10.2352/EI.2022.34.3.MOBMU-206

Chatbot Integration with Google Dialogflow Environment for Conversational Intervention

Rahul Mundlamuri, Devasena Inupakutika, Ganesh Reddy Gunnam, Sahak Kaghyan and David Akopian; Department of Electrical and Computer Engineering, University of Texas at San Antonio, San Antonio, USA

Abstract
Chatbots are computer programs that execute protocols for supporting human-machine conversations and perform various functions such as searching in the web, ordering food, making appointments, and many more. To facilitate timely responses and actions, and also enable interactive human-like conversations, chatbots use Natural Language Processing (NLP) to understand user's messages and respond appropriately. NLP is an area of computer science and artificial intelligence that interprets the interactions between computers and human languages. Google Dialogflow is a chatbot development platform example that provides NLP and other supporting tools and third-party interfaces. Many chatbot applications are developed for short-term conversations, but in order to assist medical applications, we need a long-spread conversation. DashMessaging is a smart chatbot platform that enables the creation of chatbots for long-spread conversations, so we try to mimic this long-spread conversation with the frame-based platforms such as Google Dialogflow. In this paper, we explore the feasibility and challenges of using these tools by a DashMessaging chatbot development platform.

Index Terms: *long-spread conversation, chatbots, Dialogflow*

Introduction
A chatbot mimics human conversation in machine-human interactive dialogs. The chatbot architecture combines a linguistic model with computational techniques to simulate basic or advanced natural language chat conversation between a human and a machine. In the present era, chatbots have become quite popular. The core of the chatbot is a dialog management unit, that is responsible for the conversation's state and flow. Paper [1] explored different approaches for dialog management in conversational agents. While they classified these approaches into many categories, we limit the scope of discussion to frame-based in this paper. Additionally, [2] categorized these frame-based approaches as AI-based because majority of the chatbot platforms have built-in natural language processing (NLP) modules [3], [4].

Figure 1 shows the frame-based platforms (AI-based alternatively) such as Google Dialogflow[5], Amazon Lex[6], IBM Watson[7], and Microsoft LUIS[8]. Since these chatbots can use NLP with training data which includes intent, context, and entity [2], in the following section, we will discuss these chatbots briefly. Nowadays, rule-based chatbots usage is more because they work based on a decision-tree model and follow a set of rules. As the name indicates, they act like a flowchart. Rule-based chatbots are vulnerable when it comes to free text-based conversations. So, to resolve this, we need NLP to efficiently analyze the free text. The AI-based chatbot establishes communication with humans utilizing AI technology and underlying built-in computer program. Additionally, these are defined as a particular type of chatbots that are designed for turn-by-turn conversations with human users based on the textual input[9]. Unlike rule-based chatbots, AI-based are more valuable and adopted into several businesses and industries such as customer service, retail, medical, among others. The chatbots may incorporate both knowledge-based and rule-based components, but we follow a simplified terminology.

The integration of NLP to these chatbots enables more human-like conversation. However, many of the current chatbots support only short-term conversation, which will be enough for most applications such as airlines, banking, and retail customers service. Nevertheless, when it comes to medical applications, we need a long-spread conversation to follow up with appointments and medication-related questions[9]. In this paper, we demonstrate a chatbot developed with a platform called DashMessaging[10] which supports long-spread conversation with the help of a scheduler. Further, we propose a methodology to incorporate this long-spread conversation concept into the rule-based case-study DashMessaging chatbot by using a frame-based (AI-based) platform, Google Dialogflow and its NLP service.

The remainder of the paper is structured as follows: The following section discusses the state-of-the-art frame-based chatbot development platforms and draws comparisons based on a set of features that are characteristic of longer conversation-based bots. The subsection further highlights the existing challenges regarding long-spread conversations development in domains such as healthcare. The methodology section first covers the case-study chatbot developed using the frame-based platform and presents the Dialogflow integration with the created chatbot. The results section presents the demo conversation and observations from the integration. Finally, the concluding remarks are presented along with future directions of the current work.

State of the art
As mentioned above, Google Dialogflow, Amazon Lex, IBM Watson, and Microsoft LUIS are the popular frame-based chatbot development platforms available in the modern era because of the built-in NLP service as shown in Table 1. NLP is the study of extracting purpose and relevant information from the user's text, and it uses intents and entities. Intents are utterances that learn to interpret what the user might say into a category that can reflect an action. In order to take advantage of specific textual details, the aforementioned frame-based platforms provide a way to define entities that will represent this information in the user's text.

Table 1: Longer multi-turn conversation-based chatbot features

Features	DashMessaging bot	Bot with Dialogflow	Bot with Amazon Lex	Bot with IBM Watson	Bot with Microsoft LUIS
Persist user state between messages over daily sessions	Yes	Not a native feature	Not a native feature	Not a native feature	Not a native feature
Referring back to past day messages etc. using interaction variables	Yes	Not a native feature	Not a native feature	Not a native feature	Not a native feature
Initiating new/ next sessions at specific times	Yes	Not a native feature	Not a native feature	Not a native feature	Not a native feature
Natural Language Processing Services	Custom integration via microservices	Yes	Yes	Yes	Yes
Expose NLP services as microservice	No	Yes	Yes	No	Yes
Proactive Messages (Bot initiating conversation or starting after a pause)	Yes (consists as part of a scheduler microservice)	Yes (consists as part of a scheduler microservice)	Yes (consists as part of a scheduler microservice)	Not a native feature	Not a native feature
Languages	Bilingual	Multiple	Only English	Multiple	Multiple

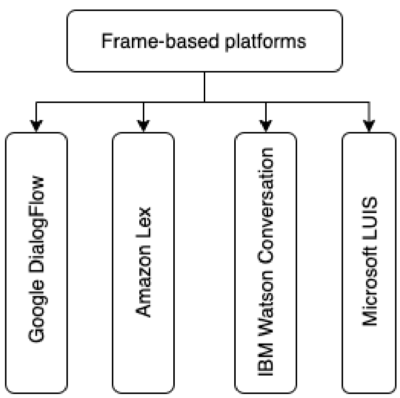

Figure 1: Classification of platforms considered in the scope of this work

Machine learning or deep learning models are used to recognize those entities [11]. Dialogflow, IBM Watson Assistant, and LUIS Bot services support intents and entities. On the other hand, Amazon Lex allows users to use standard built-in intents and slot types, where a slot type is a set of values used by Amazon Lex to train its machine learning model to detect slot values. Except IBM Watson, the remaining platforms allow their NLP services as microservice to support third-party chatbot platforms like DashMessaging.

All these platforms provide integration services with popular messaging channels like Facebook Messenger, Twilio, and SMS and support multiple languages to chat except Amazon Lex, which supports only English for now. However, most frame-based platforms are designed for small conversations because they were designed for business purposes. For implementing long-spread or bot initiating conversations, users need high-level programming skills to design a scheduler and deploy it as a microservice. Next, we discuss the deployment challenges in longer conversation chatbots and the rationale behind the choice of Dialogflow as a development platform for the same in this work.

Challenges in longer conversation chatbots

The demand for chatbots has increased exponentially in the healthcare for setting up an appointment, prescription refills, motivational messages, and detecting the type of illness with symptoms. However, healthcare-specific chatbots require long-spread conversations with participants (e.g. behavioral intervention studies). Such chatbots typically have bidirectional chat initiations for more patient engagement and disease management through the tracking of symptoms [12]. Not all the chatbots are bidirectional (chat initiation from both machine- and human-end), nor do they support long-spread protocols (typically spread over weeks/ months). Even the existing frameworks summarized in the previous section, do not allow a current user session state to extend beyond 24 hours, thus resetting the current user state in the conversation flow. The need for preserving the user state is required for the type of conversations in this work. To cater to these challenges, we present a method, that uses google Dialogflow NLP service with chatbot created utilizing the DashMessaging platform [10] to support longer conversations (through the delay or scheduling like service) and recognizes users' free text (with Dialogflow's built-in NLP).

Dialogflow constitutes NLP capabilities that enhance rule-based dialogs and enables easier integration. Thus, this work utilizes Dialogflow to enable the rule-based chatbot spinned from the DashMessaging platform, to recognize users' free text more efficiently. This integration is made possible in Dialogflow through the features such as webhook and fulfillment for easier deployment of customized intent recognition logic. Table 1 presents the features of rule-based bot with DashMessaging and Dialogflow respectively. Thus, the desired features of a case-study chatbot presented in this paper are achieved with the combination of best of the both frameworks.

Methodology: Dialogflow Integration

This section presents the process of integrating Dialogflow's NLP service with a case-study rule-based conversation management unit that is developed using DashMessaging platform.

Case-study chatbot with rule-based engine (DashMessaging)

As mentioned earlier, we used DashMessaging, a chatbot platform service developed at the SCNS lab in UTSA. DashMessaging platform was designed for long-spread conversations, and unlike existing chatbot platforms in the market, designing a long-spread is easy and straightforward with DashMessaging platform. In DashMessaging, the conversation will be triggered by keywords from the participants. From there, the conversation flow follows respective branch based on participant's responses. The key feature in the proposed platform is timing element. The timing element is for initiating a message to be sent to users or continue the dialog even after a long pause (e.g. a day or 2 later) which is typical in daily human conversations. As shown in figure 2, we designed a simple three-day medical appointment protocol for the case-study integration with Dialogflow.

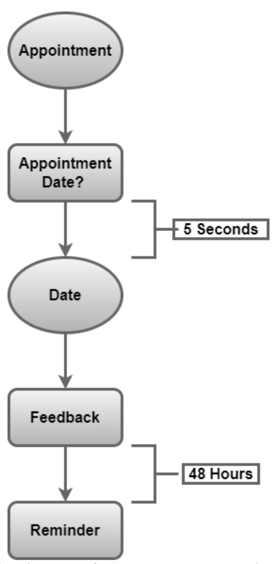

Figure 2: Medical appointment protocol implemented with DashMessaging platform

In Figure.2, the oval shape represents the keywords and responses from the participant, and rectangles represent actions from DashMessaging platform. Since the proposed chatbot is rule-based, the responses from the participant should match precisely in the chatbot system to trigger the action. So, we utilize the NLP service provided by the Dialogflow to determine the precise keyword from the participant response.

Chatbot Integration with Dialogflow

Dialogflow has built-in natural language understanding module with training phrases provided for each intent (keywords for a conversational sequence), uses context conditions for each intent and the corresponding responses to be sent to users. This way it facilitates the development of conversational interfaces.

Dialogflow has three main natural language understanding(NLU) module specific concepts that are constituent of a typical communication in a chatbot: 1) Intents: It is a specific action that is invoked through the use of sentences matching the NLU model. A set of training examples are provided for each intent on which an agent is trained. Thus, based on the user's message, the agent maps it to a specific intent for a corresponding response from the chatbot. Each intent is in a way a dialog turn of the conversation. 2) Entities: Terms corresponding to the intents. These provide a specific context for the intents. These are the keywords for identifying and extracting valuable information from user inputs. Dialogflow consists of various predefined system entities, such as dates, times, cities, colors, or units of measure. However, it lets developers define custom entities (with type and a value). 3) Contexts refers to the current state of the interaction. They carry information between intents and can be combined to direct the conversation, which defines conditions required to access an intent i.e. input contexts or conditions defined after accessing them e.g. output contexts. The fulfillment functionality of the Dialogflow makes it possible to connect natural language understanding and processing for each intent (oval shape content in figure 2) to a business logic, such as accessing third-party APIs, querying databases or using machine-learning-based models to predict apt response with the context provided. We utilize this functionality to integrate our case-study rule-based chatbot for more human-like conversations.

Figure 3 shows the integrated architecture highlighting the message communication sequence at two different dialog states (times). As it is observed, the architecture enables integration of external and customized services to facilitate the interaction between Dialogflow's NLU and dialog management of the rule-based DashMessaging chatbot. Firebase cloud functions are utilized to implement customized actions for the identified intents. As opposed to a decision tree-based model, intents are not used to retrieve a predefined response, but to extract the context that is fed to the NLU model with the appropriate dialog history. Cloud function takes the context to obtain the current conversation state from the previous interaction one with the dedicated user (user-state tracking in longer conversations). A separate scheduler service handles the timing of the messages at selected times in the long-spread conversations, thus engaging them in any follow-up dialogs. Furthermore, Dialogflow stores the entire chatbot structure [4] (with intents (including follow-ups), entities, and contexts) as a standard javascript object notation (JSON) format for cross-framework portability.

Results

This section presents the results of validation of the successful integration of a cloud-based chatbot framework (Dialogflow). We trained Dialogflow's NLP module by defining the set of intents, entities, and parameters for the case study. We defined 7 intents (such as *DiseaseIntensity, DiseaseIntensity-span,DiseaseIntensity-TobaccoCraving, DiseaseIntensity-Medicine, DiseaseIntensity-Suggestion, AppointmentTime, among others*) in total, each related to a specific request or message from the user in order to book a medical appointment for execessive smoking symptoms. For each of the different intents, specific handlers were implemented for the cloud function. Based on the previously described architecture in figure 3, the first step

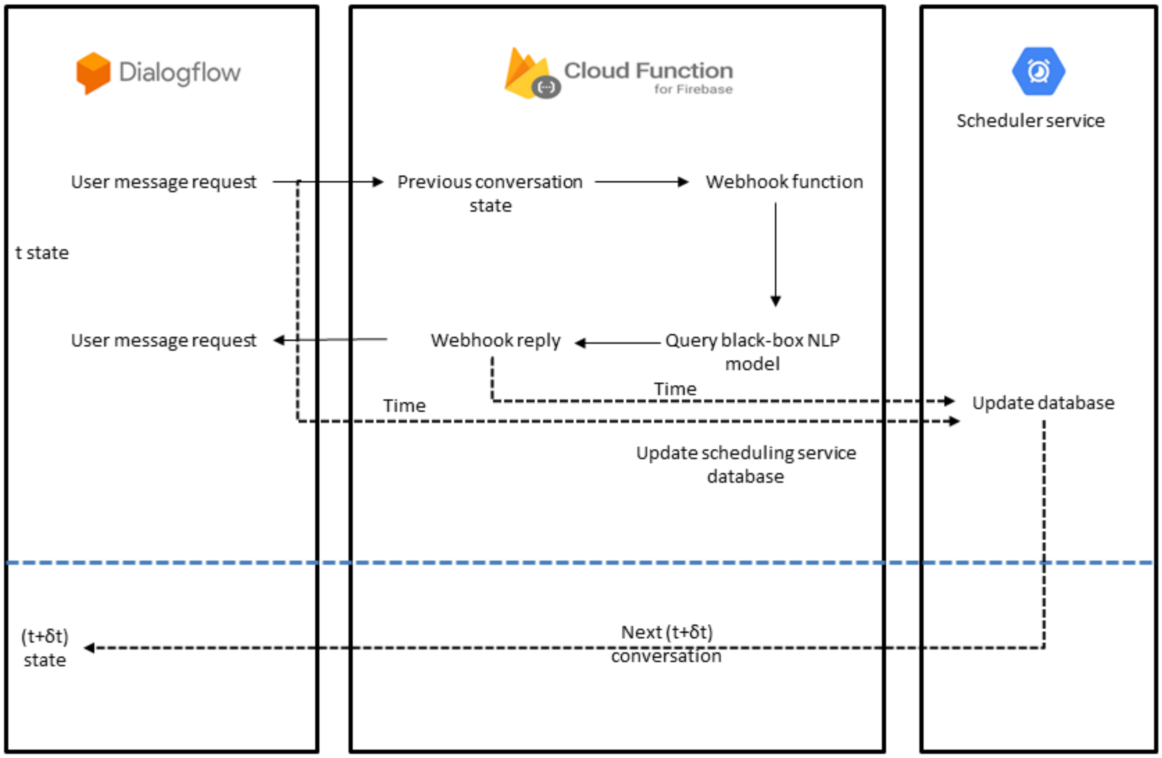

Figure 3: Rule-based chatbot and Dialogflow Integrated Architecture and Communication

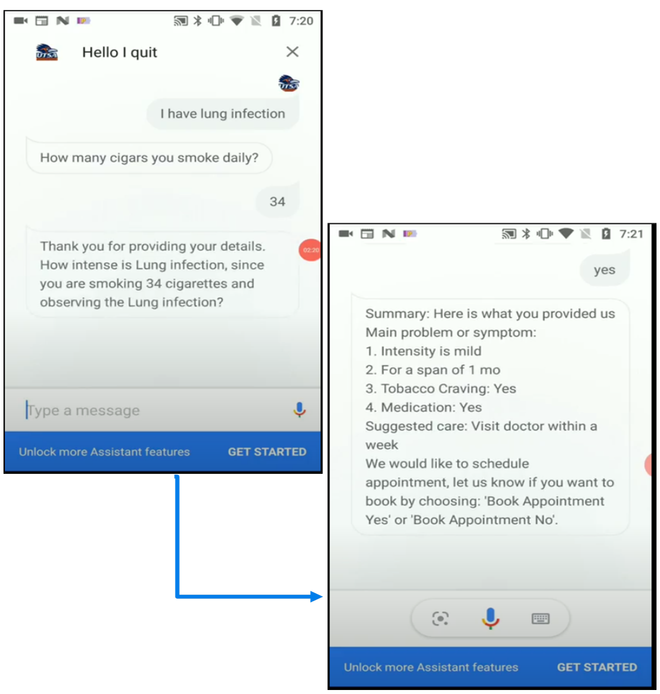

Figure 4: Keyword recognition communication user interface flowchart with Dialogflow integration

is to access the database to obtain the previous user response and state of conversation, as well as the already stored information for the interaction.

Figure 4 shows a part of successful conversation sequence for medical appointment (see figure 2) booking for symptoms associated to excessive smoking extracted from one of the tests. The interface is of the Google Assistant channel linked to the Dialogflow agent.

Conclusions and Future Work

In this work, we present a method, a cloud-based bot developed with the integration of a custom rule-based chatbot with Dialogflow. Chatbot framework such as Dialogflow are not typically designed for healthcare domain use-cases. But this work shows the feasibility of utilizing the functionality of existing chatbot development platforms. Furthermore, we demonstrate that it is still possible to fulfill the requirements of healthcare like domains (especially with regards to long-spread interventions that involve multi-turn conversations) with Dialogflow like platforms.

However, despite the fast development cycles associated with these platforms, the compatibility issues arise and there is additional overhead of addressing frequent API changes linked to these external platforms. As a future work, we intend to extend the current implementation to enable platform agnostic porting of created chatbot agents.

References

[1] Jan-Gerrit Harms et al. "Approaches for Dialog Management in Conversational Agents". In: *IEEE Internet Computing* 23.2 (2019), pp. 13–22. DOI: `10.1109/MIC.2018.2881519`.

[2] Nudtaporn Rosruen and Taweesak Samanchuen. "Chatbot utilization for medical consultant system". In: *2018*

3rd technology innovation management and engineering science international conference (TIMES-iCON). IEEE. 2018, pp. 1–5.

[3] D Inupakutika et al. "Integration of NLP and Speech-to-text Applications with Chatbots". In: *Electronic Imaging* 2021.3 (2021), pp. 35–1.

[4] Daniel Braun et al. "Evaluating Natural Language Understanding Services for Conversational Question Answering Systems". In: *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*. Saarbrücken, Germany: Association for Computational Linguistics, Aug. 2017, pp. 174–185. DOI: `10.18653/v1/W17-5522`. URL: `https://aclanthology.org/W17-5522`.

[5] *Google Dialogflow*. `https://cloud.google.com/dialogflow`. Accessed: 2022-02-05.

[6] *Amazon Lex*. `https://docs.aws.amazon.com/lex/index.html`. Accessed: 2022-02-20.

[7] *IBM Watson*. `https://www.ibm.com/watson`. Accessed: 2022-02-05.

[8] *Microsoft Language Understanding (LUIS)*. `https://www.luis.ai/`. Accessed: 2022-02-20.

[9] Rajasshrie Pillai and Brijesh Sivathanu. "Adoption of AI-based chatbots for hospitality and tourism". In: *International Journal of Contemporary Hospitality Management* (2020).

[10] David Akopian, Rodrigo David Escobar Palacios, and Sahak Igor Kaghyan. *Interactive mobile service for deploying automated protocols*. US Patent 10,819,663. Oct. 2020.

[11] Alejandro Figueroa. "Exploring effective features for recognizing the user intent behind web queries". In: *Computers in Industry* 68 (2015), pp. 162–169. ISSN: 0166-3615. DOI: `https://doi.org/10.1016/j.compind.2015.01.005`.

[12] Asbjørn Følstad, Marita Skjuve, and Petter Bae Brandtzaeg. "Different Chatbots for Different Purposes: Towards a Typology of Chatbots to Understand Interaction Design". In: *Internet Science*. Ed. by Svetlana S. Bodrunova et al. Cham: Springer International Publishing, 2019, pp. 145–156. ISBN: 978-3-030-17705-8.

Author Biography

Rahul Mundlamuri received a bachelor's degree in electronics and communications engineering from JNT University, India, in 2014, and a master's degree from the University of Houston, Houston, TX, USA, in 2016. He is currently pursuing the Ph.D. degree in electrical engineering with The University of Texas at San Antonio, TX. From 2017 to 2018, he was a Data Engineer with VRN Technologies, Austin, TX. His current research interest includes neural network-based localization services and data processing. For correspondence. Email: Rahul.mundlamuri473@gmail.com.

Devasena Inupakutika is working with Samsung Semiconductor Inc. She received Ph.D. in the Department of Electrical Engineering at the University of Texas at San Antonio (UTSA). Her research is in the development and performance analysis of systems and methods for enhancing mobility. Her research interests include web and mobile application development, cloud-IoT integration and deep learning based wireless LAN indoor positioning systems. For correspondence. Email: Devasena.inupakutika@my.utsa.edu.

Ganesh Reddy Gunnam is a Ph. D. candidate in the Department of Electrical Engineering at the University of Texas at San Antonio (UTSA). He received his MS degree in electrical engineering from the University of Texas Rio Grande Valley (UTRGV) in 2017. He is pursuing his studies in the Smart Deep-logic Chatbot Design, Development and Performance Assessment Methodology. His research interests include Chatbot cloud deployment, virtualization and cloud computing. For correspondence. Email: Ganeshreddy.Gunnam@utsa.edu.

Sahak Kaghyan is a Postdoctoral Research Scientist at the University of Texas at San Antonio (UTSA). He received his Ph.D. in Computer Science from Russian- Armenian University in 2014. His research interests include Full Stack Web Development, Mobile application development, Conversational AI design and development, Machine Learning and Software Engineering.

David Akopian is a Professor and Associate Dean of Research at the University of Texas at San Antonio (UTSA). Prior to joining UTSA, he was a specialist with Nokia from 1999 to 2003. From 1993 to 1999, he was a staff member at the Tampere University of Technology, Finland, where he received his Ph.D. degree in 1997. His current research interests include signal processing algorithms for communication and navigation receiver