1, pp. 64-65, January 2017.

[10] A. Pradana, G. O. Sing dan Y. J. Kumar, "SamBot - Intelligent Conversational Bot for Interactive Marketing with Consumer-centric Approach," *International Journal of Computer Information Systems and Industrial Management Applications,* vol. 6, pp. 265-275, December 2017.

[11] N. Mhatre, K. Motani, M. Shah dan S. Mali, "Donna Interactive Chat-bot acting as a Personal Assistant," *International Journal of Computer Applications,* vol. 140, no. 10, pp. 6-11, April 2016.

[12] M. Tufail, M. Danish dan P. I. Shahzad, "Interactive College Enquiry Using Bot," *International Journal of Research in Information Technology,* vol. 2, no. 3, pp. 1-7, March 2018.

[13] Lasmadi, Freddy Kurniawan, Denny Dermawan, Gilang N. P. Pratama, "Mobile Robot Localization via Unscented Kalman Filter," dalam *2019 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*, Yogyakarta, 2019.

[14] K. Hercegfi, "Heart Rate Variability Monitoring during Human-Computer Interaction," *Acta Polytechnica Hungarica,* vol. 8, no. 5, pp. 205-224, 2011.

[15] S. Mujeeb, M. H. Javed dan T. Arshad, "Aquabot: A Diagnostic Chatbot for Achluophobia and Autism," *(IJACSA) International Journal of Advanced Computer Science and Applications,* vol. 8, no. 9, pp. 209- 216, 2017.