

**ASD.AI – SINHALA DIALOGUE MANAGEMENT TOOL
TO SCREEN KIDS WITH AUTISM SPECTRUM
DISORDER**

Gunawardhana M.D. R.T.

(IT16090804)

Bachelor of Science (Honors) in Information Technology

Specializing in Information Technology

Department of Information Technology

Sri Lanka Institute of Information Technology

Sri Lanka

October 2021

**ASD.AI – SINHALA DIALOGUE MANAGEMENT TOOL
TO SCREEN KIDS WITH AUTISM SPECTRUM
DISORDER**

Gunawardhana M.D.R.T.

(IT16090804)

Dissertation submitted in partial fulfillment of the requirements for the Bachelor of
Science (Honors) in Information Technology

Department of Information Technology

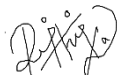
Sri Lanka Institute of Information Technology

Sri Lanka

October 2021

DECLARATION

I declare that this is my own work and this dissertation I does not incorporate without acknowledgment any material previously submitted for a degree or Diploma in any other University or institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgment is made in the text. Also, I hereby grant to Sri Lanka Institute of Information Technology, the non-exclusive right to reproduce and distribute our dissertation, in whole or in part in print, electronic or another medium. I retain the right to use this content in whole or part in future works (such as articles or books).



13th Oct 2021

Signature (Gunawardhana M.D.R.T.)

Date

The above candidate has carried out research for the bachelor's degree dissertation under my supervision.

Signature of the Supervisor

Date

Signature of the Co-Supervisor

Date

ABSTRACT

In today's world, computer-based conversational systems are quite popular. It may either be a human-powered or a human-machine conversational system. Voice assistants are well-known human-machine conversation systems. Some of the most popular and sophisticated voice assistants can act nearly organically. However, the majority of voice assistants are only available in English. However, the population that is unable to make full use of the available voice assistants owing to a language barrier may miss out on receiving assistance from conversational agents/voice assistants. Furthermore, while speaking with a bot, they are not required to conduct a formal discussion; otherwise, users may avoid chatting freely owing to a lack of confidence in asking inquiries.

This study aims to develop a solution, such as a Sinhala voice assistant, that may be used in a certain domain. A generic voice assistant has a significant likelihood of failing to answer to the majority of inquiries, rendering it useless and unproductive. As a result, with a domain-specific voice assistant, users may ask inquiries and solve problems without wasting time by pointing to the particular point. So, by being language particular (Sinhala) and domain specific, building '**A Sinhala voice assistant for screening kids with autism spectrum disorder**' primarily focuses on tackling two challenges.

Rather of answering basic inquiries, this voice assistant focuses on determining the purpose of the person who has asked the question and replying appropriately. As a result, this voice assistant is capable of handling contextual discussions. In comparison to a typical rule-based, template-based, or keyword-based bot, the replies become more reliable and acceptable. The RASA platform, an open-source dialogue management technology, was used to create the voice assistant. RASA Core and the RASA NLU are the framework's two primary components, with RASA Core providing more complex conversations and chats to its users. It enables the dataset to be trained in a variety of methods, including interactive learning and supervised machine learning. In voice assistants/dialogue models, RASA NLU handles natural language processing tasks including intent categorization and entity extraction.

ACKNOWLEDGMENT

Apart from the efforts of the project team, the success of any project rests largely on the concerted endeavors of many others. First and foremost, we would like to thank our supervisor Prof Koliya Pulasinghe for providing insight and expertise that greatly assisted the research and also for supervising our project. We also thank our co supervisor Ms. Vijani Piyawardana for her sincere and selfless support, prompt and useful advice during our research. We also thank our colleagues from Sri Lanka Institute of Information Technology who have given their invaluable support and comments for our research project to make it a success.

TABLE OF CONTENTS

| | |
|---|-----|
| DECLARATION | i |
| ABSTRACT | ii |
| ACKNOWLEDGMENT | iii |
| TABLE OF FIGURES | v |
| LIST OF ABBREVIATIONS | vii |
| 1. INTRODUCTION | 1 |
| 1.1 Background Literature | 1 |
| 1.1.1 History of Voice assistants | 1 |
| 1.1.2 Recent Voice assistants | 5 |
| 1.1.3 Sinhala Voice assistants | 6 |
| 1.1.4 The Voice assistant System | 7 |
| 1.1.5 Available Machine Learning Tools for Voice assistants | 9 |
| 1.1.6 Advancement of Artificial Intelligence | 11 |
| 1.1.7 The Sinhala Language | 16 |
| 1.1.8 Autism Spectrum Disorder | 17 |
| 1.2 Research Gap | 19 |
| 1.3 Research Problem | 20 |
| 1.4 Research Objectives | 23 |
| 1.4.1 Main Objective | 23 |
| 1.4.2 Specific Objectives | 24 |
| 2. RESEARCH METHODOLOGY | 25 |
| 2.1 Requirement Gathering and Analysis | 25 |
| 2.2 System Diagram | 26 |
| 2.3 Methodology | 28 |
| 2.3.1 Tools and software | 30 |

| | |
|---|----|
| 2.3.2 The Rasa Assistant..... | 33 |
| 2.3.3 Architecture | 34 |
| 2.4 Testing and Implementation | 36 |
| 2.4.1 Training Data Formats | 36 |
| 2.4.2 Machine Teaching..... | 39 |
| 2.4.3 Visualization of Dialogue Graphs..... | 40 |
| 2.4.4 Deployment..... | 40 |
| 2.5 Commercialization of the Product..... | 40 |
| 3. RESULTS AND DISCUSSION | 42 |
| 3.1 Results | 42 |
| 3.2 Research Findings and Discussion | 43 |
| 4. CONCLUSION..... | 45 |
| REFERENCES..... | 47 |
| APPENDICES | 49 |

TABLE OF FIGURES

| | |
|---|-------------------------------------|
| Figure 1.1: Google Assistant..... | Error! Bookmark not defined. |
| Figure 1.2: Intent Classification..... | Error! Bookmark not defined. |
| Figure 1.3: Entity Extraction..... | Error! Bookmark not defined. |
| Figure 1.4: Spoken Sinhala Vowel Classification | Error! Bookmark not defined. |
| Figure 1.5: Spoken Sinhala Consonant Classification..... | Error! Bookmark not defined. |
| Figure 2.1: High Level Architecture | Error! Bookmark not defined. |
| Figure 2.2: System Architecture | Error! Bookmark not defined. |
| Figure 2.3: Components of Open-Source Voice Assistant | Error! Bookmark not defined. |
| Figure 2.4: Sample Convocation with Rasa Assistant | 33 |
| Figure 2.5: Architecture of Voice Assistant..... | 34 |

| | |
|---|----|
| Figure 2.6: Dialogue Management..... | 35 |
| Figure 2.7: Sample story graphs with and without simplification | 40 |
| Figure 3.1: Confusion Matrix..... | 43 |

LIST OF ABBREVIATIONS

| Abbreviation | Description |
|--------------|-----------------------------------|
| 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 |

1. INTRODUCTION

The goal of this research is to create a Sinhala voice assistant capable of detecting user intent and extracting the essential elements in a phrase. The Statement of Problem, Motivation, and Aims & Objectives, as well as the Significance/Novelty/Contribution, are all explained in this chapter. The scope of the study is then explained. The chapter finishes by outlining the dissertation's structure.

1.1 Background Literature

Voice assistants have become the most common method of acting as representatives for any business. And, since the dawn of AI, creating a voice assistant that performs flawlessly has been one of the most difficult tasks.

The primary emphasis of this chapter is to look at the history of voice assistants. Then there's a discussion of contemporary voice assistants as well as the literature on Sinhala voice assistants. Following that, the framework of a voice assistant is identified, as well as NLU voice assistant components. A short glance at the various machine learning technologies for voice assistants is covered at the end of the chapter.

1.1.1 History of Voice assistants

I. ELIZA

ELIZA is a nondirective psychotherapist simulation voice assistant software developed in the early 1990s. It generates responses that mimic the user's input utterances using smart handwritten templates. Decomposition hand-written rules are used to evaluate input sentences, which are activated by key terms in the input text. ELIZA's natural language comprehension abilities were severely restricted.

II. PARRY

Psychiatrist Kenneth Colby created PARRY, an early voice assistant output, in 1972. It was created with the hopes of simulating a person suffering from paranoid

schizophrenia. The behavior of a person with paranoid schizophrenia, such as thoughts, conceptualizations, and beliefs, is the basis for the rudimentary model that has been built for this voice assistant. Psychiatrists examined actual patients and computers running PARRY, and transcripts of the discussions were sent to another group of psychiatrists to determine which patients were real and which were not. By guessing, they were only able to identify 48 percent of the people.

III. ALICE

ALICE (Artificial Linguistic Internet Computer Entity) is a software robot or computer with whom users may converse in natural language. To detect user input, ALICE employs a pattern-matching algorithm that employs depth-first search techniques. It has also passed the Turing test for the past two years. ALICE was created using the Artificial Intelligence Markup Language (AIML). And the most recent iteration of this language is based on the Pandora platform, with "Mitsuku" as the well-known voice assistant.

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>
```

The outputted reply would be the sentence between the <template> brackets if the user's input sentence matches the sentence between the <pattern> brackets. The "*" (star) sign is used to replace words in the following example. In this example, whatever word comes after the word like will appear in the answer at the place indicated by the <star/> token:

```
<category>
```

<pattern>I like *</pattern>

<template>I too like <star/></template>

</category>

IV. Elizabeth

Elizabeth is said to be a spin-off of Eliza. Elizabeth saves her information in a text file as a script, with each line beginning with a script command notation. Elizabeth can generate a sentence grammar structure analysis using a collection of input transformation rules to represent grammar rules.

Those are the major tiers of the voice assistant's history, in terms of the aforementioned early methods. The majority of early voice assistants relied on "keyword-based" or "template-based" matching. 'Rule-Based' or basic statistical methods are also used in certain voice assistant systems [3].

Handwritten rules may or may not be effective in all situations. According to Kumar Shridhar, BotSupply's Co-Chief AI Scientist, "In a rule-based approach, a bot responds to queries depending on some pre-programmed rules. The stated rules might range from simple to complicated. The development of these bots is very simple when utilizing a rule-based method, however the bot is ineffective when answering queries whose pattern does not fit the rules on which it was taught ".

Furthermore, the bots' capacity to respond exclusively to user inputs defined in pre-determined templates is limited by the template matching technique. Because today's languages are so sophisticated, manipulating templates for virtually all user inputs is difficult. As a result, there's a good probability that a bot won't be able to respond to most user inputs in a legitimate or helpful manner. Because Sinhala is an agglutinative language with a high rate of affixes or morphemes per word, using the aforementioned early methods alone is impracticable.

The majority of early voice assistants relied on "keyword-based" or "template-based" matching. 'Rule-Based' or basic statistical methods are also used in certain voice

assistant systems [3]. And neural networks are now the most popular paradigm for creating conversational bots. Many advanced voice assistants have been created for the English language.

Because of these limits in presence, it has become clear that old techniques must be replaced. Neural network-based methods have supplanted them. Traditional machine learning approaches are only employed as additional techniques when neural networks are used as the backbone of conversational modeling.

The existence of a learning algorithm in the latter scenario distinguishes traditional rule-based and neural network-based methods. Instead of utilizing hand-written rules, deep learning models use matrix multiplications and non-linear functions with millions of parameters to convert input words into responses.

V. SmarterKid

The move to messaging systems outside of standalone applications was the next step in the history of voice assistants. SmarterKid, a sarcastic voice assistant that debuted in 2001, is the most well-known example of this. SmarterKid was ahead of its time, coming closer to the experience we enjoy with voice assistants today. It introduced NLP to SMS networks and AOL Instant Messenger, as well as serving as a first introduction to voice assistants for many consumers.

SmarterKid performed a variety of helpful functions in addition to conversing, like delivering news, weather, stock information, sports scores, and much more. This showed and promoted voice assistants' ability to act as smart digital assistants through popular messaging platforms. It's also worth mentioning that SmarterKid was only one of many voice assistants created by Active Buddy, which was later purchased by Microsoft. Active Buddy also provided a variety of advertising bots (including an Austin Powers-themed bot), redefining the voice assistant as a new marketing tool rather than merely a conversation buddy.

1.1.2 Recent Voice assistants

The accuracy and efficacy of natural language processing have considerably increased because to recent advancements in machine learning, making voice assistants a realistic choice for many businesses. This advancement in NLP has sparked a flurry of new research, which should lead to continuing advancements in voice assistant efficacy in the next years. An FAQ (frequently asked questions) may be loaded into voice assistant software to build a basic voice assistant. The voice assistant's capabilities may be enhanced by connecting it with the company's business software, allowing it to answer more personal queries like "What is my balance?" or "What is the status of my order?"

For natural language processing, the majority of commercial voice assistants rely on systems developed by IT behemoths. Amazon Lex, Microsoft Cognitive Services, Google Cloud Natural Language API, Facebook DeepText, and IBM Watson are among the services available. Facebook Messenger, Skype, Slack, Twitter, Kik, WhatsApp, and Viber are just a few of the platforms where voice assistants are used.

Deep learning methods are used in today's voice assistants. Deep learning models use matrix multiplications and non-linear functions with millions of parameters to directly convert input words into responses. Conversational models based on neural networks may be split into two types: retrieval-based and generative models. The former just computes the most likely answer to the current input utterance based on a scoring function, which may be implemented as a neural network [11], or simply computes the cosine similarity between the word embedding of the input utterances and the candidate responses from the dataset. In contrast, generative models' synthesis the response one word at a time by computing probabilities over the whole lexicon [12].

There have also been techniques that combine the two types of dialog systems by comparing a created and retrieved response and judging which is more likely to be a superior response. Apple's Siri [4], Google's Assistant, and Microsoft's Cortana are three of the most popular voice assistants today. The Google Assistant is seen in Figure 1 [7].

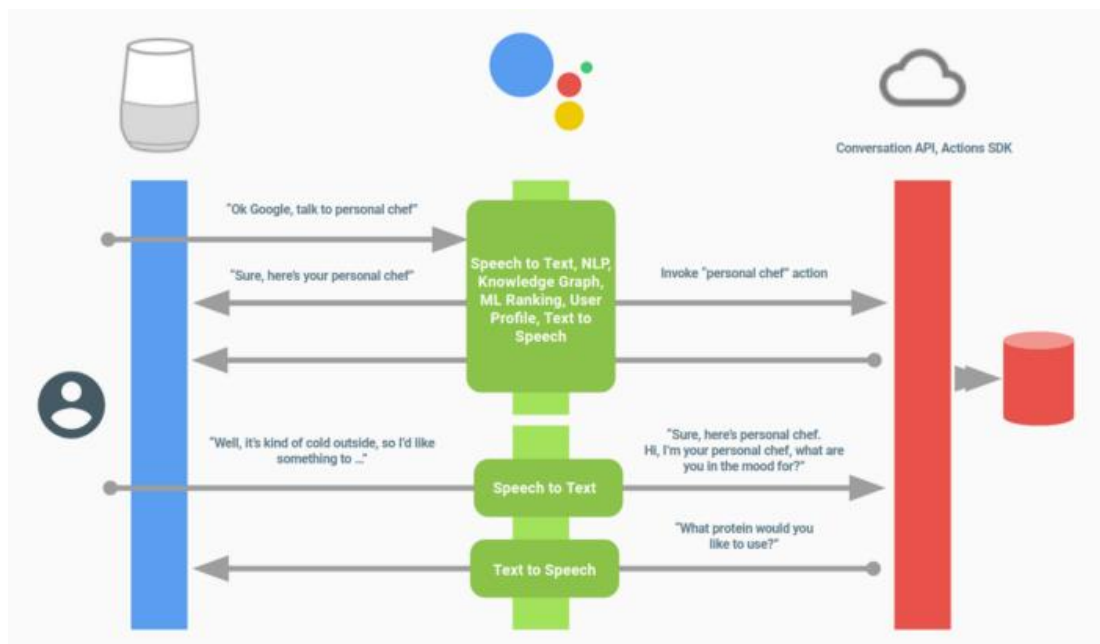


Figure 1.1: Google Assistant

1.1.3 Sinhala Voice assistants

I. Hettige, B. and Karunananda, A. Voice assistant

In 2006, the first Sinhala voice assistant was launched, which was meant to answer simple queries and was not tied to a specific domain. It's a voice assistant prototype that employs a Sinhala language parsing system that includes a Morphological analyzer, Sinhala parser, Sinhala composer, and lexical dictionaries. This system was created with Java and SWI-Prolog, and it operates on both Linux and Windows. This system is built on a client-server architecture, in which the server houses all of the resources and engine modules, while clients use the network to retrieve the data. The client-server architecture is utilized to allow several users to search for information using the voice assistant system at the same time.

To obtain the proper response, a knowledge identification engine is utilized to determine the relevant pattern. In a sentence, the system can recognize the subject, object, and verb. The pattern/3 Prolog predicate is then used to save the patterns. To

discover appropriate responses, this uses the normal Prolog matching and unification method [2]. As a result, it can be stated that this study was conducted in order to build the entire framework that is required.

II. GIC Voice assistant

This is a button-based voice assistant that is integrated in the Government Information Center's website and is available in English and Sinhala. The user can select from a pre-defined set of button-like alternatives in this voice assistant. Customers benefit from a significantly quicker experience because they don't have to type a single query. Instead, users may simply select a button from a menu of options. However, the drawback of this type of voice assistant is that users are confined to a small number of alternatives, thus only a limited range of topics is covered. The bot user is limited to a set of information intentions and cannot freely ask inquiries.

In order for a bot to respond to a user's message, it must complete two primary tasks. Intent Classification and Entity Extraction are the two topics covered in the following parts of this Literature Review. Wit.ai and LUIS are two built-in apis that can do the aforementioned two jobs. Those technologies make it easier to develop voice assistants by completing two main jobs, particularly for English-speaking users. However, because https calls are sluggish, using APIs might slow down the program, and users always are restricted by the design decisions made for API endpoints. There's also the possibility of the libraries being compromised. As a result, there may be a security risk. And, in order to be processed, the entire dataset must be transferred to a third party. The general-purpose APIs must be capable of solving any problem, but we just need to solve ours [1].

1.1.4 The Voice assistant System

The following three features should be included in a voice assistant system:

- Natural Language Understanding by Computer

A method for the bot to comprehend what the user is asking for should be provided in this section. First and foremost, the bot must be able to distinguish between when the user requests information from the bot and when the user sends information to the bot. Once it has been determined, the user query's purpose must be captured. In addition, particular parameters/entities in the user query may need to be retrieved in order to provide an appropriate answer to the user.

- Define and design the voice assistant's knowledge base.

Once the bot has detected the user's speech, it must respond appropriately to the user. The knowledge base comes into play at this point. A well-organized knowledge base should be available to get the appropriate outputs for the user.

- Develop appropriate pattern matching algorithms.

There may be a specified format that fits inside a collection of user utterances for a certain request or answer to be delivered. If a distinctive format can be found, it is most beneficial in entity extraction from that user's utterance. Regular Expressions are a well-known method for matching patterns like this. Regular expressions may also be defined in NLU frameworks like RASA.

I. Intent Classification

One of the most important functions of a voice assistant is intent classification. This focuses on determining what the user desires (recognizing the intent). Figure 2 depicts how user inputs are categorized based on their intent.



Figure 1.2: Intent Classification

II. Entity Extraction

We utilize Entity Extraction to try to figure out what people are talking about. The entity extraction process in DialogFlow [8] is illustrated in Figure 3. The user inputs "11am," "tomorrow," "California," and "\$40" are extracted as sys.time, sys.date, sys.geo-state-us," and sys.unit-currency entities, according to DialogFlow.

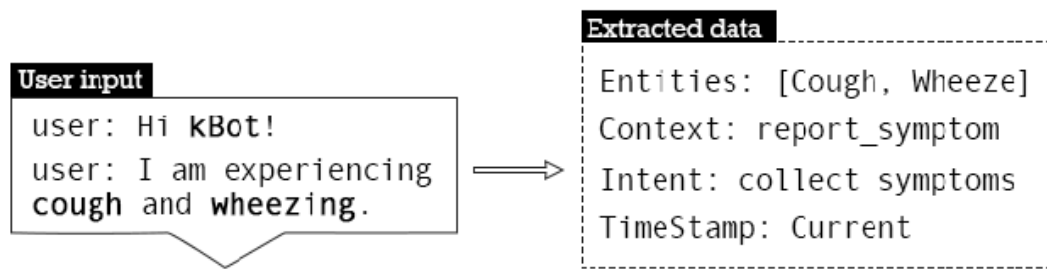


Figure 1.3: Entity Extraction

1.1.5 Available Machine Learning Tools for Voice assistants

Natural Language Understanding should be a voice assistant's primary capability (NLU). Some of the tools that come with Intent Classification and Entity Extraction include the following tools/platforms. Only a few open-source Natural Language Understanding Tools, such as RASA NLU and SNIPS NLU, are available for free, whereas the majority of the rest require a subscription for more advanced features.

I. RASA STACK

Developers may use RASA Stack's machine learning frameworks to build contextual AI assistants and voice assistants that go beyond simple queries. Thousands of community members promote open-source natural language processing and dialogue management [18].

RASA Core and RASA NLU are the two major components, with RASA Core serving as a voice assistant framework with machine learning-based dialogue management and

RASA NLU serving as a natural language understanding library with intent categorization and entity extraction.

Both of these elements are self-contained. RASA Core and RASA NLU may both be utilized with various NLU frameworks and dialog management frameworks/tools.

RASA NLU is responsible for interpreting the user's message based on prior training data when a user message is received. Intent Classification and Entity Extraction are two techniques used to accomplish this. The term "intent categorization" refers to the process of interpreting meaning based on specified intentions. The purpose of a user's utterance is determined based on the intent returned with the highest confidence rate. Entity Extraction is a technique for identifying structured data (the entities and their values).

RASA Core is called into action next. What happens next in the discourse is decided by the core. Based on NLU input, conversation history, and training data, its machine learning-based dialogue management predicts the next optimal action.

Because this is the research platform, it is discussed in greater detail later.

II. SNIPS NLU

Snips NLU is an open-source Python module for natural language understanding that allows you to parse natural language phrases and extract structured data [6]. The Snips NLU engine is able to extract structured data after it has been properly trained. With this, the same technique as with RASA NLU is feasible. Identifying the purpose and extracting the entities is what this entails. It works with both built-in and custom entities.

However, the reason SNIPS NLU cannot be utilized in our study is because it requires language resources for the language we are using, and there are presently no language resources available for Sinhala.

III. NLULite

NLULite is a developer-friendly database that combines a natural language parser with a graph database to scan texts and answer queries about them [15].

IV. SyntaxNet

SyntaxNet is a TensorFlow toolbox for natural language understanding (NLU) driven by deep learning [16].

V. DialogFlow

Dialogflow is a Google-owned developer of natural language conversation-based human–computer interface technology. The firm is best known for developing the Assistant, a virtual assistant for Android, iOS, and Windows Phone that does tasks and answers queries in natural language. This NLP framework provides a strong natural language understanding (NLU) engine to process and comprehend natural language input, allowing conversational interfaces to be built on top of goods and services [9].

VI. LUIS

Microsoft Azure's Language Understanding Intelligent Service (LUIS) provides a quick and easy approach to integrate language understanding to applications [10].

1.1.6 Advancement of Artificial Intelligence

In recent years, the evolution of artificial intelligence voice assistants with traditional text-based interface has become a new phenomenon on the market. The lack of human feeling is, nevertheless, the major flow connected with voice assistants. Voice assistants appear strange and troublesome due to their lack of human feeling. As a response to this problem, voice-activated voice assistants and products entered the market, with virtual assistants like Amazon Alexa, Google Assistant, and Apple Siri becoming extremely popular.

The following components make up most voice-enabled voice assistant frameworks.

- Speech Recognition
- Natural Language Processing
- Conversational Artificial Intelligence (Dialogue Management)
- Speech Synthesis

Researchers have done several studies in the aforementioned areas in order to get the best feasible outcomes in each application area. The following are some of the most important discoveries made in the above fields.

I. Speech Recognition

The basic goal is to teach a computer to recognize spoken language. Understanding is acting correctly and converting the incoming voice into another medium, in this case writing. Voice-to-text is the term used to describe speech recognition (STT). Despite the fact that this issue appears to be under investigation by a large number of academics at the moment, it has been around since 1920, when machine recognition was first presented. Engineers and scientists have been working on a variety of approaches and patterns since then, which have evolved through time. Some of these include:

- Acoustic phonetics-based speech recognition
- Hardware-based recognition
- Recognition of speech based on patterns
- Continuous word recognition for speech recognition
- A mix of statistical and connectionist approaches (HMM/ANN) is used to recognize speech
- Variational Bayesian (VB) estimate for speech recognition

The following are some of the current research projects. According to [9] The Aurora framework, which is establishing standards for Distributed Speech Recognition (DSR), speech analysis is performed in the telecommunication terminal and recognition is performed at a central point in the telecom network. The framework is presently being used to compare several front-end feature extraction ideas [3].

Furthermore, for end-to-end voice recognition, [10] an empirical comparison of the CTC, RNN-Transducer, and attention-based Seq2Seq models has been indicated. The result would be to stress that, on the prominent Hub5'00 benchmark, Seq2Seq and RNN-Transducer models both beat the best-known CTC models with a language model.

II. Natural Language Processing

Natural language processing research has gotten a lot of attention in recent years. Natural language processing (NLP) is a tremendously active area of study and development since it is a computerized technique to interpreting text. [6] Paraphrase that has been formalized Once upon a time; research suggested a platform that combined deep learning approaches with GPU-based mind training. The study focuses on basic semantic issues, such as efforts to generalize semantic role labeling to all words, models for generic coreference resolution, semantic parsers that produce relatively competitive meaning representations, and semi-supervised learning of better word representations. In contrast to the foregoing, research [7] proposes a detached natural language processing (Rasa NLU) from the dialog management unit. Its API is built on scikit-learn and Keras, with a consistent API taking precedence over strict inheritance. For text classification, Rasa NLU employs a fastText method that combines pre-trained word embeddings with trained intent classifiers. When compared to competing systems, Rasa NLU comes out on top in benchmarks [8].

III. Conversational Artificial Intelligence (Dialogue Management)

Traditionally, conversation management systems have been designed using a unique pipeline that combines distinct modules for language interpretation, state monitoring, action selection, and language creation. With the method described above, each module must be trained separately with labeled data. However, the engineering process's complexity and closely linked module dependencies compelled researchers to seek alternate alternatives. Researchers were able to build methods that infer a latent representation of the state thanks to recent advances in recurrent neural networks, but they lacked a generic way to inject domain information and restrictions. By [1] training

a recurrent neural network on text transcripts of discussion, a latent representation of state may be inferred, eliminating the requirement for state labels. Furthermore, rather than enabling the dialogue management system to acquire the domain knowledge for the scenario from the conversation, [1] provides a method in which the developer has the ability to represent the domain knowledge for the scenario using software and action templates. This method promotes concern separation by allowing domain information and restrictions to be embodied in software and control flow to be learnt from these inputs.

It gives developers more control and only requires a small amount of data to train the system. It outperforms the performance of simply learnt models and rule-based systems, according to the findings of the aforementioned study. Integration of reinforcement and supervised learning in the existing approach is also recommended as a way to improve the system.

According to [2,] present neural model-based conversation production appears to be a tried-and-true approach, with the sole drawback being that the dialogue generating process entirely ignores the discussion's future conclusion. The aforesaid issue was addressed using reinforcement learning algorithms, which allowed produced conversations to take into account the response's future result via a specified reward function. Researchers were able to enhance reward-based interactive answers that encourage a more prolonged discussion using deep reinforcement learning, as mentioned in the above study paper.

[3] The addition of a transfer learning technique to an existing goal-oriented conversation system in a closed domain can increase the system's learning rate by 5 to 10 times, with a response generation success rate of more than 20%. This method, as opposed to a deep reinforcement learning strategy, may be utilized in a space with a limited volume of data. This technique also greatly increases the system's success rate, even when large amounts of domain-specific data are available. The majority of the study was split between flat reinforcement learning agents and rule-based agents for dialogue management.

The latter technique, which is based on a hand-coded approach, relies on a deterministic set of rules and lacks a high-level description of a conversation system [4]. Complex tasks [5] are formulated in a mathematical framework of options over Markov Decision Processes (MDPs), and a hierarchical deep reinforcement learning approach to learning a dialogue manager that operates at various temporal scales outperforms a flat reinforcement learning agent and rule-based agents significantly.

This study introduces a dialogue management engine with three components: a top-level sub-task selector that chooses a subtask or option for the given input, a low-level dialogue policy that chooses primitive actions for the above-selected sub-task, and finally, a global state tracker object that spans the entire conversation tree to keep track of the dialogue's future outcome. According to the research's test results, the agent's hierarchical structure increased the discussion flow's coherence.

IV. Speech Synthesis

Each spoken word is formed by combining a set of vowel and consonant speech sound components in a phonetic manner. Voice Synthesis is the technique of turning any arbitrary text in any language into a matching speech sound unit [11], [12]. Text-To-Speech is another name for this (TTS). Many academics select this issue for their studies since it is now a hot topic in the world of information technology. However, this is an issue that has been discussed in this field since the 18th century [13]. However, scholars in this sector are more interested in the machine learning method. Many different forms of study have been conducted in this subject since its inception, using a variety of languages. Slovak [14], Indian languages - Tamil, Hindi, Malayalam, and Telugu [15], Devanagari script Indian languages [16], Indonesian [17], Moroccan Arabic [18], and so forth.

Based on the objective of their research and the language they concentrated on, they employed a variety of methodologies and procedures. The primary goal, however, remains the same: to transform a text format into a sound signal. There are a variety of speech synthesis methods available, such as [12].

- Unit Selection Synthesis: This approach makes use of a vast database of recorded words, which gives the product a more genuine feel.
- Diaphone Synthesis: This approach maintains tiny units of speech and uses a smaller database than unit selection. As a result, the product is less natural than that produced by unit selection synthesis.
- Domain Specific Synthesis: This approach is typically employed in systems that require a limited vocabulary.
- Formant Synthesis: The source-filter model is used in this approach. Cascade and parallel structures are the two sorts of structures. These two kinds can also be combined to improve performance.
- Articulatory Synthesis: This is based on human speech production system modeling and is difficult to execute. [12], [19]
- Hidden Markov Model

Researchers shifted to a context-independent method since the input is unpredictable when employing voice synthesis. Even humans can learn to pronounce new words quickly by studying the pronunciation of existing ones. This is the foundation of this context-agnostic method.

1.1.7 The Sinhala Language

The Sinhala language dates back over two thousand years. It is a north Indian language similar to Hindi, Bengali, and others. Its most closely related language is Divehi, which is spoken in the Maldives islands (Pannasara and Arachchi, 2011). Contemporary Pali, Sanskrit, Tamil, Portuguese, Dutch, and English are only a few of the languages that have impacted Sinhala. The Sinhala alphabet is a Brahmic family script that is employed in the Sinhala writing system. It is one of the world's longest alphabets. Sinhala is one of Sri Lanka's official languages and the mother tongue of 74 percent of the country's people. Sinhala has 40 segmental phonemes, 14 vowels, and 26 consonants in spoken form.

| | Front | | Central | | Back | |
|------|-------|------|---------|------|-------|------|
| | Short | Long | Short | Long | Short | Long |
| High | i | i: | | | u | u: |
| Mid | e | e: | ə | ə: | o | o: |
| Low | æ | æ: | | | a | a: |

Figure 1.4: Spoken Sinhala Vowel Classification

| | | Labial | Dental | Alveolar | Retroflex | Palatal | Velar | Glottal |
|----------------------------|-----------|--------|--------|----------|-----------|---------|-------|---------|
| Stops | Voiceless | p | t | | t̪ | | k | |
| | Voiced | b | d | | d̪ | | g | |
| Affricates | Voiceless | | | | | c | | |
| | Voiced | | | | | j | | |
| Pre-nasalized voiced stops | | ḃ | ḋ | | Ḍ | | ḡ | |
| Nasals | | m | | n | | ɲ | ŋ | |
| Trill | | | | r | | | | |
| Lateral | | | | l | | | | |
| Spirants | | f | s | | | ʃ | | h |
| Semivowels | | v | | | | y | | |

Figure 1.5: Spoken Sinhala Consonant Classification

In Sinhala, there are four nasalized vowels that appear in two or three words. They are /ã/, /ã:/, /æ̃/ and /æ̃:/. /æi/, /iu/, /eu/, /æu/, /ou/, /au/, /ui/, /ei/, /oi/ and /ai/ are all diphthongs in spoken Sinhala.

1.1.8 Autism Spectrum Disorder

ASD is a complicated developmental disease characterized by chronic difficulties with social communication, limited interests, and repetitive conduct. While autism is a lifelong condition, the degree to which these problems affect one's ability to operate differs from person to person with autism [19].

Before a kid turns one year old, parents/caregivers or physicians can detect early indications of this condition. However, by the time a youngster is two or three years old, symptoms are usually more persistent. In certain circumstances, the functional impairment associated with autism may be modest and not noticeable until the kid begins school, after which their impairments may become more evident when they are with their classmates [19].

Autism spectrum disorder (ASD) is a social interaction, communication, and behavior problem. ASD can be detected at any age, although symptoms usually appear within the first 24 months of life. In low- and middle-income countries (LMICs) like Sri Lanka, however, there is little evidence supporting ASD prevalence estimates. Due to research and financial constraints, ASD diagnosis in LMICs is lower than in developed nations. Sri Lanka has only conducted little study to establish how many of its inhabitants are autistic, and health authorities who claim ASD is not present in the region are likely unaware of how to recognize it. This is a really serious problem, and additional assistance and services for these people in this country are desperately needed. Early detection of ASD in kids allows for aggressive intervention prior to the completion of neural pruning.

I. Symptoms of Autistic Kids

Autistic kids may have difficulty relating to and communicating with others. They may acquire language more slowly, have no language at all, or have substantial difficulties understanding and using spoken language. They may not utilize gestures to compensate for their difficulties with language. Autistic youngsters typically talk to ask for something or to express their dissatisfaction. For social reasons, such as sharing information, they are less inclined to communicate. They also have a hard time recognizing when and how to interact with others in a socially acceptable manner. They might not establish eye contact or allow another person to have a turn in a discussion, for example.

Kids must be able to understand what others say to them (receptive language), express themselves using words and gestures (expressive language), and use their receptive

and expressive language abilities in socially appropriate ways in order to communicate effectively.

II. Autism Screening Tools

The advent of technology allows for a variety of methods to autism screening techniques, both official and informal. These might be anything from casual observations to official evaluations. The following are some of the most widely used autism screening tools:

- The M-CHAT (Modified Checklist for Autism in Kids, Revised) is a common 20-question exam for toddlers aged 16 to 30 months. According to new research, the M-CHAT may be less successful in screening females, minorities, urban youngsters, and kids from low-income families.
- The Ages and Stages Questionnaire (ASQ) is a developmental screening instrument that looks at developmental issues at different ages.
- STAT (Screening Instrument for Autism in Toddlers and Young Kids) is a twelve-activity interactive screening tool that evaluates play, communication, and imitation.
- PEDS stands for Parents' Evaluation of Developmental Status and is a developmental parent interview that looks for deficits in motor, language, self-help, and other areas.

1.2 Research Gap

Speech Recognition, Conversational Artificial Intelligence, and Speech Synthesis, all of which are linked to the suggested solution presented in this research, have all seen a considerable amount of research. Many studies have shown that the products that are similar to the approach provided in this study can perform the following tasks.

Because of advancements in speech recognition technology, systems can now handle real-time speech recognition in a variety of languages. The majority of platforms, according to the study conducted on each platform, do not support the voice

recognition procedure. Users of currently accessible speech synthesis platforms are confined to the voices given by the platform, making the majority of them sound to the listener like automated presences. Finding a framework that supported training unique voices to offer more lifelike voice output after the procedure was tough. Despite the fact that most of these platforms accept languages from all over the world, we have yet to identify a single platform that supports Sinhala.

Researchers have used a variety of techniques to reach the end objective of task completion in relation to Dialogue Management, with the bulk focused on the English language. Some research focuses on establishing an initial knowledge base that may be used to map conversations using entities and actions. The output is created by mapping the input source content into knowledge base entities and actions in those systems.

The capacity to control the state of the present discussion has been added to these sorts of studies, transforming the traditional stateless dialogue management into stateful dialogue management. The development of GPU processing and machine learning allowed researchers to focus on a deep learning-based neural network technique for dialogue management. This technique has been the focus of most recent research in this subject. The issue of Sinhala language support was also present in these study areas.

However, the suggested platform would largely focus on voice detection, conversational AI, and speech synthesis in Sinhala. The platform is a stand-alone solution that can be installed locally and modified to fit the needs of the organization. In addition, rather than the general approach used by most other platforms, the machine learning algorithms is taught using current data to produce a domain-specific representation.

1.3 Research Problem

Technology is improving every day, making people's life simpler. Smart devices are a great platform for a computer-aided tool since they are widely available and widely used. Due to users' interest and need for voice assistants, they may get a lot of attention

as a result of this reality. A voice assistant is a built-in software application that allows users to converse with one other using natural language. Voice assistants are employed for a variety of reasons, but its primary goal is to recognize and reply to users' words.

E-commerce, insurance, health care, retail hospitality, and logistics are just a few of the businesses that employ voice assistants. User support, sales and marketing, order processing, and social networking are the most common corporate activities that voice assistants may perform [13]. Voice assistants are typically programmed to react to Frequently Asked Questions (FAQ) in a certain subject.

Enterprises, banks, and financial institutions are enticed to employ voice assistants for a variety of reasons. They can, for example, lower the cost of giving information to their clients while increasing the speed and success rate. They may also get important information from users or bot users and experiment with new ideas as a result of employing voice assistants.

It's important to maintain track of voice assistant interactions since they might lead to track the progress. Users may also offer feedback on their experience, whether positive or negative. The comments are valuable in influencing future decisions. Voice assistants, if properly built, may remove repetitive activities, making workloads lighter, simpler, and faster while increasing user satisfaction for industries that use them.

Furthermore, if users have to wait a long period for a bot to respond, their experience with voice assistants becomes less valuable, and they may feel annoyed. As a result, whatever internal processing is required to interpret the user's message and create a suitable response, the user's waiting time should be reduced to a minimum. Users may feel dissatisfied if this does not happen. Advanced voice assistants have been developed as a result of advances in the disciplines of Machine Learning and Artificial Intelligence. However, because most voice assistants are only available in English, there are certain users who are unable to fully utilize the existing voice assistants owing to a language barrier. Because users are not required to participate in a formal discussion while speaking with a bot in their native language, the level of difficulty in

communicating decreases. Otherwise, users may be hesitant to ask inquiries, preventing them from openly talking. Furthermore, there are no voice assistants that target a specific domain that have been developed for the Sinhala language.

A generic voice assistant has a significant likelihood of not replying to the majority of inquiries, rendering it useless. Users may immediately ask inquiries and solve difficulties by pointing to the particular point when using a domain-specific voice assistant instead of wasting time exploring the internet. As a result, building 'A Sinhala voice assistant for screening kids with autism spectrum disorder' primarily addresses two issues by being both language and domain specific (Sinhala). However, this solution should work in any domain, with only the dataset and a few custom components changing to match the domain.

In most conversational platforms, individuals tend to utilize basic phrases. As a result, the linguistic complexity issue in voice assistant systems, such as in extensive talks, is less of an issue. This opportunity is also available for the Sinhala voice assistant. In addition, if the user's utterance falls short of the bot's ability to determine the user's purpose, the bot can ask the user to repeat what they've already stated.

By increasing the engagement until the user obtains an acceptable response, the dialogue becomes more lifelike. Furthermore, voice assistants do not need to worry about a language's complicated written grammatical structures, as basic verbal grammar suffices in most situations [2].

Retrieval-based bots and generative-based bots are the two primary types of bots. A retrieval-based bot selects the best appropriate response for a user input using a preset collection of replies and a heuristic. In contrast to a retrieval-based bot, a generative model seeks to transform a given user input to an output by creating an appropriate answer rather than using a preset collection of replies.

A growing number of industries are attempting to integrate a voice assistant into their everyday operations. According to recent studies, voice assistants may be used to automate some activities in areas such as medical, education, information retrieval, business, e-commerce, and entertainment. [14].

Technology is improving every day, making people's life simpler. However, there are certain aspects that remain unexplored. ASD awareness is poor in LMICs like Sri Lanka due to cultural reasons. Due to a lack of resources, people with ASD are frequently left untreated for lengthy periods of time after being diagnosed. To enhance the clinical results of small infants with ASD, early detection and diagnosis are critical. Smart gadgets are a great platform for a computer-aided tool since they are widely available and widely used. The majority of currently available screening solutions automate basic screening checklists like M-CHAT/F. To make a choice, only the ASD.AI program uses an intelligent machine learning model. Using a culturally appropriate symptom checklist and an integrated machine learning model, this research offers a unique method for ASD screening. On PAAS data acquired clinically, a variety of supervised learning models were trained. The suggested application outperformed conventional paper-based techniques in terms of prediction (PAAS). By allowing non-specialist healthcare providers to test for ASD during home visits, the new software helps to raise ASD awareness and identification. Furthermore, useful data on the prevalence of ASD in LMICs (which is currently sparse) may be obtained, and resources can be allocated appropriately to reduce treatment delays.

1.4 Research Objectives

The main aim of this study is to create a Sinhala voice assistant that can be used to assess kids for autism spectrum disorder. So that it may function as a conversational agent. The key objectives are to eliminate the language barrier in voice assistants by making Sinhala available and allowing the voice assistant to understand and react to in-domain inquiries.

1.4.1 Main Objective

The main objective of this solution is to develop a machine learning-based automated autism screening tool that decrease or remove the need for error-prone and wasteful human involvement in the field. The capacity of the system is to support both English and Sinhala. The efficiency and reliability of the system is having a direct influence on the service quality.

1.4.2 Specific Objectives

To achieve the main objective, specific objectives that need to be attained are as follows:

I. Increase availability

This platform's intelligent agents may be operational and easily engage with their stated purpose 24 hours a day, 365 days a year. Humans are emotional beings, and their present state of mind may have a direct influence on the level of service they give. On the other hand, the intelligent agent has a 0% probability of this happening.

II. Process many requests at once

Intelligent agents are deployed using this platform based on the current load of requests to be handled, and the agents is able to handle numerous conversations at once, unlike their human counterparts.

III. Lowered Cost

The system is simple to configure to fulfill a variety of demands throughout time. When compared to existing systems, intelligent agents are having low to no known maintenance costs once installed. Because of the modular foundation on which it was developed, the system is straightforward to adapt to various languages.

IV. Enhancing the service's overall productivity

Intelligent agents are reactive and efficient once deployed via the suggested platform. Users of the system can gain productivity, time, and scalability thanks to the intelligent agent's capacity to communicate with numerous users at once.

2. RESEARCH METHODOLOGY

2.1 Requirement Gathering and Analysis

Obtaining and evaluating requirements is the most essential aspect of any software development project. Because the answer to the problem is unknown, gathering and assessing needs is essential, particularly in research. As a result, a requirement must be stated clearly, realistically, and correctly. It's crucial to have a solid grasp of the research issue. Before going on to the design and implementation phases, it is important to ensure that the proposed system is the best solution for the identified problem.

So, a comprehensive collection and analysis of requirements is necessary. The project's needs were gathered by consultation with a physician, and then a research was undertaken to better understand the speech difficulties of ASD kids aged 1-4 years. This phase requires all of the criteria that have been gathered and assessed to be presented and recorded.

2.2 System Diagram

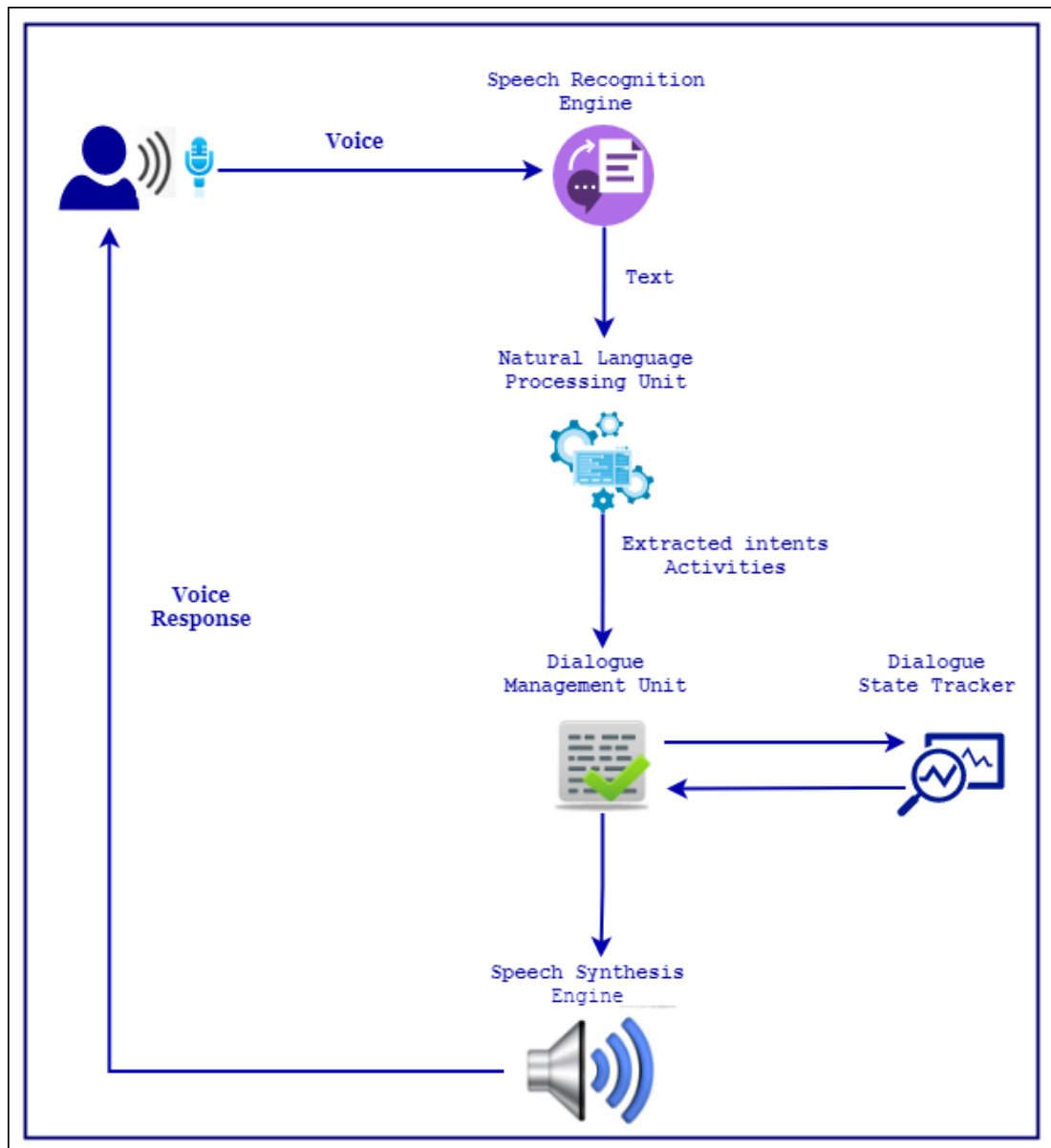


Figure 2.1: High Level Architecture

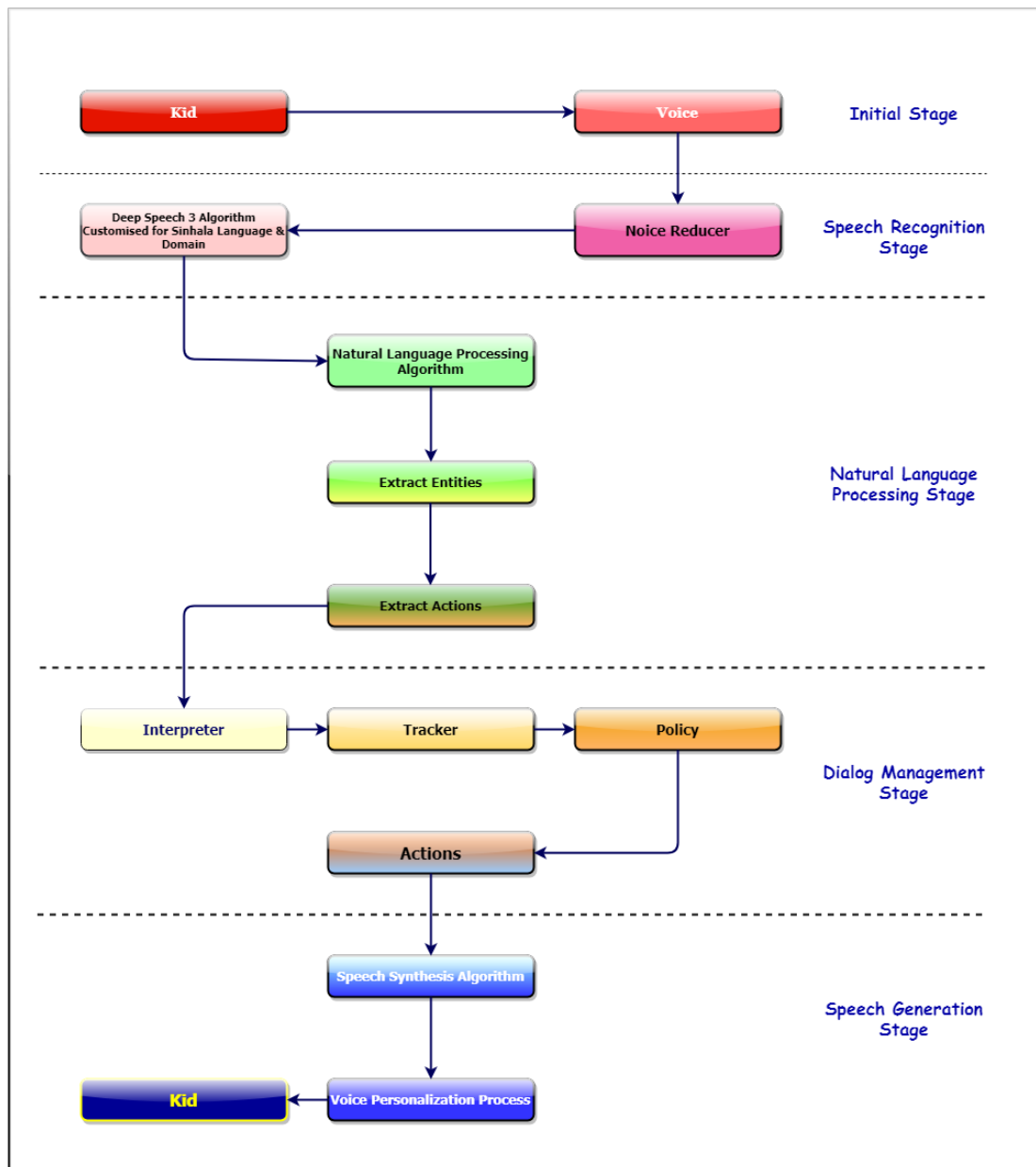


Figure 2.2: System Architecture

2.3 Methodology

The suggested solution was built on existing research in key platform areas such as Speech Recognition, Conversational Artificial Intelligence, and Speech Synthesis, as well as the unique capacity of Sinhala Language support and a standalone decoupled solution. The approach relies significantly on machine-learning algorithms to attain human-level intelligence in decision-making, conversation creation, and taking actions that increase the likelihood of accomplishing the final objective of autism screening.

Initially, the device is programmed with the kids's voices. The noise reduction in the system eliminates any noise associated with the current voice file. Then, using a speech recognition engine based on Baidu researchers' deep speech implementation [20] and the most recent version of the implementation, version 3, [10], a speech recognition engine extracts entities and actions related with the present voice data into text format. To support Sinhala voice recognition, the supplied implementation is modified accordingly. The architecture offered by the aforementioned implementation is the most important aspect in deciding which implementation to choose. The chosen technique is a well-optimized RNN training system that leverages several GPUs to train on a huge quantity of diverse data, rather than typical text-to-speech engines with laboriously built processing pipelines and poor performance in noisy situations. This allows us to train Sinhala language-specific data without the need for bespoke pipelines and phoneme dictionaries. In addition, the implementation emphasizes the suggested system's capacity to manage difficult loud situations. The chosen method even allows for training with a labeled transcript data set. The Sinhala language's machine learning model for voice recognition is trained using a corpus of existing Sinhala conversations. Additionally, Mozilla's common voice data collection is used for the English language.

This platform's natural language processing capabilities add Sinhala language processing features to the opensource implementation given by the Rasa framework [7]. The natural language processing engine extract the purpose, entities, and any other structured information from the given text in the Sinhala language after obtaining the audio data as text via speech recognition. The conversation management unit receives

the processed data from the natural language processing unit [7]. The dialogue management unit, which is in charge of creating replies to user inquiries with the end aim of job completion, was built using various methods proposed by recent research for high accuracy and speed. The architecture of the conversation management engine closely resembles that of the Rasa dialog management unit. The major benefit of utilizing the aforementioned design is that it allows users to respond to whether the engine's anticipated behavior is right or incorrect, depending on the circumstance. These features enable the platform to recommend actions to human agents while the platform is still evaluating itself. In addition, the platform can view a graph of training conversations, which may subsequently be turned into a business domain knowledge base. The suggested dialog management unit [7] was improved by including deep reinforcement learning [2], which has the capacity to create utterances that maximize future reward, thereby capturing the global features of a good discussion. This change enables for more engaged and different replies, resulting in a longer dialogue. This change enables for more engaged and different replies, resulting in a longer dialogue.

This approach then uses speech synthesis to turn the dialogue management unit's answer to a human voice, resulting in a human-like discussion between kids. To maintain the dialogue between the client and the bot more genuine, the speech synthesis engine should allow real-time Sinhala text to speech conversion with the very least degree of latency in the process. A voice synthesis engine based on the Deep Voice 3 [10] implementation suggested by Baidu researchers are constructed to accomplish the above-mentioned efficiency and performance. On a single GPU server, the above architecture can support 116 queries per second, which can enable the above use case of real-time text to speech communication with minimal or no latency. This architecture can identify each voice dialogue via multi-speaker speech synthesis, which eliminates the robotic answer generation found in currently available commercial systems.

Custom voice assistants are simple to create using platforms like Google Assistant. However, the free source Rasa, Mozilla DeepSpeech, and Mozilla TTS technologies may be used to create a local assistant that protects data privacy. Voice-first assistants

are likely to be the next big thing for user interactions across many sectors, with platforms like Google Assistant and Alexa growing increasingly popular. However, unless hosted off-the-shelf solutions are utilized, voice assistant creation presents a new set of problems that extend beyond natural language understanding and dialogue management. In addition, speech-to-text and text-to-speech components, as well as the frontend, must be taken care of. From the backend to the frontend, a voice assistant that operates locally and maintains data security is created using solely open-source technologies.

2.3.1 Tools and software

Following tools and software were used for the implementation.

1. Language: Python 3.6
2. Frameworks/Libraries:
 - RASA Core Version: 0.13.0
Rasa is a software development platform for creating conversational applications. Rasa utilizes a machine learning model to build up the bot's logic instead of dealing with a collection of if/else statements, and the bot is trained on sample tales or dialogues.
 - RASA NLU Version: 0.14.0
Rasa NLU is an open-source natural language processing technology that may be used in chatbots to classify intent and extract entities.
 - RASA SDK
This is the Rasa SDK for creating custom actions.
`http://localhost:5055/webhook` is the address of the action server.
3. Chat Platform: Slack
4. RASA Core allows third-party chat services to be integrated. Slack will be utilized as the conversation platform in this study. Interactive features may be utilized in Slack.

5. Anaconda Navigator Version: 1.9.6

Anaconda Navigator is a data science distribution that is open source. Anaconda is an Anaconda Inc. product. It has features for R and Python programming languages. Anaconda is a brand name for a product made by Anaconda Inc. It acts as a single point of administration for all required packages, many of which are already preloaded. It is possible to install tools, libraries, and their dependencies more quickly and easily. Anaconda's ability to manage different environments based on Python version is another helpful feature. This became quite handy when Rasa's versions evolved on a regular basis.

By retaining several Rasa Environments, the bot may be tested for new features and enhanced as a result of those version updates. While the bot is being operated in one Anaconda environment with rasa core and rasa nlu, another Anaconda environment is being built up to host the Action server, which is necessary to perform the custom actions. Rasa-sdk is installed in the next environment. When utilizing the Rasa Core component, Rasa also recommends Anaconda.

6. SQLite Database Version: 3.28.0

SQLite is a SQL database engine that is compact, quick, self-contained, high-reliability, and full-featured. We utilize custom actions that query the database to get information because it is impractical to cover the entire domain in direct message messages. Because SQLite is a lightweight database management system, it will deliver the required information in the shortest time possible.

7. Ngrok For Local Testing

Ngrok enables the creation of a local webhook from the bot's deployment machine, allowing it to be made publicly available on the internet and utilized with apps such as Slack, Facebook, and others.

Five main components were used to build ASD.AI.

- A frontend that users use to connect with the assistant is known as a voice interface (web).
- Speech-to-text (STT) is a voice processing feature that accepts user input in an audio format and converts it to text.
- NLU is a component that accepts text input and extracts structured data (intents and entities) to help the assistant understand what the user wants.
- Dialogue management is a feature that assesses how an assistant should reply at a certain point in a discussion and then creates that response in text format.
- Text-to-speech (TTS) is a component that converts a text answer from an assistance into a vocal representation, which is then delivered back to the user.

Here, Mozilla DeepSpeech is used, which a speech-to-text framework that accepts user input in an audio format and converts it to a text format that can then be analyzed by NLU and conversation systems using machine learning. Mozilla TTS does the opposite: it takes a text input (in our example, the ASD.AI response generated by the conversation system) and converts it to an audio representation using machine learning.

The backend of the ASD.AI is made up of NLU, conversation management, and speech processing components. The open-source voice assistant includes Rasa, Mozilla DeepSpeech, Mozilla TTS, and Rasa Voice Interface.

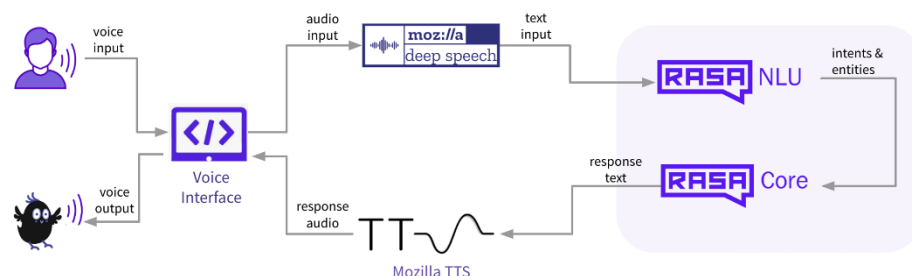
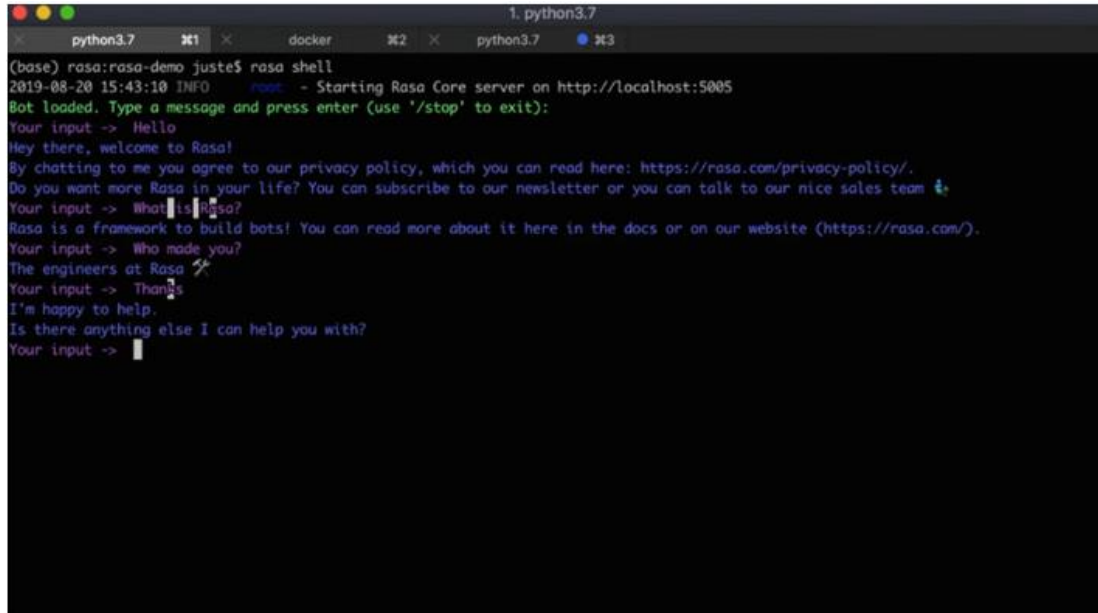


Figure 2.3: Components of Open-Source Voice Assistant

2.3.2 The Rasa Assistant

Sara, an existing Rasa assistant, was utilized for this project. It's a Rasa-powered open-source assistant that can answer a variety of queries and it helps to get started with the Rasa framework. Here's an example of a conversation had with Sara:

A screenshot of a terminal window with three tabs: 'python3.7', 'docker', and 'python3.7'. The active tab is 'python3.7'. The terminal shows the following text:

```
(base) rasa:rasa-demo juste$ rasa shell
2019-08-20 15:43:10 INFO rasa: - Starting Rasa Core server on http://localhost:5005
Bot loaded. Type a message and press enter (use '/stop' to exit):
Your input -> Hello
Hey there, welcome to Rasa!
By chatting to me you agree to our privacy policy, which you can read here: https://rasa.com/privacy-policy/.
Do you want more Rasa in your life? You can subscribe to our newsletter or you can talk to our nice sales team 📧
Your input -> What is Rasa?
Rasa is a framework to build bots! You can read more about it here in the docs or on our website (https://rasa.com/).
Your input -> Who made you?
The engineers at Rasa 🦊
Your input -> Thanks
I'm happy to help.
Is there anything else I can help you with?
Your input -> 
```

Figure 2.4: Sample Convocation with Rasa Assistant

Steps for installing Sara on a local PC are as follows:

- Clone the Sara Repository
git clone https://github.com/RasaHQ/rasa-demo.git
cd rasa-demo
- Install the Necessary Dependencies
pip install -e .
- Train the NLU and Dialogue Models
rasa train --augmentation 0
- Test Sara
docker run -p 8000:8000 rasa/duckling
rasa run actions --actions demo.actions
rasa shell --debug

2.3.3 Architecture

- A microphone is used to record the user's voice.
- A speech-to-text service converts spoken words into text (Botium Speech Processing)
- A natural language understanding engine (NLU) pulls information from text (Rasa)
- A text answer is created by a conversation engine (Rasa)
- A Text-To-Speech service converts written text into audio (Botium Speech Processing)
- The user hears the audio file.

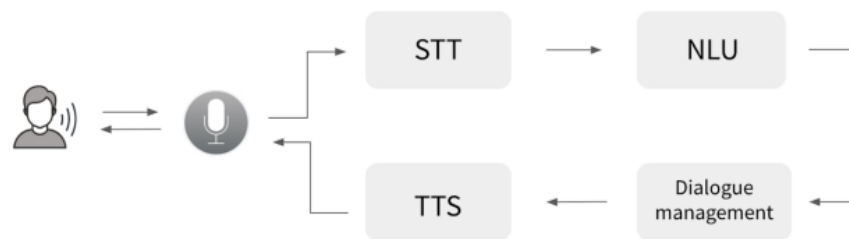


Figure 2.5: Architecture of Voice Assistant

Rasa's architecture is designed to be modular. This makes it simple to integrate with other systems. Rasa Core, for example, may be used as a conversation manager with NLU services other than Rasa NLU. While the code is written in Python, both services may expose HTTP APIs so that applications written in other languages can simply access them.

Here, the status of the dialogue is stored in a tracker object. Each discussion session has one tracker object, which is the system's single stateful component. A tracker keeps track of slots as well as a history of all the events that led up to that point in the dialogue. By repeating all of the events in a discussion, the state of the conversation may be recreated. When Rasa receives a user message, she follows the procedures

shown in Figure 1. Rasa NLU takes care of the first step, while Rasa Core takes care of the rest.

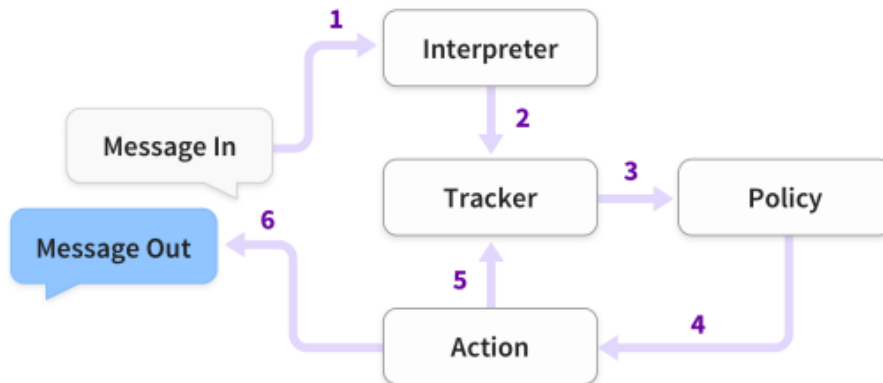


Figure 2.6: Dialogue Management

1. A message is received and sent to an interpreter (e.g., Rasa NLU) for the purpose of extracting the intent, entities, and any other structured data.
2. The tracker keeps track of the status of the dialogue. It is notified when a new message has arrived.
3. The policy receives the tracker's status.
4. The policy determines the next course of action.
5. The tracker records the specified action.
6. The activity is carried out (this may include sending a message to the user).
7. Return to step 3 if the expected action is not 'listening'.

The dialogue management problem is stated as a classification issue. Rasa Core guesses the action to take from a specified list at each cycle. An action can be as basic as sending a message to the user, or as complex as executing an arbitrary code. When an action is performed, it is given a tracker instance, which allows it to use any relevant information gathered throughout the dialogue's history: slots, past utterances, and the outcomes of prior actions. Actions can't change the tracker directly, but they can return a list of events when they're run. The tracker uses these events to keep track of its status. There are several distinct sorts of events, such as `SlotSet`, `AllSlotsReset`, `Restarted`, and so on.

A policy's duty is to choose the next action to do based on the tracker object. A policy is created coupled with a featurizer, which uses the tracker to produce a vector representation of the current discussion state.

The typical featurizer combines features that describe:

- What was the most recent action that you took?
- the intent and entities of the most recent user message
- What slots have been established thus far?

A slot's feature set may differ. A slot is represented by a single binary vector element in the simplest instance, indicating whether it is full. Categorical variables are stored as a one-of-k binary vector; continuous variables can define thresholds that impact featurization, or they can simply be provided to the featurizer as a float.

The hyperparameter max history defines how many prior states should be included in the featurization. The states are layered by default to provide a two-dimensional array that may be processed by a recurrent neural network or other sequence model. In practice, it's been discovered that a max history setting of 3 to 6 works well for most issues.

2.4 Testing and Implementation

2.4.1 Training Data Formats

The dataset is quite important in the creation of the voice assistant. To predict the test's purpose or other human utterances in the future, the bot requires a big dataset to train on.

The participants in this study are autistic kids under the age of four. A speech therapy doctor was enlisted to help collect data. The information gathered is compiled into a single excel document. The entire dataset is then examined to find all of the intents and entities. After that, the intent and entities are manually tagged on the whole data set. This diagram shows how the Sinhala Voice assistant dataset is structured.

The data is initially entered into an excel spreadsheet. The whole dataset is then examined to determine the intents and entities for the gathered data. Some queries have to be classified as being outside the scope of the project. Furthermore, several questions needed further information in order to provide a response.

It is impossible to gather data in an accurate manner due to the technique used to acquire it. In such cases, we're teaching the bot to produce user stories that deal with obtaining information in an engaging way.

The benefit of a voice assistant model is that the response is grammatically correct and, more importantly, relevant to the user. The flaw is that an open domain makes it hard to gather answers. As a result, only one domain is examined in this study.

All API solutions for voice assistants use English as the standard language, and those that support other languages only give a Beta version in those languages. This restricts the API options accessible if the voice assistant is to be implemented in a different language. Furthermore, it has an impact on the voice assistant's quality. Information retrieval and prediction is poorer to those of other languages since the NLP system lacks most of the training.

The voice assistants' prediction system is also based on machine learning, which means that the more data it has to train on, the more accurate predictions it can make. This characteristic means that the voice assistant's predictions are based on more data the more user input it receives. This implies it is able to not only better differentiate the context of an input, but it is able to modify its predictions dynamically as the user inputs change.

Because the prediction method is based on machine learning, it need training in order to detect and extract essential information from user input. To build a voice assistant that can handle a range of inputs, the system requires a data set to train on, which

should be the same as the user input. It is almost impossible to interview users directly and maintain a discussion on the domain in order to collect data.

Human-readable training data types are used by both Rasa NLU and Core. A collection of utterances annotated with intents and entities is required by Rasa NLU. These can be supplied as a json structure or as a markdown document. Many text editors and web apps, such as GitHub, can generate markdown syntax, which is particularly short and simple to read. The json format is slightly more difficult to understand, but it is not sensitive to whitespace and is better suited for sending training data between apps and servers.

```
"intents": [  
    {  
        "question": "බබා, ඔයාට පුලුවන්ද එකේ ඉඳන් දහයට ගනන් කරන්න ",  
        "answer": "1 2 3 4 5 6 7 8 9 10",  
        "reply": "බෝහම හොඳයි බබා, හරියටම කීවා",  
        "reply_negative_q": "බබා ට මම කියන්නද හරි උන්නරේ ?",  
        "reply_negative_a_reply": " එක, දෙක, තුන, හතර, පහ, හය, හත, අට, නවය, දහය"}  
    ]
```

Markdown is used by Rasa Core to define training dialogues (also known as 'stories'). The beginning of a story begins with a name followed by two hashes. The name is chosen at random, although it can be useful for troubleshooting. A story's body is made up of a series of events separated by newlines.

```
##1 greet and thanks
```

```
* greet
```

```
-action_greet
```

```
*ask_name{"name":"නම"}
```

```
-utter_name
```

```
*thank_you
```

```
-utter_welcome
```

For voice recognition, natural language processing, and speech synthesis, the system requires a large amount of label data because it relies largely on deep learning algorithms. In order to train machine learning models, we require voice chats with transcripts for the proposed system. To meet the aforementioned need, a current customer service provider for the Sinhala language would employ roughly 1000 hours of voice interaction data with its produced transcript. Public datasets such as Mozilla Common Voice will be used for English language-based training. The first round of testing will be carried out with the help of human agents.

Based on the continuous dialogue with the kid, the algorithm will first recommend actions to the human agent. Each action will be marked as correct or wrong by the human agent. The system will adjust its replies in reaction to the human agent's responses, and a performance score will be provided for each agent. The evaluation will be conducted out using a performance score provided by the agent as a result of the human agent's behaviors being corrected.

2.4.2 Machine Teaching

A graph of training conversations may also be shown using Rasa Core. A narrative graph is a directed graph with nodes that represent actions. The user utterances that occur between the execution of two actions are identified on the edges. The edge label is removed if there is no user interaction between two successive activities. Each graph contains a START node at the beginning and an END node at the end. It's worth noting that the graph doesn't depict Rasa Core also offers a machine teaching method, in which developers correct actions taken by the system, in addition to supervised learning. This is a realistic technique for producing training data and quickly exploring the space of plausible dialogues, according to our findings.

A new training data point is created when you choose the right action. Rasa Core then advances the conversation to the next level by partially training the dialogue policy. The trained model is stored to a file when it has been completed, and the freshly produced training data is saved to a file.

2.4.3 Visualization of Dialogue Graphs

the entire conversation state, and that not all potential walks along the edges are included in the training set. A heuristic is employed to combine related nodes to simplify the display. Following figure depicts a produced graph before and after the simplification. It's a simple illustration of how narrative graphs may be simplified. Two tales with the identical initial encounter are included in the training data. As a result, these nodes are considered equal and merged.

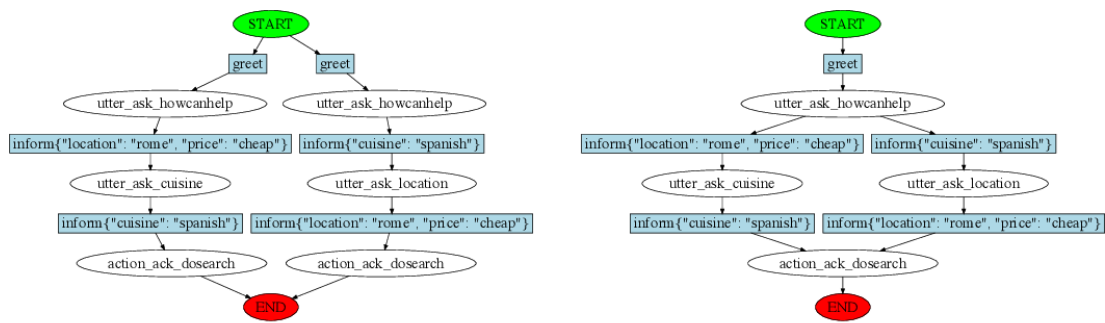


Figure 2.7: Sample story graphs with and without simplification

2.4.4 Deployment

Docker files for creating static virtual machine (VM) images can be found in both the Rasa NLU and Core repositories. This helps with repeatability and deployment across a wide range of server configurations. The HTTP API web servers enable thread-based and process-based parallelism, allowing them to manage huge request volumes in a production setting.

2.5 Commercialization of the Product

A conversational AI system might cost anything from hundreds to thousands of dollars. A speech pathologist should evaluate an autistic kid to select the appropriate technology for them, program the device with their own language, and teach them how to use it properly. Medicare may cover up to 20 sessions of visiting a therapist about using a speaker system, according to the expert providing the consultation. Some private health insurance policies may pay a portion of the consultation fee.

Conversational AI software allows those with visual impairments or reading difficulties to listen to text printed on a smartphone or computer. A visually challenged user can utilize an aural interface to comprehend and execute computer tasks when a conversational AI system is coupled with a screen reader. As a result, this system serves as an assistive device, allowing these persons to use information and communication technology. Governments all around the globe are increasingly looking for innovative methods to support kids with autism spectrum disorder.

Services are expected to account for the bulk of income in the future quarters. The functioning of conversational AI software is reliant on third-party services. They are managed by solution, platform, and service providers and are an integral component of the tool deployment process. Leading companies across a wide range of sectors are employing Conversational AI to cope with the ever-increasing amount of audio/video-based content. This aids organizations in identifying new ways to tap into the huge volumes of data accessible in order to create new goods, services, and processes, giving them a competitive edge.

It used to be simple to contact doctors and medical executives. All that was required of a firm was to send a sales representative to the practice and inform them about the medical devices. Doctors are busier than ever, and many practices no longer let salespeople into their offices. Sales and marketing strategies that worked a decade ago are no longer successful in this industry. Medical device businesses now require new strategies that match how doctors interact with marketing today. Commercial advertising can be used to bring the system in front of a specified audience. This technique may be used to reach a broader audience at various phases of development. Newspaper advertisements are also effective and may target a wide variety of qualities. Social media platforms may also be utilized to create a digital marketing platform.

3. RESULTS AND DISCUSSION

3.1 Results

Speech perception, spontaneity, computational complexity, and other factors may all be used to evaluate dialogue management systems. New assessment criteria, such as sentimental impact on the user, ability to convince the user to act, mastery of language production, and if the system analyzes environmental circumstances and adapts its actions appropriately, are likely to be required for acoustic intelligence applications. Dialogue management is a typical form of assistive technology in which a computer or tablet reads aloud letters and phrases to the user. This method is common among kids with literacy problems, particularly those who have difficulty decoding.

The youngsters can focus on the meaning of the phrase instead of spending all of their mental energy to sound them out since the words are presented in an aural manner. While this technology can assist kids in overcoming reading difficulties and gaining access to educational resources, it does not assist in the development of reading skills. However, the study found that utilizing the ASD.AI program benefitted kids with ASD. This team gave small-group software training to kids for six weeks, and they saw improvements in motivation to study, comprehension, and pronunciation. Another study found that ASD.AI was effective in aiding kids in accessing reading content and was well-liked by kids who used it, especially those aged 1-4 years.

Hidden Markov model-based systems were chosen over voice conversion unit selection techniques as being more convenient for the original speaker, according to the assessment. There is, nevertheless, a distinction in interpretation. These biases favoring hidden Markov model-based systems over voice-based systems are complex, to say the least.

A summary of prediction outcomes for the classification issue was taken using the confusion matrix. As the key to the confusion matrix, the number of accurate and wrong guesses is summed with count values and broken down by class. If there are an uneven number of observations in each class or if your dataset has more than two

classes, classification accuracy alone might be deceptive. Calculating a confusion matrix can help you see what your classification model is getting right and where it's going wrong.

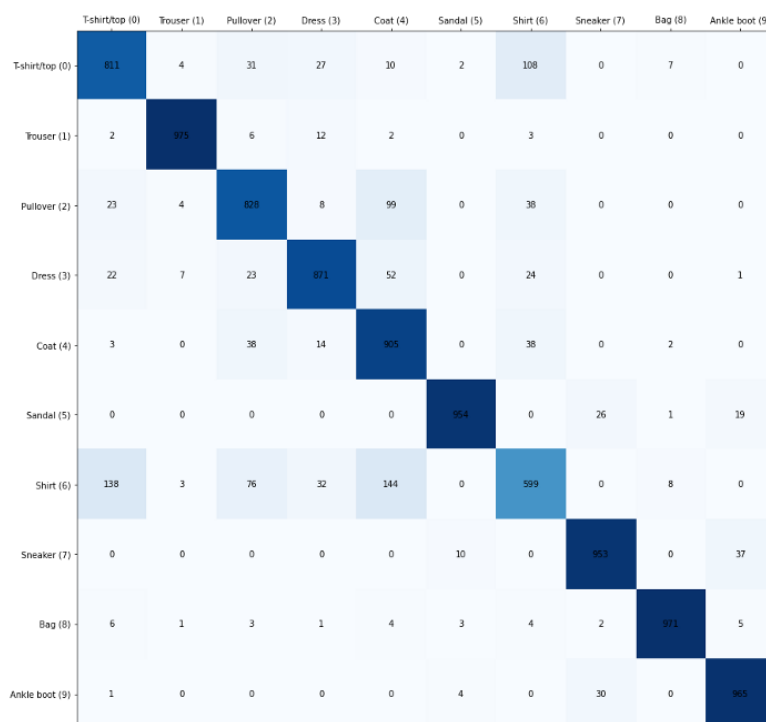


Figure 3.1: Confusion Matrix

3.2 Research Findings and Discussion

Ten youngsters who are differently abled and have speech impairments The study included 5 ASD kids (5 females, 5 boys) and 12 TD kids (4 females, 8 males) ranging in age from 4 to 6 years, as well as 5 TD younger kids ranging in age from 2 to 3 years (3 males). In the end research, all of the 4- to 5-year-old kids with ASD had a verbal age of 40 months or higher. All of the kids, according to parent assessments, spoke Sinhala fluently and had minimal exposure to other languages. Because their cognitive age was less than 40 months, data from two more kids who were previously enrolled in the ASD group was removed.

Finally, data from one extra participant in the ASD group and two extra kids originally selected for the TD group were removed due to lengthy exposure to a second language. We investigated brain responses to speech and non-speech sounds in kids with ASD who were verbally matched separately using an HMM paradigm. The goal of this research was to understand more about the brain processes that support speech detection and processing in this group of people.

This is the only ERP research we know of that looks at the detection and discrimination of speech from non-speech in 4- to 6-year-old kids with ASD without using oddball stimuli or concomitant attentional orienting responses.

4. CONCLUSION

This study offers an automated screening method for detecting children with autism that outperforms human agents in terms of efficiency, performance, productivity, and availability, with a focus on Sinhala language support. Deep learning models with multi-GPU based huge data set training, paired with a data-driven conversational system, allow our approach. We think that as computational power and Sinhala language-based data sets increase in the future, our technique will continue to improve, revolutionizing the screening procedure for children with autism.

Core and Rasa NLU are both in ongoing development. As such, they will never be "completed." They serve as a framework for making practical research in conversational AI usable by non-specialist developers. A variety of subjects are currently being worked on, including better reinforcement learning support, making NLU more resilient against typos and slang, and supporting new languages.

Analyzing the ASD.AI's results lead to the following conclusion:

- Rasa may be used to create a number of different voice assistants.
- To construct conversational models like Voice assistants, it is not necessary to have preset word vectors for the language.
- Entity Extraction works nicely with Conditional Random Fields (CRFEntityExtractor in Rasa).

Testing difficulties unique to voice assistant apps are different from testing issues unique to other apps. Separate testing of the speech sections of the application from the rest of the application is possible and essential. The performance of both voice input (speech to text) and voice output (speech to speech) was evaluated in this dissertation (text to speech). To summarize, the speech synthetic systems that were attempted to install in order to aid ASD children with their speech impairments will assist them during their youth and will help them socialize with society much better than they do currently.

In this environment of data scarcity and domain specialization, even seemingly modest changes in the quantity and similarity of supplementary data can have a significant impact on model performance. Data quality over quantity was more important, with a restricted age range that matched the objective data outperforming alternative combinations with higher variation and number. It was also shown that if domain similarity can be detected and elaborated upon, model performance may be enhanced even with greater amounts of data.

When considering future work, people frequently say many things in a message, thus enhancing a voice assistant's ability to grasp various intents will provide the voice assistant with even more value. Only one purpose per communication is recognized in this study, and it is reacted to appropriately. In addition, including some Sinhala NER tagging in the bot's training will help it recognize well-known named things.

- The bot does not refer to or keep previously acquired material in its knowledge to answer future queries or throughout the discussion.
- Even if a user asks the same question several times, the bot will answer each time without displaying a notice indicating that the question has already been addressed.

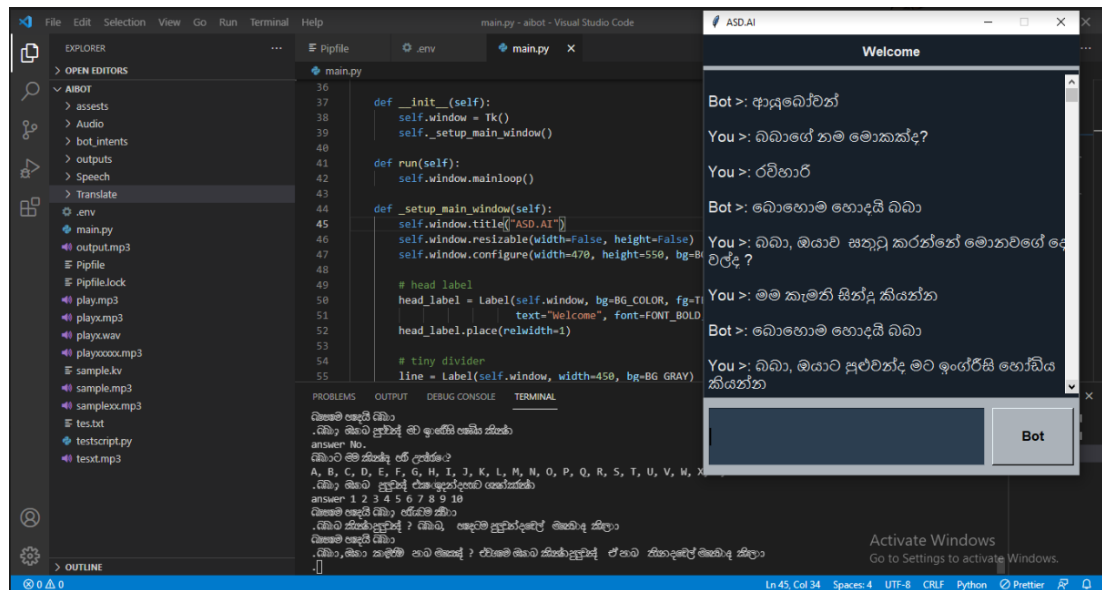
REFERENCES

- [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-developers2e2da2817f3d>. [Accessed: 09- Sep- 2018].
- [2] Hettige, B. and Karunananda, A. (2006). First Sinhala Voice assistant 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 Voice Assistant Models. [online] Research Gate. Available at: https://www.researchgate.net/publication/323587007_Deep_Learning_Based_Voice_assistant_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", Snipsnlu.readthedocs.io. [Online]. Available: <https://snips-nlu.readthedocs.io/en/latest/>. [Accessed: 11- Sep- 2018].
- [7]"An Overview of Voice assistant – Voice assistants Life", Voice assistants Life. [Online]. Available: <https://voiceassistantslife.com/an-overview-of-voice-assistant-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].

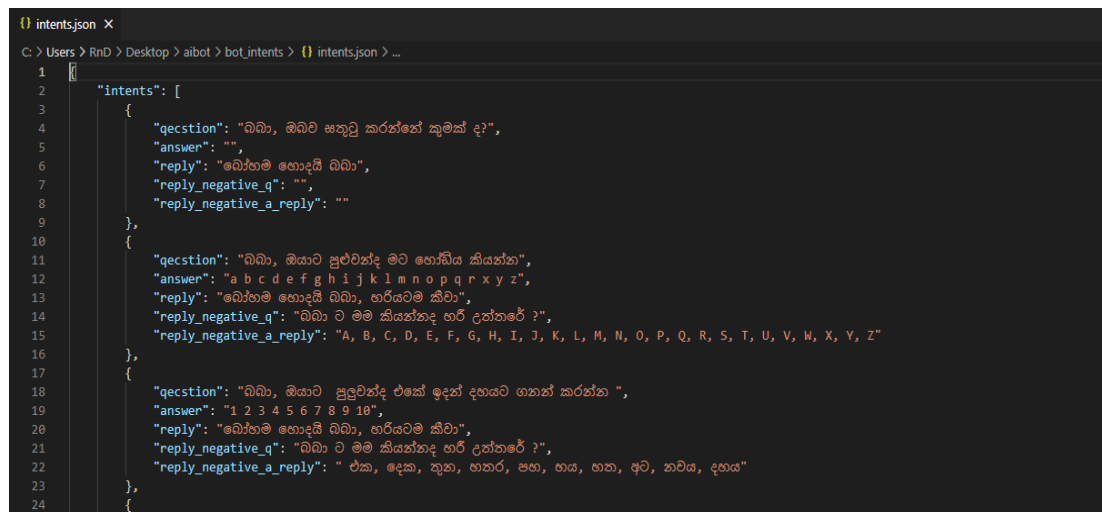
- [10] "LUIS: Language Understanding Intelligent Service", Luis.ai, 2018. [Online]. Available: <https://www.luis.ai/home>. [Accessed: 14- Sep- 2018].
- [11] Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using 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] "Voice assistant Market Survey- 2017: Mindbowser Info Solutions", Mindbowser, 2018. [Online]. Available: <http://mindbowser.com/voice-assistant-market-survey-2017/>. [Accessed: 10- Aug- 2018].
- [14] M. KOTTORP and F. JÄDERBERG, Voice assistant 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].

APPENDICES

Appendix 1 - Implementation



Appendix 2 - Intents File (intents.json)



Appendix 3: Stories File (stories.md)

```
File Edit Selection View Go Run Terminal Help
stories.md x
C: > Users > RnD > Desktop > aibot > bot_intents > stories.md > ...
1  ##1 greet and thanks
2  * greet
3  -action_greet
4  *ask_name{"name":"කම"}
5  -utter_name
6  *thank_you
7  -utter_welcome
8
9  ##2 greet
10 * greet
11 -action_greet
12 *ask_name[{"name":"කම"}]
13 -utter_name
14
15 ##3 greet and thanks
16 * greet
17 -action_greet
18 *thank_you
19 -utter_welcome
20
21 ## say thanks
22 *thank_you
23 -utter_welcome
24
```

Appendix 4: Plagiarism Content

evturnitin.com/app/carta/en_us/?lang=en_us&student_user=1&co=1672368973&ss=1&u=1115812093

feedback studio Ravihari Gunawardhana Final Report

ASD.AI – SINHALA DIALOGUE MANAGEMENT TOOL
TO SCREEN KIDS WITH AUTISM SPECTRUM
DISORDER

Gunawardhana M.D. R.T.
(IT16090804)

Match Overview

14%

| | | |
|---|---|-----|
| 1 | Submitted to Postgrad... Student Paper | 5% |
| 2 | Submitted to Sri Lanka ... Student Paper | 2% |
| 3 | blog.rasa.com Internet Source | 1% |
| 4 | Submitted to University... Student Paper | 1% |
| 5 | deepai.org Internet Source | 1% |
| 6 | Submitted to University... Student Paper | 1% |
| 7 | Submitted to Coventry ... Student Paper | <1% |
| 8 | Submitted to Athlone L... Student Paper | <1% |
| 9 | www.ncbi.nlm.nih.gov Internet Source | <1% |