



**A Sinhala chatbot for user inquiries
regarding Degree Programs at
University of Ruhuna**

**A dissertation submitted for the Degree of Master
of Computer Science**

U.E. Kumanayake

University of Colombo School of Computing

2019



Declaration

The thesis is my original work and has not been submitted previously for a degree at this or any other university/institute.

To the best of my knowledge it does not contain any material published or written by another person, except as acknowledged in the text.

Student Name: U.E.Kumanayake

Registration Number: 2015/MCS/042

Index Number: 15440421

Signature:

Date:

This is to certify that this thesis is based on the work of

Ms. U.E. Kumanayake

under my supervision. The thesis has been prepared according to the format stipulated and is of acceptable standard.

Certified by:

Supervisor Name: Dr. A.R. Weerasinghe

Signature:

Date:

Abstract

Computer based chat system is very popular in the modern world. It can be either Human-Human chat system or human-machine chat system. Human-Machine chat systems are well-known as chatbots. There exists some popular and advanced chatbots which are capable of performing almost naturally. But most of the chatbots are only available for English Language. But the set of population who are incapable of getting the maximum use with the available chatbots due to this language barrier can miss the chance of getting support from conversational agents/chatbots. And also they don't have to have a formal conversation when chatting with a bot, otherwise users may avoid chatting freely, due to the diffidence to ask questions.

This research targets on coming up with a solution as a Sinhala Chatbot applicable for a specific domain. A generic chatbot has a high chance to be not responding for most of the questions, making it unusable and ineffective. Therefore, when having a domain specific chatbot, the users can directly ask the questions and solve their problems referring the exact point without wasting the time searching the internet. So, developing '**A Sinhala chatbot for user inquiries regarding Degree Programs at University of Ruhuna**' mainly targets to solve two problems by being language specific (Sinhala) & domain specific.

Rather than a chatbot answering simple questions, this chatbot targets to identify the intent of the user asking the question and respond accordingly. Therefore, this chatbot can handle contextual conversations. That way the reliability & the acceptability of the responses become high with respect to a traditional rule based, template-based or keyword-based bot. The chatbot is developed using RASA platform which is an open source dialogue management tool. RASA Core and the RASA NLU are the two main components of this framework, where RASA Core provides its users for more advanced dialogues / chats. It allows training the dataset in many ways such as interactive learning and supervised machine learning. RASA NLU takes care of natural language processing tasks such as intent classification and entity extraction in chatbots/dialogue models.

Acknowledgements

This thesis is the result of me being fortunate to have the unconditional assistance of several people who have been extremely supportive in various ways. My First humble gratitude should be offered to Dr.

Ruvan Weerasinghe, my research Supervisor, for been extremely supportive and for the guidance given to me.

I thank my beloved parents for bringing me in to this world and guiding with the love and affection to fulfil my education.

And also, I want to thank my Batchmate, Mr. Lakal Nagasinha for the given support in collecting data for this research. Without your help it would be a very difficult task for me. I should thank all the students/parties who co-operated in collecting the dataset.

Finally, I like to thank all the people who had helped me throughout this project.

Contents

Chapter 1: Introduction	1
1.1. Statement of Problem	1
1.2. Motivation	3
1.3. Aims and objectives	3
1.4. Significance/ Novelty/ Contribution	4
1.5. Scope	4
1.6. Structure of the Dissertation	5
Chapter 2: Background	6
2.1. History of Chatbots	6
2.1.1. ELIZA	6
2.1.2. PARRY	6
2.1.3. ALICE	7
AIML - Example	7
2.1.4. Elizabeth	7
2.1.5. SmarterChild	8
2.2. Recent Chatbots	9
2.3. Sinhala Chatbots	10
2.3.1. Hettige, B. and Karunananda, A. Chatbot	10
2.3.2. GIC Chatbot	11
2.4. The Chatbot System	11
2.4.1. Intent Classification	12
2.4.2. Entity Extraction	13
2.5. Available Machine Learning Tools for Chatbots	14
2.5.1. RASA STACK	14
2.5.2. SNIPS NLU	15
2.5.3. NLULIt	15
2.5.4. SyntaxNet	16
2.5.5. DialogFlow	16
2.5.6. LUIS	16
Chapter 3: Analysis & Design	17
3.1. Data Set	17
3.2. Proposing model/design	18
3.3. Methodology	19
Chapter 4: Implementation	21
4.1. Used tools	21

4.2. Using RASA to implement the chatbot	23
4.2.1. Intent Classification & Entity Extraction of Training Data	25
4.2.2. RASA Pipelines	25
4.2.3. Entity Extraction	27
4.2.4. Custom Entities	28
4.3. Formatting/Structuring the Dataset	29
4.3.1. Markdown Format	29
4.4. Actions	31
4.4.1. Default Actions	31
4.4.2. Utterance Actions	31
4.4.3. Custom Actions	32
4.4.4. Fallback Action	33
4.5. Slot filling	34
4.6. Stories	34
4.7. Important Code Snippets	35
4.4.1. Training the Bot – Online	35
4.4.2. Running the ChatBot	35
Chapter 5: Evaluation	36
5.1. Evaluation Approach 1	36
5.1.1. Evaluating Models	38
5.1.2. Evaluating Intent Classification	38
5.1.3. Classification Report	39
5.1.4. Confusion Matrix	40
5.1.5. Intent Prediction Confidence Distribution	41
5.1.6. Evaluation of Entity Extraction	42
5.2. Evaluation Approach 2 – User Evaluation	43
5.3. Chapter Summary	43
Chapter 6: Future Works & Conclusion	44
6.1. Deploying the Chatbot	44
6.2. Conclusion	44
6.2. Future Works	44
References	46
Appendix	

List of Figures

Figure 1.....	10
Figure 2[8].....	13
Figure 3.....	14
Figure 4 – Rasa Architecture.....	23
Figure 5.....	25
Figure 6.....	35
Figure 7.....	35
Figure 8 – Intent Classification Example	37
Figure 9 - Classification Report of Intents.....	40
Figure 10 – Confusion Matrix	41
Figure 11 – Intent Prediction Confidence Distribution.....	42
Figure 12- Classification Report of the Entity Extraction.....	43

Acronyms & Abbreviations

NLU - Natural Language Understanding

NLP - Natural Language Processing

RASA NLU - An Open Source NLU tool

SNIPS NLU - An Open Source NLU tool

API - Application Programming Interface

ML - Machine Learning

AI – Artificial Intelligence

FAQ - Frequently Asked Questions

SVM – Support Vector Machines

NER – Named Entity Recognition

CRF – Conditional Random Fields

SDK – Software Development Kit

Chapter 1: Introduction

Developing a Sinhala chatbot with the ability of identifying the user intent and extracting the key entities in a sentence has been a target of this study.

This chapter explains the Statement of Problem, Motivation, and Aims & Objectives along with the Significance/Novelty/Contribution. Then it explains the scope of the Research. The chapter concludes by providing the Structure of the Dissertation.

1.1. Statement of Problem

It is identified that people gradually tempt to chat using any social media or available platforms rather than personally speaking or having phone calls. Therefore, text communication has become socially acceptable form of personal interaction. With this truth chatbots could gain a considerable attention, due to users' attraction & demand for those chat agents. A chatbot which is a software tool that is built-in with the capability to communicate with people using their natural language. Chatbots are used for various purposes, while their key target remains to identify users' utterances and respond accordingly.

Some of the industries that take advantage of the chatbots are E-Commerce, Insurance, Health Care, and Retail Hospitality & Logistics. And the business functions that chatbots can be mostly applicable are Customer Service, Sales/Marketing, order processing and Social Media [13]. Most of the times chatbots are trained to answer the Frequently Asked Questions (FAQ) in a corresponding domain.

There are reasons why enterprises, banks, and financial institutions are tempting to use Chatbots. Mainly they can reduce cost in providing information to their clients, to improve speed and the success rate. One extra benefit they gain by using chatbots they can gather valuable information from customers/ the bot users and experiment with new ideas.

It is beneficial to keep track of the chatbot conversations, since it can be a “new transaction” or “business idea”. It is not the important thing that each conversation should go ahead with a sales purpose, but to consider the vast amount of knowledge they can collect on their customers desires, favorations, ‘look-for items’, preferences, colors, designs, etc. This knowledge can be used, to improve the business products, services, etc. And also the

customers may provide their feedback on their experience, whether it is great or bad; the feedback will be useful to impact future sales of the business/product. Other than that, customers chatbots as an enhanced/advanced feature of a company. If the chatbot is designed correctly, chatbots eliminate repetitive tasks – making workloads lighter, easier, and faster – while promoting greater customer satisfaction for the organizations that deploy them.

And also if the waiting time for a bot's response is high, the users experience with chatbots becomes less worthy and user may get frustrated. Therefore no matter whatever processing inside has to carry on to understand the user message and produce an appropriate response, the waiting time for the user should be minimum. Otherwise users may become unsatisfied. With the growth in the fields of Machine Learning and Artificial Intelligence, advanced chatbots have been built up. But since most of the chatbots are only available for English Language, there exists a set of population who can't get the maximum use with the available chatbots due to the language barrier. When chatting with a bot using native language the amount of difficulty in communicating becomes low, since they don't have to engage in a formal conversation. Otherwise users may avoid chatting freely, due to the diffidence to ask questions. And also still there are no available chatbots developed for Sinhala Language targeting a specific domain.

A generic chatbot has a high chance of not being responding for most of the questions, making it unusable. Therefore when having a domain specific chatbot, the users can directly ask the questions and solve their problems referring the exact point without wasting the time searching the internet. So developing 'A Sinhala chatbot for user inquiries regarding Degree Programs at University of Ruhuna' mainly targets to solve two problems by being language specific (Sinhala) & domain specific. This bot will be helpful for the outside community/students to get the information on degree programs at Faculty of Technology, University of Ruhuna. So the target audience for this chatbot will be the Advanced level technology stream students, their parents and any interested parties on the degree programs. But this solution should be applicable in any domain, only the dataset and some custom components being replaced according to the corresponding domain.

Mostly, people tend to use simple sentences in chat systems. Because of that language complexity issue won't be a huge issue in chatbot systems, like in detailed discussions. This chance comes for the Sinhala chatbot as well. And also, if the user utterance is not up to bots capacity of identifying the intent, bot can request user to rephrase what is said previously. This build ups the realistic nature of the conversation by enhancing the interactivity until user

receives a satisfactory response. Furthermore, chatbot does not need to bother about complex written grammatical structures of a language, but simple verbal grammar would be sufficient for most of the instance [2].

The two main types of bots are retrieval based and generative based. A retrieval based bot uses a predefined set of responses and a heuristic to select the most suitable response for a user input. A generative model behaves conversely to a retrieval based bot and tries to convert from a given user input to an output by generating a fitting response by avoiding the use of a predefined set of responses.

Nowadays, a considerable number of companies try to have a chatbot integrated into their daily processes. According to the recent studies, implementation of chatbots can be successful in order to automate certain processes related to education, information retrieval, business, e-commerce and amusement. Companies like Pizza Hut, Wholefoods and HBO have implemented chatbots as a way to interact and guide customers through their customer service [14].

1.2. Motivation

Nowadays conversational AI has become much popular, mostly in the field business as a marketing strategy. The support of real agent is not always available where the bots are always available. Due to the importance of this conversational agents, it is evolving faster, with all latest machine language capabilities included. With the advancements in NLU, the ability and the capacity of a chatbot could also be enhanced.

Even Though there exists plenty of chatbots developed for English language, no enough research work has been done for developing Sinhala Chatbots. And since chatbots has become much popular nowadays, A Sinhala Chatbot will be a very useful application. It becomes highly useful in local business industry since Sinhala is the conversational language used in this.

1.3. Aims and objectives

The main goal of this research is to develop a Sinhala Chatbot for user inquiries regarding degree programs at Faculty of Technology, University of Ruhuna. So that it can serve as an effective conversational agent.

The key objectives are to fill the gap of language barrier in chatbots by making it available for Sinhala Language & enable the chatbot to understand the in-domain questions & respond accordingly.

1.4. Significance/ Novelty/ Contribution

Previous work on Sinhala Chatbots doesn't cover a specific domain. This chatbot will be covering a specific domain only.

Since the work carried out in Natural Language Processing field for Sinhala language is in considerable stage, the novel findings could be used in developing an effective Sinhala Chatbot.

1.5. Scope

Having a generic chatbot will be useless since a chatbot cannot be trained for almost all the domains. Therefore the effectiveness of a generic chatbot can be less.

The knowledge base of the chatbot that is implemented in this study is limited to the information about the degree programs at Faculty of Technology, University of Ruhuna.

Therefore the chatbot is not expected to generate successful responses for questions those are out of scope. It will produce a common response, saying it is not aware of instead.

The importance of using a narrow scope is producing successful responses for in scope questions.

And also the chatbot will accept only the user inputs in Unicode & not the singlish format.

Yet the bot will accept English terms in the user utterances. For an example

ET වලට නියෝගේ මොන subjects ද?

Rather than expecting the exact Sinhala Wordings for the terms such as ET, it is user friendly when we train the bot to understand such terms. But this will be limited to set of terms that is in this scope. And the bot may not be able to understand all the English terms/abbreviations in user utterances.

And also it is not expected that the users will be dealing with the chatbot using written Sinhala grammar. So the bot is only trained for spoken Sinhala sentences.

Assumptions - It is assumed that the users ask independent questions that do not rely on the context of the previous questions (Ex: එය (It), එම (It), අර (that), ඔහු (he)).

1.6. Structure of the Dissertation

Chapter 2 surveys the background, related researches and the tools and technologies used in the literature. Chapter 3 outlines the system analysis and design of the proposed prototype. Chapter 4 describes the implementation of the prototype and the chosen tools and technologies. In chapter 5 evaluation and testing procedures, results, analysis and interpretations are presented. Chapter 6 discusses the conclusions, limitations, improvements and future works of the study.

Chapter 2: Background

The chatbots have been a most popular mechanism to serve as an agent for any organization. And developing a chatbot that serves perfectly well has become one of the toughest challenges ever since the birth of AI.

In this chapter first focus is offered to look into the History of Chatbots. Then about the Recent Chatbots and also about the available literature on Sinhala Chatbots. The structure of a chatbot is identified next together with NLU chatbot elements.

At the end of the chapter a quick look into the Available Machine Learning Tools for chatbots is reviewed.

2.1. History of Chatbots

2.1.1. ELIZA

ELIZA is an early Artificial Intelligent chatbot program used to simulate a nondirective psychotherapist. It uses clever handwritten templates to generate replies that resemble the user's input utterances. Input sentences are analyzed based on decomposition hand-written rules, which are triggered by key words appearing in the input text. ELIZA had very limited natural language processing capabilities.

2.1.2. PARRY

PARRY is an early output as a chatbot, and it was implemented in 1972 by psychiatrist Kenneth Colby. It has been implemented with the attempt to simulate a person with a mental disorder called paranoid schizophrenia. The basis for the crude model that is developed for this chatbot is the behavior of a person with paranoid schizophrenia such as concepts, conceptualizations and beliefs.

Real patients and computers running PARRY were analysed by psychiatrists and the transcripts of the conversations were given to another set of psychiatrists to identify which of the patients were real and which were of computers'. They were only capable of successfully identifying 48% by guessing.

2.1.3. ALICE

ALICE (Artificial Linguistic Internet Computer Entity) is a software robot or program that one can chat with using natural language. ALICE uses pattern-matching algorithm to identify user input and this algorithm uses depth-first search techniques. And it has passed the Turing test in two consecutive years. Artificial Intelligence Markup Language (AIML) is used in developing ALICE. And the latest derivative using this language is built upon pandorabots platform and the famous chatbot is "Mitsuku".

AIML - Example

A simple example on using AIML is shown below.

```
<category>
    <pattern>What is your name?</pattern>
    <template>My name is Alice</template>
</category>
```

If the input sentence from user, matches the sentence in between <pattern> brackets, the outputted reply would be the sentence in between <template> brackets.

The next example below shows the usage of '*' (star) symbol for replacing words. In this case whatever word follows the word like it will be present in the response at the position specified by the <star/> token:

```
<category>
    <pattern>I like *</pattern>
    <template>I too like <star/></template>
</category>
```

2.1.4. Elizabeth

Elizabeth is known as an adaptation of Eliza. Elizabeth stores knowledge as a script in a text file, where each line is started with a script command notation. Elizabeth has the ability to produce a grammar structure analysis of a sentence using a set of input transformation rules to represent grammar rules.

With respect to the above early approaches, those are the primary level of the chatbot history. Most of the early chatbots had been based on 'Keyword Based' or 'Template Matching'. And also some chatbot architectures also rely on 'Rule-Based' or simple statistical mechanisms

[3]. The hand-written rules may not work well for all cases. According to Kumar Shridhar, Co-Chief AI Scientist at BotSupply, “In a rule-based approach, a bot answers questions based on some rules on which it is trained on. The rules defined can be very simple to very complex. The creation of these bots are relatively straightforward using some rule-based approach, but the bot is not efficient in answering questions, whose pattern does not match with the rules on which the bot is trained”.

And also the template matching approach limits the bots’ ability to respond only for user inputs defined in pre-defined templates. Since the languages today are very complex, it is an impossible task to manipulate templates for almost all user inputs. So most probably there’s a high chance for bot to be not able to output a valid/useful response to most of the user inputs. Since Sinhala is an agglutinative language, it has a high rate of affixes or morphemes per word which makes the above early approaches are impractical to use sole.

Most of the early chatbots had been based on ‘Keyword Based’ or ‘Template Matching’. And also some chatbot architectures also rely on ‘Rule-Based’ or simple statistical mechanisms [3]. And current dominating models for building conversational agents are based on Neural Networks. There are many advanced chatbots developed for English Language.

Due to those limitations in presence, the huge necessity to replace the traditional approaches was emerged. And that place was taken by the neural network based approaches. Neural networks have been the backbone of conversational modeling and traditional machine learning methods are only rarely used as supplementary techniques.

The main difference between traditional rule-based and neural network-based approaches is the presence of a learning algorithm in the latter case. Instead of using hand-written rules deep learning models transform input sentences into replies directly by using matrix multiplications and non-linear functions that contain millions of parameters.

2.1.5. SmarterChild

The next step in the history of chatbots is when a transition was made to messaging platforms outside of standalone applications. A most well-known example of this is SmarterChild, a sassy chatbot that launched in 2001. Closer to the experience we have with chatbots today, SmarterChild was ahead of its time. It brought NLP to SMS networks and AOL Instant Messenger and served as many users’ introduction to chatbots.

In addition to making conversation, SmarterChild carried out many useful tasks, providing news, weather, stock info, sports scores and much more. This demonstrated and popularized chatbots' potential to serve as sophisticated digital assistants, all via popular messaging channels. It's also worth noting that SmarterChild was only one of many chatbots developed by Active Buddy, who was eventually acquired by Microsoft. Active Buddy offered many promotional bots as well (including an Austin Powers-themed bot), changing the chatbot definition of the time by establishing them as novel marketing tools, not just conversation partners.

2.2. Recent Chatbots

Recent advances in machine learning have greatly improved the accuracy and effectiveness of natural language processing, making chatbots a viable option for many organizations. This improvement in NLP is firing a great deal of additional research which should lead to continued improvement in the effectiveness of chatbots in the years to come.

A simple chatbot can be created by loading an FAQ (frequently asked questions) into chatbot software. The functionality of the chatbot can be improved by integrating it into the organization's enterprise software, allowing more personal questions to be answered, like "What is my balance?", or "What is the status of my order?"

Most commercial chatbots are dependent on platforms created by the technology giants for their natural language processing. These include Amazon Lex, Microsoft Cognitive Services, Google Cloud Natural Language API, Facebook DeepText, and IBM Watson. Platforms where chatbots are deployed include Facebook Messenger, Skype, Slack, Twitter, Kik, WhatsApp, and Viber among many others.

The chatbots today are equipped with deep learning approaches. Deep learning models transform input sentences into replies directly by using matrix multiplications and non-linear functions that contain millions of parameters. Neural network based conversational models can be further divided into two categories, retrieval-based and generative models.

The former simply returns a reply from the dataset by computing the most likely response to the current input utterance based on a scoring function, which can be implemented as a neural network [11] or by simply computing the cosine similarity between the word embedding of the input utterances and the candidate replies. Generative models on the other hand synthesize the reply one word at a time by computing probabilities over the whole vocabulary

[12]. There have also been approaches that integrate the two types of dialog systems by comparing a generated reply with a retrieved reply and determining which one is more likely to be a better response.

Some of the most popular chatbots nowadays are Apple's Siri [4], Google's Google Assistant & Microsoft's Cortana

The Figure 1[7] depicts how the Google Assistant works.

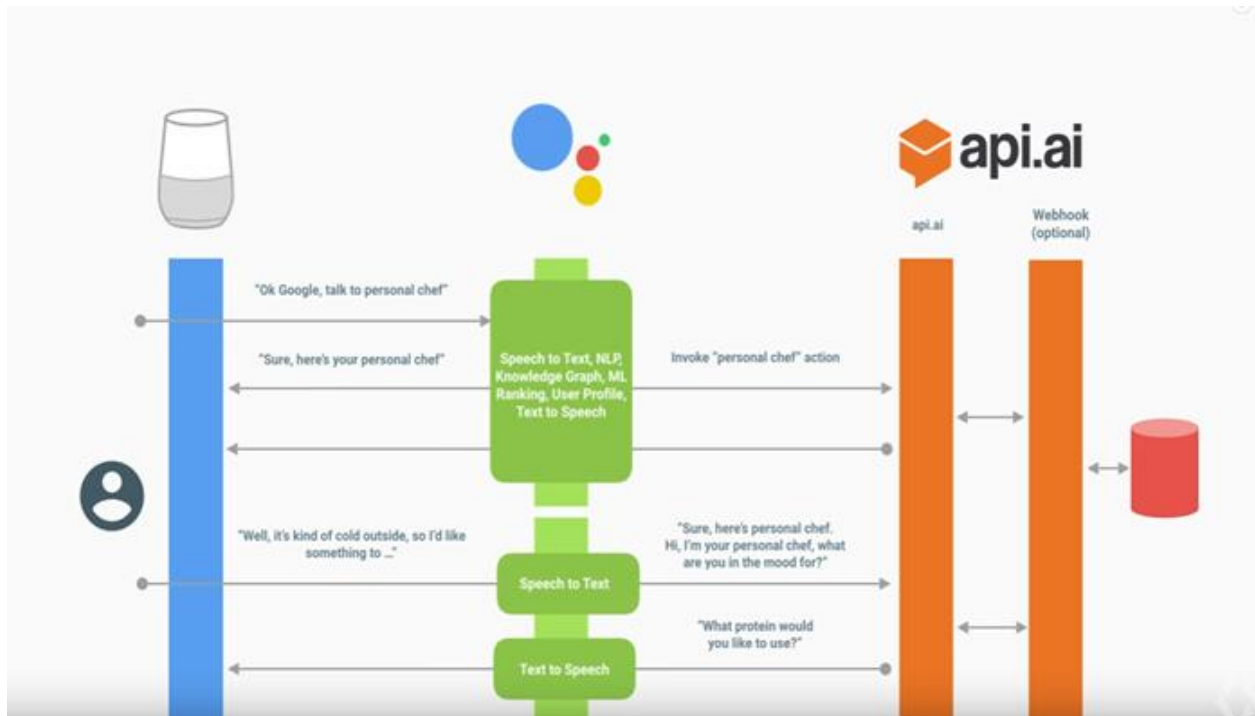


Figure 1

2.3. Sinhala Chatbots

2.3.1. Hettige, B. and Karunananda, A. Chatbot

The first Sinhala chatbot is implemented in 2006, which is designed to answer simple questions and not designed based on a particular domain. It is a prototype of a Chatbot system that uses a Sinhala language parsing system which consists of Morphological analyzer, Sinhala parser, Sinhala composer and Lexical dictionaries. This system is developed using Java and SWI-Prolog that runs on Linux and Windows environments. This system is designed to work on the client server model, where the server contains all the resources and engine modules by allowing client to access the information through the network. The client server architecture used with the purpose of giving access to many people to find the information through the Chatbot system at the same time.

Knowledge identification engine is used to identify appropriate pattern to find the suitable answer. The system can identify subject, object and verb in a sentence. Then the patterns are stored by using spattern/3 Prolog predicate. This uses the standard Prolog matching and unification procedure to find suitable answers. [2].

So, it can be concluded that this research is done to develop the whole structure that is needed

2.3.2. GIC Chatbot

This is a button input based chatbot embedded in Government Information Center website and is available for two languages English & Sinhala.

In this chatbot there is a pre-defined set of button-like options, a user can choose from. It helps to provide a much faster experience to the customers as they don't have to type a single query, instead, they can just choose a button like an option given to them. But the limitation of this kind of chatbot is users being restricted to limited options, therefore only limited scope is covered. The bot user can't freely ask questions and have to be restricted to a set of information intents.

There are two main tasks a bot has to perform in order to be able to reply for a user message. Those are Intent Classification & Entity Extraction which will be discussed in the subsequent sections of this Literature Review. There are already built apis to perform above two tasks such as Wit.ai & LUIS. Those tools make it easier to build chatbots by performing two major tasks especially for English language. But the usage of APIs can cause to the slowness of the application, since https calls are slow, and users will be always constrained by the design choices that went into the API endpoints. And also there is a risk of the used Libraries being hacked. So there can be a security risk as well. And the whole dataset has to be sent to a third party in order to process it. The general-purpose APIs have to do well on every problem, but we only have to do well on ours [1].

2.4. The Chatbot System

A chatbot system should consist of following 3 features.

- Computer-based of Natural Languages Understanding.

What should be included in this part is a mechanism for the bot to understand what user is asking for. In the first-place bot should be able to understand if the user asks for some

information from the bot or user provides some information. Once it is distinguished, the intent of the user query must be grabbed. And, there can be specific parameters/entities in the user query to be extracted to produce a proper response to the user.

- Define and design knowledge base for the chatbot

Once the user utterance is identified by the bot, it must produce a proper respond back to the user. This is where the knowledge base comes into action. There should be a well-organized knowledge base, to retrieve the necessary outputs for the user.

- Develop suitable algorithms for pattern matching.

In a set of user utterances there can be a specific format that lies for a certain request to be made or response to be provided. If such distinguish format can be identified then it will be useful mostly in Entity Extraction of that user utterance. The well-known procedure for such pattern matching is using Regular Expressions. The NLU frameworks like RASA also allows regular expressions to be defined.

2.4.1. Intent Classification

Intent Classification can be specified as one of the major tasks of a chatbot. This targets to identify what does the user want (recognize the intent)? Figure 2 shows how the user inputs are classified by the intent of that input.


Intents	Example phrases
 Check prices	<ul style="list-style-type: none"> • How much is your best road bike? • What does a tune-up cost? • How much is it for repair?
 Book a repair	<ul style="list-style-type: none"> • I'd like to get my bike fixed! • My mountain bike is broken. • Can you fix my bike this week?
 Get opening hours	<ul style="list-style-type: none"> • What are your hours? • How late are you open? • When do you close?

Figure 2[8]

2.4.2. Entity Extraction

In Entity Extraction we try to identify the things users mention.

Figure 3 explains how entity extraction is done in DialogFlow [8]. According to DialogFlow, it extracts user inputs “11am”, “tomorrow”, “California”, “\$40” as sys.time, sys.date, sys.geo-state-us and sys.unit-currency entities respectively.

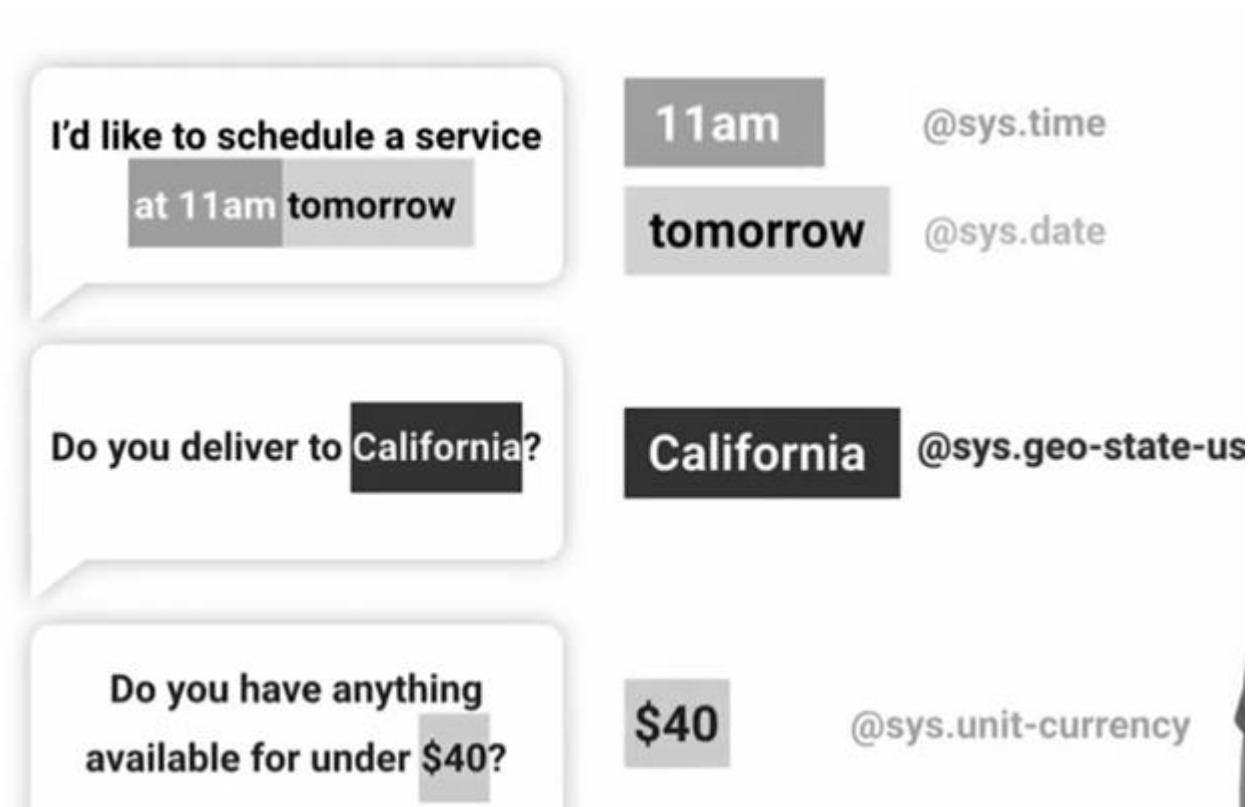


Figure 3

2.5. Available Machine Learning Tools for Chatbots

The main possibility of a chatbot should be Natural Language Understanding (NLU). Following tools/platforms are some examples for the tools that comes with Intent Classification and Entity Extraction. There are only few open source Natural Language Understanding Tools such as RASA NLU and SNIPS NLU where most of the others comes with a premium version for enhanced tools.

2.5.1. RASA STACK

RASA Stack provides Machine learning libraries for developers to create contextual AI assistants and chatbots that go beyond simple questions. Open source natural language processing and dialogue management that is supported by thousands of community members[18].

It comes with two main components called, RASA Core & RASA NLU where RASA Core acts as a chatbot framework with machine learning-based dialogue management and RASA NLU is a library for natural language understanding with intent classification and entity extraction.

Both of these components work independently. RASA Core can be used with other NLU frameworks and RASA NLU can be used with other dialog management frameworks/tools.

When a user message is received, RASA NLU is responsible for understanding the user's message based on the previous training data. This is done using two mechanisms, which is called Intent Classification & Entity Extraction. Intent classification means interpreting the meaning based on predefined intents. The intent returned with maximum confidence rate will be chosen as the intent of a user utterance. Entity Extraction is for recognizing structured data (the entities and their values).

Next is where RASA Core comes into action. Core decides what happens next in the conversation. Its machine learning-based dialogue management predicts the next best action based on the input from NLU, the conversation history and the training data.

Since this is the platform used in this research, it will be further explained later.

2.5.2. SNIPS NLU

Snips NLU is a Natural Language Understanding open source python library that allows to parse sentences written in natural language, and extract structured information [6].

When it is properly trained, the Snips NLU engine will be able to extract structured data. So the same approach is possible just like in RASA NLU with this. That is identifying the intent and extracting the entities. It supports both Builtin entities and Custom entities.

But the limitation why SNIPS NLU cannot be used in this research is it require language resources available for the language we are going to use & currently no language resources available for Sinhala Language.

2.5.3. NLULite

NLULite is a developer-friendly database that can read texts and answer questions about them, by combining a natural language parser and a graph database.[15]

2.5.4. SyntaxNet

SyntaxNet is a TensorFlow toolkit for deep learning powered natural language understanding (NLU)[16].

2.5.5. DialogFlow

Dialogflow is a Google-owned developer of human–computer interaction technologies based on natural language conversations. The company is best known for creating the Assistant, a virtual buddy for Android, iOS, and Windows Phone smartphones that performs tasks and answers users' question in a natural language. This NLP framework gives the opportunity to build conversational interfaces on top of products and services by providing a powerful natural language understanding (NLU) engine to process and understand natural language input [9].

2.5.6. LUIS

Language Understanding Intelligent Service (LUIS) by Microsoft Azure offers a fast and effective way of adding language understanding to applications [10].

Chapter 3: Analysis & Design

3.1. Data Set

The dataset plays a main role in the successive of the chatbot. The bot requires considerable dataset for training, so that it can successively predict the intent for the test/other user utterances.

Target Audience: Target Audience of this research is the Advanced Level Students of Technology Stream & the university students following Technology degree programs. They are the highly interested parties to know about the degree programs of Faculty of Technology, University of Ruhuna. The collected data is taken into a single excel sheet. Then the whole dataset is analyzed to identify all the intents & entities. Then, this full data set is manually tagged with the intent and the entities. This is how the dataset is organized to be used for the Sinhala Chatbot.

Collecting the Dataset Using a printed leaflet, the target audience was asked to note down the questions in a paper which are related to the domain. And the medium they were asked to write the questions is Sinhala. This mechanism seems effective, rather than interviewing them, since the exact way they would ask the questions from the chatbot is written in the leaflet.

Collected dataset is first taken into an excel sheet. Then the whole dataset is analyzed to identify the intents and the entities for the collected dataset. Some questions had to be considered as out of domain; since we only consider the questions related to the domain “The degree programs in Faculty of Technology, University of Ruhuna”. And also, some questions required further information in order to produce an answer.

For an example, they ask “උපාධිය අවසානයේ පිවිසිය හැකි රැකියා මොනවාද?”. In this case the Degree or the Stream is not defined. Therefore, the bot can't produce an answer in this state without knowing the Degree or the Stream. Due to the mechanism the data is collected, it is unable to collect the information in an exact manner. For such situations we are training the bot using interactive way, to generate the user stories that deals with gathering those information.

The strength of a chatbot model is that the response will be of correct grammar, and most likely relevant to the user. The weakness is that an open domain make collecting responses an impossible task. Therefore, in this research only a selected domain will be considered.

The standard language for all API solutions for chatbots is English, and those who support other languages only provide a Beta-version in the available languages. This limits the API solutions available if one would like to implement the chatbot in a different language.

Furthermore, it also affects the quality of the chatbot. The NLP system is lacking much of the training and its information retrieval and prediction will therefore not perform with equal quality as for other languages.

And also the chatbots' prediction algorithm is based on ML, which means that the more data it trains on, the more accurate predictions it is able to perform. This property entails that the more user input the chatbot gets, the more data it bases its predictions on. This results in that it will not only be able to better distinguish the context of an input, it will also dynamically change its predictions if the user inputs changes.

As mentioned, since the prediction algorithm is based on ML, which requires training to recognize and retrieve key information from user input. To develop a robust chatbot, which can handle a variety of inputs, the system therefore needs a data set to train on, which should be equivalent of the user input. To obtain this data, it is practically difficult to interview the users directly and keep a conversation on the domain to gather data. Therefore users were requested to submit their questions in written form. This may miss the realistic nature of a conversation in the dataset.

3.2. Proposing model/design

Every potential user input is marked with its entities and intents which is necessary for the bot to make future predictions. The proposed chatbot model is based on Intent Classification & Entity Extraction. In Intent Classification, we determine the target/intent of the sentence, whilst Entity extraction targets to extract the parameters, a.k.a. the *slots* of the sentence.

The two terms will be demonstrated using a suitable example.

Intent

In the context of information extraction, an *intent* corresponds to the action or intention contained in the user's query, which can be more or less explicit.

Lets' consider for instance the following sentences:

මෙම පාඨමාලාවල පුහුණු කාලසීමාව කොපමණද?

පාඨමාලාවලට අමතරව පුහුණු කාලසීමාවක් තිබේද, තිබේ නම් කොපමණද ?

Intent : getTrainingPeriod

Entity : පුහුණු කාලසීමාව

They both express the same intent which is getTrainingPeriod, but they are expressed in two different ways.

Thus, the first task in intent parsing is to be able to detect the *intent* of the sentence or say differently to classify sentences based on their underlying *intent*.

Entities

Once the intent is known, next task is to extract the parameters that may be contained in the sentence.

For example, let's consider this sentence:

තොරතුරු තාක්ෂණ පාඨමාලාවේ පුහුණු කාලසීමාව කොපමණද?

There is an additional piece of information which is contained in the words තොරතුරු තාක්ෂණ පාඨමාලාවේ.

This intent contains one slot, which is the *course name* of which the training period is asked.

This information is known as the 'Entity'. The specialty and the importance of identifying the entity is, different answers must be produced for different entity values of the same intent.

Therefore, the entity extraction plays a major role in this chatbot development.

3.3. Methodology

- ❖ Firstly, the dataset is gathered from the target audience.
- ❖ A leaflet is given, and written questions will be collected.
- ❖ The questions/ the gathered dataset is used to identify & annotate the intents & the entities for the chatbot to train.
(And a set of data will be kept for testing the chat bot.)
- ❖ After annotating the dataset, the chatbot is trained using Machine Learning.
- ❖ In this study, RASA NLU Apis will be chosen as the platform for implementing the chatbot.

- ❖ Then for a new data/sentence/question the chatbot will generate a response using already seen data.

Chapter 4: Implementation

Since there exists plenty of frameworks/APIs/Tools either Open Source or Commercial, developing a chatbot from the scratch is no more essential. A chatbot based on. Going beyond from developing simple chatbots such as conditional/template based, nowadays there are frameworks which support identifying the user intent and extracting the entities. This is known as Natural Language Understanding (NLU) which is a subtopic of Natural Language Processing (NLP).

In this research also, a framework which is handy with NLU is used, RASA NLU for the intent classification and entity extraction. And RASA Core is used as the responsible framework for the handling the conversation.

4.1. Used tools

Language: Python 3.6

Frameworks/Libraries:

➤ **RASA Core**

Version: 0.13.0

Rasa is a framework for building conversational software. Instead of dealing with set of if/else statements, Rasa uses a machine learning model to build up the logic of the bot and bot is trained on example stories/conversations.

➤ **RASA NLU**

Version: 0.14.0

Rasa NLU is an open-source natural language processing tool for intent classification and entity extraction in chatbots.

➤ **RASA SDK**

Version: 0.12.2

This is the SDK used to develop custom actions in Rasa. The action server is hosted at <http://localhost:5055/webhook>

Chat Platform: Slack

RASA Core provides the feature to integrate third party chat platforms. In this research Slack will be used as the chat platform. Slack allows interactive elements to be used

Anaconda Navigator

Version: 1.9.6

Anaconda Navigator is an open source data science distribution. It provides features for the programming languages R and Python, Anaconda is a product of the manufacturer of Anaconda Inc. It serves as a one point access for the administration of all necessary packages, many of which are already preinstalled, tools, libraries, and their dependencies can be installed and faster and easier. Another important feature Anaconda provides is it allows keeping any number of different environments, based on python version. This became really useful, since Rasa itself improved its versions, periodically. With those version changes the bot can be checked for more features and improved accordingly, by keeping multiple Rasa Environments.

While running the bot in one environment of Anaconda with rasa core and rasa nlu, another environment is set up to run the Action server that is required to run the custom actions. In this subsequent environment rasa-sdk is installed.

Rasa also recommends the use of Anaconda when using the Rasa Core component.

SQLite Database

Version: 3.28.0

SQLite is a small, fast, self-contained, high-reliability, full-featured, SQL database engine. Since it is not practical to cover the whole domain information in direct message texts, we use custom actions that access the database to retrieve information. Since SQLite is a light-weight database management system it will use minimum time to return the requested information.

Ngrok For Local Testing

Ngrok allows to create a local webhook from the machine that the bot is deployed in; so that it can be publicly available on the internet and can be used with applications like Slack, Facebook, etc.

4.2. Using RASA to implement the chatbot

Figure 4 shows, how a chatbot built up using the Rasa Stack responds to a user message, in a step wise manner.

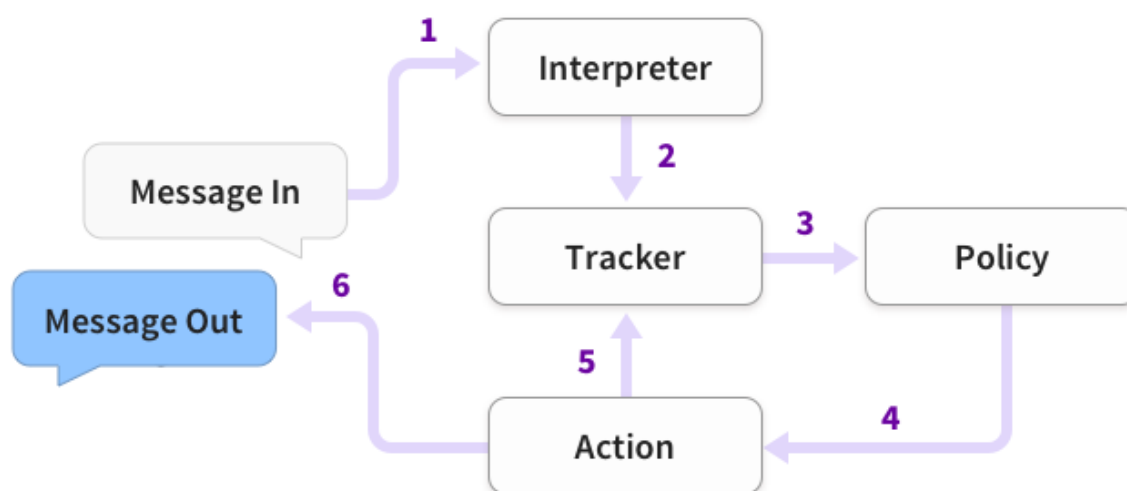


Figure 4 – Rasa Architecture

The process how a user receives a response, for a given input is described below.

Step 1 – The '**Interpreter**' receives the message. Then it converts the user input in to a dictionary format by adding the original message, intent together with the entities found. Rasa NLU takes care of this step.

Step 2 – Separate Object called '**Tracker**' is maintained, on order to follow up on the conversation state. Once a new message is entered it gets to know about that.

Step 3 – The current state of '**Tracker**' is then sent to '**Policy**'.

Step 4 – It is upto '**Policy**' to decide what to do next.

Step 5 – The action chosen by the ‘**Policy**’ is then logged by the ‘**Tracker**’.

Step 6 – Finally response is sent to the user.

Intents for the Chatbot

When identifying intents for a particular domain, it is important to cover almost all the areas in the domain. Otherwise the bot will not be outputting effective responses. We have gathered domain specific questions from nearly 100 students, so that we are able to cover most of the areas in this domain.

The intents used in this chatbot are recognized as below. When finalizing/selecting the intents overall dataset is considered. The average consideration/interested level of the target users to certain areas has been considered when selecting the intents. And also if there is a variety of ways to ask for specific point, it will be selected as an intent.

To handle out of scope/out of domain messages in user utterances we have defined a separate intent as ‘out_of_domain’. And, we have added sample messages from our training dataset that are just bogus or non-relevant for the domain that is labelled as ‘out_of_domain’.

Altogether here we represent 13 intents in our selected domain.

- greet
- thank_you
- ask_course_duration
- ask_job_opportunities
- ask_course_requirements
- ask_internal_conditions
- ask_course_content
- provide_self_info
- ask_information
- ask_degree_recognition
- out_of_domain
- ask_medium

For each of the above intents minimum of 10 user utterances/examples are available in the training data set.

4.2.1. Intent Classification & Entity Extraction of Training Data

Figure 4 shows how the intents and the entities of two sentences are organized in a json object for RASA NLU to be used in training.

```
{
  "rasa_nlu_data":
  {
    "common_examples": [
      {
        "text": "වසර 4ක් ම උසාඩි සාට්‍මාලාව හැලවීම අනිවාර්යද?",
        "intent": "ask_mandatory_course_duration",
        "entities": [
          {
            "start":0,
            "entity":"duration",
            "value": "වසර 4ක්",
            "end":6
          }
        ]
      },
      {
        "text": "වසර 3 කින් උසාඩිය නිම කර නොහැකිද?",
        "intent": "ask_possibility_course_duration",
        "entities": [{
          "start":0,
          "entity":"duration",
          "value": "වසර 3 කින්",
          "end":8
        }]
      }
    ]
  }
}
```

Figure 5

4.2.2. RASA Pipelines

Selecting an Appropriate Pipeline/Component for Intent Classification

There are two main available components to perform Intent Classification in RASA NLU. Those are,

- **Pretrained Embeddings (intent_classifier_sklearn)**
- **Supervised Embeddings (intent_classifier_tensorflow_embedding)**

The biggest difference between them is that the pretrained_embeddings_spacy pipeline uses pre-trained word vectors from either GloVe or fastText. Instead, the supervised embeddings pipeline doesn't use any pre-trained word vectors, but instead fits these specifically for your dataset.

The standard way that is used to do intent classification is to represent sentences as a sum of word vectors, and then train a classifier on that representation. Mostly it's the quality (or appropriateness) of the word vectors that matters and using a neural network instead of a support vector machine (SVM) doesn't make any difference.

This mechanism is called **Pretrained Embeddings (intent_classifier_sklearn)**, which require enough language resources. The bag-of-words vector approach is known as a good baseline, even though it has some limitations. Most of the times it is impossible to have word vectors for some important words, such as jargon specific to the domain, or common typos. In this study, that's especially true since Sinhala is such an informal language, which hasn't any pre-trained word vectors for the language. On the other hand when using tens of thousands of word vectors stored in memory, those data may be never used since most of the conversational AI deals with a narrow domain.

Tensorflow Embedding Pipeline

The tensorflow embedding pipeline instead using any pre-trained word vectors, it fits these specifically for the dataset. The advantage of using this pipeline is the word vectors are being customised for the domain. And the other specialty with this pipeline is that this does not use a language specific model. Instead it is applicable for any language that is tokenizable using whitespace.

This pipeline doesn't use pre-trained word vectors, and should work on any language, which make is possible to be used for Sinhala Language.

This pipeline doesn't use any pre-trained vectors, instead learns embeddings for both the intents and the words simultaneously. And instead of training a classifier, these embeddings are used to rank the similarity between an input sentence and all of the intents. This means we will not get stuck with out-of-the-box pre-trained word vectors, but learn our own specifically for the domain. One more advantage of using this pipeline is it even works robustly for small datasets.

And also, it has been experimentally proven that, 'tensorflow_embedding' architecture becomes effective rather than the 'spacy_sklearn' architecture [16].

Further, we can't use 'Pretrained Embeddings' classifier since no predefined word vectors available for Sinhala Language in Spacy.

Therefore we will be using '**Tensorflow Embedding Pipeline**' in this study.

4.2.3. Entity Extraction

Understanding the user's intent is only a part of the problem. It has the same importance to extract relevant information from user's message. This process of extracting the different required pieces of information is called entity recognition/entity extraction. For this requirement also, there are available components in RASA NLU.

- **Entity recognition with SpaCy language models: ner_spacy**

The spaCy library offers pretrained entity extractors. As with the word embeddings, only certain languages are supported. If the required language is supported, the component ner_spacy is the recommended option to recognize entities like organization names, people's names, or places. But since spacy doesn't support Sinhala Language yet, there is no possibility of using it.

- **Rule based entity recognition using Facebook's Duckling: ner_http_duckling**

This library supports extracting any number related information, such as amounts of money, dates, distances, or durations. Duckling was implemented in Haskell and is not well supported by Python libraries. To communicate with Duckling, Rasa NLU uses the REST interface of Duckling.

- **Training an extractor for custom entities: ner_crf**

Both ner_spacy and ner_duckling do not require us to annotate any of the training data, since they are either using pretrained classifiers (spaCy) or rule-based approaches (Duckling). The ner_crf component trains a conditional random field which is then used to tag entities in the user messages. Since this component is trained

from scratch as part of the NLU pipeline we must annotate our training data ourselves.

‘ner_crf’ can be used whenever it is impossible to use a rule-based or a pretrained component. Since this component is trained from the scratch, it has to be careful when annotating the training data. Because any incorrect annotation may result in generating an erroneous-model.

There are some other concerns as well. It is recommended to provide enough examples (> 20) per entity so that the conditional random field can generalize and pick up the data. Even if the entity might not be relevant for the intent it is better to annotate the training examples everywhere in the training data.

And also this study is based on a dataset of Sinhala Language, for which no predefined entities are available. Therefore using the CRFEntityExtractor is the best possible extractor in this scenario. By using Regular Expressions(regex) we can help the CRF model learn to classify entities. CRFEntityExtractor requires sklearn-crfsuite which provides a scikit-learn compatible estimator.

When a list of regular expressions specified, each of which provides the CRFEntityExtractor with an extra binary feature, which says if the regex was found (1) or not (0).

4.2.4. Custom Entities

We are using user annotated data set in this study, and also this study has no previous dataset for reference or to take use of, due to the language used is being Sinhala. Therefore it is obvious that we have to use custom entities. The CRFEntityExtractor is having the ability to identify custom entities in any language, when given enough annotated training data.

It is a manual process to identify the custom entities, by examining the whole dataset. So when choosing one as a custom entity, the frequency of occurring that word category within the domain is considered.

The custom entities that is used in this study are listed below.

- Stream
- Medium
- Condition
- Activities

- Course
- Info_type
- Info
- Job_market
- Duration

4.3. Formatting/Structuring the Dataset

When it comes to a conversational chatbot which is based on machine learning, the dataset is an undoubtedly mandatory and should be covering all the features and factors in the domain that it pertains. Therefore, a special consideration should be applied in preparing the dataset for the chatbot.

The dataset is transformed into following formats when preparing the final required data format.

Written paper chits -> Excel -> Markdown File (nlu.md)

4.3.1. Markdown Format

Markdown format is a convenient format that can be used in annotating the dataset used by Rasa NLU which is human readable and writable. Examples can be listed using the unordered list syntax, e.g. minus -, asterisk *, or plus +. Examples are grouped by intent, and entities are annotated as markdown links.

Following are some examples extracted from the markdown file created for this study.

As shown in below example, Entity value is marked using square brackets('[']') and entity name is marked using parenthesis('(')').

Eg :-

```
## intent:ask_medium
```

- උගන්නන්නේ [සිංහලෙන්](medium)ද [ඉංග්‍රීසි](medium) වලින්ද?
- උගන්නන්නේ [සිංහල](medium) භාෂාවෙන්ද නැත්නම් [ඉංග්‍රීසි](medium) භාෂාවෙන්ද?
- ඉගෙන ගන්න වෙන්නේ [ඉංග්‍රීසි](medium) වලින් නේද?
- දේශන පවත්වන්නේ සිංහලෙන්ද? [ඉංග්‍රීසි](medium)යෙන්ද ?
- Campus එකේ උගන්වන්නේ [English](medium) medium ද ?

intent: ask_job_opportunities

- [ET](stream) වලට ඇති රැකියා අවස්ථා මොනවාද ?
- [ET](stream) සඳහා ඇති රැකියා අවස්ථා මොනවාද ?
- [ජීව පද්ධති තාක්ෂණවේදය](stream) සඳහා ඇති රැකියා අවස්ථා මොනවාද ?

RASA NLU allows to structure the training data into different parts; such as examples, synonyms, regex features and lookup tables.

Synonyms: Maps extracted entities into same name. As shown in above example mapping 'ICT' to 'තොරතුරු හා සන්නිවේදන තාක්ෂණවේදය'. But this is only possible, after entities are extracted. Therefore it is required to include enough examples with the synonyms present, for RASA NLU to pick them up.

synonym: ICT

- තොරතුරු හා සන්නිවේදන තාක්ෂණවේදය
- Information & Communication Technology
- Information and Communication Technology
- Information Technology
- ict

Lookup tables: Can be specified either lists or as text files containing new line-separated words or phrases. Upon loading the training data, these files are used to generate case-insensitive regex patterns that are added to the regex features. For example, in this case a list of 'stream' (of degree programs) names is supplied so that it is easier to pick out this entity.

lookup:stream

- ET
- ICT
- BST

Regex features: CRFEntityExtractor serves as the only component that uses regex features in its Entity Extraction mechanism. But if any other entity extractor is used in extracting entities such as MitieEntityExtractor or SpacyEntityExtractor those extractors do not take use of any

defined regex features. Generated features and their presence will not improve entity recognition for these extractors. Currently, all intent classifiers make use of available regex features.

4.4. Actions

When the bot identifies a user message, then the response to the user is sent by executing the actions. In Rasa there are different kinds of actions.

4.4.1. Default Actions

There are 8 default actions that are built in Rasa, used for various purposes. Any of these default actions can be overridden.

e.g:- **action_listen** : It waits for a user input to be received, without going to predict more actions.

action_default_fallback : Bot says the user that it couldn't understand the previous message and avoids keeping track of the previous message.

4.4.2. Utterance Actions

There can be simple cases like the bot can directly respond to a user input, without having to consider about the slots or the entities. By simply adding utterance template in domain file these actions can be defined.

In our study we have such utterance actions as well.

utter_greet

- text: "ආයුබෝවන්"
- text: "හෙලෝ"
- text: "හලෝ"
- text: "හායි"

utter_welcome

- text: "එය මගේ රාජකාරිය"
- text: "සුඵ දෙයක්"
- text: "සතුටුයි ඔබ සමග කතා කරන්නට ලැබීම ගැන"

utter_tell_medium

- text: "සියලුම අධ්‍යයන කටයුතු කෙරෙන්නේ ඉංග්‍රීසි මාධ්‍යයෙන්"
- text: "අධ්‍යයන කටයුතු ඉංග්‍රීසි මාධ්‍යයෙන් පවත්වා ගෙන යනු ලබන්නේ"

4.4.3. Custom Actions

Most of the time we need to respond the user based on the specific entities in user input. Therefore custom message has to be generated. In this case Custom actions can be used. Custom actions are run using an action server.

Following are some examples of custom actions used in this study.

Custom Action used to greet the user: action_greet

```
class ActionGreetUser(Action):  
    def name(self) -> Text:  
        return "action_greet"  
  
    def run(self, dispatcher: CollectingDispatcher,  
            tracker: Tracker,  
            domain: Dict[Text, Any]) -> List[Dict[Text, Any]]:  
  
        dispatcher.utter_message(tracker.latest_message["text"])  
  
        return []
```

Custom action to provide information of job opportunities: action_get_job_opportunities


```

class ActionGetJobs(Action):
    def name(self):
        return 'action_get_job_opportunities'

    def run(self, dispatcher, tracker, domain):

        conn = sqlite3.connect("db/bot.db")

        c = conn.cursor()

        entity = tracker.latest_message["entities"][0]["entity"];
        entity_value = tracker.latest_message["entities"][0]["value"];

        if entity_value == "ET":
            c.execute('SELECT job_field FROM job_fields WHERE stream={}'''.format(entity_value))
            rows = c.fetchall()

            job_fields = ""

            for row in rows:
                job_fields = job_fields + row[0] + '\n'

            dispatcher.utter_message("{} ಕರ್ತವ್ಯಗಳನ್ನು ಕಂಡು ಹಿಡಿಯಲು ಈ ಕೆಳಕಂಡ ವಿಷಯವನ್ನು ಹೇಳಿ".format(entity_value,job_fields))

        return []

```

4.4.4. Fallback Action

Not all the time the bot will be able to understand the exact intent of the user utterance, may be because of the way user has asked/stated is not familiar to the nlu model, so that it misses hitting the intent. And also not all the questions will be able to be answered by the bot, because user has asked/said something from outside the bot's domain. In both of these cases bot may not predict an intent which is upto minimum threshold.

Therefore Rasa allows defining two threshold values.

Nlu_threshold - the acceptable minimum score for the confidence of an intent prediction

Core_threshold - the acceptable minimum score for the confidence of predicting an action using any dialog policy

If the confidence is below either of the nlu_threshold or core_threshold, Rasa Core allows defining a Fallback Policy for such situations. When the Fallback Policy is applied we also define the fallback_action to be executed in a case explained above. Because of this, even for a worst case scenario the bot can inform the user that either user has to rephrase the previous statement or it is beyond the scope of bot's knowledgebase.

4.5. Slot filling

Slots are considered as the memory of a chatbot. It stores the information in a key-value manner. In a conversation the information collected from a user is stored in slots. When the conversation continues with a user, previously provided information may need to be referred. When we do not have ask back, what is already provided by the user, the realistic nature of the conversation is preserved.

Slots can be defined with an initial value, or set by button clicks, and by the custom actions. There are different types of slots such as text, bool, float, list or custom.

We can simply assume slots as the variables of our chatbot program.

As we know the real conversations doesn't always go in the simple question and answer manner. When the conversation progresses bot may require to know some more information from the user, in order to answer the previous question. In this case bot also will be asking questions from the user, and should be effective when collecting users' responses, since it is not better to ask the same question again and again.

Therefore Rasa has introduced a new policy called, "FormPolicy" which allows to cover all the happy paths within a one story.

4.6. Stories

Rasa's Core Dialog system needs training examples, for the core model to be trained. Those training examples are identified as the 'stories' and stored in a separate file.

- Stories are separated using '##'
- Messages of the user starts with '*', and the format will be like `intent{entity1 : value1, entity2 : value2 }`
- Responses from the bot, probably the actions starts with '-'
- If any action return an event, it will be placed immediately after the particular action.

4.7. Important Code Snippets

4.4.1. Training the Bot – Online

Following code snippets used to train the bot online.

```
from __future__ import absolute_import
from __future__ import division
from __future__ import print_function
from __future__ import unicode_literals

import logging

from rasa_core.agent import Agent
from rasa_core.policies.keras_policy import KerasPolicy
from rasa_core.policies.memoization import MemoizationPolicy
from rasa_core.interpreter import RasaNLUInterpreter
from rasa_core.train import interactive
from rasa_core.utils import EndpointConfig

logger = logging.getLogger(__name__)

def run_sinhalabot_online(interpreter,
                        domain_file="sibot_domain.yml",
                        training_data_file='data/stories.md'):
    action_endpoint = EndpointConfig(url="http://localhost:5055/webhook")
    agent = Agent(domain_file,
                  policies=[MemoizationPolicy(max_history=2), KerasPolicy(max_history=3, epochs=3, batch_size=50)],
                  interpreter=interpreter,
                  action_endpoint=action_endpoint)

    data = agent.load_data(training_data_file)
    agent.train(data)
    interactive.run_interactive_learning(agent, training_data_file)
    return agent

if __name__ == '__main__':
    logging.basicConfig(level="INFO")
    nlu_interpreter = RasaNLUInterpreter('./models/nlu/default/fotnlu')
    run_sinhalabot_online(nlu_interpreter)
```

Figure 6

4.4.2. Running the ChatBot

```
from rasa_core.channels.slack import SlackInput
from rasa_core.agent import Agent
from rasa_core.interpreter import RasaNLUInterpreter
import yaml
from rasa_core.utils import EndpointConfig

nlu_interpreter = RasaNLUInterpreter('./models/nlu/default/fotnlu')
action_endpoint = EndpointConfig(url="http://localhost:5055/webhook")
agent = Agent.load('./models/dialogue', interpreter = nlu_interpreter, action_endpoint = action_endpoint)

input_channel = SlackInput('xoxb-519207642247-518888236868-RYJIhP5qHGwYzKA0GuKKZJhJ' #your bot user authentication token
)

agent.handle_channels([input_channel], 5004, serve_forever=True)
```

Figure 7

Chapter 5: Evaluation

The implemented chatbot solution is evaluated to check if it can respond the user requests in a satisfactory level. And, it is needed to be tested if the bot can identify the user intent properly and extract important information components (Entities) within it.

When evaluating the bot, since we do not have other built-in bots for Sinhala Language at the moment, there is no possibility to compare this with another product. Therefore, the possible mechanism of evaluation is to check how far the bot is capable of properly tracking the intent of the user utterance. And if the entities defined are also successfully tracked the bots response can guaranteed to be successive, since based on the different entity values for the same intent, different answers will be generated.

5.1. Evaluation Approach 1

Evaluate the Successive Intent Classification the most important evaluation to be done in this research is to evaluate the accuracy of the Intent Classification using Machine Learning. Accuracy of the Entity Extraction becomes the next important thing to evaluate.

Figure 5 shows the bot trainers' intent classification output for the sentence,

“ඒව පද්ධති තාක්ෂණවේදය සඳහා ඇති රැකියා අවස්ථා මොනවාද ?”

```

{
  "intent": {
    "name": "ask_job_opportunities",
    "confidence": 0.9775900840759277
  },
  "entities": [
    {
      "start": 0,
      "end": 21,
      "value": "ඒම සද්ධති තාක්ෂණවිද්‍යා",
      "entity": "stream",
      "confidence": 0.848031698672048,
      "extractor": "CRFEntityExtractor"
    }
  ],
  "intent_ranking": [
    {
      "name": "ask_job_opportunities",
      "confidence": 0.9775900840759277
    },
    {
      "name": "goodbye",
      "confidence": 0.03767079859972
    },
    {
      "name": "out_of_domain",
      "confidence": 0.03216104209423065
    },
    {
      "name": "ask_course_content",
      "confidence": 0.00901554524898529
    },
    {
      "name": "ask_course_requirements",
      "confidence": 0.0
    },
    {
      "name": "greet",
      "confidence": 0.0
    },
    {
      "name": "thank_you",
      "confidence": 0.0
    }
  ],
  "text": "ඒම සද්ධති තාක්ෂණවිද්‍යා සඳහා ඉතිරි රැකියා අවස්ථා මොනවාද?"
}

```

Figure 8 – Intent Classification Example

As shown in Figure 7, for each user utterance, ‘**intent_rankings**’ array is generated which contains the **name** and the **confidence** for each intent. And the intent with maximum confidence becomes the final intent for that user utterance.

5.1.1. Evaluating Models

Evaluating NLU Model

RASA provides two mechanisms for evaluating the nlu models, one is using training data set and a test data set. The other mechanism is using the whole data set with a cross validation technique.

In this study we follow the standard technique in machine learning which is to keep some data separate as a test set. Therefore the whole data set is split into two as train and test set.

Training Data

80% of the tagged dataset (questions with manually identified intents and entities) will be used as the training dataset.

Test Data

The other 20% of the dataset will be used as the test dataset. The output intents & entities will be checked against the corresponding manually tagged intents and the entities.

Using this the accuracy percentage can be calculated. Questions in the Test dataset is passed to the Bot. The output intent & entities are retrieved.

5.1.2. Evaluating Intent Classification

Using the Rasa's evaluation script, we can get a report, confusion matrix and confidence histogram for the model.

The report logs precision, recall and f1 measure for each intent and entity, as well as providing an overall average.

Precision: This basically tells us that out of the results classified as positive by our model, how many were actually positive.

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

Recall: This measure tells us how many true positives (points labelled as positive) were recalled or found by our model.

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

F1-score: This is a metric which takes into account both, precision and recall as we can't always evaluate both and then take the higher one for our model. It is the harmonic mean of precision and recall. It tells us about the balance that exists between precision and recall.

Above evaluation metrics when trained our NLU model using '**supervised_embeddings**' pipeline is shown below.

F1-Score: 0.977110157367668

Precision: 1.0

Accuracy: 0.9656652360515021

5.1.3. Classification Report

	precision	recall	f1-score	support
	0.00	0.00	0.00	0
affirm	1.00	1.00	1.00	5
ask_course_content	1.00	1.00	1.00	8
ask_course_duration	1.00	1.00	1.00	16
ask_course_requirements	1.00	1.00	1.00	1
ask_information	1.00	1.00	1.00	38
ask_internal_conditions	1.00	1.00	1.00	20
ask_job_opportunities	1.00	1.00	1.00	72
ask_medium	1.00	1.00	1.00	20
deny	1.00	1.00	1.00	6
goodbye	1.00	1.00	1.00	4
greet	1.00	0.71	0.83	14
mood_great	1.00	1.00	1.00	8
mood_unhappy	1.00	1.00	1.00	10
out_of_domain	1.00	1.00	1.00	5
thank_you	1.00	0.33	0.50	6
micro avg	0.97	0.97	0.97	233
macro avg	0.94	0.88	0.90	233
weighted avg	1.00	0.97	0.98	233

Figure 9 - Classification Report of Intents

Figure 8 shows the Classification scores for each of the intents in our domain. The ‘**support**’ measure displayed in it is the number of occurrences of each intent in the test dataset.

5.1.4. Confusion Matrix

Following confidence matrix displayed in Figure 9 – which is also calculated against the model for evaluating if any intent is predicted incorrect.

This helps to identify which are identified or interpreted to the nlu model in ambiguous way.

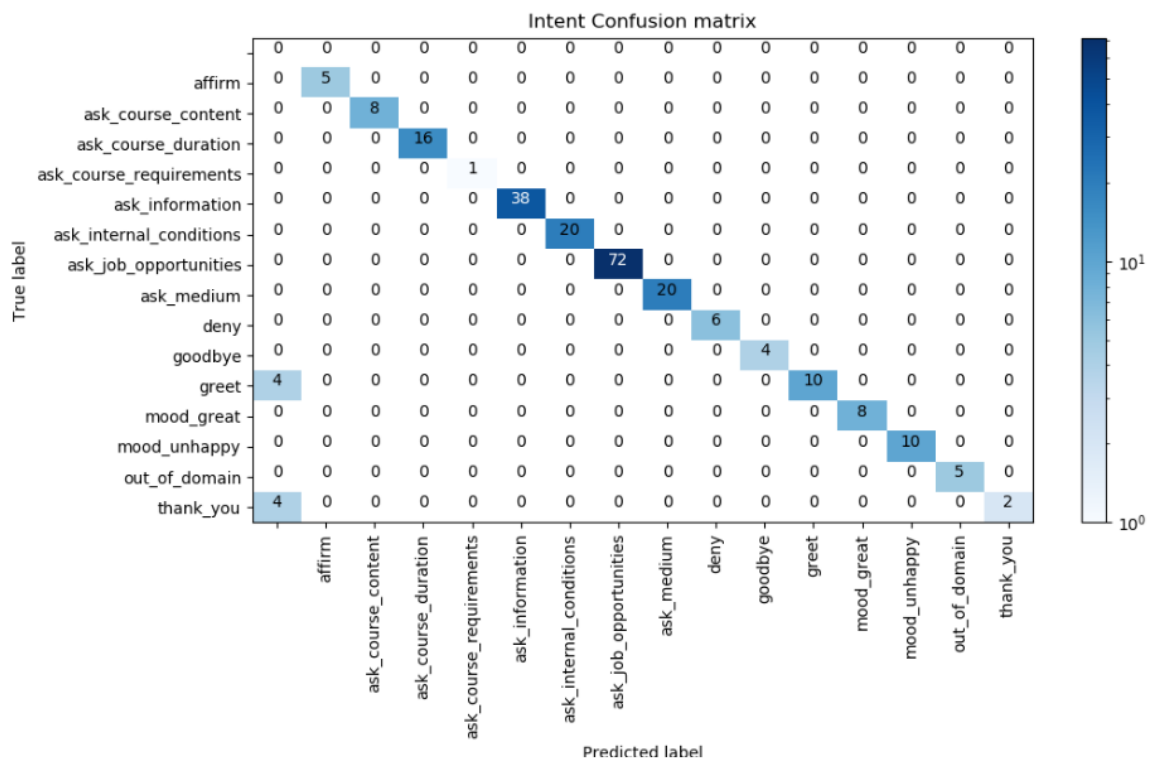


Figure 10 – Confusion Matrix

5.1.5. Intent Prediction Confidence Distribution

The histogram that is generated using Rasa’s evaluation method allows us to visualize the confidence distribution for all predictions, with the volume of correct and incorrect predictions being displayed by blue and red bars respectively.

Therefore by improving the quality of our training data the blue histogram bars can be moved to the right and the red histogram bars to the left of the plot.

Figure 10 below illustrates the confidence distribution for all predictions.

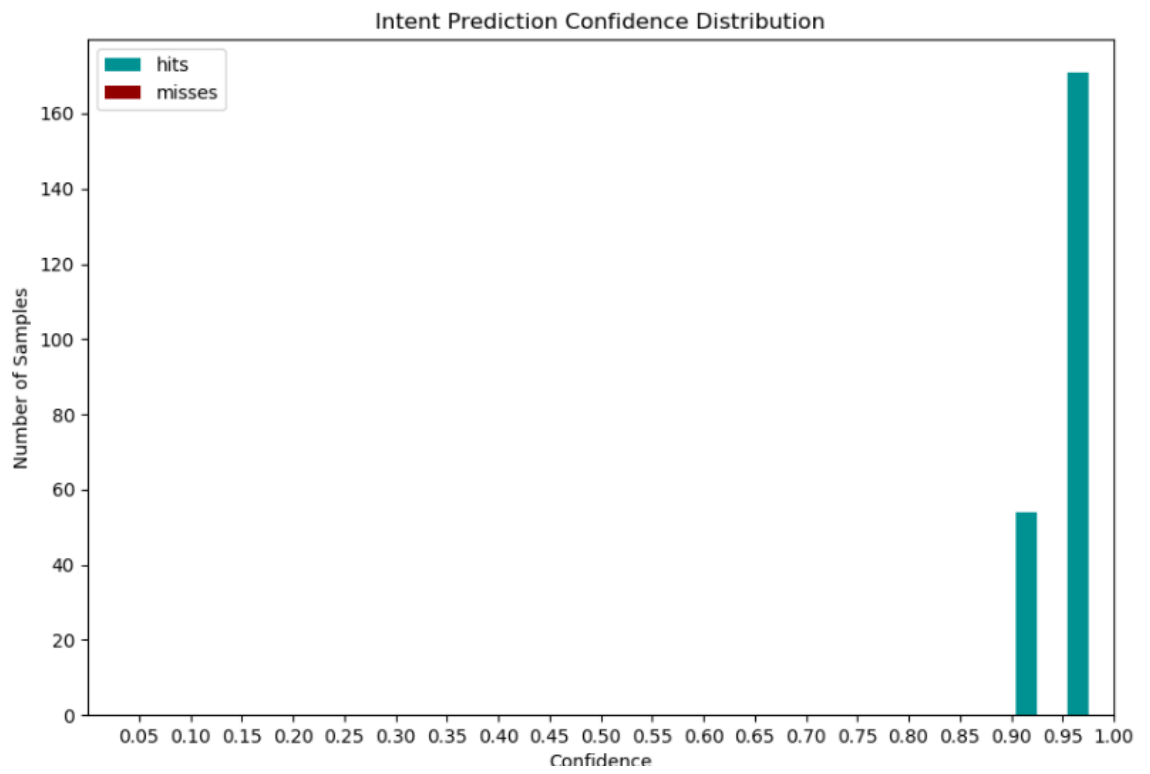


Figure 11 – Intent Prediction Confidence Distribution

5.1.6. Evaluation of Entity Extraction

	precision	recall	f1-score	support
activities	1.00	1.00	1.00	4
ask_if	0.00	0.00	0.00	8
condition	1.00	0.62	0.77	80
content_type	1.00	0.33	0.50	12
course	0.00	0.00	0.00	8
duration	0.00	0.00	0.00	38
info	1.00	0.49	0.66	196
info_type	1.00	0.64	0.78	25
job_field	1.00	1.00	1.00	4
job_market	0.77	0.37	0.50	65
medium	1.00	0.10	0.18	82
no_entity	0.87	0.99	0.93	2544
number	1.00	0.33	0.50	12
period	1.00	0.56	0.71	9
stream	0.94	0.84	0.89	230
subject	1.00	1.00	1.00	1
year	1.00	0.50	0.67	2
micro avg	0.88	0.88	0.88	3320
macro avg	0.80	0.52	0.59	3320
weighted avg	0.88	0.88	0.86	3320

Figure 12- Classification Report of the Entity Extraction

5.2. Evaluation Approach 2 – User Evaluation

Since the chatbot is a user centric application the most effective way to evaluate the chatbot is User Experience. Therefore, user satisfaction on the conversation with the bot is considered as the second evaluation approach.

But due to the time limitation, the bot could not be evaluated in that way.

5.3. Chapter Summary

It is very important that the evaluations fail if any of our training data set is annotated incorrectly with intents and entities.

Chapter 6: Future Works & Conclusion

6.1. Deploying the Chatbot

Since the domain the chatbot is developed for is ‘the Degree programs at Faculty of Technology, University of Ruhuna ’, there are several deployable places for this chatbot. The chatbot can be deployed in the Faculty Website, as a chat widget. It can be used as a user attractive feature and will be the best place to be deployed. So that the bot can be enhanced with new training data. And, this can be deployed in Facebook page of the Faculty as a messenger bot.

6.2. Conclusion

Following conclusion can be made by analyzing the results of the chatbot.

- Rasa can be used to develop a successive chatbot.
- It is not a mandatory thing to have a predefined word vectors for the language to implement a conversational models/Chatbots.
- Conditional Random Fields (CRFEntityExtractor in Rasa) serve well in Entity Extraction

6.2. Future Works

People often say more than one thing in a message, so improving a chatbot which can understand multiple intents will add an additional great value to the chatbot.

In this research only one intent per message is identified and respond accordingly.

And using some Sinhala NER tagging for the Bot training will make the bot rich with Recognizing well known named entities.

- Previously gathered information is not referred or, kept in bot’s knowledge to answer subsequent questions or throughout the conversation.
- Even if a user repeats the same question in different ways, bot will repeatedly answer, without outputting a message mentioning this question has already been answered.

In the time of the final phase of this study, which lies on last week of May 2019, Rasa introduced their new product as RASA and also another tool as RASA X. RASA is the

combined version of RASA NLU & RASA Core, where the user friendliness is much higher than previous versions. Rasa X is a tool to learn from real conversations and improve the assistant. So that by studying the applicability of RASA X with our bot, the bot can be improved.

References

- [1] "Do-it-yourself NLP for bot developers – Rasa Blog – Medium", Medium, 2016. [Online]. Available: <https://medium.com/rasa-blog/do-it-yourself-nlp-for-bot-developers-2e2da2817f3d> . [Accessed: 09- Sep- 2018].
- [2] Hettige, B. and Karunananda, A. (2006). First Sinhala Chatbot in action. [online] Staffweb.sjp.ac.lk. Available at: <http://staffweb.sjp.ac.lk/sites/default/files/budditha/files/budditha2006.pdf> [Accessed 23 May 2018].
- [3] Deep Learning Based Chatbot Models. [online] Research Gate. Available at: https://www.researchgate.net/publication/323587007_Deep_Learning_Based_Chatbot_Models [Accessed 21 Jul. 2018].
- [4]"iOS - Siri", Apple. [Online]. Available: <https://www.apple.com/ios/siri/> . [Accessed: 10- Sep- 2018].
- [5]"Rasa: Open source conversational AI", Rasa.com. [Online]. Available: <https://rasa.com/> [Accessed: 11- Sep- 2018].
- [6]"Snips Natural Language Understanding — Snips NLU 0.16.5 documentation", Snips-nlu.readthedocs.io. [Online]. Available: <https://snips-nlu.readthedocs.io/en/latest/> . [Accessed: 11- Sep- 2018].
- [7]"An Overview of Chatbot – Chatbots Life", Chatbots Life. [Online]. Available: <https://chatbotslife.com/an-overview-of-chatbot-a539b5fc55d3> . [Accessed: 11- Sep- 2018].
- [8] Dialogflow Entities: Identify things your users mention [Basics 2/3]. Google, 2018.
- [9]"Dialogflow", Dialogflow, 2018. [Online]. Available: <https://dialogflow.com/> . [Accessed: 14- Sep- 2018].

[10] "LUIS: Language Understanding Intelligent Service", Luis.ai, 2018. [Online]. Available: <https://www.luis.ai/home>. [Accessed: 14- Sep- 2018].

[11] Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation.

[12] Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

[13] "Chatbot Market Survey- 2017 : Mindbrowser Info Solutions", Mindbrowser, 2018. [Online]. Available: <http://mindbrowser.com/chatbot-market-survey-2017/>. [Accessed: 10- Aug- 2018].

[14] M. KOTTORP and F. JÄDERBERG, *Chatbot as a potential tool for businesses*. KTH SKOLAN FÖR INDUSTRIELL TEKNIK OCH MANAGEMENT, 2017, pp. 4-18.

[15]"NLUlite", *Nlulite.com*, 2019. [Online]. Available: <http://nlulite.com/>. [Accessed: 04- Apr- 2019].

[16] "Supervised Word Vectors from Scratch in Rasa NLU", *Medium*, 2019. [Online]. Available: <https://medium.com/rasa-blog/supervised-word-vectors-from-scratch-in-rasa-nlu-6daf794efcd8>. [Accessed: 24- Apr- 2019].

[17]"tensorflow/models", *GitHub*, 2019. [Online]. Available: <https://github.com/tensorflow/models/tree/master/research/syntaxnet>. [Accessed: 24- Apr- 2019].

[18] "Rasa Stack: Open source conversational AI", *Rasa.com*, 2019. [Online]. Available: <https://rasa.com/products/rasa-stack>. [Accessed: 11- May- 2019].

Appendix

Appendix 1 - Stories File (Stories.md)

```
▼ ## 1 greet and ask medium then thanks
* greet
  - action_greet
* ask_medium{"medium":"ဒွံ့ဖ်•ဝဲမိ"}
  - utter_tell_medium
* thank_you
  - utter_welcome

▼ ## 2
* greet
  - action_greet
* ask_medium{"medium":"ဒွံ့ဖ်•ဝဲမိ"}
  - utter_tell_medium

▼ ## 3 greet and thanks
* greet
  - action_greet
* thank_you
  - utter_welcome

▼ ## 4 say thanks
* thank_you
  - utter_welcome

▼ ## 5
* greet
  - action_greet
* ask_course_duration{"duration":"မိပွဲ ချဲဝဲင်"}
  - utter_tell_course_duration

▼ ## 6
* ask_course_duration
  - utter_tell_course_duration

▼ ## 7
* greet
  - action_greet
* ask_course_duration{"duration":"ဝဲမိဝ် 4ဆ"}
  - utter_tell_course_duration
* thank_you
  - utter_welcome

▼ ## 8
* greet
  - action_greet
* ask_job_opportunities
  - action_get_job_opportunities

▼ ## 9
* ask_job_opportunities
  - action_get_job_opportunities
* thank_you
  - utter_welcome

▼ ## 10
* greet
  - action_greet
* ask_job_opportunities
  - action_get_job_opportunities
* thank_you
  - utter_welcome|
```


Portion of the dataset

intent:ask_medium

- උගන්නන්නේ [සිංහලෙන්](medium)ද [ඉංග්‍රීසි](medium) වලින්ද?
- උගන්නන්නේ [සිංහල](medium) භාෂාවෙන්ද නැත්නම් [ඉංග්‍රීසි](medium) භාෂාවෙන්ද?
- ඉගෙන ගන්න වෙන්නේ [ඉංග්‍රීසි](medium) වලින් නේද?
- දේශන පවත්වන්නේ සිංහලෙන්ද? [ඉංග්‍රීසි](medium)යෙන්ද ?
- Campus එකේ උගන්වන්නේ [English](medium) medium ද ?
- [සිංහලෙන්](medium) දේශන පවත්වන්නේ නැද්ද?
- [සිංහලෙන්](medium) lectures කරන්නේ නැද්ද?
- [සිංහලෙන්](medium) ලෙක්චර්ස් කරන්නැද්ද?
- ලෙක්චර්ස් තියෙන්නේ [සිංහලෙන්](medium) නෙමෙයිද?
- Campus ගියහම සියල්ල [English](medium) භාෂාවෙන්ද උගන්වන්නේ ?

intent: ask_course_duration

- සියලුම සිසුන්ට [සිවු අවුරුදු](duration) පාඨමාලාව හදාරන්නට ලැබෙනවාද?
- [වසර 4ක්](duration) ම උපාධි පාඨමාලාව හැදෑරීම අනිවාර්යද?
- [වසර 3කින්](duration) උපාධිය නිම කල නොහැකිද?
- යම් අයකුට සාමාන්‍ය උපාධියකින් නිම කිරීමට අවශ්‍ය නම් ඒ සඳහා ඉඩ ලැබෙනවාද?
- තාක්ෂණ විෂය ධාරාවෙන් හදාරන සිසුන්ට උපාධිය අවසන් කිරීමට වසර කීයක් ගතවේද ?
- [වසර 3 ක්](duration) ඇතුලත උපාධිය නිමකළ නොහැකිද ?
- [වසර 5 න්](duration) උපාධිය නිමකළ නොහැකිද ?
- [වසර 04 න්](duration) උපාධිය නිමකළ හැකිද ?

intent: ask_job_opportunities

- [ET](stream) වලට ඇති රැකියා අවස්ථා මොනවාද ?
- [ET](stream) සඳහා ඇති රැකියා අවස්ථා මොනවාද ?
- [ජීව පද්ධති තාක්ෂණවේදය](stream) සඳහා ඇති රැකියා අවස්ථා මොනවාද ?
- [ICT](stream) සඳහා ඇති රැකියා අවස්ථා මොනවාද ?
- [ICT](stream) වලට ඇති රැකියා අවස්ථා මොනවාද ?
- [ICT](stream) වලින් යන්න පුලුවන් jobs මොනවාද?
- [ET](stream) උපාධිධාරීන්ට ලංකාවේ රැකියා තිබෙනවාද ?
- [ජෛව පද්ධති තාක්ෂණවේදය](stream) උපාධිධාරීන්ට ලංකාවේ රැකියා තිබෙනවාද ?
- [තාක්ෂණවේදය](stream) උපාධිධාරීන් සියලු දෙනාට රැකියා අවස්ථා ලංකාවේ තිබේද?
- උපාධිය අවසානයේ පිවිසිය හැකි රැකියා මොනවාද?
- උපාධිය අවසානයේ ලබාගත හැකි රැකියා ක්ෂේත්‍ර මොනවාද?
- [තාක්ෂණ විෂය ධාරාව](stream) හැදෑරීම මගින් සිසුන්ට ලැබෙන රැකියා අවස්ථා මොනවාද ?

- [තාක්ෂණවේදී](stream) උපාධිය හැදෑරීමෙන් පසුව කලා උපාධිධාරීන් මෙන් විරැකියා පෝලීම් වල සිටින්නට සිදුවේද ?
- මෙම පාඨමාලා හැදෑරීමෙන් අපට ලබාගත හැකි රැකියා මොනවාද ?
- අප හදාල විෂය ධාරාව මගින් ලබා ගත හැකි Job මොනවාද ?
- මෙම පාඨමාලා ඉලක්ක කර ඇත්තේ [දේශීය රැකියා](job_market) වෙළඳ පොළද, [විදේශීය රැකියා](job_market) වෙළඳ පොළද ?
- විශ්ව විද්‍යාල තුල ඉගැන්වෙන මෙම [තාක්ෂණික විෂය](stream) රැකියා වෙළඳ පොළ ඉලක්ක කරගෙන තිබේද ?
- [විදේශ රැකියාවකට](job_market) යාමට මෙම උපාධිය ප්‍රමාණවත් වේද ?
- ඒ සඳහා ඇති Jobs මොනවාද ?
- [තාක්ෂණවේදය](stream) කිරීමෙන් ලභා විය හැකි රැකියා අවස්ථා මොනවාද ?
- [තොරතුරු හා සන්නිවේදන තාක්ෂණවේදය](stream) සඳහා ඇති රැකියා අවස්ථා මොනවාද ?
- [BST](stream) සඳහා ඇති රැකියා අවස්ථා මොනවාද ?
- [Engineering Technology](stream) උපාධිධාරියකුට ඇති රැකියා මොනවාද ?
- [Engineering Technology](stream) සඳහා ඇති රැකියා අවස්ථා මොනවාද ?
- [තාක්ෂණවේදී](stream) උපාධිධාරීන්ට ලංකාවේ ප්‍රමාණවත් රැකියා තිබේද ?
- [තාක්ෂණ විෂය ධාරාව](stream) හැදෑරීම මගින් සිසුන්ට ලැබෙන රැකියා අවස්ථා මොනවාද ?
- [විදේශ රැකියා](job_market) සඳහා මෙම උපාධිය ඉලක්ක කරගෙන තිබේද ?
- [විදේශ රැකියා](job_market) වෙළඳ පොළ සඳහා රුහුණු විශ්ව විද්‍යාලයෙන් ලබාදෙන [සහතිකය](info) පමණක් ප්‍රමාණවත් වේද ?
- [විදේශ රැකියාවකට](job_market) යාමට මෙම උපාධිය ප්‍රමාණවත් වේද ?
- [විදේශ රැකියාවකට](job_market) යාමට මෙම උපාධිය සමග [ප්‍රායෝගික පුහුණුවීමත්](info) අවශ්‍යද ?
- Campus යාමට හැකි වුවහොත් ඊට අදාල Jobs මොනවාද ?
- [පෞච්ච පද්ධති තාක්ෂණවේදය](stream) හදාරලා යන්න තියෙන රැකියා මොනවාද ?
- [පෞච්ච පද්ධති](stream) විෂයේ Job Field එක කොහොමද ?
- උපාධිධාරී තාක්ෂණවේදීන්ට ලැබෙන රැකියාව කෙසේ එකක්ද ?
- තාක්ෂණ විෂයෙන් යන්න පුළුවන් උපරිම රැකියාව කුමක්ද ?
- Tech කරපු අයට [ලංකාවේ](job_market) තිබෙන JOB Market එක කොහොමද ?
- Tech කරලා කරන්න පුළුවන් රැකියා අවස්ථා මොනවාද ?
- Tech කරලා යන්න පුළුවන් රැකියා ක්ෂේත්‍ර මොනවාද ?
- Tech [විදේශ](job_market) ගතවී නිරත විය හැකි රැකියා ක්ෂේත්‍ර මොනවාද ?
- Tech වලින් [Software Engineering](job_field) යන්න පුළුවන්ද ?
- Tech වලින් හොඳ Job ලැබෙනවාද ?
- Tech කරලා තිබෙන Job මොනවාද ?
- Tech කරලා [Network Engineer](job_field) කෙනෙක් වෙන්න පුළුවන්ද ?
- තාක්ෂණික පාඨමාලාවලින් අපිට ලැබෙන වෘත්තීන් මොනවාද ?
- තාක්ෂණවේදය විෂය ධාරාව හැදෑරීමෙන් අපිට කුමන රැකියා ක්ෂේත්‍රයක වැඩ කටයුතු සිදු කල හැකිද ?
- තාක්ෂණවේදය හැදෑරීමෙන් පසු රැකියා වෙළඳපොළ තුල පවතින රැකියා අවස්ථා මොනවාද ?

- තාක්ෂණවේදය විෂය ධාරාවෙන් [ET](stream) වලින් විශ්ව විද්‍යාලයට තේරුණු පසු අපට හොඳ රැකියා අවස්ථා සඳහා යා හැකිද ?

intent: ask_information

- [කොයි කාලේද](period) [අලුත් බැව් එක](info) ගන්නේ.
- [අලුතින් සිසුන්](info) බඳවා ගන්නා [කාලසීමාව](period)ක් තිබේද?
- රුහුණ තාක්ෂණ පීඨයට [ICT](stream) පාඨමාලාව සඳහා තෝරාගන්න [Z-Score](info_type) අගය කීයද?
- පසුගිය වසර වල [ICT](stream) පාඨමාලාව සඳහා සිසුන් බඳවා ගත් [Z-Score](info_type) අගයන් මොනවාද?
- [2016](year) දී, [2017](year)දී [ICT](stream) පාඨමාලාව සඳහා සිසුන් බඳවා ගත් [Z-Score](info_type) අගයන් දැනගන්න පුළුවන්ද ?
- මොකක්ද අපිට ලැබෙන [උපාධිය](info_type) ,[ICT](stream) කල ශිෂ්‍යයෙක් නම්
- [විභාග](info_type) වලදී [ප්‍රායෝගික පරීක්ෂණ](info)ක් තියෙනවද?
- මගේ [z-score](info_type) එක 1.000. [ICT](stream). මාව තේරෙයිද ?
- රුහුණ විශ්ව විද්‍යාලයේ තාක්ෂණ පීඨයට [ICT](stream) වලට [කීදෙනෙක්](number) ගන්නවද ?
- රුහුණ විශ්ව විද්‍යාලයේ තාක්ෂණ පීඨයේ [ICT](stream) පාඨමාලාව සඳහා සිසුන් [කී දෙනෙක්](number) බඳවා ගන්නවාද?
- [තාක්ෂණ විෂය ධාරාව](stream) යටතේ තෝරාගත හැකි පැතිකඩ කීයද?
- [තාක්ෂණ විෂය ධාරාව](stream) යටතේ පිරිනැමෙන [උපාධිය](info_type) කුමක්ද?
- තාක්ෂණ [උපාධි](info_type) පාඨමාලා දැනට ක්‍රියාත්මක වන [විශ්ව විද්‍යාල](info) මොනවාද?
- පාඨමාලාවලට අමතරව [පුහුණු කාලසීමාව](info)ක් තිබේද?
- [තාක්ෂණ විෂය ධාරාව](stream) හදාල අපට විශ්වවිද්‍යාලය තුළ කොපමණ [පාඨමාලා](info) ප්‍රමාණයක් හැදෑරිය හැකිද ?
- මෙම පාඨමාලා සඳහා අන්තර්ගත කර ඇති [විෂය ක්ෂේත්‍ර](info) මොනවාද ?
- [විෂය දැනුම](info) හා [ප්‍රායෝගික දැනුම](info) මෙම [පාඨමාලාව](info_type) තුළ සමබරව පවතීද ?
- විශ්ව විද්‍යාලය මගින් ලබාදෙන [Degree](info) එක [විදෙස් රටවල](info) පිළිගන්නවාද ?
- මෙම උපාධිය හැදෑරීමේදී [ප්‍රායෝගික පුහුණුවීම්](info) සඳහා [විවිධ ආයතන](info_type) වලට යොමුකරන්නේද ?
- මෙම උපාධියෙහි [අන්තර්ජාතික පිළිගැනීමක්](info) තිබේද ?
- පවතින රජයෙන් මෙම පාඨමාලා හදාරන සිසුන්ට ලබාදෙන [වරප්‍රසාද](info) මොනවාද ?
- විශ්ව විද්‍යාල තුළ [තාක්ෂණ විෂය ධාරාව](stream) හදාරන සිසුන්ට [සම අයිතිවාසිකම්](info) හිමි වේද ?
- Campus වලින් ලබාදෙන [පාඨමාලා මොනවාද](info) ?
- [Bio Tech](stream) කරද්දී වෙනම [උපාධියක්](info) ලැබෙනවද විශ්ව විද්‍යාලයේදී ?
- Tech කරලා කැම්පස් එකේ කරන්න තියෙන [උපාධි](info), යා හැකි [ක්ෂේත්‍ර](info) මොනවාද ?
- උසස් පෙළට [BST](stream) විෂය හැදෑරූ සිසුන්ට විශ්ව විද්‍යාල වල ඇති [පාඨමාලා මොනවාද](info) ?
- තාක්ෂණවේදය සඳහා ඇති [පාඨමාලා මොනවාද](info) ?
- [පෞච්ච පද්ධති තාක්ෂණය](stream) විෂයේ තියෙන [පාඨමාලා මොනවාද](info) ?

- ඇතුළත් වීමට අවශ්‍ය අවම [Z අගය](info_type) කොතරම් විය හැකිද ?
- තාක්ෂණවේදය පාඨමාලා සඳහා ගන්නා ළමයි ගාන [කොපමණද](number) ?
- [ICT](stream) වලින් ඉහලටම යන්න තිබෙන [Course මොනවාද](info) ?
- Campus ගිණිත් කරන්න තියෙන [Course මොනවාද](info) ?
- විශ්ව විද්‍යාල [පාඨමාලා මොනවාද](info)?
- තාක්ෂණවේදය කරලා අපට විශ්ව විද්‍යාලයේ තිබෙන [පාඨමාලා මොනවාද](info) ?
- විශ්ව විද්‍යාලයේ තිබෙන තාක්ෂණික [පාඨමාලා මොනවාද](info)?
- තාක්ෂණය පිලිබඳ පාඨමාලා විශ්ව විද්‍යාල මොනවාද?
- විශ්ව විද්‍යාලයට ගියාට පස්සේ තාක්ෂණවේදය යටතේ කරන්න තියෙන [කෝස් මොනවද](info)?
- විශ්ව විද්‍යාලයට ගිය පසු තෝරා ගත හැකි [විෂය ධාරාවන් මොනවද](info) ?

intent: out_of_domain

- මේ සඳහා ඇති [පෞද්ගලික විශ්ව විද්‍යාල](info) මොනවාද ?
- පාඨමාලා සඳහා [කොපමණ](number) ළමුන් පිරිසක් බඳවා ගන්නවාද ?
- පළමු උපාධිය ලෙස [ET](stream) හදාළ සිසුවෙකුට ඉන් ඉදිරියට ඇති [පශ්චාත් උපාධි පාඨමාලා](info) කවරේද?
- තාක්ෂණවේදී උපාධිය ලබා ගැනීමෙන් පසුව තවදුරටත් [පශ්චාත් උපාධිය](info) හැදෑරීම සඳහා පහසුකම් තිබේද ?
- ආණ්ඩු වෙනස් උවහොත් විශ්ව විද්‍යාල පාඨමාලාවල කෙබඳු වෙනසක් සිදුවේද ?

intent: ask_course_requirements

- තාක්ෂණික පාඨමාලා හැදෑරීමට [ICT](subject) විෂය අත්‍යවශ්‍යද ?

intent: greet

- ආයුබෝවන්
- හෙලෝ
- හලෝ
- හායි

intent: thank_you

- ස්තූතියි
- තෑන්ක්ස්
- ඔබට ස්තූතියි
- Thanks
- බොහොම ස්තූතියි
- ස්තූති
- ස්තූති
- ස්තූතියි

intent: ask_internal_conditions

- අපට [පරිගණක](condition) ලබා දෙනවාද?
- අපිට [නේවාසික පහසුකම්](condition) තිබෙනවාද?
- [නේවාසික ව සිටීම](condition) [අනිවාර්ය](ask_if)ද?
- [නවකවදය](condition) නියෝගවද?
- [විභාග අසමත් උනොත්](condition) මොකද වෙන්නේ?
- [සිංහලෙන්](medium) [උගන්නන්න](condition) කියල අපිට ඉල්ලන්න බැරිද?
- labs වල සියලුම සිසුන්ට ජ්‍යෙෂ්ඨත්ව තරම් [පරිගණක](condition) තිබෙනවාද?
- [සිංහලෙන්](medium) [උගන්නාවි](condition)ද අපි ඉල්ලුවොත් ?
- එකකොට අපිට [ඉංග්‍රීසි](course)ත් [උගන්නනවද](condition)?
- [ඉගෙන ගන්නවා ඇරෙන්න](condition) වෙන මොනවද [කරන්න](activities) නියෝගවේ?

intent: ask_course_content

- [ඉංජිනේරු තාක්ෂණවේදය](stream)ට නියෝග [subjects](content_type) මොනවද?
- [ET](stream) වලට නියෝගවේ මොන [subjects](content_type) ද ?
- [ET](stream) වලට නියෝග [විෂයන්](content_type) දැනගන්න පුලුවන්ද?
- පාඨමාලාවල [විෂයන්](content_type) ප්‍රභේද ගණන කීයද?

synonym: ET

- ඉංජිනේරු තාක්ෂණවේදය
- Engineering Technology

synonym: ජීව පද්ධති තාක්ෂණවේදය

- BST
- Biosystems Technology
- biosystems technology
- ජෛව පද්ධති තාක්ෂණවේදය
- Bio systems Technology
- Bio Systems Technology
- Bio Tech

synonym: ICT

- තොරතුරු හා සන්නිවේදන තාක්ෂණවේදය
- Information & Communication Technology
- Information and Communication Technology
- Information Technology
- ict

lookup:stream

- ET

- ICT

- BST

lookup:activities

- වෙන කරන්න

- බාහිර ක්‍රියාකාරකම්

- බාහිර දේවල්