Sabaragamuwa
University
Journal

Volume 19 Number 2 December 2021 pp 46-63
ISSN 1391-3166 eISSN 2386-2041
https://doi.org/10.4038/suslj.v19i2.7792

Towards a More Intuitive Sinhala Chatbot: Leveraging NLU for Enhanced Intent Identification and Entity Extraction

Pasan Avishka[1*], Nirubikaa Ravikumar[2], Kuhaneswaran Banujan[1] and Hansi Gunasinghe[1]

[1]Department of Computing and Information Systems, Faculty of Computing,
Sabaragamuwa University of Sri Lanka, Sri Lanka
[2]Department of Software Engineering, Faculty of Computing,
Sabaragamuwa University of Sri Lanka, Sri Lanka

pasanavishka36@gmail.com

Abstract

In the fast-paced and ever-changing world of conversational AI, chatbots have become essential interfaces for user interactions in various domains, especially when supporting different languages. This study delves into the development of chatbots and their ability to understand language, explicitly focusing on the Sinhala language. The effectiveness of two platforms, Rasa NLU and Microsoft LUIS, were compared in identifying and extracting intents. Both platforms showed proficiency, but Rasa stood out for its flexibility, cost-effectiveness and accurate intent recognition. A case study in the restaurant domain was conducted to demonstrate the system's capabilities. An architecture was created that can interpret Sinhala expressions and analyze intents using the NLU engine. The study defined four intents: Food Ordering, Get In Touch, About Restaurant and None. The findings highlight how this architecture has the potential to accurately interpret intents during chatbot development regardless of the conversational language used. This research aims to contribute insights to developers, linguists and AI enthusiasts involved in language-specific chatbot development by emphasizing its promises and challenges.

Keywords: Chatbot development, Natural Language Understanding (NLU), Sinhala language, Rasa NLU, Microsoft LUIS, Intent identification

1 Introduction

Information technology (IT) underpins the modern industrial landscape, with businesses across the spectrum leveraging its potential to achieve their objectives. Within IT, areas like Artificial Intelligence (AI), Cloud Computing, Blockchain, and the Internet of Things (IoT) have emerged as focal points (Bullinaria, 2005). AI, in particular, encompasses subfields such as Neural Networks, Evolutionary Computation, Vision, Robotics, Expert Systems, Speech Processing, Natural Language Processing (NLP), Planning, and Machine Learning. Driven by its growing prominence, industries invest heavily in AI to steer future business trajectories. AI influences numerous sectors, from Business and Politics to Entertainment and Healthcare.

A standout area within AI is Natural Language Processing (NLP). This field facilitates human-computer interaction, business analytics, and web software development at the intersection of linguistics, computer science, and AI. NLP aims to process, analyze, and understand human language, improving task efficiencies in various industries. Notably, while only 21% of data on the internet is structured, the remainder largely comprises unstructured data from platforms like WhatsApp, Facebook, and Instagram. NLP helps convert this unstructured data into a structured format (Bullinaria, 2005). Key NLP applications include Sentiment Analysis (Agarwal, Xie, Vovsha, Rambow, Passonneau, 2011), Chatbots (Adamopoulou Moussiades, 2020), Speech Recognition (Gaikwad, Gawali, Yannawar, 2010), Machine Translation (Poibeau, 2017), Spell Checking (Hodge Austin, 2003), Keyword Searching (Hildreth, 1997), Information Extraction (Cowie Lehnert, 1996), and Advertisement Matching (Xu, Wu, Li, Chen, 2015).

This research spotlights one primary application of NLP: Chatbots. Chatbots are interactive software utilities that emulate human conversation and are becoming integral to modern web applications. Their evolution traces back to the 1960s, and since then, they've been refined continually, finding utility across Education, Business, and E-commerce, among other sectors (Shawar Atwell, 2007).

Yet, despite their advantages, traditional web applications often suffer from limitations such as complex navigation and ineffective search capabilities. Chatbots address these challenges by providing a more intuitive user interface, enabling 24/7 service, reducing errors, and offering responses in multiple languages (Services, 2019).

Today, leading companies like Microsoft, Google, Apple, and Amazon offer Chatbot solutions, recognising the potential of these platforms to enhance user experiences. Given the proliferation of over 5,000 conversational languages globally, there's an increasing push to diversify language options in Chatbot technology. This trend underscores the importance of developing Chatbots that understand and communicate in major world languages, including English, Chinese, and Spanish.

NLP, as the backbone of Chatbots, remains a thriving research area. However, despite significant strides in English, Spanish, and Chinese language-enabled Chatbots, there's an evident gap in integrating other native languages. For instance, while English has international prominence, numerous native languages, such as Sinhala, Tamil, Korean, and Indonesian, remain underrepresented in digital conversational platforms (Aswani Gaizauskas, 2010; Ekbal Bandyopadhyay, 2011; Isahara, Bond, Uchimoto, Utiyama, Kanzaki, 2008; Sarkar Bandyopadhyay, 2008; Singh, 2008).

Sri Lanka's linguistic diversity, encompassing Sinhala, Tamil, and English, underscores the need for Chatbots that cater to this multilingual population. Sinhala holds particular significance as the primary language of around 13 million Sinhalese people. This research aims to bridge this gap by focusing on the development of a Sinhala-based Chatbot, proposing an innovative architectural framework to guide its creation. Utilizing existing tools, techniques, and insights, we introduce a novel Chatbot architecture tailored to Sinhala.

1.1 Significance of the Research

As pivotal web applications of the future, Chatbots offer unmatched efficiency and user-friendliness. However, the predominant focus on English restricts non-English speakers from leveraging their full potential. In Sri Lanka, a significant portion of the population lacks fluency in English, emphasizing the necessity of Sinhala-enabled Chatbots. However, the absence of Sinhala Chatbots with Natural Language Understanding (NLU) capabilities presents a glaring gap. Addressing this, the research delineates a robust architecture to foster Sinhala Chatbot development, which holds immense potential for businesses, especially e-commerce platforms, operating in Sri Lanka.

1.2 Aim and Scope of the Research

This research seeks to integrate the Sinhala language into Chatbots by devising a ground-breaking architecture. By amalgamating existing tools and NLP theories, like Tokenization and Named Entity Recognition (NER) (Vasiliev, 2020), it aims to establish a blueprint for Sinhala Chatbot development.

To ground the research in practical application, a Sinhala-based Chatbot will be developed for a restaurant setting. This prototype will include features such as;

1. NLU unit for intent identification and entity extraction.
2. Dialogue management unit with predefined restaurant-related dialogues.
3. Channel integration across platforms like Web, Facebook Messenger, and Telegram.
4. Additional features via external API integrations.

This structured approach ensures that the developed chatbot exemplifies the comprehensive capabilities envisioned in our architectural framework.

2 Literature Review and Related Work

Chatbots have rapidly transformed digital interactions in various domains. From assisting users on websites to providing customer support, chatbots are ubiquitous due to their ability to streamline operations and offer real-time assistance.

The development of chatbots traces back to the 1950s, starting with the Turing Test (Solutions, 2020). Over the years, the IT sector witnessed several milestones, including the Turing Test 1950(Harnad, 2008); ELIZA, 1966 (Bradeško Mladenić, 2012) to Bots for Messenger: Facebook Chatbots, 2016 (Dale, 2016); Tay, 2016 (Davis, 2016); Woebot, 2017 (Fitzpatrick, Darcy, Vierhile, 2017) etc. Prominent IT companies such as Apple, Microsoft, Google, and Amazon have capitalized on this trend, offering chatbot solutions integral to daily digital experiences (Solutions, 2020). Despite being a contemporary trend, the concept of chatbot development isn't new; its foundation was laid decades ago.

Chatbots fundamentally rely on converting user input into comprehensible data. If voice-enabled, they utilize Automatic Speech Recognition technology to transcribe voice data to text (Solutions, 2020). This text is then processed to elicit appropriate responses. Intricate Natural Language Processing (NLP) activities, including Natural Language Understanding and Generation, are underlying these interactions. The development lifecycle of chatbots is pivotal to ensuring effective communication. Google's paper on implementing their Meena chatbot offers invaluable insights into chatbot creation and training (Adiwardana et al., 2020). PwC EU Services' documentation provides a comprehensive guide on chatbot architecture, including essential components, operations, and service recommendations (Solutions, 2020).

Developing sophisticated chatbots was historically challenging due to a lack of advanced tools. Nevertheless, chatbots like MILABOT, XiaoIce, and Mitsuku managed to emulate human-like interactions using intricate frameworks and advanced machine learning (Shum, He, Li, 2018). By understanding their challenges and achievements, current innovations in chatbot technology can be appreciated.

Creating a chatbot specifically for the Sinhala language presents its challenges. Considering existing chatbot architectures is essential while devising a novel solution tailored for Sinhala (Nimavat Champaneria, 2017). Resources like the scientific journal by Ketakee Nimavat and Prof. Thushar Champaneria provide a holistic view of chatbot developments, shedding light on tools, algorithms, and architectures that can guide Sinhala chatbot creations (Nimavat Champaneria, 2017).

Smart campuses represent the new wave in educational infrastructure, integrating AI and IoT to transform traditional settings. An article in Sustainability illustrates how chatbots can be incorporated into these advanced environments (Villegas-Ch, Arias-Navarrete, Palacios-Pacheco, 2020). Additionally, enhancing customer experience is a primary goal for many online platforms. Chatbots, especially in customer support roles, can significantly impact this area. A master's thesis from the University of Liege discusses the intricacies of designing a customer-support chatbot from the ground up. This study could guide the development of a Sinhala-specific chatbot (Ngai, Lee, Luo, Chan, Liang, 2021).

NLP stands at the heart of chatbot functionality. With subsets like Natural Language Understanding (NLU) and Generation (NLG), comprehending these technical facets is paramount (Manning et al., 2014; Nadkarni, Ohno-Machado, Chapman, 2011; Vasiliev, 2020). Libraries such as NLTK and Stanford NLP play a crucial role in customizing the NLU components of chatbots. Considering the focus on Sinhala-based NLP, surveys like the one by N. de Silva (de Silva, 2019) offer a trove of tools and methodologies tailored for the Sinhala language. While they provide algorithms essential for NLP, direct adaptation to chatbots can be problematic. Efforts like the Sinhala-based Chatbot by B. Hettige Asoka. S. Karunananda (Hettige Karunananda, 2006) highlights the potential in this direction despite the challenges posed by the complex linguistic structure of Sinhala.

The identification of user intent is pivotal in chatbot interactions. Services like Rasa NLU provide robust solutions (Bocklisch, Faulkner, Pawlowski, Nichol, 2017). Furthermore, entity extraction can be challenging, especially in languages with intricate structures like Sinhala. Many research endeavors have been undertaken to enhance NLP for the Sinhala language, aiming to improve applications (de Silva, 2015; Jayaweera Dias, 2016; Weerasinghe, Wasala, Herath, Welgama, 2008; Welgama et al., 2011).

Frameworks like Microsoft's Bot Framework offer comprehensive solutions for chatbot development (Biswas, 2018). It is integrated with Azure Bot Service and boasts features like dialogue management, media sharing, and advanced security. Syllable recognition is particularly challenging in Sinhala NLP. Specific research papers, such as the one cited (Weerasinghe, Wasala, Gamage, 2005), offer rule-based algorithms for Sinhala syllabification. Moreover, other research on Sinhala-centric applications provides insights into NLP theories and their practical applications (Vasiliev, 2020; Weerasinghe et al., 2008; Wijesiri et al., 2014).

3 Research Methodology

The primary objective of this research is to develop a Sinhala Chatbot that offers key services of a restaurant using a newly introduced architecture. A clear definition of the chatbot's scope is essential before delving into its development through the designated architecture. The chatbot's scope will be elaborated upon in subsequent sections. The journey to the findings begins with a thorough literature review delineated in previous sections. Figure 1 provides an overarching view of the methodological framework employed in this research.

3.1 Chatbot Development Environment and Tools

The foundational step in this research was identifying the apt tools and frameworks for Chatbot development. While a multitude of Chatbot development frameworks are available, with most offering robust features for development, evaluation, and maintenance, a core challenge remains - compatibility with the Sinhala language. Choosing tools must seamlessly accommodate the Sinhala language, especially within the Natural Language Understanding (NLU) components, which are pivotal to Chatbot functionality. Upon weighing various factors such as adaptability, ease of use, and compatibility, this research adopted the Microsoft Bot Framework and the Rasa stack for the ensuing experiments.

3.2 Architecture for Chatbot development

Having established the development environment, the next focus was crafting an architecture specific to Sinhala language-based Chatbots. As discussed, the Microsoft Bot Framework and Rasa NLU emerged as prime choices after an exhaustive survey. Figure 2 delineates the foundational structure of the Chatbot pipeline. Within this structure, user interaction initiates with an utterance to the chatbot. This input is then funneled to the NLU unit, equipped with modules for Intent Identification, Entity Extraction, and Context Awareness (Bocklisch et al., 2017), which deciphers and directs the query. The Bot management unit then handles user responses, cloud integrations, information storage, and dialogue flow management, ensuring the user receives the intended service. This research's crux was to mold a fitting architecture for Sinhala language-based Chatbots, integrating existing tools, frameworks, and pertinent NLP techniques like Tokenization, Lemmatization, and Named Entity Recognition. Our efforts coalesced in an architecture that supports Sinhala language-based Chatbot development, grounded in the Microsoft Bot Framework and Rasa NLU.

3.3 Intent Identification

The efficacy of a Chatbot hinges on the prowess of its Natural Language Understanding (NLU) unit. Given its importance, the research gravitated towards enhancing the NLU capabilities of the proposed architecture. Following a comprehensive literature review, Rasa NLU was chosen, though Microsoft's Azure Language Understanding Intelligent Service (LUIS) was also deemed noteworthy. The preference for Rasa NLU stemmed from its expansive NLU pipeline capabilities and open-source nature. However, exploring both was deemed valuable. At the heart of the NLU engine lies ”Intent.” This crucial feature allows Rasa NLU to interpret incoming utterances and return a corresponding intent, based on which the chatbot can formulate responses. An ”Intent” embodies the objective or motive behind a user's utterance. Table I illustrates intents paired with example utterances. While many NLUs can proficiently classify intents in English, the challenge lies in their efficacy with Sinhala. The experiments led to choosing Rasa and LUIS due to their proficiency in classifying Sinhala intents, eliminating the need to craft an intent classifier from scratch.

The above illustration depicts the English language-based intent identification. Almost every NLU can classify the English-based intents but not in the Sinhala language. Rasa and LUIS were chosen for this research by testing and experimenting because they can also classify the Sinhala intentions. Therefore, it doesn't need to develop an intent classifier from scratch. Rasa NLU will be compatible with the introduced architecture.

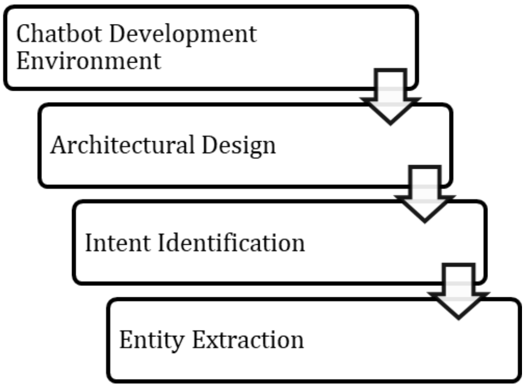


Figure 1: The high-level methodological framework

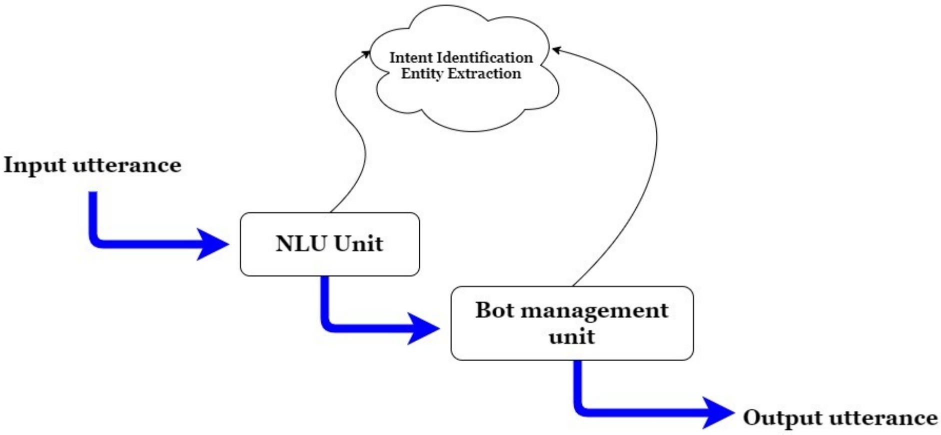


Figure 2: A basic skeleton of the Chatbot pipeline

3.4 Entity Extraction

Entity extraction stands as another critical facet of NLU units. Entities enable the extraction of specific details from utterances, often essential for subsequent stages of the chatbot's operations. After intent classification, this is the second vital step in the process. Table 2 lists entities with related example utterances. Although existing NLUs efficiently extract data in English, a conspicuous gap exists in their capability to process Sinhala utterances. Consequently, a primary challenge in this research was pioneering a method for Sinhala entity extraction. We initially concentrated on the selected NLU engine, Rasa NLU, integrating NLP techniques like Tokenization and Featurization, given Rasa's flexibility in customizing its NLU pipeline.

4 Results and Findings

4.1 Chatbot Development Environment

Upon examining various tools and frameworks outlined in the research methodology section, it became evident that many frameworks exist for Chatbot development. Numerous companies offer platforms to design, implement, test, and evaluate Chatbots. The most prominent and widely-used Chatbot development tools and frameworks include Microsoft Bot framework (Azure bot service), Azure Language Understand Intelligent Service (LUIS), Google Dialog Flow, Amazon Lex, Botkit, Wit.ai, Rasa Stack, SAP Conversational AI, IBM Watson Assistant, BotPress, BotMan, GupShup, Botsify, Pandorabots, MobileMonkey.

Out of these, the study narrowed down to a select few tools based on their capabilities, NLU strengths, recommendations from experts in the field, and online sources. The selection focused on key features like intent classification, entity extraction, sentiment analysis, and context awareness: Microsoft Bot framework (with LUIS), Google Dialogflow, Amazon Lex, Wit.ai, Rasa Stack, and IBM Watson Assistant.

Given the complexity of the Sinhala language, introducing Chatbot capabilities poses a significant

Table 1: Examples for intentsintents

Intent	Example utterances
Greeting	"Hi", "Hello", "Good morning", "How are you doing", "What's up", "Hola"
CheckWeather	"Show me the forecast in Colombo", "Is today a rainy day?", "How will be tomorrow weather?", "I would like to know the weather forecast for tomorrow."
BookFlight	"Book me a flight to India next week.", "I am going to need a ticket to go to Singapore."
JoinUs	"I would like to join your company", "Are there any job vacancies?", "Is it possible to apply to your company?", "I am interested in the job vacancy you posted."

Table 2: Example utterances with relevant entities and data

Example Utterance	Intent	Entities	Data
"Book me 3 tickets to India."	BookFlight	Quantity Destination	3, "India"
"Can you deliver me 2 Pizzas at 8 pm?"	FoodOrdering	FoodType Quantity Time	"Pizza" 2 "8.00 pm"
"Please bring me a cup of coffee at 4 pm."	FoodOrdering	FoodType Quantity Time	"Coffee" 1 "4.00 pm"

challenge. This research highlights that existing tools can be leveraged to achieve this without starting from scratch. Both the Rasa Stack and the Microsoft Bot framework were selected as they are free, open-source, and adaptable for Sinhala language integration.

4.1.1 Microsoft Bot Framework

The Microsoft Bot framework is an open-source SDK, including the Azure bot service, provided by Microsoft. It offers various tools and services catering to different Chatbot development stages. One of its main advantages is its compatibility with multiple programming languages like C, JavaScript, and Python. The framework is equipped to handle various stages of Sinhala language-based Chatbot development.

4.1.2 Rasa NLU

Given its role in understanding user inputs, NLU is crucial for Chatbot development. Rasa NLU was selected due to its capacity to support the Sinhala language. As an open-source machine learning framework, Rasa aids in constructing AI bots for both text and voice interactions. While the Microsoft Bot framework was favored for dialogue management, Rasa was preferred for its NLU capabilities. Initially designed for English, Rasa NLU caters to multiple languages and has demonstrated proficiency with Sinhala. The main focus of this research with Rasa NLU was intent identification and entity extraction. The customizability of Rasa NLU, especially in entity extraction, will be explored further in subsequent sections.

4.2 Modeling an Architecture

The primary aim of this research is to propose an architecture suitable for Sinhala Chatbot development. As depicted in Figure 3 of the methodology section, core stages of this development include:

- User Interface - For user interaction with the chatbot.
- NLU engine - Analyses user input to understand their requirements.
- Bot management unit - Oversees all bot activities, such as dialogue management.
- Cloud provider - Facilitates hosting the application.

Given these prerequisites, the chosen tools, Microsoft Bot Framework and Rasa NLU, are deemed apt.

It is essential to clarify that the term 'architecture' refers to the organization, design, construction, and maintenance of a product or structure.

Although the Microsoft Bot and Rasa stack operate as standalone Chatbot development frameworks, this research suggests that a fusion of the two can be effective for Sinhala conversational agents. With the open-source nature of the Microsoft Bot framework, there's room for customisation, making it conducive for Sinhala integration.

As previously highlighted, while the Microsoft Bot framework handles bot management features, Rasa caters to the NLU components. The foundation for the new architecture, depicted in Figure 3, is constructed based on the Chatbot pipeline skeleton detailed in Figure 2 of the methodology section.

When a user communicates with the bot, the Microsoft bot framework orchestrates the chatbot pipeline's entire workflow. Initially, the framework sends the user's message to Rasa NLU for Intent Identification. After identifying the intent, Rasa NLU sends it back, upon which the framework directs it to the Entity Extraction stage. If there are entities present, they are captured and stored. Subsequently, the Microsoft bot framework processes the final response to be delivered to the user. This sequence outlines the operation of the presented architecture.

4.3 Components of the Architecture

4.3.1 User Interface

The User Interface serves as the user-facing component, enabling interaction with the chatbot. Users can employ diverse interfaces for communication, with available channels including Web interface, Facebook Messenger, Telegram, Twilio, Skype, Slack, Emails, Kik, Line, and Microsoft Teams, among others.

4.3.2 NLU Engine

- Intent Identification: An intent encapsulates the underlying objective of a user's statement. The Intent Identification unit's role is to discern the intent of incoming user utterances.

- Entity Extraction: Entities represent specific pieces of data within user messages. The NLU engine extracts and processes these entities.

- NLU Pipeline Customization: Despite Rasa NLU's robust functionality, certain customisations were necessary to meet this research's objectives. Its open-source nature facilitates such modifications.

- Knowledge Base: This database houses training data for the NLU engine. It contains predefined intents, entities, and associated Sinhala examples.

4.3.3 Bot Management Unit

- Component Dialogs: A chatbot's operational domain, whether hospitality, HR, travel, etc., dictates the dialogues it can handle. These dialogues cater to user queries, sometimes facilitating tasks like monetary transactions.

- Dialogue Management: This central unit steers the conversation, deploying relevant component dialogues based on the identified intent.

- Bot Configuration: All bot configurations, encompassing areas like NLU settings, security, and external API links, are overseen here.

- API Client: This caters to integrations with various external APIs, enhancing user experience.

- Database Management: This ensures efficient data storage and retrieval, irrespective of whether the data resides locally or in the cloud.

- NLU Handling: This handles intents retrieved from the NLU engine, directing the dialogue management unit accordingly.

- Package Manager: Manages software packages essential for the bot's operation.

- Security: Focuses on integral security aspects like authentication, authorisation, and data protection.

4.3.4 Cloud Provider

- Web Hosting: Given that chatbots parallel modern web applications, they are typically hosted in the cloud. The benefits are more pronounced when integrated with cloud-centric channels like Facebook Messenger.

- Database: Cloud-based databases are often preferred for their efficiency and security features.

- Analytics: For certain chatbots, understanding user behavior is pivotal for optimisation. This cloud-centric component serves that purpose.

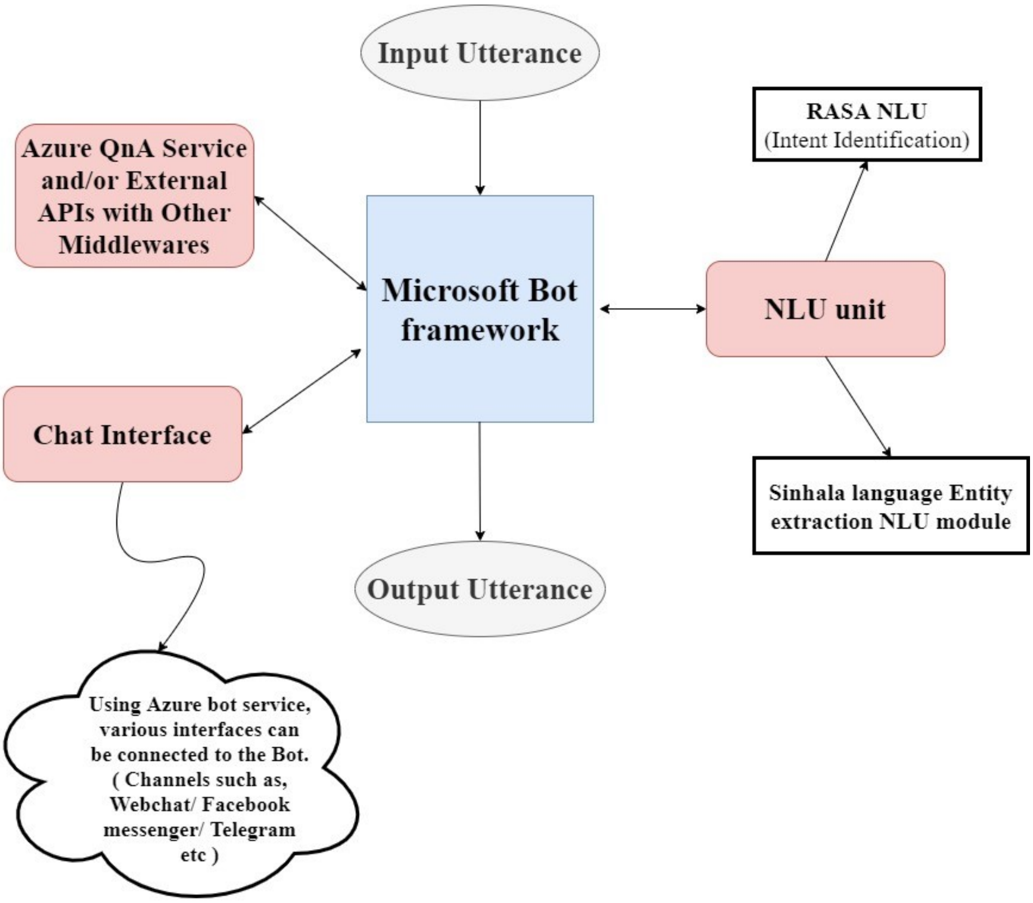

Input Utterance
Azure QnA Service and/or External APIs with Other Middlewares
Microsoft Bot framework
RASA NLU
(Intent Identification)
NLU unit
Chat Interface
Sinhala language Entity extraction NLU module
Output Utterance
Using Azure bot service, various interfaces can be connected to the Bot. (Channels such as, Webchat/ Facebook messenger/ Telegram etc)

Figure 3: A conceptual architecture for the development of the Sinhala Language-based Chatbots

- Channel Integrations: Enables chatbots to be integrated with various user interface channels. Platforms like Azure facilitate this integration.

In conclusion, Figure 4 visually represents the architecture, providing a high-level view of the integrated framework.

4.4 Intent Identifications

Although the system is equipped to manage dialogues and other operations, the focus remains predominantly on its NLU capabilities. Informed by our literature review, Rasa NLU and Microsoft LUIS were chosen for NLU experimentation. An elaboration of ”Intent” is provided in the methodology section.

Sinhala, a complex language, poses challenges in intent identification for chatbot development. However, Rasa and LUIS have demonstrated proficiency in accurately identifying Sinhala intents. Given the precision, flexibility, and cost-effectiveness (free usage) of Rasa, it is endorsed for the proposed architecture. For example, here is an utterance that is used to have in-flight operations based on a chatbot.

“මට ගුවන් ටිකට් පතක් මිළදී ගන්න ඕනි හෙට ඉන්දියාවේ යන්න.”

(I want to buy a plane ticket to India tomorrow.)

Such utterances typify queries in chatbots designed for airline operations. The language of conversation becomes secondary to the prowess of the bot’s NLU components. Recognising the intent behind the utterance is crucial. In this case, possible intentions include “Flight” or “Book_Tickets”. A chatbot developed using our architecture should accurately infer such intentions.

In the context of this study, focusing on the restaurant domain, four primary intents were identified. Upon user interaction with the chatbot, the message first navigates through the NLU engine, underscoring the NLU’s significance during chatbot development. The initial role of the NLU is to discern the appropriate intent for subsequent operations.

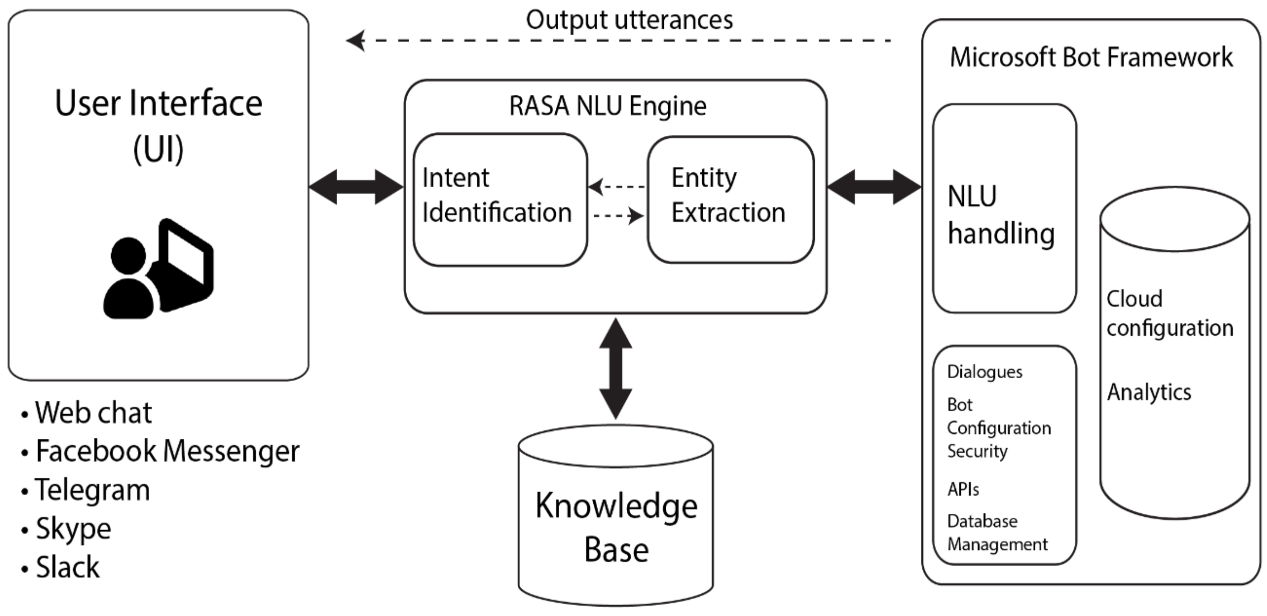


Figure 4: High-level architecture of the proposed architecture

The predefined intents for our Sinhala restaurant chatbot include:

1. Food Ordering.
2. Get In Touch.
3. About Restaurant.
4. None.

When users introduce an utterance, the system, by examining it, assigns a corresponding intent. This designated intent then guides subsequent interactions. Remark: Both Rasa and LUIS, by design, feature a “None” intent. Utterances not fitting into any predefined category default to this ”None” intent.

4.4.1 Illustrating Intents

Training the NLU machine learning model in both Rasa and LUIS is necessary to conduct this research. Initially, this research uses both NLU engines with the same data. One NLU will be chosen for the introduced architecture by comparing the accuracy and other facts. Those tests will be described in the next section. Both NLU engines return the appropriate intent by analyzing the incoming utterances.

When considering the above trained NLU models, they can return appropriate intents with scores of each intent. The following section will illustrate the testing of intent identifications using both Rasa and LUIS. And then, by making a comparison between those two, one NLU engine will be chosen for the introduced architecture.

4.4.2 Testing

As mentioned, it should first be trained on the NLU models using the data. The architecture will choose the highest-scored intent for further processes by considering the returned intents and scores. The following tables have shown the example results of intent identifications of given Sinhala utterances.

The next section of the document provides illustrations of testing those utterances in Rasa and LUIS. The next section of the document provides illustrations of testing those utterances in Rasa and LUIS. To conduct the test on Rasa, This research uses the Postman (https://www.postman.com/) software, and LUIS uses the web-based dashboard Azure provides users.

As mentioned earlier, one of the significant tasks in this research is to choose an appropriate one for the introduced architecture between these NLU engines. Therefore, this research has conducted

Table 3: Results of Intent Identifications in Rasa

Example Utterances	**Food_Ordering**	**Get_In_Touch**	**About_Restaurant**	**None**
අහාර ඇණවුම් කිරීමේදී අයකරගන්නා මුදල් පිලිබද දැනගැනීමට අවශ්‍යයි (Want to know about the amount charged while ordering food).	0.91	0.04	0.02	0.03
මම කැමතී ඔබලාගේ අවන්හල සමග සම්බන්ද වී සිටින්න (I would like to stay connected with your restaurant).	0.01	0.89	0.09	0.01
ඔබලාගේ අවන්හලෙන් සපයන විශේෂ අමතර සෙවාවන් මොනවාද (What special extras does your restaurant offer)?	0.02	0.08	0.89	0.01
ඔබලාගේ සෙවාවන් මම ගොඩක් අගය කරනවා (I appreciate your services very much).	0.01	0.03	0.10	0.86

Table 4: Results of Intent Identifications in LUIS

Example Utterances	**Food_Ordering**	**Get_In_Touch**	**About_Restaurant**	**None**
අහාර ඇණවුම් කිරීමේදී අයකරගන්නා මුදල් පිලිබද දැනගැනීමට අවශ්‍යයි (Want to know about the amount charged while ordering food).	0.93	0.04	0.01	0.02
මම කැමතී ඔබලාගේ අවන්හල සමග සම්බන්ද වී සිටින්න (I would like to stay connected with your restaurant).	0.04	0.87	0.06	0.03
ඔබලාගේ අවන්හලෙන් සපයන විශේෂ අමතර සෙවාවන් මොනවාද (What special extras does your restaurant offer)?	0.11	0.03	0.83	0.03
ඔබලාගේ සෙවාවන් මම ගොඩක් අගය කරනවා (I appreciate your services very much).	0.03	0.06	0.19	0.72

several tests using the exact data for both NLU engines. By comparing the results of that test, one NLU engine has been chosen.

Initially, the test is conducted using specific predefined Sinhala utterances. By noting every result returned by Rasa and LUIS, this test justifies that one of the NLU engines has enough power for the introduced architecture.

This research has used 50, 100, 200, and 500 utterances for scoped Chatbot (Sinhala Restaurant Bot) intents to test both NLU engines. Therefore, the most suitable NLU engine for the introduced architecture has been chosen.

The test results are listed here on both NLU engines. The following tables show the number of utterances tested and the number of utterances that both Engines correctly identified.

Figure 5 visually represents the results of Rasa and LUIS with utterances. By analyzing the above test results and comparisons between Rasa and LUIS, this research has focused on choosing the Rasa NLU engine for the proposed architecture for developing Sinhala Chatbots. Because of the accuracy illustrated above, Rasa can be chosen and recommended for Sinhala language-based intent identifications.

4.5 Entity Extractions

The next phase of this research is Entity extraction. This task is the most challenging part of the research and the current ongoing task. Even if existing NLUs can do the entity extraction in English, no NLUs have the power to do the entity extraction in the Sinhala language. Therefore, currently, it is one of the active research areas in Sri Lanka. But rather than developing an NLU from scratch for the Sinhala Entity extraction, this research focuses on customizing Rasa NLU to fulfill this task because Rasa provides a way to customize their NLU pipeline. It provides some features to customize the NLU components.

When considering entities, as described earlier, Entities can be used to store information, which is included in the Sinhala utterance. Here is the same example that is used to explain intent identification.

“මට ගුවන් ටිකට් පතක් මිළදී ගන්න ඕනි හෙට ඉන්දියාවේ යන්න.”

(I want to buy a plane ticket to India tomorrow.)

After proper training, Rasa or LUIS can identify the intent of the Sinhala utterance as “Book_Tickets” or “Flight”. But there is more information to consider inside that utterance. The user wants to go to India, or the user wants to go there tomorrow. Here is the list of Entities to be extracted.

1. Destination - “India”
2. Time/Date - “Tomorrow”

The primary goal of this research phase is to find a way to extract this information from given Sinhala utterances. Core NLP functions such as Tokenization, lemmatization, or Named entity recognition will be needed to implement this task from scratch. Therefore, this research initially focused on customizing the Rasa NLU pipeline for the introduced architecture. If it succeeds, the Rasa NLU will suit the Sinhala Entity extraction. If not, this research will focus on implementing an independent model for Entity extraction, which can integrate it with the introduced architecture.

According to the restaurant bot, many Sinhala utterances could have entities to extract. When considering the food_ordering scenario,

“මට හෙට රෑට පීසා හදකක් අවශ්‍යයි”

When analyzing the above utterance, Rasa NLU can return the relevant intent as “food_ordering’. But there is more information that can be extracted from the above utterance. To extract that information, there should be an Entity extraction unit. Then, it can extract entities such as Time, Food Type, and Quantity. For the above scenario, the entities are,

1. Time/date = “Tomorrow”,
2. Food Type = “Pizza”
3. Quantity = “2”.

The following section briefly describes the Core functionality of the Rasa NLU pipeline for Sinhala entity extraction.

4.5.1 Customizing Rasa NLU pipeline for Sinhala entity extractions

Considering the Rasa NLU architecture it is defined with the concept of pipeline. That means there is a set of steps inside the core NLU model of

Table 5: Results of Rasa and LUIS with 50 utterances.

Intent	Result in Rasa NLU	Result in LUIS
Food_Ordering	13/14	12/14
Get_In_Touch	11/12	11/12
About_Restaurant	11/12	12/12
None	10/12	9/12
	45/50 – 90%	44/50 - 88%

Table 6: Results of Rasa and LUIS with 100 utterances.

Intent	Result in Rasa NLU	Result in LUIS
Food_Ordering	20/25	21/25
Get_In_Touch	23/25	22/25
About_Restaurant	23/25	23/25
None	21/25	20/25
	87/100 - 87%	86/100 – 86%

Table 7: Results of Rasa and LUIS with 200 utterances.

Intent	Result in Rasa NLU	Result in LUIS
Food_Ordering	43/50	41/50
Get_In_Touch	44/50	40/50
About_Restaurant	40/50	39/50
None	35/50	37/50
	162/200 - 81%	157/200 – 78.5%

Table 8: Results of Rasa and LUIS with 500 utterances.

Intent	Result in Rasa NLU	Result in LUIS
Food_Ordering	116/125	111/125
Get_In_Touch	119/125	118/125
About_Restaurant	119/125	120/125
None	99/125	95/125
	453/500 - 90.6%	444/500 - 88.8%

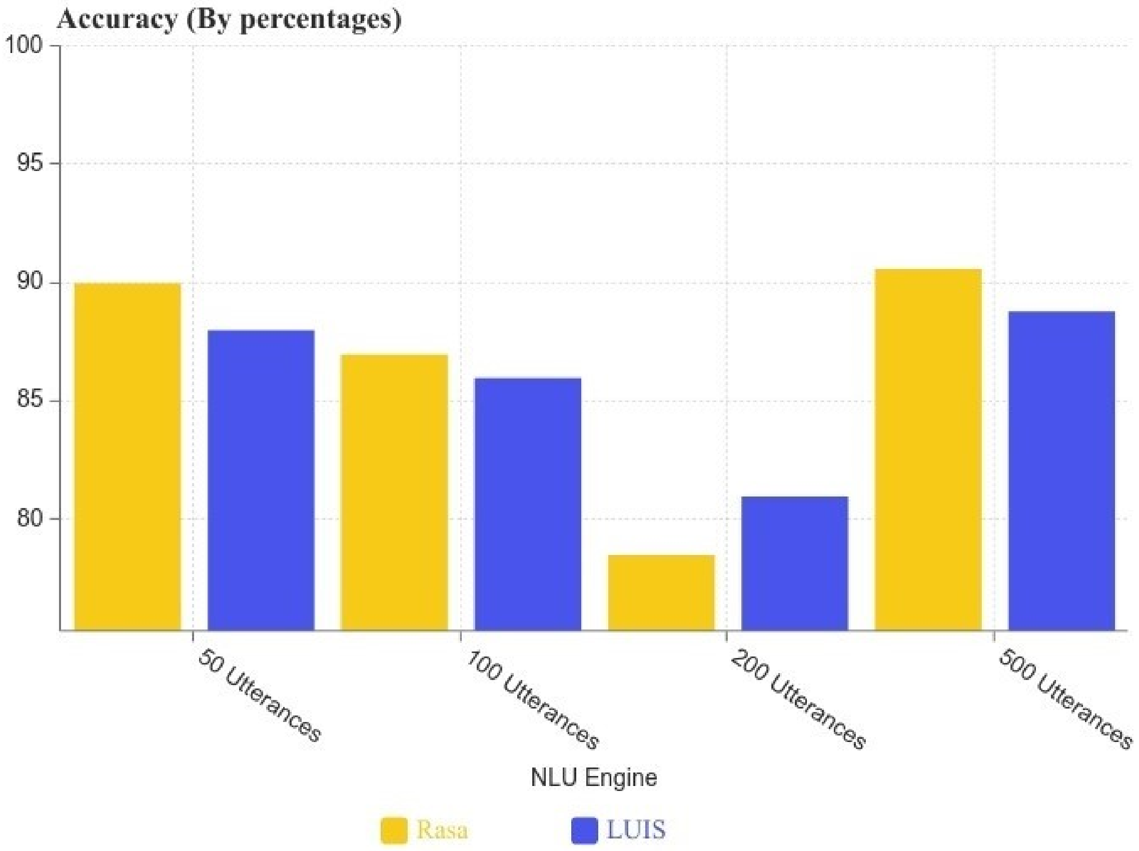


Figure 5: Results of Rasa and LUIS with utterances.

Rasa. The main advantage of this stack is providing the customization facility of the pipeline of Rasa.

Here are the main parts of the Rasa NLU pipeline.

1. Tokenization
2. Featurization
3. Entity recognition/ Intent classification/ Response selectors

The first step is the process of tokenizing the incoming utterance. Then, developers should decide what type of NLU model can be used for the further process. And finally, Rasa's intent classification units and other related units will be functioning . For these NLU pipelines, Rasa uses the units called components. The Rasa has provided all the components used for the NLU pipeline, including intent identification and other related NLU features. Therefore, developers do not need to worry about these components.

The main target of this research task is to customize the NLU pipeline for Sinhala Entity extractions. This research tries to create a separate component that includes all the functionalities for Sinhala language-based Entity extractions by using Python programming language.

According to the above Figure 6, components can be added or removed according to the Chatbot applications' domain and scope. Therefore, this research suggests and focuses on adding a complete component into the NLU pipeline, which can be used for the Sinhala language-based Entity extractions.

4.5.2 Alternative ways for Sinhala Entity extractions

When considering this task, the Sinhala language can be considered one of the world's most complex languages. Therefore, it is not a trivial task to implement NLPs for the Sinhala language. According to the literature survey, many Sinhala language-based types of research in Sri Lanka can be found.

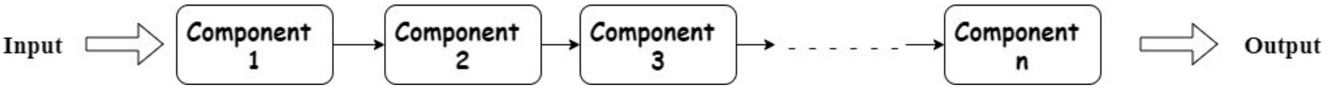


Figure 6: Structure of Components in Rasa NLU pipeline

Here are some libraries to develop NLP functions for the Sinhala language.

1. Python NLTK library (https://www.nltk.org/)

2. Stanza NLP library (An NLP library for many human languages - https://github.com/stanfordnlp)

3. Microsoft NLTK NuGet package (https://www.nuget.org/packages/NltkNet/)

Using those libraries, this research can build a unit for Sinhala entity extractions. However, this research tries to find alternatives for this task because it mainly focuses on existing tools and frameworks.

Another alternative for Entity Extraction can be finding the intersection of prebuilt word arrays with incoming tokenized word arrays. That means, According to the Rasa pipeline, the step is tokenization . The NLU engine temps to tokenize the incoming user utterance by using Rasa's tokenizers . After that, automatically create an array of tokenized words. Developers can initiate the prebuilt word arrays according to the domain. When considering the following scenario,

“මට පිසා දෙකක් ගෙදරට ගෙන්වා ගන්න ඕනි!”

(I want to take two pizzas home!)

By analyzing the above utterance, Rasa NLU can identify the intent as “Food_Ordering”. But there are entities to be extracted.

1. Food_type - “පිසා”

2. Quantity – 2

To extract those two data from the utterance, the developer should predefine the array of possible entities as in the following example,

Food_type = [“චිකන් පිසා”, “පිසා”, “බනිස්”, “බිස්කට්”, “කේක්”, “කිරි”, “යෝගට්”, “පලතුරු යුශ”, “ඇපල්”, “දොඩම්”]

And here is the tokenized word array of the above example utterance.

User_Input = [“මට”, ‘පිසා”, “දෙකක්”, “ගෙදරට”, “ගෙන්වා”, “ගන්න”, “ඕනි”]

When comparing the above two arrays (Food_type and User_Input), the identical items are stored in those.

The intersection of the above two arrays - [“පිසා”]

Thereby, the word “ ” can be extracted as the food type of the above Sinhala utterance.

This approach can identify and extract specific information from any given Sinhala utterance. This approach can be considered one of the possible alternatives for fulfilling the aim of Entity extraction.

However, the Entity extraction part of this research is an ongoing research area. It will be one of the significant future works of introducing an architecture for Sinhala language-based Chatbot developments.

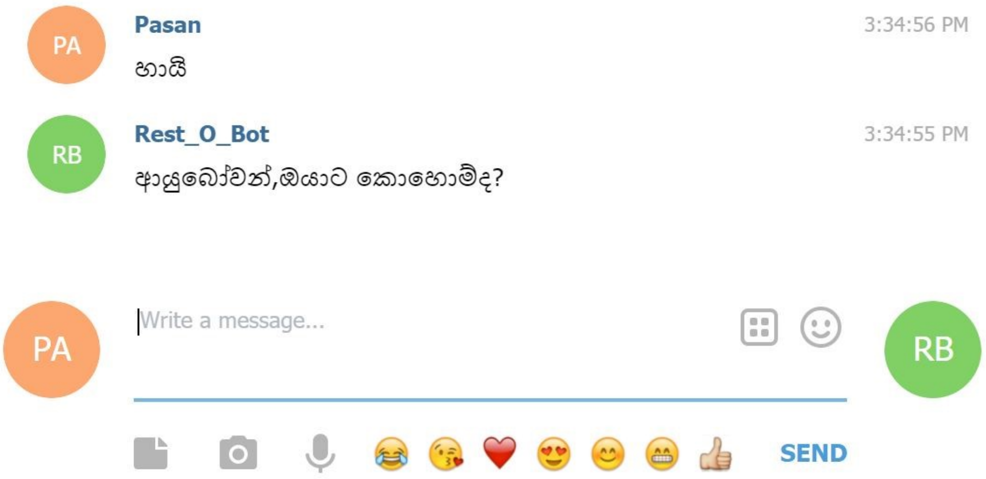
4.6 Results from the RestaurantBot

Using introduced architecture, this research developed a Chatbot for a restaurant that includes the facilities as clarified the scope. This chatbot has been integrated into the Telegram channel using Azure cloud Channel integration. Here are some screenshots of the developed Chatbot. The Screenshots of RestaurantBot 1, RestaurantBot 2, RestaurantBot 3, RestaurantBot 4, RestaurantBot 5 and RestaurantBot 5 are shown in Figure 7, Figure 8, Figure 9 and Figure 10 respectively.

Note - This Chatbot was developed to ensure the reliability of the introduced novel architecture and only supports a few basic operations for now.

5 Discussion and Conclusion

Incorporating the Sinhala language into conversational agents presents a distinct challenge, primarily due to the limited resources available for Sinhala language-based Natural Language Processing (NLP). In this context, the research significantly contributes to technological develop-



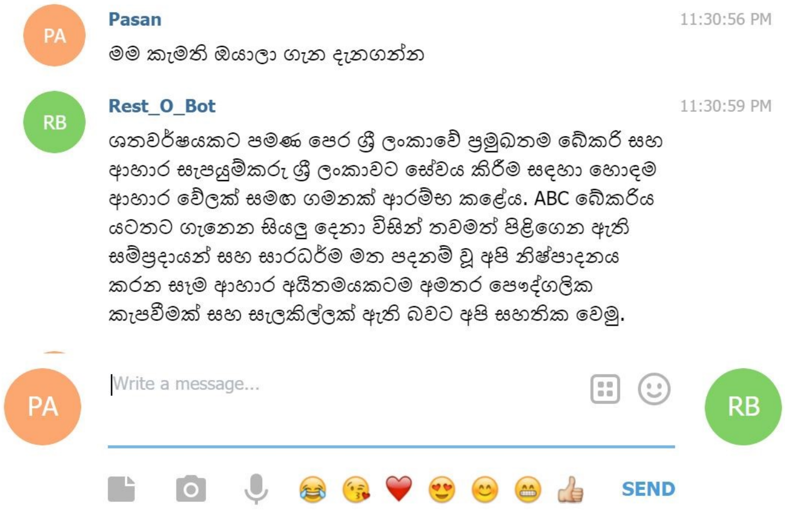
Pasan 3:34:56 PM

හායි

Rest_O_Bot 3:34:55 PM

ආයුබෝවන්,ඔයාට කොහොම්ද?

Figure 7: Screenshot of the RestaurantBot 1



Pasan 11:30:56 PM

මම කැමති ඔයාලා ගැන දැනගන්න

Rest_O_Bot 11:30:59 PM

ශතවර්ෂයකට පමණ පෙර ශ්‍රී ලංකාවේ ප්‍රමුඛතම බේකරි සහ ආහාර සැපයුම්කරු ශ්‍රී ලංකාවට සේවය කිරීම සදහා හොදම ආහාර වේලක් සමඟ ගමනක් ආරම්භ කළේය. ABC බේකරිය යටතට ගැනෙන සියලු දෙනා විසින් නවමත් පිළිගෙන ඇති සම්ප්‍රදායන් සහ සාරධර්ම මත පදනම් වූ අපි නිෂ්පාදනය කරන සෑම ආහාර අයිතමයකටම අමතර පෞද්ගලික කැපවීමක් සහ සැලකිල්ලක් ඇති බවට අපි සහතික වෙමු.

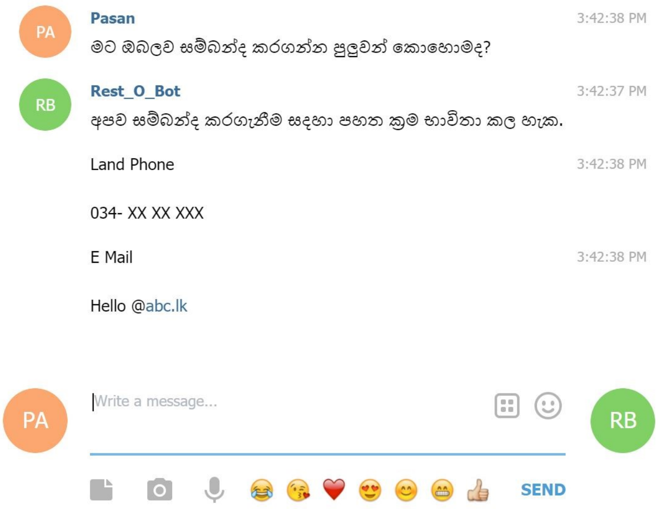
Figure 8: Screenshot of the RestaurantBot 2

ments assisted by the Sinhala language. To infuse the Sinhala language into intelligent conversational agents was aimed by introducing a comprehensive architecture tailored for Sinhala language-based chatbot development. This study was methodically divided into four primary segments:

1. Establishing a chatbot development environment.
2. Designing a comprehensive architecture.
3. Intent identification techniques.
4. Methods for entity extraction.

Existing tools and frameworks were identified and leveraged to shape the study's architecture by conducting an extensive literature survey. The Microsoft Bot Framework was identified and utilized as the bot management unit, while Rasa NLU was selected for its prowess as the NLP engine. Integrating these components, the proposed architecture was formed.

While the primary objective was architecture modeling, the importance of intent identification and entity extraction within any NLP system was recognized. The emphasis on these components stems from their inherent complexity and critical role in determining the efficacy of a conversational agent. Commendable accuracy was achieved up to the intent identification phase. Nonetheless, entity extraction emerged as a formidable challenge in this research trajectory. Despite these hurdles, this



Pasan 3:42:38 PM
මට ඔබලව සම්බන්ද කරගන්න පුලුවන් කොහොමද?

Rest_O_Bot 3:42:37 PM
අපව සම්බන්ද කරගැනීම සදහා පහත ක්‍රම භාවිතා කල හැක.

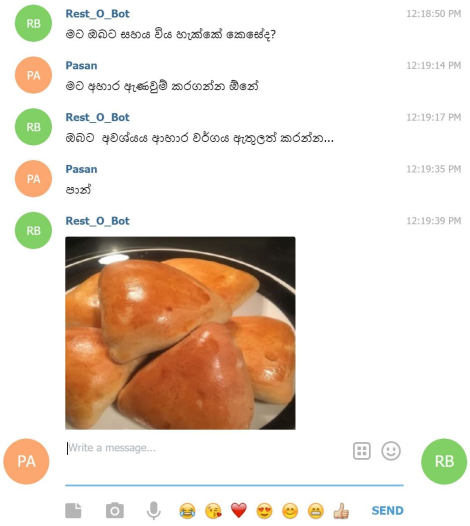
Land Phone 3:42:38 PM

034- XX XX XXX

E Mail 3:42:38 PM

Hello @abc.lk

Figure 9: Screenshot of the RestaurantBot 3



Rest_O_Bot 12:18:50 PM
මට ඔබට සහය විය හැක්කේ කෙසේද?

Pasan 12:19:14 PM
මට අහාර ඇණවුම් කරගන්න ඕනේ

Rest_O_Bot 12:19:17 PM
ඔබට අවශ්‍යය ආහාර වර්ගය ඇතුලත් කරන්න...

Pasan 12:19:35 PM
පාන්

Rest_O_Bot 12:19:39 PM

Figure 10: Screenshot of the RestaurantBot 4

function was successfully integrated into the Rasa NLU pipeline and further proposed alternative methodologies to achieve the same.

As we look forward, enhancing the NLP engine, especially in entity extraction, remains a pivotal area for further research and development. The proposed architecture provides a foundation for Sri Lankan organizations . With this, they can develop Sinhala chatbots equipped with NLP capabilities, negating the need to build chatbots from the ground up.

References

Adamopoulou, E., & Moussiades, L. (2020) Chatbots: History, technology, and applications. Machine Learning with Applications, 2, 100006.

Adiwardana, D., Luong, M.-T., So, D. R., Hall, J., Fiedel, N., Thoppilan, R., . . . Lu, Y. (2020) Towards a human-like open-domain chatbot. arXiv preprint arXiv:2001.09977.

Agarwal, A., Xie, B., Vovsha, I., Rambow, O., & Passonneau, R. J. (2011) Sentiment analysis of twitter data. Paper presented at the Proceedings of the workshop on language in social media (LSM 2011).

Aswani, N., & Gaizauskas, R. J. (2010) Developing Morphological Analysers for South Asian Languages: Experimenting with the Hindi and Gujarati Languages. Paper presented at the LREC.

Biswas, M. (2018) Microsoft Bot Framework Beginning AI Bot Frameworks (pp. 25-66): Springer.

Bocklisch, T., Faulkner, J., Pawlowski, N., & Nichol, A. (2017) Rasa: Open source language understanding and dialogue management. arXiv preprint arXiv:1712.05181.

Bradeško, L., & Mladenić, D. (2012) A survey of chatbot systems through a loebner prize competition. Paper presented at the Proceedings of Slovenian language technologies society eighth conference of language technologies.

Bullinaria, J. A. (2005) The roots, goals and subfields of AI. URL https://www. cs. bham. ac. uk/ jxb/IAI/w2. pdf.

Cowie, J., & Lehnert, W. (1996) Information extraction. Communications of the ACM, 39(1), 80-91.

Dale, R. (2016) The return of the chatbots. Natural Language Engineering, 22(5), 811-817.

Davis, E. (2016) AI amusements: the tragic tale of Tay the chatbot. AI Matters, 2(4), 20-24.

de Silva, N. (2015) Sinhala Text Classification: Observations from the Perspective of a Resource Poor Language.

de Silva, N. (2019) Survey on Publicly Available Sinhala Natural Language Processing Tools and Research. arXiv preprint arXiv:1906.02358.

Ekbal, A., & Bandyopadhyay, S. (2011) Named entity recognition in Bengali and Hindi using support vector machine. Lingvisticæ Investigationes, 34(1), 35-67.

Fitzpatrick, K. K., Darcy, A., & Vierhile, M. (2017) Delivering cognitive behavior therapy to young adults with symptoms of depression and anxiety using a fully automated conversational agent (Woebot): a randomized controlled trial. JMIR mental health, 4(2), e7785.

Gaikwad, S. K., Gawali, B. W., & Yannawar, P. (2010) A review on speech recognition technique. International Journal of Computer Applications, 10(3), 16-24.

Harnad, S. (2008) The annotation game: On turing (1950) on computing, machinery, and intelligence (published version bowdlerized).

Hettige, B., & Karunananda, A. S. (2006) First Sinhala chatbot in action. Proceedings of the 3rd Annual Sessions of Sri Lanka Association for Artificial Intelligence (SLAAI), University of Moratuwa.

Hildreth, C. R. (1997) The use and understanding of keyword searching in a university online catalog. Information technology and libraries, 16(2), 52.

Hodge, V. J., & Austin, J. (2003) A comparison of standard spell checking algorithms and a novel binary neural approach. IEEE transactions on knowledge and data engineering, 15(5), 1073-1081.

Isahara, H., Bond, F., Uchimoto, K., Utiyama, M., Kanzaki, K. (2008) Development of the japanese wordnet.

Jayaweera, M., & Dias, N. (2016) Comparison of part of speech taggers for sinhala language.

Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J. R., Bethard, S., & McClosky, D. (2014) The Stanford CoreNLP natural language processing toolkit. Paper presented at the Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations.

Nadkarni, P. M., Ohno-Machado, L., & Chapman, W. W. (2011) Natural language processing: an introduction. Journal of the American Medical Informatics Association, 18(5), 544-551.

Ngai, E. W., Lee, M. C., Luo, M., Chan, P. S., & Liang, T. (2021) An intelligent knowledge-based chatbot for customer service. Electronic Commerce Research and Applications, 50, 101098.

Nimavat, K., & Champaneria, T. (2017) Chatbots: An overview types, architecture, tools and future possibilities. Int. J. Sci. Res. Dev, 5(7), 1019-1024.

Poibeau, T. (2017) Chatbots: Machine translation: MIT Press.

Sarkar, S., & Bandyopadhyay, S. (2008) Design of a rule-based stemmer for natural language text in bengali. Paper presented at the Proceedings of the IJCNLP-08 workshop on NLP for Less Privileged Languages.

Services, P. E. (2019) Architecture for public service chatbots ISA2 Programme, 100. Retrieved from.

Shawar, B. A., & Atwell, E. (2007) Chatbots: are they really useful? Paper presented at the Ldv forum.

Shum, H.-Y., He, X.-d., & Li, D. (2018) From Eliza to XiaoIce: challenges and opportunities with social chatbots. Frontiers of Information Technology Electronic Engineering, 19(1), 10-26.

Singh, A. K. (2008) Named entity recognition for south and south east Asian languages: taking stock. Paper presented at the Proceedings of the IJCNLP-08 Workshop on Named Entity Recognition for South and South East Asian Languages.

Solutions, A. (2020) Chatbots: the definitive guide (2020). Recuperado de https://es. scribd. com/document/491470029/Chatbots-the-definitive-guide-2020-pdf.

Vasiliev, Y. (2020) Natural language processing with Python and spaCy: A practical introduction: No Starch Press.

Villegas-Ch, W., Arias-Navarrete, A., & Palacios-Pacheco, X. (2020) Proposal of an Architecture for the Integration of a Chatbot with Artificial Intelligence in a Smart Campus for the Improvement of Learning. Sustainability, 12(4), 1500.

Weerasinghe, R., Wasala, A., & Gamage, K. (2005) A rule based syllabification algorithm for Sinhala. Paper presented at the International Conference on Natural Language Processing.

Weerasinghe, R., Wasala, A., Herath, D., & Welgama, V. (2008) Nlp applications of sinhala: Tts ocr. Paper presented at the Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-II.

Welgama, V., Herath, D. L., Liyanage, C., Udalamatta, N., Weerasinghe, R., & Jayawardana, T. (2011) Towards a sinhala wordnet. Paper presented at the Proceedings of the Conference on Human Language Technology for Development.

Wijesiri, I., Gallage, M., Gunathilaka, B., Lakjeewa, M., Wimalasuriya, D., Dias, G., . . . De Silva, N. (2014) Building a wordnet for sinhala. Paper presented at the Proceedings of the Seventh Global WordNet Conference.

Xu, G., Wu, Z., Li, G., & Chen, E. (2015) Improving contextual advertising matching by using Wikipedia thesaurus knowledge. Knowledge and Information Systems, 43, 599-631.