

A characterization of chatbot development platforms for deep-logic operations

Ensieh Modiridovom; Department of Electrical & Computer Engineering, The University of Texas at San Antonio, Texas,
David Akopian; Department of Electrical & Computer Engineering, The University of Texas at San Antonio, Texas, USA

Abstract

Today the use of chatbots has proliferated across various sectors and applications, significantly enhancing customer interaction and satisfaction through real-time communication. However, there remains a critical need to explore further advancements in their development. With the progress in Natural Language Processing (NLP) and Natural Language Understanding (NLU), several major platforms—such as Google, Amazon, and IBM—have introduced a variety of tools and features for chatbot creation. In this paper, we will conduct a comparative analysis of representative chatbot development platforms, and provide some extension capabilities in the context of time-persistent(deep-logic) chatbot capabilities. All the state-of-the-art task-oriented chatbot platforms focus on facilitating connection to multiple messaging channels such as Facebook Messenger, Instagram, WhatsApp, Slack and SMS. They provide user-friendly interfaces for chatbot creation and automation. Still, the operation of long conversations, often referred to as deep-logic, brings additional challenges that are not typically addresses by many existing systems. The paper aims to provide insights into the strengths and limitations of each platform, ultimately contributing to the ongoing development of more effective and intelligent chatbots.

1. Introduction

Chatbots have emerged as indispensable tools in modern human-computer interaction, offering intelligent interfaces powered by advanced artificial intelligence (AI) technologies such as Natural Language Processing (NLP) and Natural Language Understanding (NLU). These systems simulate human conversations across various platforms, including websites and mobile applications, providing 24/7 virtual customer service. At their core, chatbots function as interactive agents designed to facilitate seamless question-and-answer exchanges, significantly enhancing user engagement and operational efficiency [1].

One of the most critical aspects of chatbot technology is its ability to comprehend and process user dialogues effectively. Since the early stages of chatbot development, engineers have focused on creating robust algorithms capable of handling diverse sentence structures, including informal and grammatically incorrect inputs. NLU has become a cornerstone of this effort, enabling chatbots to decode user intent, process natural language, and generate meaningful, context-aware responses. This capability has elevated chatbot performance in domains such as customer support, e-commerce, and virtual assistance, delivering richer and more intuitive user interactions [2].

Natural Language Understanding (NLU), a pivotal subdomain of NLP, equips machines to interpret the meaning, context, and intent behind spoken or written input. By bridging

the gap between human communication and machine comprehension, NLU facilitates fluid, human-like interactions in chatbot systems. Complementing NLU, NLP encompasses a broader scope, focusing on developing algorithms that enable machines to both understand and generate natural language, serving as the foundation for AI-driven conversational systems [3].

In this paper, we present a comparative analysis of various chatbot development platforms, with a specific focus on deep-logic chatbot capabilities. We explore two primary categories of platforms. The first category leverages machine learning (ML)-based methods to develop conversational dialogs, exemplified by platforms such as Google DialogFlow [5] and Amazon Lex [6]. These platforms provide extensive libraries for ML/NLP integration and a suite of development tools. The second category focuses on deterministic dialog trees supplemented by ML/NLP for enhancing user interactions, with systems like ManyChat [8] and Chatfuel [9] as representative examples. These platforms excel in creating structured conversations, deploying chatbots across multiple channels, and offering intuitive interface for automation.

All of these platforms aim to connect seamlessly with various messaging channels, including Facebook Messenger, Instagram, WhatsApp, Slack, and SMS. However, despite their capabilities, challenges persist—particularly with handling long, complex conversations or "deep-logic" scenarios. Issues such as session expiration disrupt user interactions, a problem commonly encountered in platforms like Google DialogFlow and Amazon Lex. This paper addresses these challenges by exploring strategies to ensure session continuity and improve chatbot usability.

2. Related works

As chatbot technology continues to evolve, development platforms have adopted distinct approaches to cater to diverse business needs. Task-oriented chatbots generally fall into two categories:

- ML-Focused Conversational Systems rely on machine learning to analyze large conversational datasets, predict user intents, and adapt responses dynamically.
- Dialog Tree Systems with ML/NLP Integration use structured frameworks to guide conversations along predefined paths, incorporating AI techniques to enhance user interactions. While these systems have revolutionized conversational AI, limitations such as session time restrictions remain a significant concern, disrupting the continuity of user interactions. For instance, platforms like Google DialogFlow and Amazon Lex enforce session time limits that require users to restart conversations.

This paper investigates these limitations, comparing two platforms from each category, and proposes an approach to maintain session continuity through conversation summaries.

The remainder of this paper is organized as follows: Section 2 reviews the state-of-the-art research on chatbot comparisons. Section 3 explores two categories of chatbots and their associated platforms. Section 4 introduces a method for extending session length in Google DialogFlow. Finally, Section 5 concludes with a summary of the findings.

Previous research on chatbots has been limited in terms of direct comparisons, analysis of their limitations, and exploration of possible improvements. Most studies have focused on evaluating different chatbot features to identify their applications and determine the most suitable types for specific use cases. The following section highlights some key contributions in this field.

Dagkoulis et al. [10] studied different Chatbot Development and defined criteria and calculate scores based on requirement assumptions to help people interested in using these platforms decide which is the best one for their case. Furthermore, Srivastava et al. [11] in his study provided a list of desirable features that platforms like Google Dialogflow, IBM Watson and Amazon Lex should exhibit in order to cater to their mixed user base. In another study, López-Morales et al. [12] present a web platform named Asymob which enables the measurement of chatbots using a suite of 20 metrics. The tool features a repository supporting chatbots built with different technologies, like Dialogflow and Rasa. Asymob's metrics help in detecting quality issues and serve to compare chatbots across and within technologies. Peyton et al. [13] also compare intent classification results of two popular chatbot frameworks to a state-of-the-art Sentence BERT (SBERT) model that can be used to build a robust chatbot. Luo et al. [14] in their research review the usability and applications of chatbots for various business sectors and analyze the various computational approaches used to develop state-of-the-art chatbots, but also thoroughly review the usability and applications of chatbots for various business sectors. In this paper, we examine two categories of chatbots, highlight a key limitation of each, and propose a potential solution to address it.

3. State-of-the-art representative task-oriented chatbot development platforms

3.1 ML-focused Systems Category:

Conversational dialogs that exploit machine learning (ML) based associations are rapidly evolving, driven by advancements in AI technologies and natural language processing (NLP). These technologies allow for sophisticated interactions between users and conversational agents, typically found in chatbots and digital assistants.

Machine learning plays a critical role in improving dialog systems. It allows these systems to:

- **Pattern Recognition:** Machine learning algorithms can identify and learn from patterns within large datasets.

This enables a dialog system to comprehend user intents and generate appropriate responses.

- **Adaptive Learning:** Over time, ML-based systems become smarter by learning from user feedback and interaction history. They can continuously refine their language models to improve accuracy and relevance.
- **Dynamic Response Generation:** Instead of relying solely on scripted responses, ML-based conversational agents can generate dynamic replies based on context and previous interactions, making conversations more fluid and natural.

3.1.1. Representative platforms that belong to ML-focused Systems category:

GoogleDialogflow: Dialogflow is a powerful development suite designed for creating virtual agents and conversational interfaces, enabling organizations to deliver interactive customer experiences through advanced AI functionalities, including both deterministic and generative AI. Available in two versions, EX and CX, the CX version offers enhanced capabilities with fewer limitations compared to EX. A key feature of this platform is its capability to develop hybrid conversational agents that integrate deterministic responses with generative AI, enabling more adaptive and context-aware interactions. Additionally, the platform facilitates seamless integration with various communication channels, including web, mobile applications, Google Assistant, and Slack, thereby enhancing its applicability across multiple domains and industry-specific use cases.

The platform's versatility enables businesses to create chatbots for tasks ranging from simple FAQs to more complex interactions that require contextual understanding, such as customer service inquiries, scheduling, and online orders. The development process is simplified through the Dialogflow console, which offers an intuitive interface for defining intents and managing conversations. Additionally, integration with Google Cloud enables seamless deployment and connectivity with messaging services. Dialogflow incorporates features such as multilingual support and performance analytics, which assist in assessing user interactions and optimizing system performance to enhance response accuracy and user interaction. [15].

Limitations of GoogleDialogflow: Google Dialogflow, while powerful, has several limitations that may impact its suitability for certain use cases. Its Natural Language Understanding (NLU) models are managed by Google, allowing limited customization, and it is deeply embedded within the Google Cloud ecosystem, which may limit flexibility and increase dependency on a single provider. The platform struggles with handling complex workflows, multi-turn conversations, and long session durations due to session timeouts. Although it supports multiple languages, its multilingual capabilities are limited, particularly in real-time language switching. Dialogflow's cloud-based nature requires a stable internet connection, making offline applications unfeasible. Moreover, the cost can increase significantly for high-traffic. Although the platform supports multiple languages, its multilingual capabilities remain constrained, particularly in real-time language switching. Its cloud-based architecture necessitates a stable internet connection, rendering offline deployment impractical. Furthermore, the cost may escalate substantially for high-traffic applications, and integration with non-Google ecosystems can present

significant challenges. Leveraging advanced functionalities often demands programming expertise, thereby introducing a steep learning curve for non-technical users. Additionally, its limited capacity for handling complex logical operations and reasoning frequently necessitates reliance on external APIs or custom back-end solutions.

Amazon Lex. Amazon Lex is a fully managed artificial intelligence service that leverages advanced natural language processing (NLP) models to facilitate the development, testing, and deployment of conversational interfaces within applications. By automating key aspects of dialogue management, Lex streamlines the interface design process, enabling organizations to enhance user engagement and interaction efficiency. The implementation of such automation significantly enhances user experience and satisfaction by enabling organizations to deploy conversational systems capable of effectively addressing user needs with high levels of responsiveness and adaptability. Among the available tools, Amazon Lex stands out as a prominent platform, with its Version 2 (V2) offering advanced functionalities and increased flexibility compared to the earlier iteration. These improvements facilitate more robust and scalable conversational interfaces, underscoring the critical role of evolving technologies in meeting the growing demands for efficient and user-centric digital interactions. Amazon Lex V2 introduces advanced features that extend the platform's capabilities, making it more accessible for developers with varying skill levels. The version supports conditional branching, which allows for the design of complex conversation flows without requiring deep learning expertise or extensive coding knowledge. A significant advantage of Amazon Lex V2 is its seamless integration with other AWS services, such as AWS Lambda, Amazon Comprehend, and Amazon Kendra, enabling bots to access pre-built connectors for smooth data functionality. Furthermore, Amazon Lex V2 demonstrates exceptional scalability, effectively eliminating the necessity for developers to manage underlying infrastructure complexities. This scalability enables developers to allocate their efforts toward optimizing and enhancing the functional capabilities of conversational bots, thereby fostering innovation and improving the overall quality of user interactions. Such architectural advancements highlight the platform's capacity to support large-scale deployments while maintaining operational efficiency, which is critical for meeting the dynamic demands of modern digital ecosystems. The platform utilizes a pay-as-you-go pricing model, where users are charged solely for the text or speech requests processed. This pricing structure, in conjunction with the AWS Free Tier, provides a cost-effective framework for organizations of different sizes to explore and implement conversational AI technologies without substantial upfront costs. Such a model reduces entry barriers for businesses and

encourages the wider adoption and experimentation with AI-driven solutions, promoting innovation and scalability in the development of conversational systems. [16].

Limitations of Amazon Lex: Amazon Lex has notable limitations despite its robust features. It requires a solid understanding of AWS, making it less accessible to non-technical users. Its capabilities for managing intricate, multi-step conversations and advanced logic are limited without substantial coding effort. The platform's multilingual support is basic, lacking flexibility for dynamic or mixed-language interactions. Integration challenges arise when connecting with non-AWS services, and it depends entirely on internet connectivity, ruling out offline usage. Additionally, session management is constrained for extended conversations, and the costs can escalate for applications with high usage, potentially impacting budget-conscious projects. A comparative analysis of the two platforms, assessing their distinct features and capabilities. The results of this comparison are shown in Table 1.

3.2. Dialog Trees focused category:

Deterministic dialog trees are a structured approach used to effectively manage conversational dialogs by defining clear paths based on user inputs. These trees facilitate human-computer interaction by enabling a predetermined dialogue flow, minimizing ambiguity, and ensuring that users navigate through coherent and contextually relevant exchanges. By employing these dialog structures, systems can provide a more intuitive experience, as users receive predictable responses depending on their prior inputs, effectively reducing confusion and enhancing engagement.

3.2.1. Representative platforms that belong to Dialog Trees focused category:

ManyChat ManyChat represents an innovative chat marketing platform designed to automate and streamline customer interactions across widely used messaging applications, including Instagram, WhatsApp, and Facebook Messenger. By leveraging advanced automation capabilities, the platform facilitates efficient and personalized communication, enhancing user engagement and optimizing customer relationship management in digital environments. By leveraging advanced automation techniques, it helps businesses enhance engagement, boost sales, and improve conversion rates. The platform provides a comprehensive set of functionalities, including automated responses to frequently asked inquiries, real-time interaction with messages, reactions, and mentions, as well as data collection capabilities for contact information.

Table 1: Different features of Google Dialogflow and Amazon Lex

Feature	Google Dialogflow	Amazon Lex
Platform Type	A chatbot development platform supporting multiple communication channels	Service dedicated to developing conversational interfaces for specific applications
Capabilities	Accommodates a wide range of applications	Focused on specific conversational applications
NLP Approach	Combines machine learning and rule-based systems	Relies on machine learning for input classification
Pricing Model	A multi-tiered structure offering varying levels of complimentary usage	A usage-based model calculated on the volume of requests
Integration	Facilitated integrations with Google Cloud and messaging platforms	Facilitated integration with AWS services for custom logic
Language Support	Supports 123 languages	Currently supports 18 languages
Machine Learning	Custom model creation for flexibility	Pre-built models for specific domain tasks
Development Environment	Includes web GUI, CLI, and APIs	Web-based GUI for building interfaces
NLP Features	Advanced natural language understanding (NLU) features, including sentiment analysis	Basic NLU features, less robust than Dialogflow
User Interface	Visually intuitive and user-friendly interface	Features an intuitive focused on easy navigation

These features facilitate user engagement and information retention for future interactions. Additionally, users can configure workflows using predefined templates or AI-driven assistants, contributing to the enhancement of operational efficiency in customer service and communication management.

The implementation of ManyChat provides several advantages, including potential improvements in conversion rates, cost reductions through decreased customer support expenditures, and scalability to accommodate increasing demands as an organization expands. With widespread adoption by businesses globally, ManyChat has become a recognized tool in the realm of chat-based marketing, demonstrating both reliability and effectiveness in enhancing customer interactions and optimizing marketing strategies.

Limitations of ManyChat: ManyChat presents several limitations that may affect its applicability in certain contexts. The platform offers restricted advanced customization options, which can hinder the ability to manage complex workflows or incorporate sophisticated logic without the integration of additional tools or external coding. The platform's functionality is closely tied to social media channels such as Facebook Messenger, Instagram, and WhatsApp, which may limit its adaptability for organizations seeking to extend beyond these platforms. While a free plan is available, it is accompanied by considerable limitations in terms of features and automation capabilities. ManyChat also faces challenges in scaling for large enterprises and offers limited support for multilingual interactions, which may reduce its effectiveness for businesses with a global reach. Furthermore, the platform

lacks voice interaction capabilities, advanced AI or machine learning features, and flexibility in integrating with specialized third-party applications. Additionally, changes in the policies of social media platforms can have a direct impact on the platform's functionality.

Chatfuel. Chatfuel is a dynamic platform designed to create AI chatbots tailored for automated sales and enhanced customer engagement, leveraging its strong partnership with Meta. This cutting-edge tool enables businesses to build chatbots that seamlessly interact with customers, often replacing human agents while maintaining a natural conversational flow. Chatfuel's chatbots learn quickly, adapt to a wide range of customer requests, and support functionalities such as booking appointments, qualifying leads, responding to customer comments, and offering product recommendations. The platform boasts features that streamline customer interaction and enhance business efficiency. Chatfuel facilitates efficient query resolution by enabling rapid response times. Its scalability feature allows multiple agents to manage conversations through a single WhatsApp business number, ensuring consistent service quality and improved response efficiency. Additionally, the integration of ChatGPT technology enhances the platform's capability to provide interactions that closely resemble human communication, contributing to more engaging customer experiences and potentially improving sales outcomes. Integrating Chatfuel into sales and marketing strategies can lead to improvements in conversion rates and customer retention. The platform supports upselling and cross-selling techniques in a manner that does not disrupt the user

experience, ensuring a smooth interaction throughout the process. Through the use of automated messaging campaigns, businesses can develop targeted marketing approaches that respond to customer behavior, such as inactivity or cancellations, thereby enhancing retention efforts. Furthermore, Chatfuel's capacity to collect and analyze customer feedback provides businesses with valuable insights for the continuous refinement of their offerings.

Limitations of Chatfuel: Chatfuel, while widely used for building chatbots, has several limitations. It offers a constrained set of advanced features and customization options, particularly for managing complex workflows and intricate logic, which may necessitate the use of external APIs or custom coding. The platform's reliance on Facebook Messenger limits its flexibility for businesses seeking cross-platform integration or broader audience reach. While it provides an intuitive interface, the free plan imposes notable restrictions on features and automation, limiting its applicability for large-scale implementations. Furthermore, Chatfuel lacks advanced AI or machine learning capabilities, hindering its ability to handle complex user queries or adapt based on previous interactions. Additionally, its multilingual support is limited, making it less suitable for businesses targeting diverse global markets. The features of ManyChat and Chatfuel are compared in Table 2 [17].

4. Extending Dialogflow Session Length: Enhancing Chatbot Interactions for deep-logic operations

Chatbots like Google Dialogflow are designed for short, task-oriented interactions, such as ordering a meal or retrieving weather updates. These conversations typically last less than

24 hours, aligning with Dialogflow's default session duration of 20 minutes. Once this limit is reached, a new session begins, and all previous conversation data is discarded. If users attempt to continue their dialogue after the session timeout, the chatbot may default to fallback intents, offering generic responses like "Could you please repeat that?" or "I don't know what you mean." This limitation, common across many chatbot platforms, disrupts user experiences and can lead to the loss of essential conversational context [16].

To address this challenge, we implemented a solution that extends session functionality while maintaining conversational continuity. By leveraging Google's methods and enhancing them with a summarization model, we generate a brief recap of prior interactions. This summary reminds users of their earlier exchanges and the chatbot's responses, enabling a seamless transition into continued dialogue. This method significantly improves the user experience in scenarios requiring longer conversations, such as customer support or complex queries.

The steps of this method, originally developed by Google, are illustrated in Diagram 1. Additionally, we have incorporated a summarization step, as mentioned earlier. This diagram illustrates the extended workflow of integrating Dialogflow with a custom service to manage user interactions. It shows the process of handling user input, detecting intents, performing actions, and maintaining conversation continuity through a summarization model, ensuring a seamless and context-aware user experience [19]. The enhanced workflow as it is shown in Figure 1 includes the following steps:

1. **User Input:** The user enters or speaks a message.
2. **Detect Intent Request:** The service sends the message to Dialogflow via a DetectIntent API call.

Table 2: Different features of ManyChat and Chatfuel

Feature	Chatfuel	ManyChat
Ease of Use	The Flow Builder is intuitive but can lag with complex automations.	Intuitive drag-and-drop interface
Available Channels	Facebook, Instagram, WhatsApp	Facebook, Instagram, Telegram, SMS, Email
Live Chat Functionality	Basic live chat capabilities	Advanced live chat system with team functions
Analytics	Focus on unanswered questions	Includes detailed analysis of conversion tracking
AI Features	Integration with ChatGPT, includes AI Agents	AI Flow Builder Assistant and AI Text Improver
Contact Management	Includes basic functionality for managing custom fields	Enables the management of unlimited custom fields and tags
Growth Tools	Message templates for Instagram and WhatsApp	Banners, pop-ups, and additional growth tools
API and Integrations	Direct integrations but lacks an App Store	Own App Store with over 61 applications

3. **Intent Detection:** Dialogflow processes the request and identifies the matched intent, returning details like actions, parameters, and replies.
4. **Chat History:** A brief summary of prior interactions is generated and presented to the user to ensure continuity.
5. **Task Execution:** The service performs necessary tasks, such as querying a database or making API calls.
6. **Response Delivery:** The processed response is sent back to the user.
7. **User Interaction:** The user views or hears the chatbot's reply, benefiting from a seamless and context-aware conversation.

4.1. Implementing Session Persistence for this method:

To effectively implement these steps, a cloud environment is needed for configuring and integrating with Dialogflow. The process begins by creating a project on a cloud platform, such as Google Cloud. Once the project is established, the Dialogflow API must be enabled, and an agent should be set up. Secure communication with the platform's API is facilitated by generating the required authentication credentials.

The next stage involves developing a microservice that interfaces with Dialogflow and manages session persistence. This microservice plays a pivotal role in the system and includes several core components: an API endpoint for accepting user inputs, including session IDs and messages; integration with Dialogflow's DetectIntent API to perform

intent detection and retrieve responses; a persistence layer for storing session contexts in a database to maintain conversational continuity across requests; and a response generator to process the API responses and deliver enriched, context-aware outputs to users.

To ensure seamless conversation flow, session contexts are stored in a database. The choice of database depends on the specific needs of the application. Several options are available including:

- **Document-Based Databases:** like MongoDB which is ideal for storing session contexts as flexible JSON-like documents, making it suitable for dynamic data structures. CouchDB is another option with similar benefits, along with built-in replication features for distributed systems.
- **Key-Value Stores:** like Redis which offers fast, in-memory storage with support for expiration, making it suitable for short-lived sessions. Amazon DynamoDB is another one in this category, a serverless and scalable solution, which is highly available and ideal for key-value storage.
- **Relational Databases:** MySQL and PostgreSQL provide structured storage with tables and relationships, making them ideal for applications that require strict schema management. SQLite, a lightweight solution, is suitable for smaller-scale projects that need local storage.
- **Graph Databases:** Neo4j is suitable for managing complex relationships, such as interconnected or hierarchical session contexts.

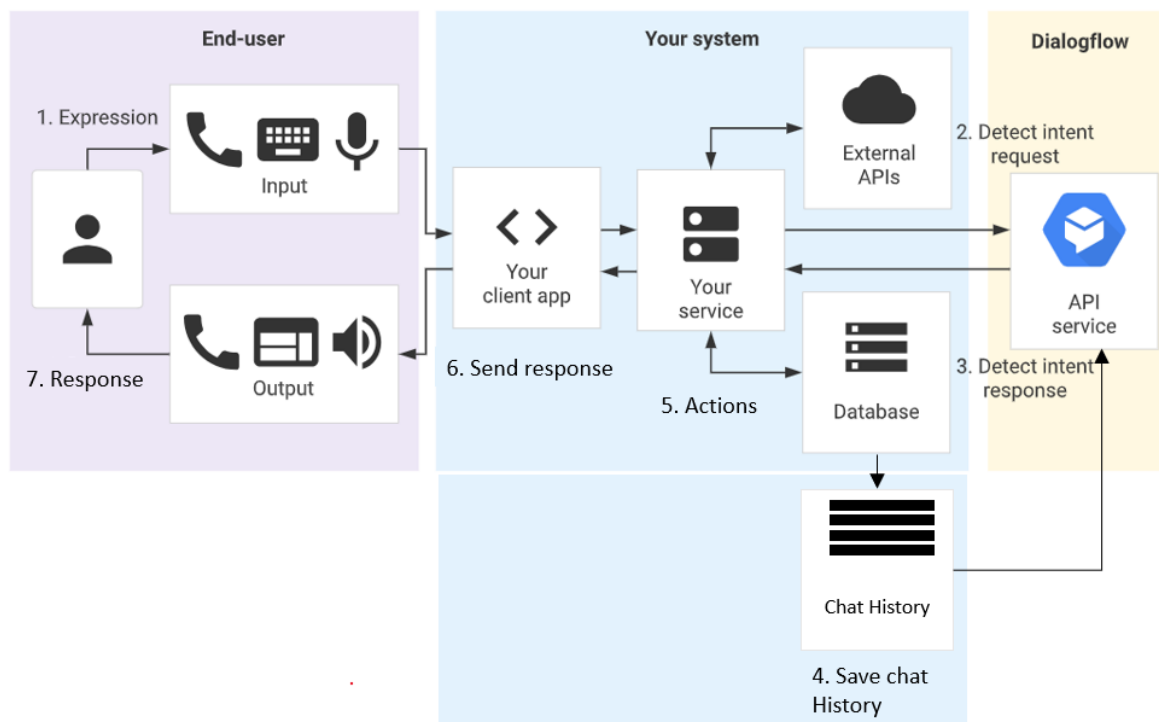


Figure 1: Dialogflow Extended Session Flow

- **Time-Series Databases:** Influx DB is ideal for tracking session-related events with timestamps, especially when historical data analysis is required.
- **Cloud-Based NoSQL Databases:** Firebase Realtime Database and Firestore integrate seamlessly with Dialogflow, optimized for real-time updates. Google Cloud Datastore is scalable, user-friendly, and particularly well-suited for applications hosted on Google Cloud.

For this implementation, MongoDB was chosen due to its document-oriented design, which aligns well with Dialogflow's dynamic and hierarchical data structures. MongoDB's flexibility allows for efficient storage and retrieval of session contexts, making it an optimal choice for this use case. The following pseudocode illustrates how Dialogflow can be integrated with MongoDB to manage session context:

Integrating Dialogflow with a database

```
def handle_user_input(session_id, user_message):
    # Step 1: Send message to Dialogflow
    intent_response = dialogflow.detect_intent(session_id,
    user_message)
    # Step 2: Process the detected intent and generate a response
    intent = intent_response['intent']
    parameters = intent_response['parameters']
    response_text = intent_response['response']
    # Step 3: Store session context in MongoDB
    session_context = {
        "session_id": session_id,
        "intent": intent,
        "parameters": parameters,
        "last_response": response_text
    }
    mongodb.collection('session_contexts').update_one(
    {"session_id": session_id,
    {"$set": session_context},
    upsert=True
    )
    # Step 4: Return the response to the user
    return response_text
```

4.2. Operational Steps

1. **Input Handling:** Users send input (e.g., a message) along with a session ID to the microservice.
2. **Context Retrieval:** The system retrieves the session's stored contexts from the database.
3. **Dialogflow Interaction:** Input and contexts are passed to the Dialogflow API, which returns an appropriate response.
4. **Context Update:** The system updates session contexts in the database based on Dialogflow's output.

By implementing this enhanced approach, the user experience in long-duration interactions can be significantly improved, ensuring seamless conversation continuity. This innovation represents a crucial step forward in creating more

sophisticated and engaging chatbot experiences, capable of handling complex, multi-step processes that extend beyond standard session duration.

5. Conclusion

In conclusion, the advancement of chatbot technologies, driven by machine learning (ML) and natural language processing (NLP), has significantly enhanced human-computer interactions across various industries. The comparative analysis of platforms such as Google Dialogflow, Amazon Lex, ManyChat, and Chatfuel reveals the diverse approaches in developing conversational agents. While platforms like Dialogflow and Lex leverage sophisticated ML-based algorithms for dynamic responses, ManyChat and Chatfuel focus on deterministic dialog trees and automation for improved user engagement and marketing effectiveness. Despite their strengths, challenges such as session continuity in long conversations remain a concern. However, by implementing solutions like interacting with the API, the continuity of user interactions can be improved, enhancing the overall user experience. This study contributes valuable insights into the strengths and limitations of these platforms, guiding future developments in chatbot technology to ensure more intelligent, adaptable, and user-friendly systems.

References

- [1] GeeksforGeeks, "10 best chatbot development platforms for conversational AI," 2024. [Online]. Available: <https://www.geeksforgeeks.org/10-best-chatbot-development-platforms-for-conversational-ai/>
- [2] FastBots.ai, "Why NLU is the backbone of modern chatbots," FastBots.ai. [Online]. Available: https://fastbots.ai/blog/why-nlu-is-the-backbone-of-modern-chatbots?utm_source=chatgpt.com. [Accessed: Jan. 24, 2025].
- [3] Zendesk, "What is NLP chatbots and how do they work?" 2024. [Online]. Available: <https://www.zendesk.com/blog/nlp-chatbot/>.
- [4] The Weather Company, "The ultimate guide to machine-learning chatbots and conversational AI," 2022. [Online]. Available: <https://www.weathercompany.com/blog/the-ultimate-guide-to-machine-learning-chatbots-and-conversational-ai/>.
- [5] Google Developers, "Build a Dialogflow CX Google Chat app that understands and...", 2024. [Online]. Available: <https://developers.google.com/workspace/chat/build-dialogflow-chat-app-natural-language>.
- [6] Amazon Web Services, "Amazon Lex - AI Chat Builder," 2022. [Online]. Available: <https://aws.amazon.com/lex/>.
- [7] Y. Sun, "Unsupervised learning of deterministic dialogue structure with...", Semantic Scholar. [Online]. Available: <https://www.semanticscholar.org/paper/Unsupervised-Learning-of-Deterministic-Dialogue-Sun-Shan/f2f984eab9e50a73462d1817f91a66c6dab15d77>.
- [8] ManyChat, "Chat Marketing Made Easy with Manychat," 2024. [Online]. Available: <https://manychat.com/>.
- [9] Chatfuel, "Chatfuel | AI agents for automated sales | Meta's partner," 2024. [Online]. Available: <https://chatfuel.com/>.

- [10] I. Dagkoulis and L. Moussiades, "A comparative evaluation of chatbot development platforms," in Proc. 26th Pan-Hellenic Conf. Informatics, 2023, pp. 293–296. [Online]. Available: <https://doi.org/10.1145/3575879.3576012>.
- [11] S. Srivastava and T. V. Prabhakar, "Desirable features of a chatbot-building platform," in 2020 IEEE Int. Conf. Humanized Comput. Commun. Artif. Intell. (HCCAI), Irvine, CA, USA, 2020, pp. 61–64. [Online]. Available: <https://doi.org/10.1109/HCCAI49649.2020.00016>.
- [12] J. M. López-Morales, P. C. Cañizares, S. Pérez-Soler, E. Guerra, and J. de Lara, "Asymbot: A platform for measuring and clustering chatbots," in Proc. ICSE '22 Companion, 2022, pp. 1–2. [Online]. Available: <https://doi.org/10.1145/3510454.3516843>.
- [13] K. Peyton and S. Unnikrishnan, "A comparison of chatbot platforms with the state-of-the-art sentence BERT for answering online student FAQs," Results Eng., vol. 13, p. 100856, 2022. [Online]. Available: <https://doi.org/10.1016/j.rineng.2022.100856>.
- [14] B. Luo, R. Y. K. Lau, C. Li, and Y.-W. Si, "A critical review of state-of-the-art chatbot designs and applications," Wiley Interdiscip. Rev. Data Min. Knowl. Discovery, vol. 12, no. 1, p. e1434, 2022. [Online]. Available: <https://doi.org/10.1002/widm.1434>.
- [15] Google Developers, "Build a Dialogflow CX Google Chat app that understands and...", 2024. [Online]. Available: <https://developers.google.com/workspace/chat/build-dialogflow-chat-app-natural-language>.
- [16] Amazon Web Services, "What is Amazon Lex V2? - AWS Documentation," 2021. [Online]. Available: <https://docs.aws.amazon.com/lexv2/latest/dg/what-is.html>.
- [17] Chatimize, "Manychat vs Chatfuel: Which platform is better in 2025?," 2024. [Online]. Available: <https://chatimize.com/manychat-vs-chatfuel/>.
- [18] P. Sudhaus, "Extending Dialogflow session length to infinity," Medium. [Online]. Available: <https://medium.com/@p.sudhaus/extending-dialogflow-session-length-to-infinity-8a27fa162db8>. [Accessed: Jan. 23, 2025].
- [19] Google Cloud, "Dialogflow ES API overview," Google Cloud. [Online]. Available: <https://cloud.google.com/dialogflow/es/docs/api-overview>. [Accessed: Jan. 23, 2025].

Author Biography

Ensieh Modiridovom received her BS in Information Technology and MS in computer science from the Azad University of Iran and currently she is a PhD student in Electrical and computer Engineering in University of Texas at San Antonio. Her main research interest is different area of Artificial Intelligence such as Natural Language Processing.

David Akopian received the Ph.D. degree from Tampere University, Finland. He is currently a Professor with The University of Texas at San Antonio (UTSA). Before joining UTSA, he was a Senior Research Engineer and a Specialist with Nokia Corporation. His current research interests include mobile computing, digital signal processing for communication and navigation receivers, positioning, dedicated hardware architectures, and platforms for software-defined radio. He is a fellow of U.S. National Academy of Inventor.

JOIN US AT THE NEXT EI!

electronic IMAGING

Imaging across applications . . . Where industry and academia meet!



- **SHORT COURSES • EXHIBITS • DEMONSTRATION SESSION • PLENARY TALKS •**
- **INTERACTIVE PAPER SESSION • SPECIAL EVENTS • TECHNICAL SESSIONS •**

www.electronicimaging.org

