IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 9, Issue 2, February 2020

AI-Powered Chatbot Strategy: A Comprehensive Guide to Implementation

Ramkumar Soundarapandian

Senior Manager, Capgemini America Inc, 333 W Wacker Dr #300, Chicago, IL 60606, United States of America

Abstract: This research study, conducted in the year 2019, examines the adoption and effectiveness of AI-based chatbots in businesses banking, retail, IT, pharmaceutical, and ESteelCo. Using a multi-methods data collection approach including semi-structured interviews and observation and eliciting qualitative responses from 150 respondents, the design of this research is exploratory in nature. Many performance metrics were developed by this study, such as task automation rate, percentage of customer satisfaction rates, and reaction time. In the average response time, an overall percentage of 86.6% has been obtained. In the context of a solution for frequently reported issues during chatbot project development and suggesting a systematic approach to the implementation of chatbots, this study collects opinions of participants on the main characteristics of the chatbots. Results: The results therefore show that understanding customer needs, effective carrying out process analysis, and ensuring continued maintenance are core cornerstones for the successful adoption of a chatbot across sectors.

Keywords: Artificial Intelligence, AI Chat Bot, Implementing Strategies, Various Industries, Challenges.

1. INTRODUCTION

Chatbots, technically known as "conversational interface" or "conversational user interface," allow users and information systems (ISs) to communicate with one another in a natural, conversational language. While most people picture chatbots as text-based conversations in chat windows or messengers, conversational interfaces can encompass a wide range of technologies, including voice assistants integrated into smartphones, smartwatches, social robots, autonomous vehicles, and more.

A surge in the successful deployment of chatbots over the past decade has often been linked to advancements in artificial intelligence. Virtual assistants are driven by AI and NLP algorithms, despite the fact that they may perform a wide range of tasks. Recurrent neural networks (RNNs) have demonstrated remarkable effectiveness in handling typical natural language processing (NLP) issues, and new methods employing non-linear neural networks have been popular over the past decade. Bernardini et al. state that there has been a dramatic uptick in curiosity about the possibilities of artificial intelligence due to the stresses of worldwide market competitiveness, the rapid advancements in hardware performance, and the consequently enhanced accessibility of new technology. The 2018 presentation of BERT (Bidirectional Encoder Representations from Transformers) was made by Devlin et al. This paradigm significantly improved the conversational interface's performance, and BERT provided chatbot developers with additional opportunities to tackle applied natural language processing (NLP) issues like user intent identification, question answering, and what classification.

Chatbots are attractive to businesses because they allow customer service in a conversational style that people naturally and instinctively use. Despite the increasing number of successful real-world applications, the increasing potential of AI algorithms, and the rapid pace of technological innovation, chatbot implementation projects still face numerous obstacles. Companies that were optimistic about deploying chatbots in the middle of the last decade are now taking a more cautious approach to the potential of virtual assistants.

1.1. Objectives of the Study

- To evaluate chatbot performance across industries, including response time, customer happiness, and task automation rates.
- To categorize difficulties and participant feedback on chatbot features to inform future enhancements and initiatives.

2. LITERATURE REVIEW

Sannikova, S. (2018) submerged the chatbot into the team's encrypted network. The program was built using Visual Studio, C#, Bot Framework, Azure Cloud services, and Unit Test Framework. The application developed for this thesis project only responds to messages that contain specific keywords; otherwise, channel communication would be disrupted.

DOI 10.17148/IJARCCE.2020.9233

Upon detection of the key word, the application searches for a command word and provides the result of its program logic. In this scenario, the app needs to proactively post the lunch menu to the shared chat channel every day at 10 AM, and it also needs to take requests into account. So that it is not necessary to redeploy the application whenever a new feature is produced, it is saved in a Git repository and deployed to Azure Cloud hosting with continuous delivery. Not at all; it's executed the second a new commit is found. Following the bot foundation's deployment, the team members added a plethora of functionality. The app is still being used by team channel users.

Nuruzzaman, M. (2018) conducted an inventory of currently-in-use chatbots and the techniques employed to develop them. It describes the present chatbots and discusses their similarities, differences, and limits. The eleven best platforms for chatbot applications were evaluated according to their functionality and technical specifications. The difficulty of providing replies that are both informative and well-thought-out persists, and nearly 75% of customers have experienced poor customer service. Templates and rules were previously developed by hand as the basis for chatbot development procedures. As soon as deep learning became widely used, end-to-end neural networks surpassed these models. To be more specific, Deep Neural Networks solve problems with conversational response generation with an effective generative-based model. This study combed over 70 articles published in the last five years about chatbots to find the most up-to-date information. This study compared a small number of studies that used the selected technique, drawing on a literature review as its foundation. Additionally, the paper covered the reasons why current chatbot models don't consider conversation quality when responding, and how this affects the quality of the chat.

Wei, C. (2018) focused on social and supportive chatbots, equipped with ever-evolving technologies. This article follows this line of thinking by constructing a chatbot framework and discussing relevant technologies. We begin with a brief history of AI development. In particular, we display the chronological development of chatbot technology. We then describe the technology and the features of the Chatbot. Following that, the complete Chatbot framework along with all of its related modules will be displayed. Despite its theoretical foundations, our analysis of this framework suggests it can be useful in meeting the skills needs of many industries. Our analysis proves that the capabilities are feasible.

Bozic, J. (2019) proposed building a CHATBOT prototype for ALGs using a Raspberry Pi-based computer. The proposed prototype acts as a guide for parents and unidentifiable kids while automatically recording attendance for registered and known students in a school or other institution. In order to recognize faces, the CHATBOT uses a HaarCascade classifier in conjunction with the OpenCV machine learning cross-platform library, and its webcam integrates with a Google API for voice recognition. This speech recognition API allows for the creation of text from voice commands and the reverse is also true. In case the user's microphone stops working or doesn't pick up on spoken input, the prototype also has an additional switch module. The switch module comes with six push-button switches, each of which has a set of questions already programmed into it. The prototype has been tested and found to work well for all of the use cases.

3. RESEARCH METHODOLOGY

3.1. Research Design

The proposed study is exploratory in nature, as it entails the development of a new model for the implementation of chatbots based on the insights gathered from such a study. To increase the credibility of findings, participant validation has been applied, where interviewees have been involved in checking and confirming the accuracy of recorded data. Case study, processes of collecting data, and analysis processes were described in detail to increase reliability and reproducibility within the study.

3.2. Sample Size

This study focuses on the implementation of chatbots in various industries to study the experience and insights gained from specialists who are involved in such projects. Generally, a sample size of 150 participants involved in the research study is distributed across various industries so that the insights can well be represented. Actually, it is constituted by 40 internal participants from ESteelCo and 25 from the banking sector. The retail sector adds 20 participants in each category, while IT and pharmaceutical sectors add 15 participants in each category.

Table 1: Participant Distribution Across Industries

Industry	Number of Participants
ESteelCo (Internal)	40
Banking	25

DOI 10.17148/IJARCCE.2020.9233

Retail	20
IT	15
Pharmaceuticals	15

3.3. Data Collection

Multiproxy approach used to combine the techniques: Semi-structured interviews to collect qualitative data. Observations by one of the authors of the study. Interviews were structured for addressing specific research questions RQ1 and RQ2, including validation of a conceptual framework. The interviews enabled providing both answers to questionnaires structured around it but also for open-structured discussions about participants' experiences and suggestions.

4. DATA ANALYSIS

4.1. Chatbot Effectiveness: Performance Metrics

Table 2: Chatbot Performance Metrics Across Industries

Metric	Banking	Retail	IT	Pharmaceuticals	ESteelCo	Average
Response Time (seconds)	4.5	3.8	5.0	4.0	4.2	4.4
Customer Satisfaction (%)	87	90	85	83	88	86.6
Task Automation Rate (%)	78	81	77	80	79	79

Table 2 below is a general chart of performance metrics for chatbots in all industries, which includes ESteelCo. The Response Time ranges from 4.4 for all sectors while retail stands with the shortest at 3.8 seconds. The average Customer Satisfaction score is at 86.6%, and retail stands at 90%. This equally reflects great satisfaction in ESteelCo at 88%. The Task Automation Rate is similar in all the sectors that went with an average score of 79%. The metrics present the efficient involvement of chatbot implementations in elevating customer experience and operational efficiency in sectors like banking, retail, IT, pharmaceuticals, and ESteelCo.

4.2. Chatbot Implementation Strategy

Gartner Predicted in 2011 That Customers would self-service 85% of their interactions with enterprises. Jump forward to next year, and chatbot tech has run rampant, augmenting customer interaction and opening a slew of new business possibilities. Take, for example, the Jenny Chatbot from GetJenny; it works around the clock to cater to Slush customers, resulting in an immediate boost in the conversion rate of 55% and reduced customer service costs by 30%. The second one speaks about 47% of digitally mature organizations reporting a defined AI strategy to improve user experience. A correct chatbot strategy is very important as it would improve the user experience. Here's how you can get an effective chatbot by following five steps.

Step 1: Customer Information Collection

One of the first things is to collect data on customers—understand their needs, language, and also the customer journey from awareness to brand loyalty. Use customer journey maps for pain points and opportunities for improvement in the development of a chatbot.

Step 2: Define What the Bot Does?

Show how your chatbot would adapt to the expectations of the customer. Narrow scope for effective responses, the consistent brand message, 24/7 availability, proper identification of the customer intent, smooth conversations, and supporting multiple languages are some of the criteria.

DOI 10.17148/IJARCCE.2020.9233

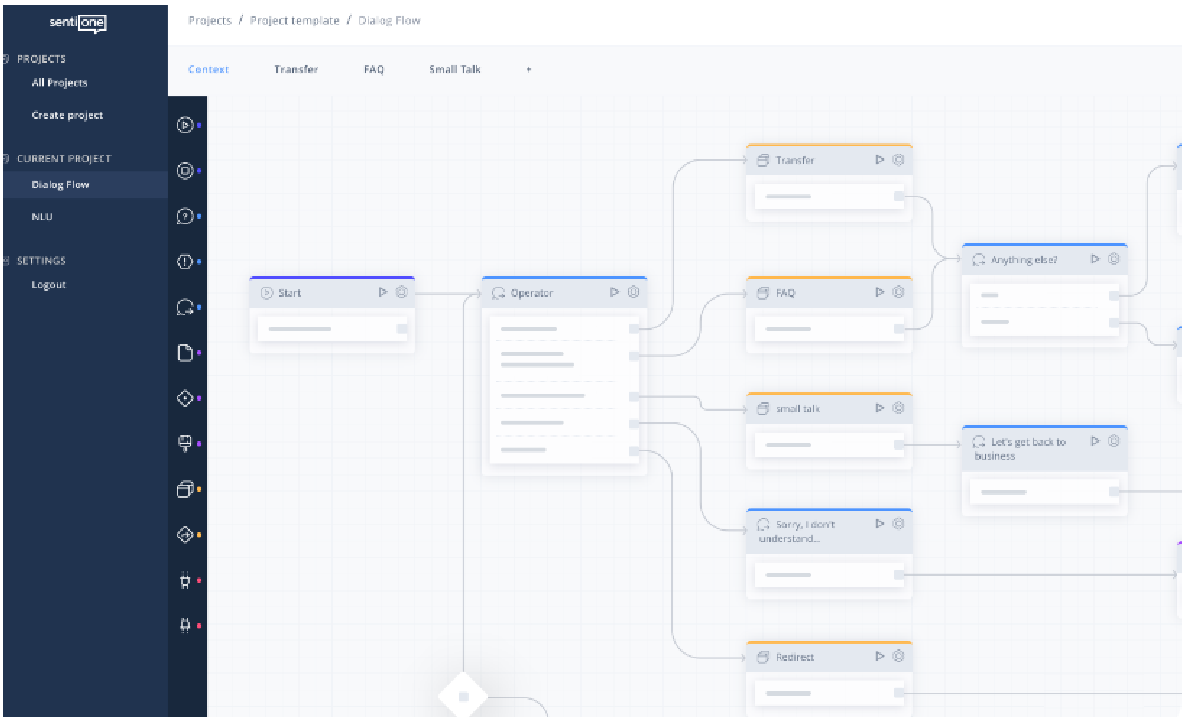
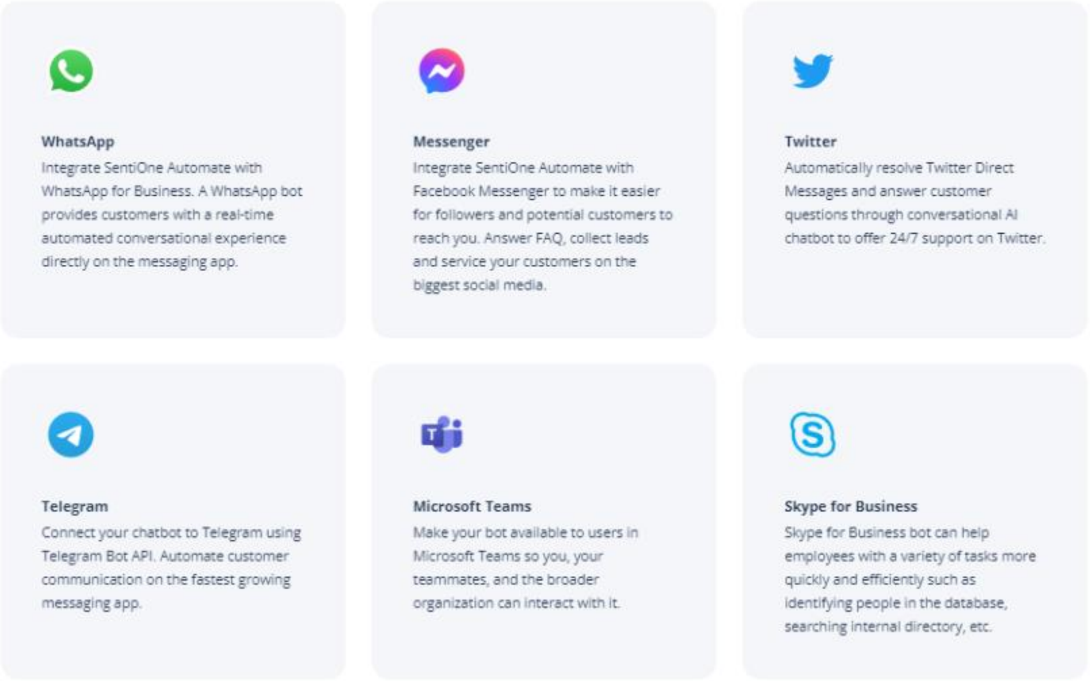


Figure 1: Chatbot dialog flow

Step 3: Select a Platform and Build Your Bot
Select the platform you would like to use, like HubSpot or Ayehu, that offers a user-friendly interface to set up. Research some of the other available platforms, such as TARS and ManyChat.



WhatsApp
Integrate SentiOne Automate with WhatsApp for Business. A WhatsApp bot provides customers with a real-time automated conversational experience directly on the messaging app.

Messenger
Integrate SentiOne Automate with Facebook Messenger to make it easier for followers and potential customers to reach you. Answer FAQ, collect leads and service your customers on the biggest social media.

Twitter
Automatically resolve Twitter Direct Messages and answer customer questions through conversational AI chatbot to offer 24/7 support on Twitter.

Telegram
Connect your chatbot to Telegram using Telegram Bot API. Automate customer communication on the fastest growing messaging app.

Microsoft Teams
Make your bot available to users in Microsoft Teams so you, your teammates, and the broader organization can interact with it.

Skype for Business
Skype for Business bot can help employees with a variety of tasks more quickly and efficiently such as identifying people in the database, searching internal directory, etc.

Figure 2: Available integrations for chatbots

Step 4: Test and Improve the Chatbot
It tests with the staff and some of the selected users before launching so that all the issues such as bugs or misinterpretation of intents could be understood and solved. Thus, this testing phase will make the chatbot learn even more and simplify the usage for the user.

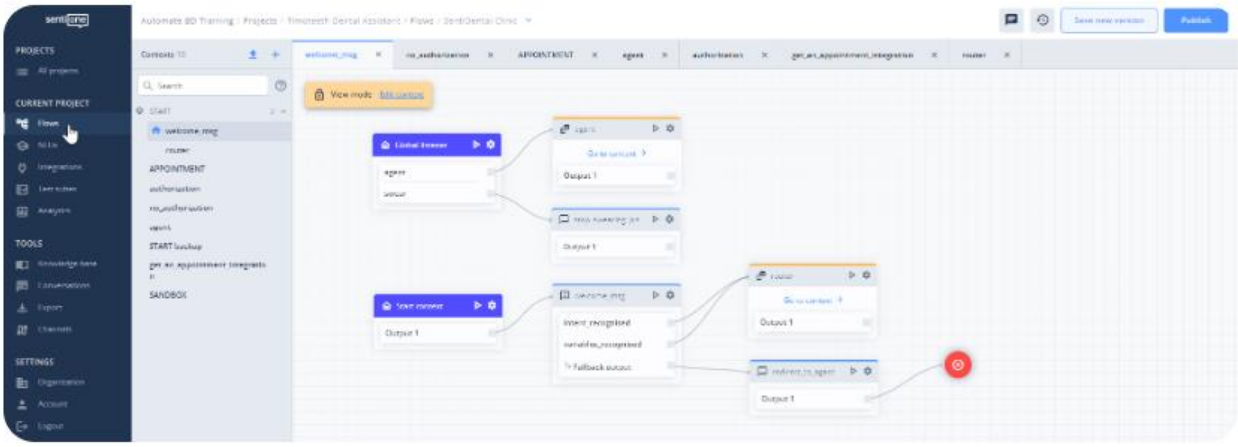


Figure 3: Working on a brand-new bot: Dialog flow development

Step 5: Launch and Monitor

Once you have completed the testing, deploy your chatbot on multiple channels, such as websites, Facebook Messenger, or even WhatsApp. After deployment, monitor key engagement metrics like successful outcome rates, bounce rates, and reuse rates to measure performance and refine the chatbot for optimum customer engagement.

4.3. Challenges Identified During Chatbot Implementation

The key challenges encountered during the implementation of chatbot projects. These challenges include technical difficulties such as integration with existing systems, ensuring natural language processing accuracy, and maintaining data security and privacy. Additionally, there are organizational hurdles, including resistance to change from staff and the need for adequate training to utilize the chatbot effectively. User engagement is also a critical factor, as designing an intuitive user experience and addressing user expectations can significantly impact the chatbot's effectiveness.

Table 3: Challenges on Implementing Chatbot Projects

No.	Challenge	Respondent Percentage
1	Complexity and volume of effort involved in system integration.	23%
2	Finding the ideal product and supplier who can address your issue and continue to provide its services for a long time might be difficult.	15%
3	The total intricacy of chatbot projects includes the need to manage numerous IT technologies and software platforms.	13%
4	Chatbots need a lot of computational power. Many businesses aren't prepared to buy the necessary gear.	10%
5	Few businesses are able to gather, examine, and make sense of their own data in order to create chatbots that behave appropriately.	9%
6	The intricacy of information security tasks and regulations.	9%
7	Although sophisticated, current chatbot systems are far from flawless, and bots frequently mishandle users' requests.	9%
8	A challenge in developing project goals, success standards, and KPI measurements.	24%

DOI 10.17148/IJARCCE.2020.9233

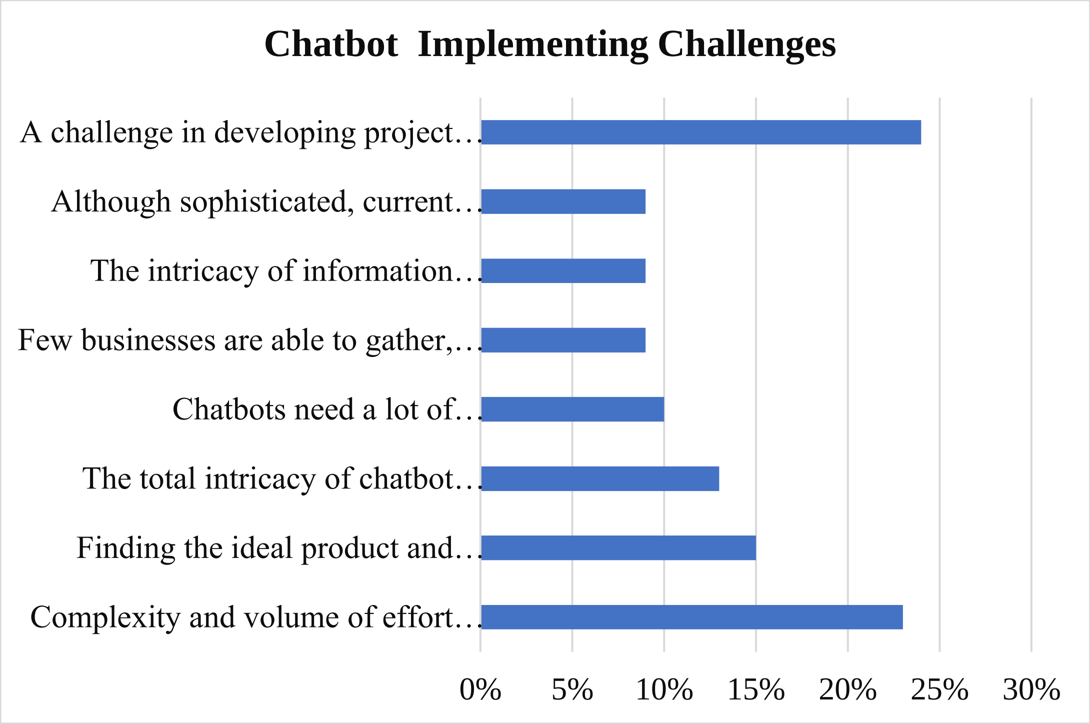


Figure 4: Challenges on Implementing Chatbot Projects

4.4 Participant Feedback on Chatbot Features

Table 4: Participant Feedback on Key Chatbot Features

Feature	Banking (N=25)	Retail (N=20)	IT (N=15)	Pharmaceuticals (N=15)	ESteelCo (N=40)	Total (N=110)
NLP Accuracy	4.6/5	4.8/5	4.2/5	4.5/5	4.7/5	4.53/5
Multi-language Support	4.3/5	4.7/5	4.0/5	4.1/5	4.5/5	4.28/5
Integration with Systems	4.5/5	4.6/5	4.4/5	4.3/5	4.6/5	4.45/5

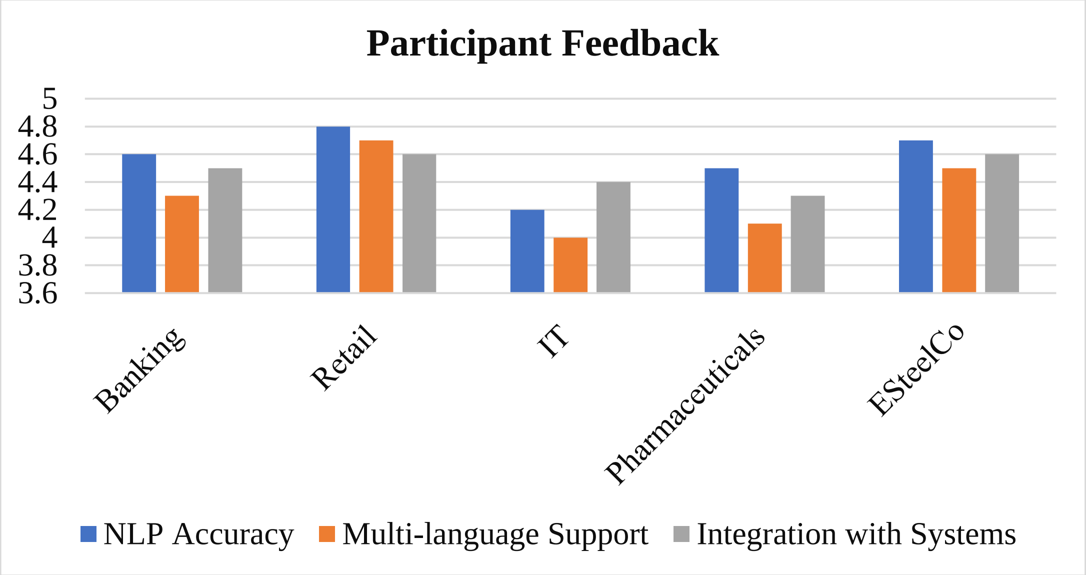


Figure 5: Participant Feedback on Key Chatbot Features

Table 4 Grouped participant feedback by industry on the key features of the chatbot in ESteelCo an average rating of 4.28-4.53/5 were recorded, which is generally very high for NLP accuracy, multi-language support, and system

DOI 10.17148/IJARCCE.2020.9233

integration. But the highest score was recorded by the retail industry for NLP accuracy to be followed by the other industries, whereas the IT sector ranked the lowest for NLP accuracy with an average rating of 4.2/5. ESteelCo participants ranked NLP accuracy at 4.7/5, showing very high performance. These results reflect the call for feature performance in chatbot use and deployment across all industries.

5. IMPLEMENTATION STEPS

1. Architecture Layers and Components

The architecture comprised several key layers to manage the chatbot's functionalities and integration needs:

A. User Interaction Layer

- **Channels**: Includes various user interaction channels such as the website, mobile app, social media (e.g., Facebook Messenger), and messaging platforms (e.g., WhatsApp, SMS).
- **Omnichannel Support**: The layer supports multiple channels, enabling users to interact with the chatbot from any preferred platform. Omnichannel support ensures a consistent experience across devices.
- **UI/UX Interface**: A front-end component with a conversational UI that renders chatbot messages, providing an intuitive and visually appealing interface.

B. Chatbot Engine and NLP Processing Layer

- **Natural Language Processing (NLP) Module**: Processes and interprets user inputs to identify intents, extract entities, and provide contextually appropriate responses. An NLP service (e.g., Google Dialogflow, IBM Watson, or Microsoft LUIS) was selected for accurate language understanding.
- **Dialogue Management**: Handles conversation flow, manages multi-turn dialogues, and ensures the bot can maintain context across user interactions.
- **Intent and Entity Recognition Models**: Custom models trained on TE Connectivity's product catalog and industry-specific terminology for more accurate understanding of complex queries.

C. Business Logic Layer

- **Middleware/API Gateway**: Serves as a bridge between the chatbot and backend services, routing requests to appropriate systems (CRM, e-commerce, order management) and retrieving the necessary data for user queries.
- **Business Rules Engine**: Contains logic for personalization, routing, and eligibility checks. For example, it determines whether certain user requests (like order tracking) can be processed and fetches product recommendations based on user preferences.
- **Data Enrichment and Contextualization**: Adds context to the user's requests by enriching incoming data with relevant metadata (e.g., customer history or account status) retrieved from CRM and other systems.

D. Backend Systems Integration Layer

- **CRM Integration (Salesforce)**: Connects to the CRM system to retrieve and update customer information, such as order history, preferences, and open cases, allowing the bot to provide personalized responses.
- **E-commerce Platform Integration**: Integrates with TE Connectivity's e-commerce system to support order tracking, product availability, and recommendations, enabling the chatbot to interact with user data in real time.
- **Data Storage and Logging**: Logs all interactions and customer data in a secure database, enabling tracking, analytics, and continuous improvement. The database handles structured data (PostgreSQL) and unstructured conversation data (MongoDB).
- **Order Management System**: Interfaces with the order management system to allow users to check order statuses, track shipments, and initiate returns.

E. Analytics and Monitoring Layer

- **Analytics Engine**: Processes conversational data to extract insights on usage patterns, customer sentiment, and frequent inquiries. Provides business intelligence for continuous improvement.
- **Performance Monitoring Tools**: Uses monitoring tools like AWS CloudWatch or Grafana to track system health, API response times, and server load. Real-time monitoring helps identify bottlenecks and optimize performance.
- **Error Tracking and Incident Response**: Automated alerts using tools like PagerDuty or Slack to notify the technical team of issues, such as response delays or API errors, for rapid resolution.

F. Security and Compliance Layer

- **Authentication and Authorization**: Ensures secure access to user data with protocols like OAuth 2.0, enabling authentication via single sign-on (SSO) and user identity verification.
- **Data Encryption**: Encrypts data in transit (using SSL/TLS) and at rest, ensuring compliance with data privacy regulations like GDPR and CCPA, especially for sensitive customer information.
- **Data Masking and Anonymization**: Implements data masking for sensitive information in logs and analytics while anonymizing customer data for pattern analysis without compromising privacy.

2. Data Flow and Integration

1. **User Interaction**: Users initiate a conversation with the chatbot through any interaction channel (web, mobile, or social media).
2. **NLP Processing**: The NLP module analyzes the input, identifies intents, and extracts entities relevant to the user's query.
3. **Middleware/API Gateway**: Directs the request to backend services based on the type of query, whether CRM, e-commerce, or order management.
4. **Backend Data Retrieval**: Backend systems return the relevant data (order status, product details, or user information) to the middleware.
5. **Response Generation**: The bot processes this data and generates a response, combining static responses with dynamic content based on retrieved data.
6. **Feedback Loop and Analytics**: Logs the interaction data for further analysis and continuous improvement, adjusting conversation flows and updating NLP models as needed.

3. Technology Stack

- **NLP Platform**: Google Dialogflow or IBM Watson for NLP and machine learning.
- **Middleware/API Gateway**: AWS API Gateway or Kong.
- **Backend Integration**: Salesforce (CRM), TE Connectivity's e-commerce platform, and SQL/NoSQL databases (PostgreSQL, MongoDB).
- **Cloud Infrastructure**: AWS or Azure for hosting, auto-scaling, and load balancing.
- **Containerization and Orchestration**: Docker for containerization, Kubernetes for orchestration.
- **Monitoring and Logging**: AWS CloudWatch, Grafana, and PagerDuty for system monitoring, with Elasticsearch and Kibana for logs.

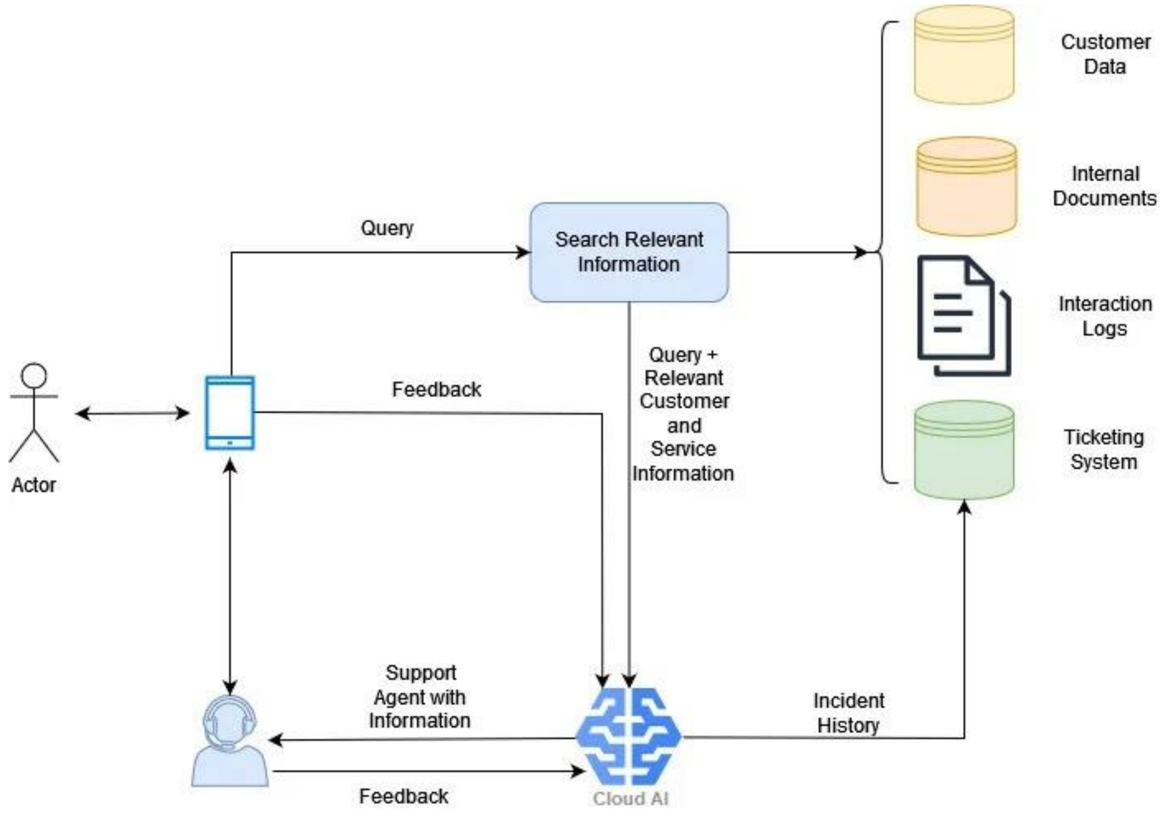


Figure 6 : Architecture

6. REFERENCE CODE

Step 1: Set Up the Environment

Make sure you have Python installed. You can create a virtual environment and install Flask:
pip install Flask

DOI 10.17148/IJARCCE.2020.9233

Step 2: Create the Flask Application

Create a file named chatbot.py and add the following code:

```
from flask import Flask, request, jsonify
app = Flask(__name__)
# Sample responses for different intents
responses = {
    "greeting": [
        "Hello! How can I assist you today?",
        "Hi there! What can I do for you?",
        "Greetings! How may I help you?"
    ],
    "farewell": [
        "Goodbye! Have a great day!",
        "See you later! Take care!",
        "Bye! Feel free to reach out anytime!"
    ],
    "thanks": [
        "You're welcome!",
        "Glad to help!",
        "Anytime!"
    ],
    "unknown": [
        "I'm sorry, I didn't understand that.",
        "Can you please rephrase?",
        "I'm not sure how to respond to that."
    ]
}

def get_intent(user_input):
    # Simple intent recognition based on keywords
    user_input = user_input.lower()
    if any(greeting in user_input for greeting in ["hi", "hello", "hey"]):
        return "greeting"
    elif any(farewell in user_input for farewell in ["bye", "goodbye", "see you"]):
        return "farewell"
    elif "thank" in user_input:
        return "thanks"
    else:
        return "unknown"

@app.route('/chat', methods=['POST'])
def chat():
    user_input = request.json.get('message')
    intent = get_intent(user_input)
    response = responses[intent]
    return jsonify({"response": response})

if __name__ == '__main__':
    app.run(debug=True)
```

Step 3: Run the Flask Application

You can run the application by executing the following command in your terminal:

```
python chatbot.py
```

This starts the Flask server on http://127.0.0.1:5000.

Step 4: Test the Chatbot

You can test the chatbot using curl or a tool like Postman. Here's an example with curl:

```
curl -X POST http://127.0.0.1:5000/chat -H "Content-Type: application/json" -d "{\"message\": \"Hello!\"}"
```

This should return a JSON response similar to:

```
{"response": ["Hello! How can I assist you today?", "Hi there! What can I do for you?", "Greetings! How may I help you?"]}
```

7. CONCLUSION

This study concludes that AI chatbots have significant potential to enhance customer service and improve organizational operational efficiency. However, their successful deployment necessitates meticulous planning and careful consideration of various parameters. Key challenges include the complexity of system integration and the need to establish clear project objectives and key performance indicators (KPIs).

Feedback from participants indicates high levels of satisfaction with the features provided by chatbots, particularly regarding the accuracy of natural language processing (NLP) and the seamless integration within existing systems. This highlights the importance of continuous improvement and adaptation of chatbot capabilities to meet the evolving needs of businesses.

The findings of this study provide a robust framework for organizations looking to effectively utilize chatbot technology. By leveraging this framework, organizations can facilitate improved business outcomes and enhance customer experiences. Implementing this architecture enables the creation of a resilient chatbot that effectively addresses user needs while remaining adaptable over time.

REFERENCES

[1]. Bozic, J., Tazl, O. A., & Wotawa, F. (2019, April). Chatbot testing using AI planning. In 2019 IEEE International Conference On Artificial Intelligence Testing (AITest) (pp. 37-44). IEEE.

[2]. Cahn, J. (2017). CHATBOT: Architecture, design, & development. University of Pennsylvania School of Engineering and Applied Science Department of Computer and Information Science.

[3]. Cīrule, D., & Bērziša, S. (2019). Use of chatbots in project management. In Information and Software Technologies: 25th International Conference, ICIST 2019, Vilnius, Lithuania, October 10–12, 2019, Proceedings 25 (pp. 33-43). Springer International Publishing.

[4]. Damnjanovic, V. (2019, September). Entry market strategy for weaver chatbot using the digital B2B model. In 2019 International Conference on Artificial Intelligence: Applications and Innovations (IC-AIAI) (pp. 40-403). IEEE.

[5]. Deshpande, A., Shahane, A., Gadre, D., Deshpande, M., & Joshi, P. M. (2017). A survey of various chatbot implementation techniques. International Journal of Computer Engineering and Applications, 11(7).

[6]. Heo, M., & Lee, K. J. (2018). Chatbot as a new business communication tool: The case of naver talktalk. Business Communication Research and Practice, 1(1), 41-45.

[7]. Lakshmi, K. N., Reddy, Y. K., Kireeti, M., Swathi, T., & Ismail, M. (2019). Design and implementation of student chat bot using AIML and LSA. International Journal of Innovative Technology and Exploring Engineering, 8(6), 1742-1746.

[8]. Lokman, A. S., & Ameedeen, M. A. (2019). Modern chatbot systems: A technical review. In Proceedings of the Future Technologies Conference (FTC) 2018: Volume 2 (pp. 1012-1023). Springer International Publishing.

[9]. Nadarzynski, T., Miles, O., Cowie, A., & Ridge, D. (2019). Acceptability of artificial intelligence (AI)-led chatbot services in healthcare: A mixed-methods study. Digital health, 5, 2055207619871808.

[10]. Nuruzzaman, M., & Hussain, O. K. (2018, October). A survey on chatbot implementation in customer service industry through deep neural networks. In 2018 IEEE 15th international conference on e-business engineering (ICEBE) (pp. 54-61). IEEE.

[11]. Owoc, M. L., Sawicka, A., & Weichbroth, P. (2019, August). Artificial intelligence technologies in education: benefits, challenges and strategies of implementation. In IFIP International Workshop on Artificial Intelligence for Knowledge Management (pp. 37-58). Cham: Springer International Publishing.

[12]. Sannikova, S. (2018). Chatbot implementation with Microsoft bot framework.

[13]. Valério, F. A. M., Guimarães, T. G., Prates, R. O., & Candello, H. (2018). Chatbots Explain Themselves: Designers' Strategies for Conveying Chatbot Features to Users. Journal on Interactive Systems, 9(3).

[14]. Wei, C., Yu, Z., & Fong, S. (2018, February). How to build a chatbot: chatbot framework and its capabilities. In Proceedings of the 2018 10th international conference on machine learning and computing (pp. 369-373).

[15]. Yan, R. (2018, July). " Chitty-Chitty-Chat Bot": Deep Learning for Conversational AI. In IJCAI (Vol. 18, pp. 5520-5526).

DOI 10.17148/IJARCCE.2020.9233