Romanized Sinhala to Sinhala Reverse Transliteration using a Hybrid Approach

A Thesis Submitted for the Degree of Master of Computer Science

T.G.D.K. Sumanathilaka

University of Colombo School of Computing

2022

DECLARATION

I hereby declare that the thesis is my original work, and it has been written by me in its entirety. I have duly acknowledged all the sources of information which have been used in the thesis. This thesis has also not been submitted for any degree in any university previously.

Student Name: TGDK Sumanathilaka

Registration Number: 2019/MCS/084

Index Number:19440847

[illegible] 20/11/2022

Signature of the Student & Date

This is to certify that this thesis is based on the work of Mr T.G.D.K. Sumanathilaka under my supervision. The thesis has been prepared according to the format stipulated and is of an acceptable standard.

Certified by,

Supervisor Name: Dr Ruvan Weerasinghe

[illegible] 23/02/2023

Signature of the Supervisor & Date

ගිනි කාෂ්ටක අව්වේ දහදිය හෙලමින් උපයා ගත් සොච්චමෙන් මා මිනිසෙකු කල මාගේ පියානන් ටත්,

ලේ කිරි කර පොවා ආදරය, කරුණාව කියා දුන් මගේ මැණියන් ටත්, මා උගතෙකු කරන්නට මාගේ

අධ්‍යාපනයට බදු ගෙවූ ශ්‍රී ලාංකිකයන්ටත් ඉමහත් ආදරයෙන්…

I would like to dedicate this thesis to the people in Sri Lanka!

ACKNOWLEDGEMENTS

It was a challenging adventure to conclude this research project in a year, but it was also a tremendous learning experience. Without the generous assistance, direction, and inspiration of numerous people who stood by my side, I would not have been able to finish my research. I want to say thank you to each one of you.

I would first like to thank my supervisor, Dr Ruvan Weerasinghe, for his guidance, and support and for pushing me to do my best Masters Thesis, from the idea initiation to the final submission. Also, my sincere gratitude goes to Mr Prasan Yapa, the External supervisor of the project, for his support and guidance throughout the project without any hesitation. Furthermore, I would like to thank all the lecturers at the University of Colombo, School of Computing for their endless knowledge and guidance throughout the master's degree period.

Next, I would like to express my sincere thanks to all evaluators, domain, and technical experts, of this research for providing their expertise and knowledge throughout the research. Further, I would like to extend my gratitude to my friends and company - Informatics Institute of Technology (IIT) colleagues for their valuable feedback and support during the project.

Finally, I would like to express my loving gratitude to my family, who has been my pillars of success throughout my life.

ABSTRACT

With the revolution of social technology, the introduction of social media platforms and instant messages strengthen the native language used for communication in electronic media. With the commencement of multi-language compatibility in the digital arena both native Sinhala and Romanized Sinhala became prominent among the general community. Machine transliteration provides the ability to transliterate the alphabet of one language to another using computational approaches. The informal shorthand language that uses in texting also known as “Singlish” makes texting easier as the words in Sinhala can be interpreted using English letters with different typing patterns. But typing “Romanized Sinhala” using ad hoc transliterations and short net acronyms and getting the expected output in native Sinhala is less accurate.

The current transliterators with a rule-based approach use a letter-level transliteration with a defined rule for the transliteration schema. But Romanized Sinhala via shortened hand-based typing is not compatible with the current system. The proposed ad-hoc schema uses multiple computational approaches Aka Hybrid Approach to accomplish the requirement of ambiguity-free transliteration. The statistical approach used in the first phase uses an N-gram tagger where the tokens are fed to Trigram, Bigram, and Unigram taggers respectively. The unknown token from the initial phase is fed to the second phase with a Rule-based Algorithm which will predict respective words. The third phase which is the finalizing phase uses a suggestion-level model implemented using a Trie and Knowledge base to find the most optimal word suggestions from the predicted words pool. This phase will solve the ambiguity of a word selection.

The Transliterator has been tested with the testing data and word level accuracy achieved was 84%. Therefore, the proposed novel transliterator which can back transliterate Romanized Sinhala to Sinhala using the Hybrid approach can use to enhance the reverse transliteration schema which will escalate the usage of Native Sinhala for communication.

Keywords: *Romanized Sinhala, Transliteration, Neural Approach, Statistical approach, Sinhala, Typing*

LIST OF PUBLICATIONS

- **Full Paper: Swa-Bhasha Romanized Sinhala to Sinhala Reverse Transliteration using a Hybrid Approach**
    - Conference: 03rd International Conference on Advanced Research in Computing(ICARC 2023)
    - **Venue:** University of Sabaragamuwa, Belihul oya, Sri Lanka
    - **Date:** 23rd February 2023

- **Extended Abstract Paper: Romanized Sinhala to Sinhala Transliteration using a Hybrid Approach**
    - **Conference:** International Conference on Innovations in Info-business & Technology (ICIIT2022)
    - **Venue:** Colombo, Sri Lanka
    - **Date:** 18th February 2022

- **Abstract Paper: Ad-hoc Transliteration Schema for Romanized Sinhala to Sinhala**
    - **Conference:** 5th Student-STAFF Research Conference (SSRC22)
    - **Venue:** London, UK (Virtual)
    - **Date:** 2nd of March 2022

- **Abstract Paper: A Rule-based Approach to Generate low-resourced Sinhala to Romanized Sinhala Ad-hoc transliterals based on Survey Data**
    - **Conference:** International conference of Innovation and Emerging Technologies 2022 (ICIET 2022)
    - **Venue:** University of Sri Jayewardenepura, Homagama, Sri Lanka
    - **Date:** 25th and 26th of November 2022

- **Abstract Paper: Sinhala word suggestion algorithm for ad hoc Romanized Sinhala transliterations using a Trie**

    - **Conference:** 10th Ruhuna International Science and Technology Conference
    - **Venue:** University of Ruhuna, Matara, Sri Lanka
    - **Date:** 18th of January 2023

- **Awards Prizes: Best Extended Abstract Paper Award**
    - **Conference:** International Conference on Innovations in Info-business & Technology (ICIIT2022)
    - **Title:** Romanized Sinhala to Sinhala Transliteration using a Hybrid Approach
    - **Venue:** Colombo, Sri Lanka
    - **Date:** 18th February 2022

- **Awards Prizes: Best Abstract Paper Award (Postgraduate)**
    - **Conference:** International conference of Innovation and Emerging Technologies 2022 (ICIET 2022)
    - **Title :** Sinhala word suggestion algorithm for ad hoc Romanized Sinhala transliterations using a Trie
    - **Venue:** University of Sri Jayewardenepura, Homagama, Sri Lanka
    - **Date:** 25th and 26th of November 2022

TABLE OF CONTENTS

Contents

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Abbreviation	**Description**
FR	Functional Requirement
LSTM	Long Short-Term Memory
NFR	Non Functional Requirement
NLP	Natural Language Processing
NLTK	Natural Language Tool Kit
MT	Machine Translation
POS	Part of Speech
SVM	Support Vector Machine

CHAPTER 1

INTRODUCTION

1.1 Chapter Overview

This chapter establishes an overview of the research project undertaken. The background of the research, scope, requirements, and objectives are detailed in this chapter. In nutshell, this chapter defines the problem, limitations of the currently existing systems, and other contributing factors that inspired and influenced the pursuit of the research.

1.2 Problem Domain / Background

1.2.1 Language Transliteration

According to the Oxford Dictionary, the word transliteration can be defined as "*the act of* ***writing words or letters using letters of a different alphabet or language.*** The word transliteration comes from the Latin word "*transliteratus"* which explains "*trans"* as across and "*Littera" as* a letter. Transliteration helps people to pronounce words and names in foreign languages. As an example, the Sinhala word *"අම්මා"* is represented using English as "Amma". Unlike translation, which tells the user about the meaning of a particular word, transliteration only provides an idea of how the word is pronounced by using a familiar language.

- Reverse Transliteration

The reverse transliteration explains, "A word or a sentence of a particular language expressed using different languages will be converted to the original language Schema. As an example, the Sinhala word "*කොහොමද"* which is expressed as Roman English "kohomdha" can be re-transliterated to the original word "*කොහොමද"* using the Reverse transliteration techniques.

This study is mainly focused on reverse transliteration where Romanized Sinhala will be translated into Sinhala characters.

- Transliteration vs. Translation

The transliteration process means that a word expressed in a character set like the Sinhala alphabet is transposed in another language model, say the English alphabets. In other words, there is no translation involved. If the source word means nothing in the given language, its transliterated form will also mean nothing, even though it will look like a word in that language as it will be written in its alphabet or syllabic system.

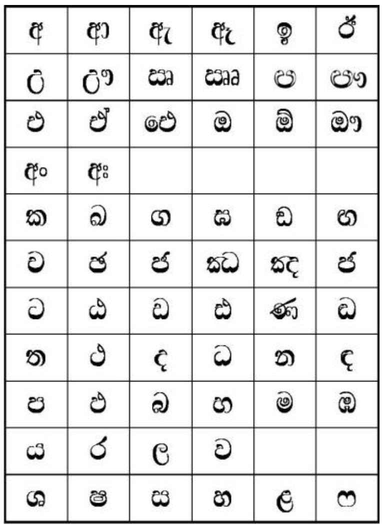
1.2.2 Romanized Sinhala

Writing letters and words belonging to Sinhala Language using the English alphabet is known as Romanized Sinhala. As an example, the Sinhala word “අම්මා” can be written as "Amma" using Romanized Sinhala Characters. This symbolic pattern was introduced to write pali in 2009 and later it was adapted to the Sinhala Language.

1.2.3 Sinhala Language

Sri Lanka is a South Asian country with 21,436,802 population (Division, 2020). Approximately 16 million people of the total population speak ‘Sinhala’ as one of their national languages. Many changes in the Sinhala Language can be identified in different communities where minor changes in pronunciation can be seen. This language is only spoken by Sri Lankans in the world. The Sinhala language is related to the Indo-Aryan language family which is a branch of the Indo-European languages (Gallege, 2010).

In the Sinhala alphabet, there are 60 basic characters with 18 vowel letters and 42 constant letters. 18 vowel letters have included 2 fricatives, 8 stops, 2 affricatives, 2 nasals, 2 glides and 2 liquids an (Gallege, 2010). This language has many grammatical rules for correct interpretation of the language were identifying the morph of a word is complex compared to the other languages. Many systems have been implemented to translate other languages into Sinhala and vice versa. With the huge growth of social media and Sinhala language-based blogging, the usage of Sinhala for the day today usage has become prominent. NLP (Natural Language Processing) technology was used for the translation process. Most often, MT (Machine Translation), a subset of NLP, is used for language translation with different algorithms and different approaches. In Machine learning multiple computational linguistics which can define as Rule-based (RB), Statistical, Example-based (EB), Human- assisted, Knowledge-based, Hybrid, Dictionary, and Agent-based (R. M. M. Shalini1, 2017) are used to build accurate models related to Sinhala Computational Domain.



අ	ආ	ඇ	ඈ	ඉ	ඊ
උ	ඌ	ඍ	ඎ	ඏ	ඐ
එ	ඒ	ඓ	ඔ	ඕ	ඖ
අං	අඃ				
ක	ඛ	ග	ඝ	ඞ	ඟ
ච	ඡ	ජ	ඣ	ඤ	ඦ
ට	ඨ	ඩ	ඪ	ණ	ඬ
ත	ථ	ද	ධ	න	ඳ
ප	ඵ	බ	භ	ම	ඹ
ය	ර	ල	ව		
ශ	ෂ	ස	හ	ළ	ෆ

Figure 1.1 Sinhala Alphabet

1.2.3 Natural Language Processing

Natural Language Processing, simply called NLP, can be categorized under Artificial Intelligence. The main task and goal of NLP are to make machines understand language the way humans understand language. (Garbade, 2018). Using this understanding of the language patterns, symbolic representation and Grammatical structure, NLP-based models feed the computer knowledge of the language model which could understand the Syntax and semantics related to a word or a sentence. This helps to bridge the language barrier between machines and human beings.

Below are some of the most famous applications which use NLP.

1. Email filters.
2. Smart assistants.
3. Text Classification.
4. Text Translation and Transliteration.
5. Topic Modelling.
6. Language Identification.
7. Grammar checkers.
8. Search results, etc.

These techniques are used with text based computation in Natural Language Processing. In this research, the author will focus on the Transliteration of Romanized Sinhala to Sinhala.

1.3 Problem Definition

With the revolution of technology, the introduction of social media platforms with instant message and commenting applications strengthens the usage of native languages for communication on electronic media. With the commencement of muti-language compatibility in digital platforms, Sri Lankans tend to use both Native Sinhala and Romanized Sinhala for their day today typing purposes. But Typing "**Romanized Sinhala**" using ad hoc transliteration or short net acronyms and getting the expected output in native Sinhala is a less accurate and time-consuming process with existing transliteration keyboards. As an example, with the available Transliteration keyboards, the word "*කොහොමද"* can be generated only using the Romanized Sinhala word "*Kohomadha*", where it follows a defined rule for the transliteration. But the proposed idea aims to transliterate short typing patterns like "*khmda ,kohomda, kohomada, khmada ,khmd*" to the native Sinhala word "*කොහොමද".* Therefore, introducing a novel reverse-transliterator that back transliterates 'Romanized Sinhala" intelligently would make typing and messaging much more convenient. This study is mainly focused on enhancing the reverse transliteration schema which will escalate the usage of the Native Sinhala for communication.

1.4 Problem Statement

Typing Romanized Sinhala using ad hoc transliterations with vowels, without vowels or including both ways and getting the expected output in native Sinhala is a less accurate and complex process with the existing Sinhala-English keyboard. But introducing a real-time reverse transliterator that transliterates Romanized Sinhala to Sinhala based on message-based typing can effectively enhance the typing experience of Sinhala users.

1.5 Research Motivation

With the usage of English to Sinhala keyboard, typing a Sinhala word using Romanized Sinhala is complicated. The pre-defined typing structure must be followed in the process where word transliteration is following a unique rule in the transliteration process. As an example, if you want to type කොහොමද on available keyboards, you are supposed to follow the defined pattern of "kohomadha". But the same word can be present in the manner of "khmda, kohomda, khmd, kohmda" where available systems don't transliterate the expected output. According to the Helakuru documentation, it has clearly stated that each Sinhala letter is transliterated using a defined pattern of Romanized Sinhala. But it has been observed many users who use Sinhala to communicate tents to use Romanized Sinhala in different patterns for day-to-day communication. Manly users try to omit the vowels in their typing.

So, this study tries to fill the gap of the complexity of using the pattern in typing Sinhala by adapting the typing schema to with or without vowels where short typing will transliterate the Romanized Sinhala word to the expected Sinhala word accurately.

1.6 Existing Work

The below table discusses the existing work related to translation and transliteration.

	Research Name and Year	**Brief of the work**	**Limitations**	**Technologies**	**Improvements**
1	(Silva and Ahangama, 2021)	Singlish to Sinhala Transliteration model which uses ruled based approach with an error correction module.	Transliteration using N gram model is less accurate and the error-correcting module depends on the Sinhala corpus.	Rule-based Method with ensemble learning for Error correction Module.	Ability to correct the errors in transliterated Sinhala words.
2	(Liwera and Ranathunga, 2020)	This is Singlish to Sinhala transliteration focused on social media comments.	Less accuracy. Focused on social media.	Rule-based method and Trigram method.	Ability to transliterate Singlish to Sinhala.
3	(Joshi et al., 2018)	Using LSTM and sequence 2 sequence model to find the optimal model with pre and post-processing	Only with 3 languages Hindi, Sindhi, and Kannada	LSTM, Sequence 2 Sequence model	Ability converts Hindi, Sindhi, and Kannada words.

4	(Wijerathna et al., 2012)	A Rule-based machine translation system that can translate sentences from Sinhala to English and vice versa.	A limited no of rules is used.	Rule-Based approach	Ability to translate Sinhala to English and English Sinhala but with less accuracy
5	(Antony PJ, 2010)	Transliterating English to Kannada language using SVM.	Less accuracy, Support only words.	SVM, Sequence labelling	Ability to translate English to Kannada.
6	(Vidanaralage et al., 2018)	A Singlish Transliterator based on a Phonetic chart and Sinhala Transliteration chart.	Character level transliteration is based on the suggestion level approach.	Phonetic rule-based and Transliteration rule-based.	A character transliteration that can transliterate Singlish to Sinhala.

1.7 Research Gap

According to research conducted By Liwera (Liwera and Ranathunga, 2020) Singlish to Sinhala Transliteration using trigram and Rule-based approach was tested and evaluated. But the proposed system has 62%word-level accuracy and 77% letter wise accuracy which is mainly trained on social media-based content. The wide variety of Romanized Sinhala patterns has not been used in the system where the ambiguity of the Singlish words is not handled. The research conducted by Vidanaralage (Vidanaralage et al., 2018) attempts to transliterate the Singlish words to Sinhala using a phonetic and Sinhala transliteration chart where the transliteration follows a rule-based approach. Short word typing without vowels and or short net acronyms cannot be transliterated accurately through the system. The research conducted by Wijerathna (Wijerathna et al., 2012) has introduced an approach for converting Sinhala to English where the input to the system is processed using Singlish typing. A rule-based approach for transliteration has been used with the system where entered Singlish will be transliterated to the respective Sinhala word. As a limitation of the system, it will follow a specific pattern in the transliteration process where the different patterns of Singlish typing will generate an error in the transliteration. So, considering the above limitation the researcher of this work attempts to build a robust model which can back transliterate Romanized Sinhala which is written in short letter format to Sinhala. The model will suggest the best possible matching in a case of ambiguity of a word where word-level predicting, and word level suggesting will be tested and evaluated in the work

1.8 Contribution to the body of knowledge

Research contributions will be further divided into domain and software engineering as elaborated below.

Contribution to the Domain

- The author of this work will be building a transliteration dataset using social media comments, and documentation data which will be available for further research.
- Adapting novel schema for Transliteration of Romanized Sinhala to Sinhala using N-gram and Rule base model.
- Enhancing the Rule-based schema in the transliteration process by introducing novel rules to the existing rules.

Contribution to Computer Science

- Sinhala is the language of a low resource where building a novel algorithmic approach for the transliteration process can enhance the usability of Sinhala among the public.

1.9 Research Challenge

The major challenges faced during the study are as follows.

- As Sinhala is a low resource language, finding the best dataset to generate transliteral is challenging.
- Collecting typing patterns from a diverse community.
- Identifying the typing patterns and building a generic approach that suits every person's typing pattern.
- Building a robust model which performs the task in real-time.

1.10 Research Questions

- RQ1: How to create a credible dataset to perform reverse transliteration.
- RQ2: How to identify the Sinhala words and their typing patterns which are frequently used by people when texting.
- RQ3: How to build the most optimal algorithm to reverse-transliterate Romanized Sinhala to Sinhala.
- RQ4: What sort of preprocessing steps and learning methods should be carried out for the main component.

1.11 Research Aim

This research aims to analyze, design, develop and evaluate a generalized system that reverse transliterates Romanized Sinhala to Sinhala using word level N-gram and rule-based module with handling ambiguity in the transliteration process by suggestion algorithm using a Knowledge base.

Further elaboration on the aim, this research project aims to back transliterate the Romanized Sinhala into Sinhala using NLP-based approaches, tools and translating algorithms and to develop an accurate and fast technique to provide native Sinhala words. Therefore

- The **Input** to the prototype will be Romanized Sinhala where Sinhala will be typed in English.
- The research **processes** are to find accurate algorithms and NLP-based techniques approach to develop this proposed system with a Romanized Sinhala- Sinhala word dataset.
- The **output** will be, native (Sinhala) Transliterals and the suggestion level Sinhala transliterals to the user.

1.12 Research Objectives

Objective	Description
The Literature Survey	This study on literature is conducted to have an idea about Natural language processing, machine learning, and technique related to it. Identification of existing systems and their gaps with various approaches and algorithms are studied through the survey. Numerous techniques,

	frameworks, libraries, tools, and methodologies to develop the project has to be identified through the critical analysis in the literature review.
Project Requirement Analysis	This chapter will be conducted to identify the exact requirements for the project and to understand the end user's expectations of the system. The functional requirements and non-functional requirements are identified in the chapter. The experts' ideas and guidance help to identify the risk involved in the system and the survey conducted among the end party contribute to identifying the typing patterns which will be used in the system development.
Project Design	According to the identified requirements building a proper model with a suitable architecture will be considered. Identified user typing patterns modelling technique, transliteral generation design, and basic UI structures are designed. The data flow of the system must be taken into consideration in the chapter.
Project Development	A prototype should be developed with relevant hardware and software resources to meet the requirements and design aspects identified in the requirements gathering formulation.
Project Testing	A prototype is tested by creating a suitable test plan and various unit and functional tests. Requirements verification is performed, including system performance and accuracy, and functional and non-functional requirements.
Project Evaluation	Both qualitative and quantitative evaluation methods will be used in every system.

1.13 Project Scope

1.13.1 In scope

- Identifying the ad hoc transliterations or short net acronyms used in Romanized Sinhala Typing.
- Researching and developing the reverse transliteration models which can perform the needful accurately.

- Choosing the best fitting approach to perform the reverse transliteration.
- Developing a GUI-based implementation to interact with the system.

1.13.2 Out scope

- Code-mixed words will not be handled since all the words will be expected to be Sinhala.
- Shorthand Typing will not be considered for the study as the input to the system has to be Romanized Sinhala.

1.14 Prototype Feature Diagram

The Proposed Solution is featured in the below prototype feature diagram.

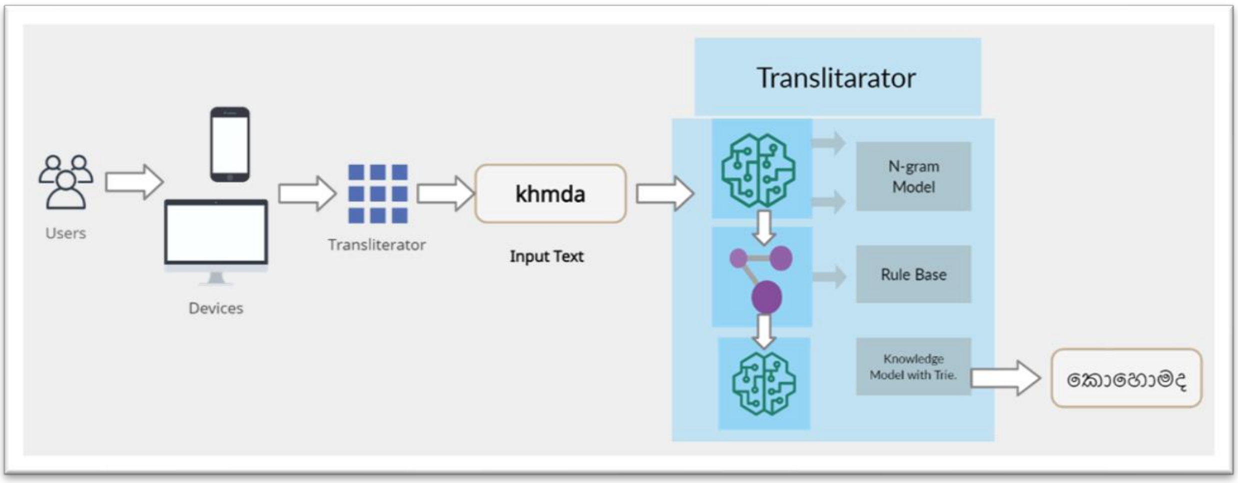


Figure 1.2 Prototype Feature Diagram of Transliterator (Self-Composed)

1.15 Chapter Summary and Structure of the dissertation

This chapter discusses the background. The introduction has been discussed. The existing work was discussed and based on the findings latest research work has been focused on Classical transliteration techniques. Therefore, the research gap was formulated by contributing a novel transliteration schema for the Romanized Sinhala to Sinhala. The objectives and aim of the research has been discussed and main research objectives were identified. The outline of the structure of the dissertation is as follows.

Chapter 01	Introduction	This chapter establishes an overview of the research project. The background of the research, scope, requirements, and objectives are detailed in this chapter. In nutshell, this chapter defines the problem, limitations of the currently existing systems, and other contributing factors.
Chapter 02	Literature Review	The literature review chapter will provide the related information on this study. Domain and Technical review with existing systems, algorithmic analysis, and evaluation methodologies are discussed. A literature review of the domain and the technology has been discussed as the major factors.
Chapter 03	Methodology	The methodology chapter provides a run-through of the research, project, and development methodologies selected to continue this research. Furthermore, identifying stakeholders, the outline of gathering requirements, and analyzing gathered information to identify and priorities the requirements are to be discussed. Initially, a rich picture diagram was drawn specifying the internal and external environments of the project that gives an overview of the stakeholders of the project.
Chapter 04	Implementation	The implementation chapter provides an overview of the technology stack and algorithms used in the implementation.
Chapter 05	Evaluation and Results	The evaluation and results chapter provides an overview of Quantitative evaluations and Qualitative evaluations based on feedback collected from domain experts, technical experts, academic experts, and researchers for different aspects of the research and prototype. Functional and non-functional requirements will be evaluated to ensure that

		system meets the expected requirement. The Numerical evaluation is presented in the chapter.
Chapter 06	Conclusion and Future Works	The conclusion chapter discusses the project's final remarks. It analyzes whether the project aims, and objectives were met successfully, the challenges encountered and the limitation with future works. The research contributions to the domain and computer science are presented.

CHAPTER 2

LITERATURE REVIEW

2.1 Chapter Overview

The literature review chapter will summarize the related information on this study. The chapter contains a domain with the existing systems, algorithmic analysis, and evaluation methodologies currently available in the literature. The chapter will mainly divide into a literature review of the domain and the technology to provide a much better analysis of the domain and the technologies. And the chapter will address the existing systems and approaches that have been used to provide a better architecture in the transliteration process in both Sinhala and other languages. This information will justify selected approaches to develop the proposed system while providing the required knowledge for the project.

2.2 Problem Domain

2.2.1 Introduction to Natural Language Processing

Natural Language Processing (NLP) is a known technology used by researchers to implement linguistic-based systems. Along with artificial Intelligent (AI), NLP can provide speech and text translation ability to computers. Humankind behaviour can be integrated with the help of context.

According to figure 2.1, the first component of the study will be morphological Processing. The inputted text will separate into paragraphs, sentences and word-level tokens. This process is known as tokenizing. The syntactic relationship among the sentences is analyzed using the syntax analyzer chapter. The semantic analysis which the third chapter uses to analyze the meaning of the sentences. The real meaning can be extracted using the pragmatic analyzer.

Ambiguity is the main concern in the domain of NLP. For example, if a sentence says, "John saw Jane at the bank with a dog". But it is difficult to explain who is with the dog. It may John, or it may be Jane. And also, the bank can be a financial bank or riverbank. So identifying the correct meaning of the word is essential. The pragmatic analysis chapter is an essential section of the NLP to solve this ambiguity in the translation processes (Diksha Khurana et al., 2017).

The multiple techniques used with NLP like Pos-Part of Speech tagging, Speech detection, Natural Language Generation, Co-reference resolution, Sentiment Analysis, Word entity recognition and the sense of words disambiguation (Education, 2020) can fulfil the task of building a proper mankind system. This domain provides the user to build a sustainable system in machine translation and Transliteration.

2.2.2 Machine Translation

Machine Translation (MT) also known as Automated translation is a well-structured systematic process which can use to translate a source language to a targeted language. The oxford dictionary says Machine translation is a process that is carried out the process of translation using a computational technique. According to Figure 2.2, the input to the system is sources text which will be divided according to its type of difficulty based on the preprocessing. The difficulty level relies on assumptions, expectations, and relations. The Text needs to

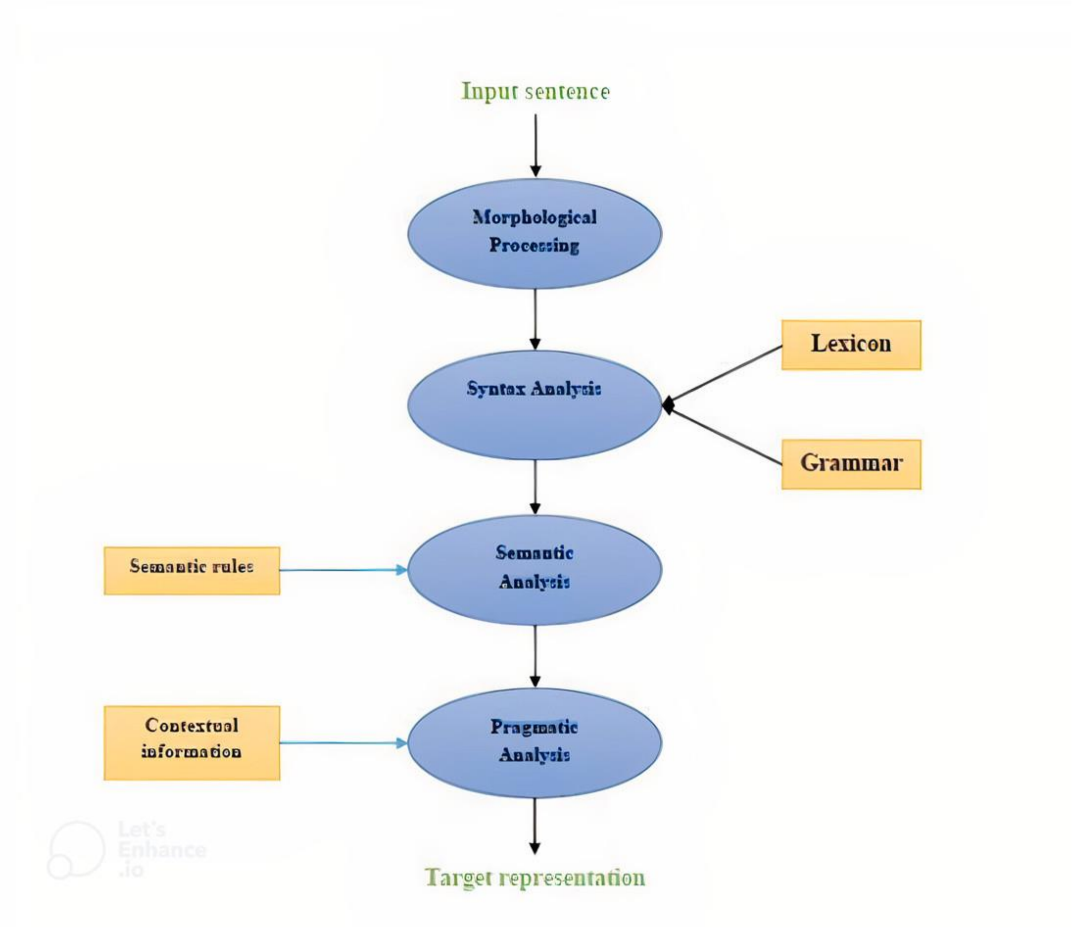


Figure 2.1 Translation overview

reformat once the translation is completed. Removing unnecessary tokens through the preprocessing and post-editing performed over the text will enhance the translation quality of the text. Two phases will control the efficiency level of an MT system. Analysis and generation of processors of the Morphological Syntactic and Semantics will do a huge contribution to the translation process. The morphological properties such as tense, grammatical form, gender, and mood will be considered about the structure of the word. And the analyzation phase and fine-tuning in the initial phase. In the Syntactic analysis phase, the Part of Speech(pos) tagging will be considered. The single token generated from the tokenizer in the syntactic phase will use lexical features in a language such that the lexical elements in each language will be analyzed concerning the language. Grammar Formalism frameworks in Machine translation will describe the structure of a language. Such frameworks are developed for different languages based on the requirement of the translation process. (Robin, 2010). Machine Translation has various types such as Statistical, Rule-based, dictionary-based, knowledge-based, example-based, and hybrid-based approaches that combinedly can be used for efficient translation and transliteration. (Okpor, 2014).

2.2.3 Computational Linguistics

Computational linguistics (CL) is the study of languages based on a computer perspective which can be derived using a scientific manner. The understanding of the writing and spoken languages are addressed in the CL domain. This is an interdisciplinary field connecting with computational modelling and NLP. CL provides language models such as knowledge-based, and data-driven. Theoretical computational linguistics includes grammar and semantics often grounded in symbolic and formal logic. Applied computational linguistics mainly branched with machine learning was discussed in the mid-2010s but with the evolvement of deep learning in 2015, it became a major framework for NLP. In terms of the NLP domain, Computer linguistics gives a vast experience in many systems. Web search engines, Text-to-speech synthesizers, and text editors are some examples of those systems. (ACL, 2020)

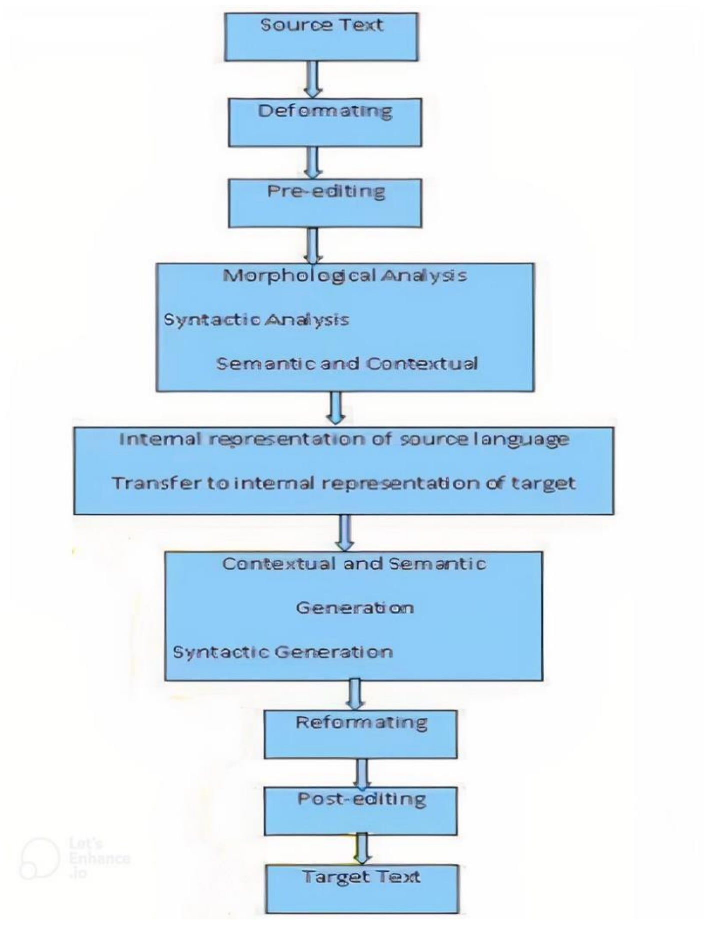
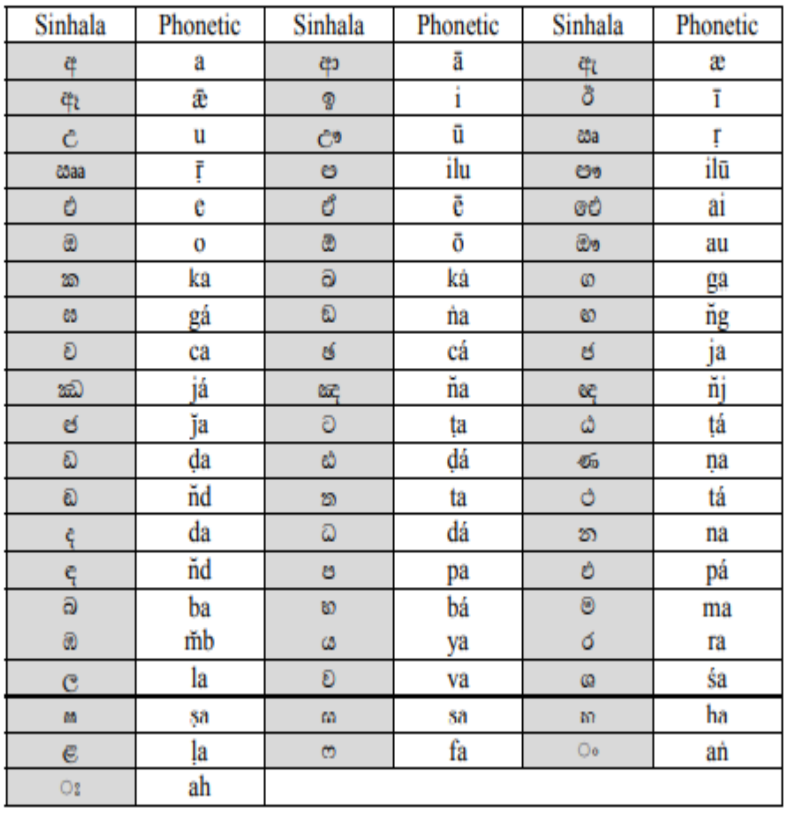


Figure 2.2 Machine Translation

2.2.4 Sinhala Language

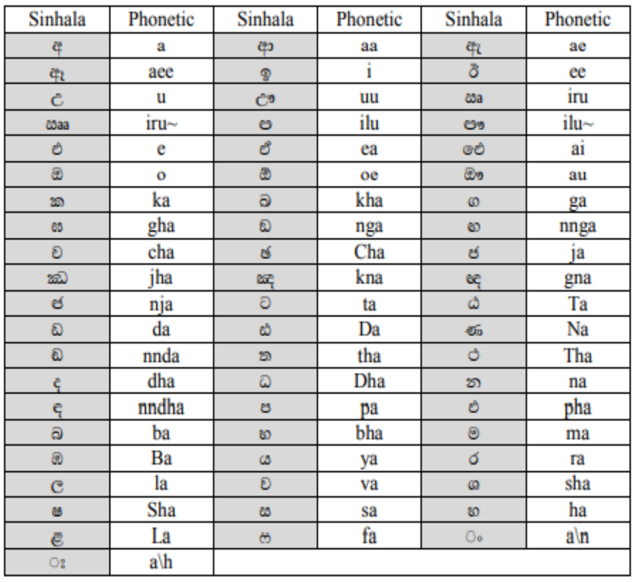
Sinhala, the major native language of the Sinhalese people of Sri Lanka who is the largest ethnic group of the county is considered a language which belongs to the globe-spanning language tree, Indo-European. (de Silva, 2021) However, due to poverty in linguistic resources, Sinhala remains a resource-poor language compared to many other existing languages. A number of research communities have engaged in numerous researched and experiments to enhance the research value of the language. However, due to various reasons, these attempts seem to lack coordination and awareness of each other.

The Sinhala language is considered to be the mother tongue of approximately 16 million people (de Silva, 2021) where the writing pattern for Sinhala originated form Indian Brahmi script. When considering the usage of the Sinhala language, the basic grammar related to the language can be interpreted using 2 formats, where one for written(literacy) and other for spoken. The written format follows Subject Object Verb (SOV) configuration where subject verb agreement is introduced to the structure of the language which produces a grammatically correct sentence based on gender, number, and person. In each category the verb of the particular sentence is decided based on gender: male or female, number: Singular or plural and person: 1st, 2nd or 3rd. But in the speaking context of the Sinhala language SOV order can be neglected. When considering the Sinhala Script, it is considered the most expansive language in the world (Vidanaralage et al., 2018) which contain 60 letters including 18 vowels, 42 consonants. But with the letter "ඦ" total no of character to be considered on 61. In Sinhala speaking style it represents 40 phoneme sounds with 14 vowels and 26 consonants sounds. However, based on the study many researchers have used phonetic chart for Sinhala and Sinhala translation chart which are represented in figure 2.3 and figure 2.4.



Sinhala	Phonetic	Sinhala	Phonetic	Sinhala	Phonetic
අ	a	ආ	ā	ඇ	æ
ඈ	ǣ	ඉ	i	ඊ	ī
උ	u	ඌ	ū	ඍ	ṛ
ඎ	ṝ	ඏ	ilu	ඐ	ilū
එ	e	ඒ	ē	ඓ	ai
ඔ	o	ඕ	ō	ඖ	au
ක	ka	ඛ	ká	ග	ga
ඝ	gá	ඞ	ṅa	ඟ	ňg
ච	ca	ඡ	cá	ජ	ja
ඣ	já	ඤ	ňa	ඦ	ňj
ජ	ja̐	ට	ṭa	ඨ	ṭá
ඩ	ḍa	ඪ	ḍá	ණ	ṇa
ඬ	ňd	ත	ta	ථ	tá
ද	da	ධ	dá	න	na
ඳ	ňd	ප	pa	ඵ	pá
බ	ba	භ	bá	ම	ma
ඹ	m̆b	ය	ya	ර	ra
ල	la	ව	va	ශ	śa
ෂ	ṣa	ස	sa	හ	ha
ළ	ḷa	ෆ	fa	ං	aṅ
ඃ	ah				

Figure 2.3 Phonetic Chart for Sinhala Alpahabet



Sinhala	Phonetic	Sinhala	Phonetic	Sinhala	Phonetic
අ	a	ආ	aa	ඇ	ae
ඈ	aee	ඉ	i	ඊ	ee
උ	u	ඌ	uu	ඍ	iru
ඎ	iru~	ඏ	ilu	ඐ	ilu~
එ	e	ඒ	ea	ඓ	ai
ඔ	o	ඕ	oe	ඖ	au
ක	ka	ඛ	kha	ග	ga
ඝ	gha	ඞ	nga	ඟ	nnga
ච	cha	ඡ	Cha	ජ	ja
ඣ	jha	ඤ	kna	ඥ	gna
ඦ	nja	ට	ta	ඨ	Ta
ඩ	da	ඪ	Da	ණ	Na
ඬ	nnda	ත	tha	ථ	Tha
ද	dha	ධ	Dha	න	na
ඳ	nndha	ප	pa	ඵ	pha
බ	ba	භ	bha	ම	ma
ඹ	Ba	ය	ya	ර	ra
ල	la	ව	va	ශ	sha
ෂ	Sha	ස	sa	හ	ha
ළ	La	ෆ	fa	ං	a\n
ඃ	a\h				

Figure 2.4 Sinhala transliteration Chart

2.2.5 Romanized Sinhala

Typing native Sinhala using English script is known as Romanized Sinhala. According to (Hettiarachchi et al., 2020) the word "*අම්මා*" can be represented as "*Amma*" with the usage of Romanized Sinhala. Instead of using letters related to the same language, a convenient language which is mainly English can be used to represent the word, where phonetics is the same and reading is the same. (Liwera and Ranathunga, 2020). With the development in social media platforms, the Sri Lankan community tends to use Romanized Sinhala for social media chatting and commenting.

Table 2.1 Most Common typing patterns in Romanized Sinhala

Sinhala Word	Pronunciation	Romanized Sinhala
මම ගෙදර යනවා	mama gedara yanavā	Mama gedara yanwa
මේ මහා කරදර කාලයක්	mē mahā karadara kālayak	Meh mahaa karadhara kaalayak
සමාජ මාධ්‍ය මිනිස්සු විනාශ කරනවා	samāja mādhya minissu vināśa karanavā	Samaaja maadhya minissu winaasha karanwa

2.3 Existing Work

With the technological and social revolution, many translation algorithms have been discovered globally. Sri Lanka as the only country that speaks Sinhala has some systems with various computational approaches on Sinhala Language even the language is considered as a low resourced. Existing machine translation-related systems and their technological review will be discussed in the below sections mainly focusing on following two Machine translations (MT) Approaches.

1. Machine Translation with Sinhala
2. Machine Translation with different languages.

2.3.1 Technology and Algorithms Review

Rule-Based Approach

The rule based (RBMT) approach was the first to be implemented, and it focuses on well-defined rules related to the source and target languages. The main approaches in the Rule-based are Direct MT, Transfer-based MT, and Interlingual MT. This approach is known to be complex in terms of effort, time, and complexity because it requires the production of linguistic rules for the translation from the source sentence to the target sentence.

Rule based approach (RB) can be performed in three phases. The Analysis phases perform while the source sentence inserted where next phase of the Transfer phase will be used in the source sentence translation into the target sentence and the final phase which is the generation phase will generate the output with the relevant grammatical rules. After inserting the source text into the system, the process moves on to the morphological analyzer, Part-of-Speech tagger (POT), lexical selection, transfer structure/lexical, morphological generator, and finally the post generator for the target language. This method focuses on the structure of both the source and target languages where the structure of the source language must be linked to the structure of the target language. As a result, RBMT employs a set of grammatical rules that can be applied to any language. Figure 2.5 depicts the basic architecture of the RBMT approach.

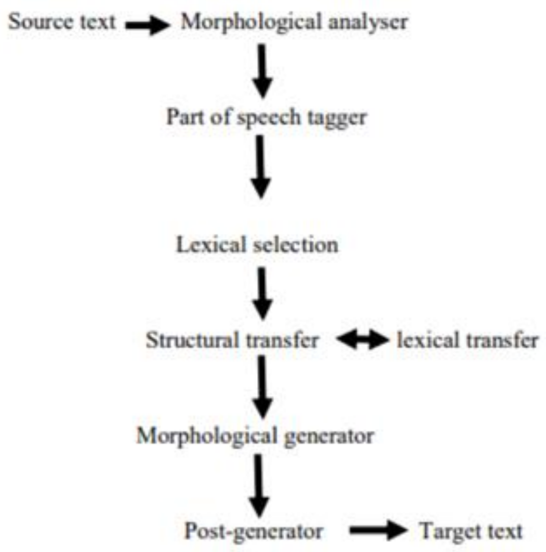


Figure 2.5 Rule Based Architecture

According to the available literature there are many developed systems which use rule-based approach. In many of cases, rule-based system has been used for the translation process of the formal work, which means for the systems which can be used in the official translations. The latest research work by Silva (Silva and Ahangama, 2021) has used rule based approach to Transliterate Singlish to Sinhala which uses an error-correcting module. The defined rule has been used to transliterate in the syllabic level which generated words are fed to the error correcting module which implements using Sinhala corpus The RBMT approach was used by A.J. Vidanaralage and the team to implement the English-Sinhala Decoder (A.J. Vidanaralage, 2018). They used a wordlist dictionary of both Sinhala and English words, as well as a set of rules, to generate meaningful target language sentences. The entered sentence (source sentence) is divided into Tokens (single words) in this system, and the morphological information of each word is analyzed using a dictionary. The Hindi-English Translator (Omkar Dhariya, 2017), a system developed by Omkar Dhariya and his team, made use of the RBMT. In the above translator, the RBMT also acts during the rearrangement phase, as well as selecting the correct nouns and tagging (PoS) using the parallel Hindi and English dictionaries.

The source language sentence (Hindi) will rearrange by using appropriate grammatical rules and produce the correct target sentence (English). The latest research by Liwera which converts Singlish to Sinhala using N gram approach use Rule based approach for the last phase of the transliteration process. The rule used for transliterating Singlish to Sinhala based on a character level is not accurate according to the results of the study.(Liwera and Ranathunga, 2020). The latest research by (Silva and Ahangama, 2021) with the error correction module uses the rule base approach for the transliteration task where the research is handling special characters like mahaprana, thaluja Characters in the Sinhala Alphabet. The main drawback of the approach is translation or transliteration process has fully relied on the defined set of rules. The latest commercial product Helakuru uses the same Rule based mechanism for their Keyboard where each character follows a defined typing pattern to transliterate in the desired character.

Statistical-Based Approach

One of the well-known approaches to MT is the statistical-based approach, which works primarily by understanding the grammatical aspects of the source and target languages. Most of the time, these statistical-based MT systems access the work with a very little number of interactions with dictionary entries and language-related resources. To facilitate training, these systems typically break sentences down into n-grams. This will eventually improve the system's accuracy and performance. From a development standpoint, training can be accomplished by increasing the number of entries in many translation scenarios. The Trigram, Bi gram and Uni gram models used in this architecture can escalate the productivity and the credibility of the results generated. The author of this work uses a ngram model to build the initial phase of the transliteration process. According to the research done on the literature using Statistical approaches, it can justify that this approach is most suitable in terms of accuracy without consuming much time and financial aspects. Google translator is one of the best examples. (Rajpirathap S, 2015).

- Character N gram

Though character n gram features were considered in text classification problems they were used only in a few number of works in the Translation domain (Liwera and Ranathunga, 2020). Character n-grams can detect character patterns in texts where misspelt words can be captured in the approach. Since code-mixed texts contain different spelling variations n gram model will be efficient for a credible answer. Character n gram features were used for feature extraction in many of the studies according to the literature.

- Word N gram

Word n-grams performed well in text classification and Transliteration since natural languages are not just words but words with structures. N gram model which is trained using word level n gram cannot avoid ambiguity in the translation and transliteration process as these models are not much focus on the meaning of the sentence but the syntax. The researcher liwera has used

Word level trigram, Bigram and Unigram for his study where accurate translation has been achieved in the study.

Example-Based Approach

Makoto Nagao founded the example-based approach as an MT-based approach in 1981. Example-based machine Translation consists of three major components: matching and testing of the separated information from the source language, identification of the separated information for translation, and combination of the separated translated information into the target language. The benefits of using this EBMT approach include the ability to use the methodology in any pair of source and target languages, which can improve the results depending on the example set available for the language model.

Knowledge-Based Approach

One of the domain-specific Machine Translation strategies is knowledge-based approach. This method has been applied to integrate machine translation associated systems during translation. When it comes to creation, this method mostly relies on ontologies and the semantic web. Nair and his team adopted Knowledge based approach in their hybrid approach (Nair et al., 2016), but this is not well explored in the literature.

Neural-Based Approach

The neural-based approach is a newly added approach to the MT-based approach family. The main difference and advancement in this Neural based approach among all the existing approaches are translation is done using a neural network. The long words sequences and short word sequences are well handled in many Neural models. The latest research has been performed using this approach with the use of encoder and decoder architecture in the translation process (Sandaruwan et al., 2021). Encoding will be directed at the source sentence and will result in a fixed-length vector. The decoder will then generate the translation to the target sentence. LSTM, Transformers like Bert and Roberta are well-known neural models for machine translations.

According to the study of Sandaruwan and the team(Sandaruwan et al., 2021), the Seq2Seq model with Recurrent neural network or LSTM were frequently used in the Neural based approaches. Seq2Seq model, Transformers and Seq2Seq model with attention mechanism are much researched in recent history concerning the translation and transliteration domain. Meta

MT, which is a meta-learning approach for Low resources language are addressed under the Neural approaches which contain model parameters and Meta parameters for fast domain adaptation.

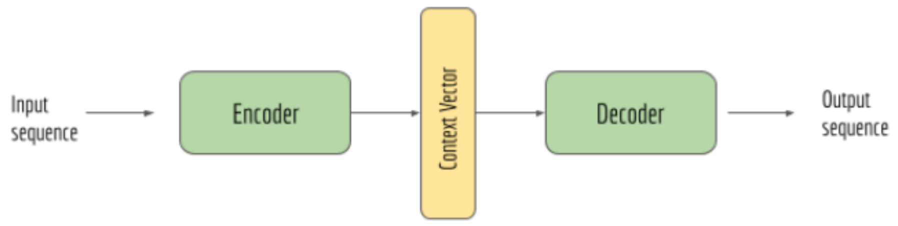


Figure 2.6 High-level encoder decoder Architecture

Hybrid Approach

The hybrid approach is one of the latest approaches in MT approaches. This will be implemented using the combination of two or many existing MT-based approaches. Two or more approached discussed in the above sections will be used in the approach where accurate answers will be produced using the approach. (Nair et al., 2016)

2.3.2 Comparison of the Algorithms

Table 2.2 Algorithm Comparison

Algorithm	Improvements	Limitations
Rule-Based/Knowledge Based Approach	• The translation has primarily been based on the vocabulary conventions of the source and target languages. • The ability to use dictionaries and grammar in both bilingual languages to find pertinent lexical information. Start with the analysis, move on to transfer, and finish with generation by following the three main basic stages. • A thorough analysis based on syntactic and semantic analysis is necessary.	• A no of rules need to be implemented. • Lack of resources and datasets. • The development of proper data sets for the process is complex. • It is more difficult to manage the rules in large-scale initiatives. • Need numerous rules for the proper translation process. • Costly, time-consuming approach and need more human effort.

		• Deep linguistic knowledge is required.
Statistical Based Approach	• Used a corpus-based bilingual lexical on both Source and target. • No complex ontology • Reusability is higher as training data can be used with similar projects.	• Bilingual data need to be managed, So the complexity is high. • For languages with a small number of parallel resources, there is a lack of viability and the feasibility • The word formatting and arrangements are different in native languages. So, the flexibility of the adaptation of the system is low.
Example Based Approach	• The ability to find the output using raw input data. • If the matching percentage is high, it will be possible to deliver structured output in a better way by utilizing examples.	• The meaning of the generated output might not meet the expectations. • The shortage of efficient texts is required for the process. • Scalability of the scope in the translation process in challenging.
Neural Based Approach	• Both source and Target can be fed to a single model for the training. • One efficient model can translate the source to Target. • Efficient outputs can be gathered.	• The speed of training models is too slow and time-consuming. • High Computational power is required for the training. • Ambiguity in the word level transliteration needs to be addressed separately.
Hybrid Approach	• Provide better efficiency in the translation process, • Better Flexibility in the translation process. • Good accuracy.	• Selecting the most suitable models for the Hybrid approach is challenging. • Experiments need to be done on model selection.

2.3.3 Tools and Techniques

There are various toolkits and libraries available for the MT process in NLP. Among them,

- The system can be implemented using the well-known open-source package "Natural Learning Toolkit (NLTK)". The following tools have been incorporated into this toolkit: sentence parsing, word segmentation, stemming, lemmatization, tokenization, N-gram tagging, and semantic reasoning. To support the rule-based approach, this toolkit also includes lexical analysis, named entity recognition, tokenization, POS tagging, parsing, and semantic reasoning.
- Sentimental analysis plus flexible models, content localization, event extraction, and intent analysis can be performed using TextBlob which is a fast tool derived using the functionalities of NLTK.
- The well-known data and sentiment analysis library "Stanford" is also used for NLP. For developers, this package also offers a wide range of NLP-based tools. This has a unique quality called "scalability." Developers may process a lot of data with less complex operations by using this module. Large-scale systems are a wonderful fit for this library. This tool's parser is widely used in translation-related applications. For the study, the English-Hindi Machine Translation (Nair et al., 2016) this module has been used.
- The Google Translation service seems to be the methodology that has been employed in the majority of translation systems. This method was applied in the Sinhala Decoder (2018) (Vidanaralage et al., 2018) translation of the source text.
- The translation method was developed using a variety of wordnets, databases, and dictionaries connected to different languages. The Sinhala Decoder (2018) (A.J. Vidanaralage, 2018) employed a lexicon of 70,000 terms in both English and Sinhala definitions to create the system. In other circumstances, the researchers must gather the necessary data sources from a variety of outside sources, including records and official papers. To avoid data inadequacy, the Si-Ta system (2018) (Ranathunga et al., 2018) has utilised data from government-related circulars, establishment codes, annual reports, and parliament-related order papers.
- Python is the primary language that can be utilized for implementation. In addition to Python, R, C#, and Java are all utilized to do a variety of tasks. However, a significant number of libraries, such as the NLTK toolkit, are readily available.

2.4 Chapter Summary

The literature review of the proposed architecture for Romanized Sinhala to Sinhala Translation is mainly separated into sections as the literature review of the domain, Existing works, and technology to discuss and justify the relevant aspects of the proposed system. Each section of the Literature review attempts to provide a clear idea of the current technologies and the existing systems which can be used for the study. It has been observed that the Transliteration process for Romanized Sinhala to Sinhala is less researched in History and the available approached are not accurate for a better typing experience. So, by identifying the drawbacks related to the current systems the author of this work will be performing research on Transliteration using statistical-based approach and Neural based approach. The chapter concluded with a summary. The next chapter will be discussed the methodology of the research project

CHAPTER 3
METHODOLOGY

3.1 Chapter Overview

This chapter provides an overview of the research, project, and development methodologies selected to continue this research. Furthermore, identifying stakeholders, the outline of gathering requirements, and analyzing gathered information to identify and priorities the requirements are to be discussed. Initially, a rich picture diagram was drawn specifying the internal and external environments of the project that gives an overview of the stakeholders of the project. Further, it consists of an analysis of different approaches followed by requirement elicitation. Use case diagrams and descriptions, and functional and non-functional requirements are specified.

3.2 Research Methodology

The research methodology section allows stakeholders to evaluate the validity and reliability of a research project critically. The Saunders research visualizes the different stages of the research work. The selection and justification for each selection are listed in the below table.

Philosophy	**Positivism** will be used where quantifiable observations and measurements will use for statistical analysis and Transliteration Schema will be implemented based on your users' comments.
Approach	A **Deductive** approach will be used with the research as the intention of the system is to use the existing theory to enhance the Transliteration of Romanized Sinhala to Sinhala. To analyze the data quantitative approach will be used.
Strategies	The primary method of collection of data will be a **quantitative** strategy where a questionnaire will be used to identify the typing patterns. As the second strategy interviews with domain expertise will be followed, where a **Qualitative approach** to verify the data. To have a wider idea of the transliteration scheme social media comments will be taken into consideration.

Choice	A **mixed method** is used as survey data must be validated with the interviews. So quantitative approach will be followed by a qualitative strategy.
Time Horizons	Data will be collected in the requirement engineering and evaluation phases in this research and will not repeatedly be collected. Hence, **cross-sectional** will be used
Techniques and other procedures.	**Questioners, interview data, reports that are published, records** related to Sinhala linguistics, and statistics will be used to validate the research project.

3.3 Development Methodology

Time, Cost and Scope must be managed properly in a research project. Thus, the development process should be structured properly with the changes in the requirements to manage cost and time. Methodologies like waterfall, agile, Rapid Application Development, RAD, and Spiral are used to manage the implementation process in the software development life cycle. But **prototype methodology**, where users are actively engaging in development is selected as errors can be found in the initial phases, missing features can be identified, and it is flexible for fluctuating requirements. Furthermore, as high evaluation interactions happen, the success of the product can be guaranteed. Hence, the **prototype methodology** is more suitable for this research project.

3.4 Project Management Methodology

Regardless of the type, any project has a scope, time duration, and cost as constraints. To produce quality output, it is necessary to manage these constraints. As many project-related processes must be followed in a controlled and manageable environment, the "**Prince2 Agile**" methodology will be used.

3.5 Data Collection Methodology

3.5.1 Data set Selection

The dataset with “Romanized Sinhala word” and respective Sinhala words based on social media content was found. (Liwera and Ranathunga, 2020). The processed dataset will be used to create transliterals required for the Study. The Dakshina dataset(Roark et al., 2020) from google which contains transliterations will be used to enhance the Transliteral set required for the unigram tagger.

A dataset on YouTube social media content that covers multi-categories will be created by the author of this work. The Dataset will cover a wide range of Romanized Sinhala and respective Sinhala words extracted from Social Media platforms. The **face pager** (Jakob Jünger, 2014) application which is freely available has been used to extract the data from YouTube. The youtube has been selected as comments are publicly available over diverse categories. Only the sentences with Romanized Sinhala will be taken into usage. The following areas are captured for the creation of the dataset.

- Social backgrounds
- Music
- Politics
- News
- Religion
- Technical

The collected dataset will be transliterated to related specific Sinhala sentences manually. Using the created dataset, the transliterals are generated which are used with Character level and Word level N-gram.

3.5.2 Data set Annotation

The online surveys are circulated among the communities to gather the typing patterns which they used to type Romanized Sinhala. Based on the 215 users' inputs the different typing patterns were identified. Identified typing patterns were analyzed to build the rule required for the Data annotation algorithm discussed below. The selected 3 sentences from the below set have been shared again with the selected group of users from the initial form to analyze the changes over time. Stratified sampling was used to identify the target audience and the related typing patterns of each group.

Table 3.1 Sinhala sentences used to identify the patterns in the survey

විවිධ අවස්ථාවල දී ක්ෂුද්‍ර ජීවීන් පාලනය සහ ජීවානුහරණය කිරීම සඳහා නොයෙකුත් ශිල්ප ක්‍රම භාවිතා කරනු ලැබේ.	කවියෙකුගේ කවි සිතුවිල්ලකට ගීත රචකයෙකුගේ ලයාන්විත ගී පබැඳුමකට සිත්තරෙකුගේ චමත්කාර සිත්තමකට හේතුවූ බොහෝ පාරිසරික පසුබිම් දැකගත හැක
ම්ලේච්ඡ ලෙස ඝාතනය කර තිබූ මිනිය සැඟවීමට උත්සාහ නොකර කෲර ලෙස විසිකර දමා තිබූ අයුරු දිස්විය	ඈත අතීතයේ සිට පැවත එන සංස්කෘතියේ එක් විශේෂාංගයක් ලෙස රජදරුවන් විසින් තැනූ වැව් සහ දාගැබ් හෙළ සිංහලයාගේ විශ්මිත හැකියාවන් පිළිබිඹු කරයි.
ඓතිහාසික වටිනාකමකින් යුක්ත වූ ඒ මහා බෝධීන් වහන්සේ කපා දැමීමට තැත් කළ සියලු දෙනා හට ස්වභාවධර්මයා විසින් දඬුවම් ලබා දුනි	බටහිර වෛද්‍ය විද්‍යාවේ ඇතැම් හිඩැස් ඌන පූර්ණය කිරීම් සඳහා ආයුර්වේද වල යොදා ගන්නා ඖෂධ සෘජුවම ඕනෑ වීම ඒවායේ ඇති ඵලදායි බව පැහැදිලි කරයි.
ප්‍රථම වතාවට පන්සල්ට පිය නැගූ පුංචි ෆාතිමා ඝන්ටාරය නාද කොට ඉන්පසුව ඊතලයක් සේ දිව ගියේ බුදු මැදුරටය	කොම්පඤ්ඤ වීදියේ පුංචි සිඤ්ඤෝ, මඤ්ඤොක්කා අරන් කඩමණ්ඩියට යන ඥානපාල ට කතා කළේ විස්කිරිඤ්ඤා දෙකක් ගෙන්න ගන්න.
දේශපාලන හැලහැප්පීම් මැද පලිගැනීම් පමණක් අරමුණු කරගත් සමාජයක, කන්නන්ගර මහතා නිදහස් අධ්‍යාපනය හඳුන්වා දෙමින් කළ සේවාව ඉමහත්ය.	

The identified patterns from the data collection chapter were used to create the data annotation algorithm which could create the Romanized Sinhala word with different typing patterns and its Sinhala word. The vowel arrangement, Consonant with Vowel usage, and the same English consonant for different Sinhala characters were identified and Rules were written. Rule-based transliterator which converts Sinhala to general Romanized Sinhala uses 60 rules for vowels and consonants, 18 rules for hal symbols, and 18 rules for special characters. The generated generalized Romanized Sinhala words were fed to the Adhoc generator which contains 12-character pattern-related rules, 6 vowel combination-related rules and 8 special character rules. The Adhoc transliteral generator will generate the Adhoc Romanized Sinhala Dictionary which will be used for the training of Trie structure. The annotation algorithm which builds upon the identified rules was used with a Sinhala Dictionary collected from the NLP society of the

University of Colombo to have a complete dataset for the process. The flow of the data annotation process and the selected rows from the dataset are as follows.

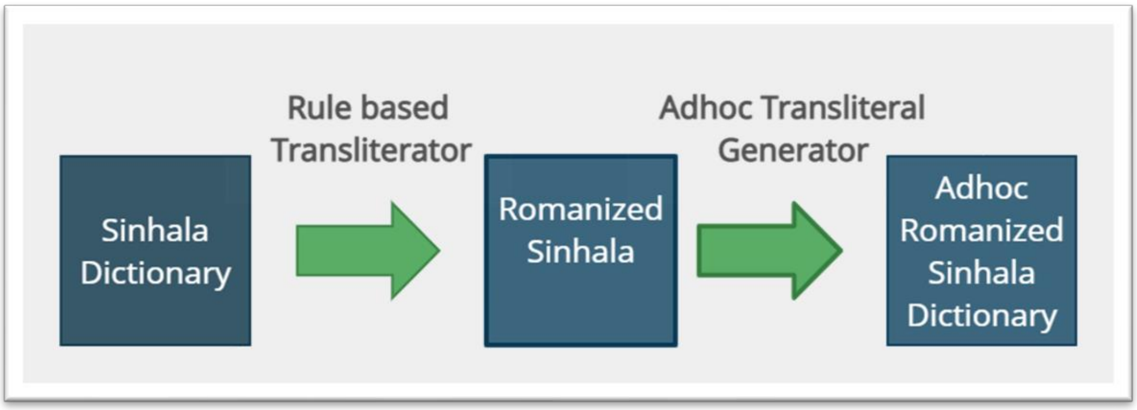

Sinhala Dictionary
Rule based Transliterator
Romanized Sinhala
Adhoc Transliteral Generator
Adhoc Romanized Sinhala Dictionary

Figure 3.1 Adhoc Transliterals generation

The Transliterated algorithm was tested with the LTRL Sinhala word list under UCSC Sinhala Corpus and all possible transliterals were generated. The generated results were evaluated using qualitative analysis and adapted with a Back transliterator. Some instances from the generator are as follows.

Table 3.2 Results of data annotation algorithm

කොහොමද	kohomada,khomada,kohmada,kohomda,kohomad,khmada,khomda,khomad,kohmda,kohmad,kohomd,khmda,khmad,khomd,kohmd,khmd
කියන්න	kiyanna,kiynna,kiyann,kiynn,kynna,kynn,kyann
පමණක්	pamanak,pmanak,pamnak,pamank,pmnak,pmank,pamnk,pmnk
යන්න	yanna,ynna,yann,ynn

3.6 Requirement Elicitation

Elicitation Techniques	Summary
Observations	The researcher of the work has conducted a thorough review of the existing commercial products which can back transliterate Romanized Sinhala to Sinhala. The following observations have been taken into consideration when defining the research gap. • The existing commercial products use ruled based mechanism where character level transliteration schema is used.

	<ul><li>The Ad-hoc transliteration schemas are not supported by many of the transliterators.</li><li>Suggestion level accuracy is bit low and suggestions are depending on the character level order.</li><li>Character level ambiguity is not handled in many of the products.</li></ul>
Literature Review	Conducting a Literature Review is important to identify the existing products and algorithmic approaches currently available in the literature. The detailed summary has been discussed under the Literature review chapter and the summary is as follows.<ul><li>As Sinhala is a low-resource language, the existing approaches' accuracy is low.</li><li>The latest research work by Liwera (Liwera and Ranathunga, 2020) which uses the N-gram method has given a reasonably good accuracy in the transliteration but which limited to a specific domain.</li><li>The Neural approach used by (Sandaruwan et al., 2021) used the Seq2Seq model in the transliteration process to convert Singlish to English. As an intermediate process, the back transliteration of Singlish to Sinhala is achieved.</li></ul>
Survey	According to the selected deductive approach, the researcher of the work has surveyed to identify the writing patterns of a diverse community. The survey is created in such a way that all the Sinhala characters and their transliterals are captured. The survey form and the results are attached to the appendix. By analyzing the responses, the rules required for the neural model will be generated and used in the transliteration process.
Interviews	The Qualitative data for research is necessary to make decisions on research. The domain experts on the NLP who work in the area has interviewed and valuable insight into the product has been captured. The proposed model has been discussed with the experts and their view and suggestions have been considered for the finetuning of the model architecture.
Brainstorming	Brainstorming is important to analyze the available techniques and

	come up with an accurate architecture. With the experts' opinions and Literature findings, using brainstorming the novel hybrid architecture has been introduced.

3.6.1 Discussion on findings from Requirement Elicitation techniques.

- Literature review
  By performing thorough research on the literature, the existing technological contribution to the field of transliteration has been discovered. A detailed discussion of the findings is discussed in the Literature review. The following sections have been identified.
  1. Rule based approach.
  2. Statistical based approach.
  3. Knowledge based approach.
  4. Example based approach.
  5. Hybrid based approach.

- Interview
  The one-to-one interviews have been conducted with the technical experts in the area and the following has been captured.

Theme	Review/Conclusion
Dataset collection	Most of the interviewers suggested using a dataset which contains data from diverse communities. Suggested to use of comments in social media such that different typing patterns can be captured.
Transliteration algorithm.	Some of the interviewers proposed using statistical approach than rule-based approach. Suggested capturing both personalized typing patterns for the transliteration process and the general rule used by the commercial product.

Use of visual artifacts in the GUI	Many interviewers suggested to use simple GUI which contains the Sinhala and Romanized Sinhala Text in a one window.
Sinhala characters and Typing patterns identifications	Some of the interviewers who were working in the Sinhala domain pointed the letters which make ambiguity in the transliteration process. Suggested to use a open survey with a diverse community such that all the Sinhala alphabets and it's Romanized Sinhala are captured.

- **Survey**

The survey was conducted in two-time phases. The changes on typing style over the time have been identified in the different phases. The users are shared with an online survey with contain 9 Sinhala sentences covering all the Sinhala alphabet and it's consonant and vowel patterns. The following results were obtained from the survey.

Sinhala Sentences	Selected User Inputs from Users	Achieved knowledge
විවිධ අවස්ථාවල දී ක්ෂුද්‍ර ජීවීන් පාලනය සහ ජීවානුහරණය කිරීම සඳහා නොයෙකුත් ශිල්ප ක්‍රම හාවිතා කරනු ලැබේ.	Wiwida awasthawala di kshudra jeewin palanaya saha jeewanuharanaya kiriima sadaha noyekuth shilpa krama bhawitha karanu labe Wiwida awasthawaladi kshudra jeewin palanaya kireema sandaha noyekuth shilpa krama bhawitha karanu lebe. Wiwidha awasthawala Dee kshudra jeewiin paalanaya Saha jeewanuharanaya kireema sandaha noyekuth Shilpa krama bhawitha karanu labe. wiwida awasthawaladi kshdra jeewin paalnaya saha jeewanuharanaya kereema sadaha noyekuth shilpa krama bhawitha karanu lAbe. Vivida awastha waladi ikshudra jiwin palanaya saha jivanuharanaya kirima sadaha noyekuth shilpa krama bawitha karanu labe. wiwida awasthawaladi kshudra jeewin palanaya saha jeewanuharanaya kirima sadaha noyekuth shilpa krama bawitha krnu labe Vivida awastha waladi kshudra jeeween paalanaya saha jeewanuharanaya kireema sandaha noyekut shilpa karama baawitha karanu labe	Letter-wise transliterals identified. Ambiguity letters were identified. වි can be represented using V or W. Vowel combinations identified.

	wiwida awasthawaladhii shrudhdha jiwiin paalanaya saha jiiwaanuharaNaya kiriima sadhahaa noyekuth shilpa krama bhaawithaa karanu lAbee. wiwida awasthawaladi kshudhra jeewin paalanaya saha jeewanuharanaya kireema sadaha noyekuth shilpa krama bhawitha karanu labe.	
කවියෙකුගේ කවි සිතුවිල්ලකට ගීත රචකයෙකුගේ ලයාන්විත ගී පබැඳුමකට සිත්තරෙකුගේ චමත්කාර සිත්තමකට හේතුවූ බොහෝ පාරිසරික පසුබිම් දැකගත හැක.	Kawiyekuge kawi sithuwillakata geetha rachakayekuge layaanwitha gee pabedumakata siththarekuge chamathkara siththamakata heethu wuu bohoo paarisarika pasubim dekagatha heka Kawiyekuge kawi sithuwillakatageetha rachakayekuge layanwitha gee pabedumakata siththarekuge chamathkara siththamakata hethu u boho parisarika pasubim dekagatha heka. Kawiyekuge Kavi sithuwillakata Geetha rachakayekuge layaanwitha gee pabaendumakata siththarekuge chamathkaara siththamakata heathy wuu boho paarisarika pasubim dekagatha haeka. kawiyekuge kawi sithuwillakata geetha rachkayekuge layanwitha gee pabedhumakata siththarakuge chamathkara siththamakata heethuwuu bohoo paarisarika pasubim dhAkagatha hAka. Kaviyekuge kavi sithuwillakata githa rachakayekuge layanwitha gi pabadumakata siththarekuge chamathkara siththamakata hethuwu boho parisarika pasubim dakagatha haka kaviyekuge kavi sithuwillakta githa rachakayekuge layanwitha gee pabadumakata siththarakuge chmathkara siththmkata hethuwu bho parisrika pasubim dakagatha hakiya Kawiyekuge kawi situwillakata geetha rachakayekuge layanwitha gee pabadumakata sithareku ge chamatkaara sithamatakata heethu wu boho paarisarika pasubim dakagatha haka. kaviyekugee kavi sithuwillakata geetha rachakayekugee layaanwitha gee pabAdhumakata siththarayekugee chamathkaara siththamakata heethuwuu bohoo paarisarika pasubim dhAkagatha hAka. kawiyekuge kawi sithuwillakata geetha rachakayekuge layanwitha gee pabadumakata siththarekuge chamathkara siththamakata hethu wu boho parisarika pasubim dakagatha haka. kawiyekugee kawi sithuwillakata giitha rachakayekugee layaanwitha gii pabadumakata siththarekugee chamathkaara	Letter wise transliterals identified. Vowel combinations identified.

	siththamakata heethuwuu bohoo paarisarika pasubim dakagatha haka kaviyekuge kavi situvillakaṭa gita rachakayekuge layanvita gi pabendumakata sittarekuge chamatkara sittamakaṭa hetuvu bohp parisarika pasubim dekagatha haka.	
ප්‍රථම වතාවට පන්සල්ට පිය නැගු පුංචි ෆාතිමා ඝන්ටාරය නාද කොට ඉන්පසුව ඊතලයක් සේ දිව ගියේ බුදු මැදුරටය	Prathama wathawata pansalata piya nagu punchi fathima ghantaraya naada kota inpasuwa eethalayak see diwa giye budu maduarataga Prathama wathawata pansalta piya negu punchi fathima ghantaraya nada kota inpasuwa ithalayak see diwa giye budu medurataya. Prathama wathawata pansalata piya naguu punchi fathima ghantaraya naada Kota inpasuwa iithalayak se Duwa giye budu maedurataya . praThama wathawata pansalata piya nAgu punchi fathima Ghantaaraya naadha kota inpasuwa eethalayak see dhiwa giye budhu mAdhuratya Prathama wathawata pansalta piya nagu punchi phathima gantaraya nada kota inpasuwa iithalayak se diwa giye budu madurataya prathama wathawata pansalata piya nagu punchi fathima gantaraya nada kota inpasuwa ithalayak se diwa giye budu madurataya Prathama wathawata pansalata piya nagu fathima gantaraya naada kota inpasuwa eethalayak se diwa giye budu madurataya. prathama wathaawata pansalata piya nAguu punchi faathimaa Ghanthaaraya naadha kota inpasuwa eethalayak see dhiwa giyee budhu mAdhurataya. prathama wathawata pansalata puya nagu punchi fathima ghantaraya nada kota inpasuwa eethalayak see diwa giye budu madurataya. prathama wathaawata piya nagu punchi faathimaa Gantaaraya naada kota inpasuwa iithalayak see diwa giyee budu madurataya prathama vatavaṭa pansalṭa piya nagu punchi fatima ghanṭaraya nada koṭa inpasuva italayak se diva giye budu medurataya	Letter wise transliterals identified. Ambiguity letters were identified. ඊ can be represented using ee or i. Vowel combinations identified.

ම්ලේච්ඡ ලෙස ඝාතනය කර තිබූ මිනිය සැඟවීමට උත්සාහ නොකර කෲර ලෙස විසිකර දමා තිබූ අයුරු දිස්විය	Milechcha lesa gathanya kra thibu miniya sagaweemata uthsha nkra krura lesa wisikara dama thibu ayuru diswiya. mleeccha lesa gaathanaya kara thibuu minima sangaviimata uthsaaha nokara kruura lesa visitara dama thibuu ayuru disviya Milichcha lesa ghathanaya kara thibu miniya sangawimata uthsaha nokara kruura lesa wisikara dama thibu ayuru diswiya Milecha lesa gathanaya kara thibu miniya segaweemata uthsaha nokara krura lesa wisikara dama thibu ayuru diswiya MileCha lesa Ghaathaya kara thibuu miniya sAgawiimata uthsaha nokara kruura lesa wisikara dhamaa thibuu ayuru dhiw wiya. Milechcha lesa gathanaya kara thabu miniya sagavimata uthsaha nokara koora lesa visikara dama thibu ayuru disviya Milecha lesa gathanaya Kara thibu miniya sagavimata uthsaha nokara kruura Lesa visikara dama thibu ayuru disviya	Letter-wise transliterals identified. Consonant and Vowel combinations were identified.

As above results, all the survey inputs were analyzed, and the rules required for the annotation algorithm were identified. As the second phase of the survey, the Sinhala sentence was given to the users. The email sent to the users and the forms are attached in the appendix.

Participant	Sinhala Sentence	Romanized Sinhala in phase 1	Romanized Sinhala in phase 2
1	විවිධ අවස්ථාවල දී ක්ෂුද්‍ර ජීවීන් පාලනය සහ ජීවානුහරණය කිරීම සඳහා නොයෙකුත් ශිල්ප ක්‍රම භාවිතා කරනු ලැබේ.	Wiwida awasthawala di kshudra jiwin palanaya saha jiwanuharanaya kirima sandaha noyekuth shilapa krama bhawitha karanau labe	Wiwida awasthawala di kshudra jeewin palanaya saha jeewanuharanaya kiriima sadaha noyekuth shilpa krama bhawitha karanu labe
	කවියෙකුගේ කවි සිතුවිල්ලකට ගීත රචකයෙකුගේ ලයාන්විත ගී පබැඳුමකට සිත්තරෙකුගේ චමත්කාර සිත්තමකට හේතුවූ බොහෝ පාරිසරික පසුබිම් දැකගත හැක.	Kawiyekuge kawi sithuwillakata geetha rachakayekuge layanwitha ger pabadumakata siththarekuge chamathkara siththamakata heethuwu boho paarisarika pasubim	Kawiyekuge kawi sithuwillakata geetha rachakayekuge layaanwitha gee pabedumakata siththarekuge chamathkara siththamakata heethu wuu bohoo paarisarika pasubim dekagatha

		dakagatha heka	heka
2	විවිධ අවස්ථාවල දී ක්ෂුද්‍ර ජීවීන් පාලනය සහ ජීවානුහරණය කිරීම සඳහා නොයෙකුත් ශිල්ප ක්‍රම භාවිතා කරනු ලැබේ	Wiwida awastha waladi shudra jeeween palanaya saha jeewanuharanaya kireema sadaha noyekuth shilpa krama bawitha karanu labe.	Wiwida awastha waladi shudra jeewin palanaya saha jeewanuharanaya kirima sadaha noyekuth shilpa krama bawitha karanu labe.
	කවියෙකුගේ කවි සිතුවිල්ලකට ගීත රචකයෙකුගේ ලයාන්විත ගී පබැඳුමකට සිත්තරෙකුගේ චමත්කාර සිත්තමකට හේතුවූ බොහෝ පාරිසරික පසුබිම් දැකගත හැක.	Kawiyekuge kawi suthuwillakata geetha rachakayakuge layanwitha gee pabadumakata sittarakuge chamathkara sittamakata hethu u boho parisarika pasubim dakagatha haka.	Kawiyakuge kawi sithuwillakata geetha rachakayakuge layanwitha gee pabadumakata siththarakuge chamathkara siththamakata hethu u boho parisarika pasubim dakagatha haka.
	ප්‍රථම වතාවට පන්සල්ට පිය නැගු පුංචි ෆාතිමා ඝන්ටාරය නාද කොට ඉන්පසුව ඊතලයක් සේ දිව ගියේ බුදු මැදුරටය	Prathama wathawata pansalata piya nagu punchi fathima ghantaraya nada kota inpasuwa ithalayak see diwa giye budu medurataya	Prathama wathawata pansalata piya naguu punchi fathima gantaraya naada kota inpasuwa iithalayak se diwa giye budu madurataya.

The above table illustrates the user inputs collected by 2 users during two-time frames in the study. It's clear that the pattern of typing changes with respect to the time and the mood of the user. So, the requirement of having Adhoc transliterals in the transliteration process is required for an efficient system. So, the author of this work proposes a hybrid model which can handle this concern in typing Sinhala using Romanized Sinhala.

3.7 System Architecture and Design Overview

The proposed system architecture, Process flow and data flow in the system are analyzed in the below subchapters. The selected layered architecture, technical stack, and the proposed data flow in the system with proper justifications are discussed in the below sections.

3.7.1 Layered Architecture

System architecture design is crucial for its ability to achieve design goals. Therefore, suitable design architecture should be followed. An analysis of selected architecture justifications for the selection is given below.

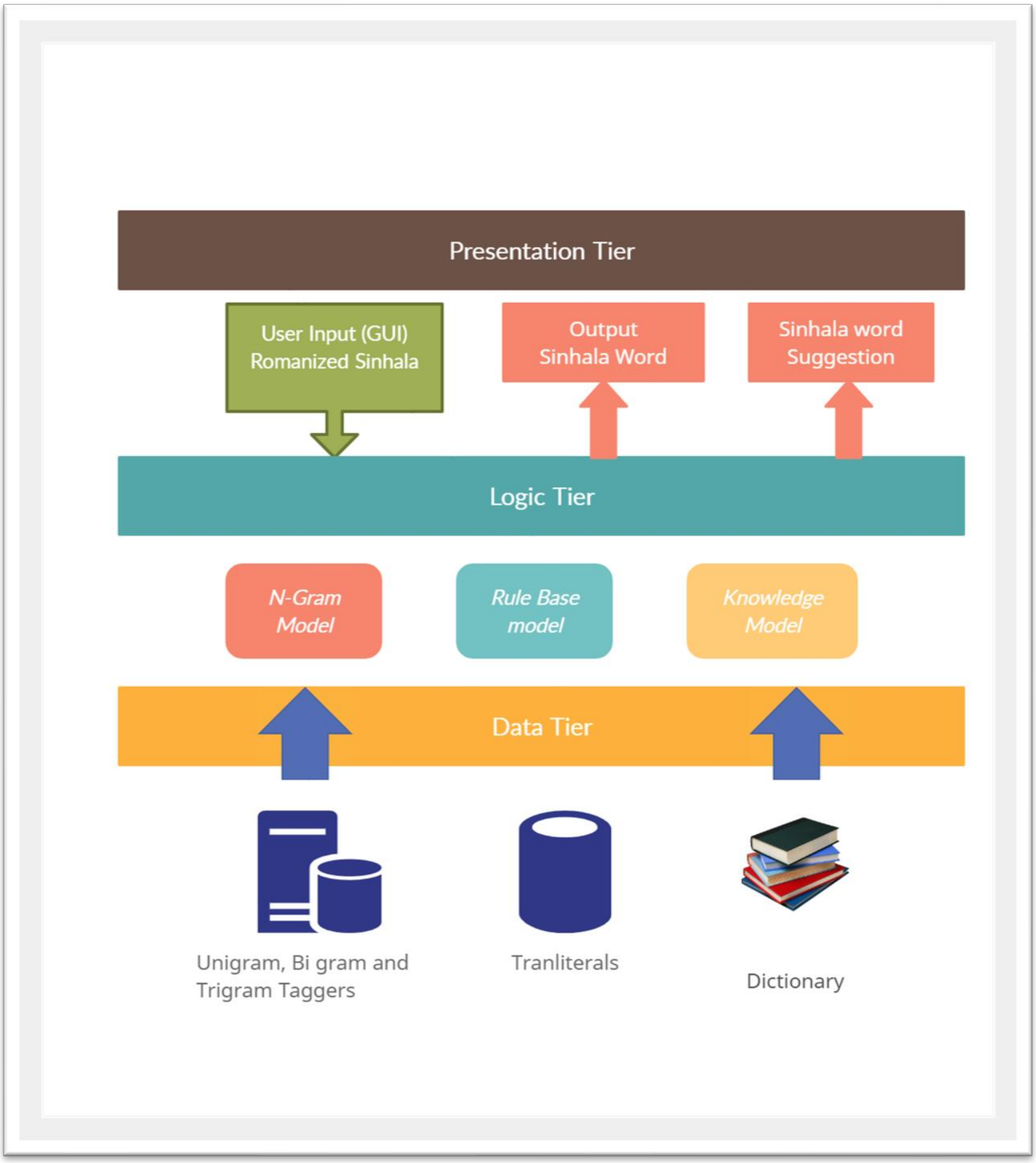


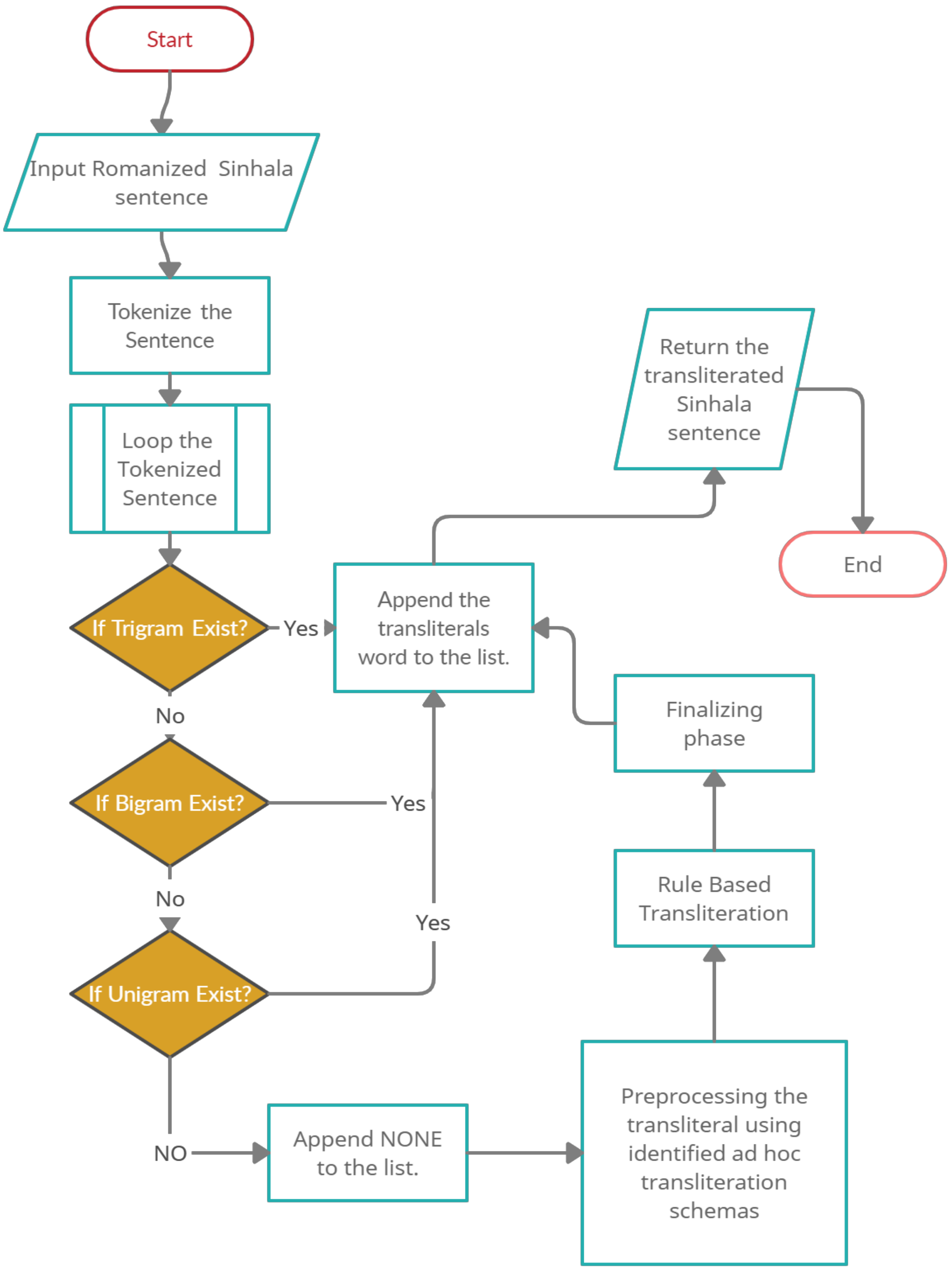
Figure 3.2 3-Tired Architecture

The proposed Model will follow the 3-tier architecture where the Components related to each section are discussed in the below table.

Presentation Layer	The User input wizard and the user output wizards are placed in the presentation layer where all the inputs required for the system will be taken through the UI Console. The Predicting level and suggesting level outputs will be given to the user via the output wizard. Error wizard in the presentation layer will prompt the invalid inputs and unrecognized transliterations schema to users.
Logical Layer	The logic layer consists of the main 3 approaches proposed in the architecture. N-gram model, Neural Model and knowledge model respectively will be used in the transliteration process. Further details are discussed in the algorithm's sections.
Data Tier	The Data layer contains all the required data resources required in the system. This contains all the datasets, Transliterals and the knowledge base used in the study.

3.8 Process Flow Chart

The process flow diagram explains the process flow in the proposed model. The Usage of the Hybrid model and the manner of data flow is discussed in the below diagram.



Start
Input Romanized Sinhala sentence
Tokenize the Sentence
Loop the Tokenized Sentence
If Trigram Exist?
Yes
No
If Bigram Exist?
Yes
No
If Unigram Exist?
Yes
NO
Append the transliterals word to the list.
Append NONE to the list.
Preprocessing the transliteral using identified ad hoc transliteration schemas
Rule Based Transliteration
Finalizing phase
Return the transliterated Sinhala sentence
End

Figure 3.3 Process Flow Diagram

3.9 Algorithms

This section gives an overview of the algorithms used for the implementation. Further information will be discussed in the implementation section.

3.9.1 The Statistical approach with N-gram tagger.

The Romanized Sinhala sentence entered by the user will be tokenized. Each token will be checked for Trigram, Bigram and Unigrm taggers. If the transliteration is available it will be appended with the Final list where the reverse transliterated Sinhala sentence will maintain. If the expected transliteral is not found the word “NONE” will be appended to the sentence. The N-gram model is using the transliterals generated in the initial processing steps. The table represents the way of transliteral generation.

Romanized Sinhala	*mama dakapu thawath nihathamani soduru minisek !*
Sinhala	මම දැකපු තවත් නිහතමානි සොදුරු මිනිසෙක් !

mama/මම	dakapu/දැකපු	*thawath/තවත්*
nihathamani/ නිහතමානි	*soduru/සොදුරු*	*minisek/මිනිසෙක්*

3.9.2 The Rule base Approach

The tokens which consist of the word “NONE” after the initial process will be examined in this chapter. Each Romanized Sinhala word related to the token will be preprocessed using the pattern identified in the ad hoc transliteration schema and fed to the Rule base module. The defined rule base will be used to generate the Sinhala word required for the transliteration process.

3.9.3 The Knowledge-based approach

In this phase, the generated transliteral is examined with a knowledge-based implementation with a Trie structure. The Knowledge-based with Romanized Sinhala/ Sinhala Transliterals along with the trained model using a Trie on Romanized Sinhala will be used to produce the most appropriate word suggestion to the user. The Romanized Sinhala words suggested by the module are searched with the knowledge-based which will return all the possible Sinhala transliterals to the user. So that ambiguity on the transliteration will be handled and the user has been given a choice to select the appropriate word from the suggestion pool.

3.10 UI Design

The UI Design of the proposed system is as follows. The prediction level and suggesting level UI interfaces are shown below.

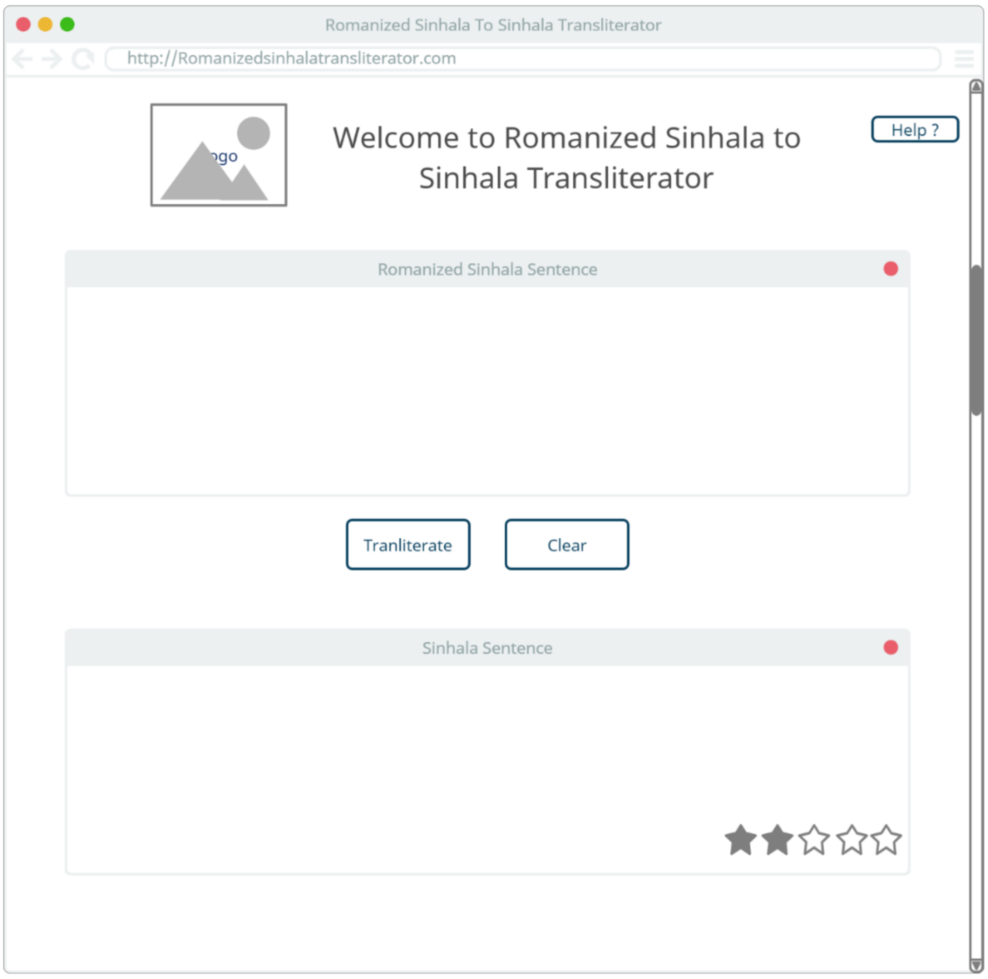

Romanized Sinhala To Sinhala Transliterator
http://Romanizedsinhalatransliterator.com
Logo
Welcome to Romanized Sinhala to Sinhala Transliterator
Help ?
Romanized Sinhala Sentence
Tranliterate
Clear
Sinhala Sentence

Figure 3.4 UI Design

3.11 Chapter Overview

This chapter outlines the design and architecture-related decisions and diagrams. Design goals were defined, and system architectures were defined to achieve identified design goals.

CHAPTER 4

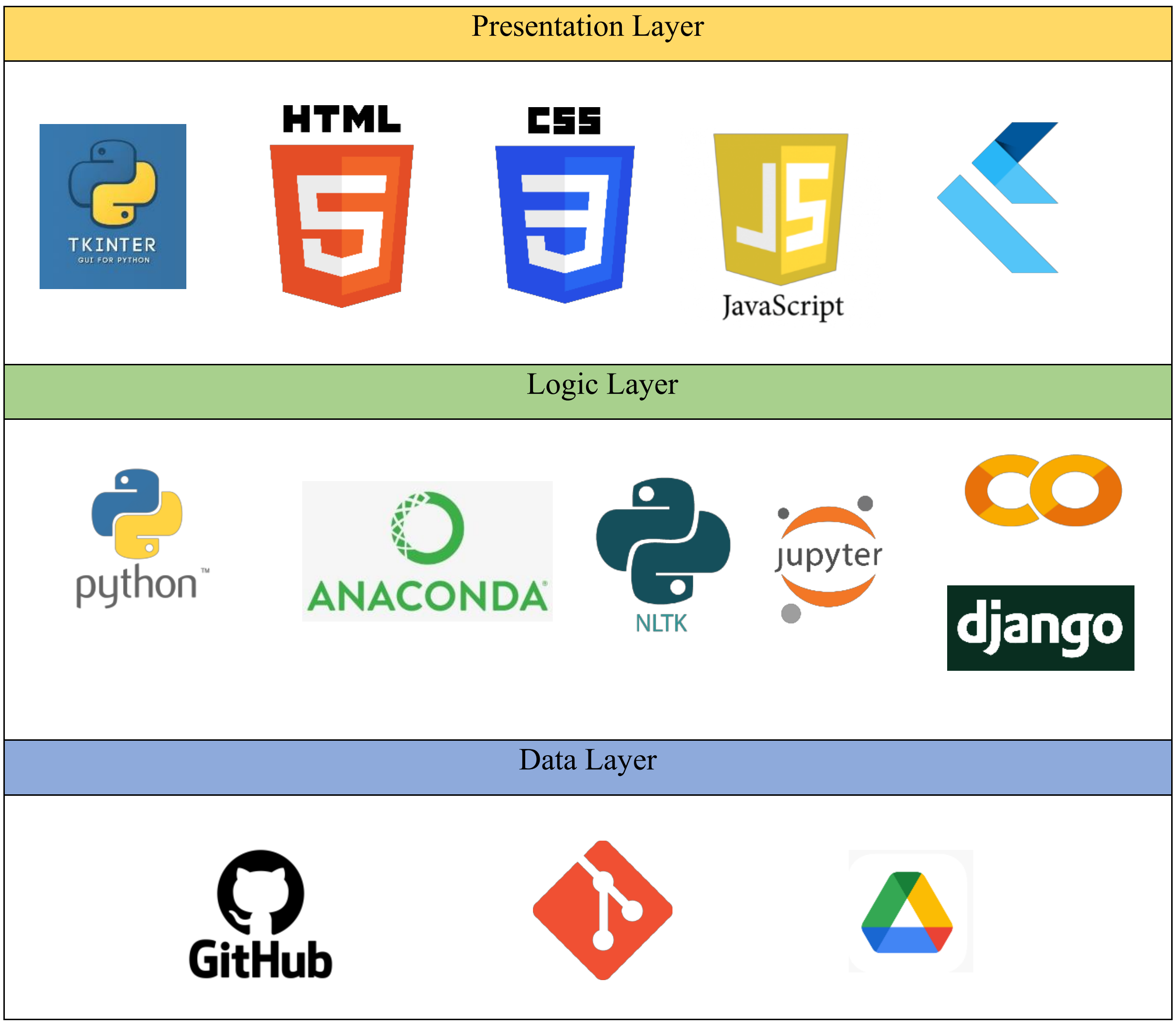
IMPLEMENTATION

4.1 Chapter Overview

This chapter will provide an overview of system implementation and decisions related to the implementation. The decisions and implementation procedures benefit from the literature review results, requirement engineering and system design, and architecture.

4.2 Technology Stack

The technology stack which has been chosen to be used in various layers of the proposed system is as per the following.



4.3 Data Annotation algorithm

In this phase, the survey data were analyzed, and data were annotated using the below 2 phases.

1. **Rule-based Transliterator (RBT)**

The Rule based generator uses to map the Sinhala dictionary to Romanized Sinhala. The following rules have been used to complete the task.

Sinhala Character	Romanized Character
'ඌ','ඕ','ඕ','ආ','ආ','ඈ','ඈ','ඈ','ඊ','ඊ','ඊ','ඊ','ඒ','ඒ','ඒ','ඌ','ඌ','ඖ','ඇ'	'oo','o\\)','oe','aa','a\\)','Aa','A\\)','ae','ii','i\\)','ie','ee','ea','e\\)','ei','uu','u\\)','au','/\a'
'අ','ඇ','ඉ','එ','උ','ඔ','ඓ'	'a','A','i','e','u','o','I'
'ඬ','ඳ','ඟ','ඨ','ඪ','ඝ','ඡ','ඵ','භ','ශ','ෂ','ඥ','ඤ','ළු','ද','ච','ඛ','ථ'	'nnd','nndh','nng','Th','Dh','gh','Ch','ph','bh','sh','Sh','GN','KN','Lu','dh','ch','kh','th'
'ට','ක','ඩ','න','ප','බ','ම',ය','ය','ජ','ල','ව','ව','ස','හ','ණ','ළ','ඛ','ඝ','ඨ','ඪ','ඵ','ඹ','ෆ','ඣ','ග'	't','k','d','n','p','b','m','y','Y','y','j','l','v','w','s','h','N','L','K','G','T','D','P','B','f','q','g'

Based on the above rules the generalized transliterals are created and passed to the Adhoc Transliteral generator.

2. **Adhoc Transliteral generator (ATG)**

The Generalized Romanized Sinhala Transliterals are fed to the ATG, where ATG can generate all the possible Romanized Sinhala patterns concerning the word.

The ATG module is consist of 12-character pattern-related rules, 6 vowel combination-related rules and 8 special character rules. The generated sequence for the Sinhala word කොහොමද through the module is as follows.

කොහොමද	kohomada,khomada,kohmada,kohomda,kohomad,khmada,khomda,khomad,kohmda,kohmad,kohomd,khmda,khmad,khomd,kohmd,khmd

The identified rules discussed in the above section are as follows.

Rule Seq	Rule description
12-character patterns	• ඇ - A, E • ඈ - Aa, Ae • ඊ - i, e • එ - e, a • ව - wa, va • ශ - sha, sa • ෂ - sa, sha • ත - tha, ta • ද - dha, da • ච - cha, ca • කි(hal) -ki, ke • ඔ - oo, o
6 vowel Combination	• a, e, i, o • ae, ea
8 special character rule	• ට - ta • ඩ - da • භ - ba • ඹ - ba • ෆ - f • ණ - na • ළ - la • ඛ - ka

1.4 Ngram Model for Romanized Sinhala to Sinhala

The Ngram model has been trained with a corpus which contains Romanized Sinhala and Sinhala sentences. The Sentences are collected through liwera's dataset, Dakshina dataset and the self-composed data materials through YouTube. Along with data in the datasets, the annotated data through the annotation algorithm discussed above was fed to the trigram model.

Collectively 12447 sentences and 7134803 words are used for training purposes. The Ngram model implementation is as follows. The NNN is defined as the default tag.

```
def ngramTranslater(train_sents, n, defaultTag='NNN'):
    t0 = nltk.DefaultTagger(defaultTag)
    if (n <= 0):
        return t0
    elif (n == 1):
        t1 = nltk.UnigramTagger(train_sents, backoff=t0)
        return t1
    elif (n == 2):
        t1 = nltk.UnigramTagger(train_sents, backoff=t0)
        t2 = nltk.BigramTagger(train_sents, backoff=t1)
        return t2
    else:
        t1 = nltk.UnigramTagger(train_sents, backoff=t0)
        t2 = nltk.BigramTagger(train_sents, backoff=t1)
        t3 = nltk.TrigramTagger(train_sents, backoff=t2)
        print(t3)
        return t3
```

1.5 Rule-based Approach

The Transliterals with "NNN" returned through the Ngram model are fed to the rule-based model, where rule based reverse transliterator uses the transliteration combination as of the rule in the appendix. Based on the rule, the Romanized Sinhala words can be transliterated to Sinhala using the algorithm. The words which were not handled by the tri-gram model can be tackled here so that the user can generate the expected output through the system.

1.6 Trie model Training and Best Word Suggestion

1.6.1 Trie Model training

The created dataset was used to train the Trie data structure. The Adhoc Romanized Sinhala patterns generated from the annotation algorithms were fed to the Trie structure to generate the trained model. The python language was used to implement the Trie structure which uses a complete binary Tree. The trained module can generate intermediate Romanized Sinhala word suggestions based on the Trie and the generated intermediate Romanized Sinhala words are fed to the Knowledge base to generate the Sinhala words. The knowledge-based is consist of multiple dictionaries which contain Romanized Sinhala to Sinhala transliterals for different writing patterns. The flow of the model training a result generation is discussed in the below diagram.

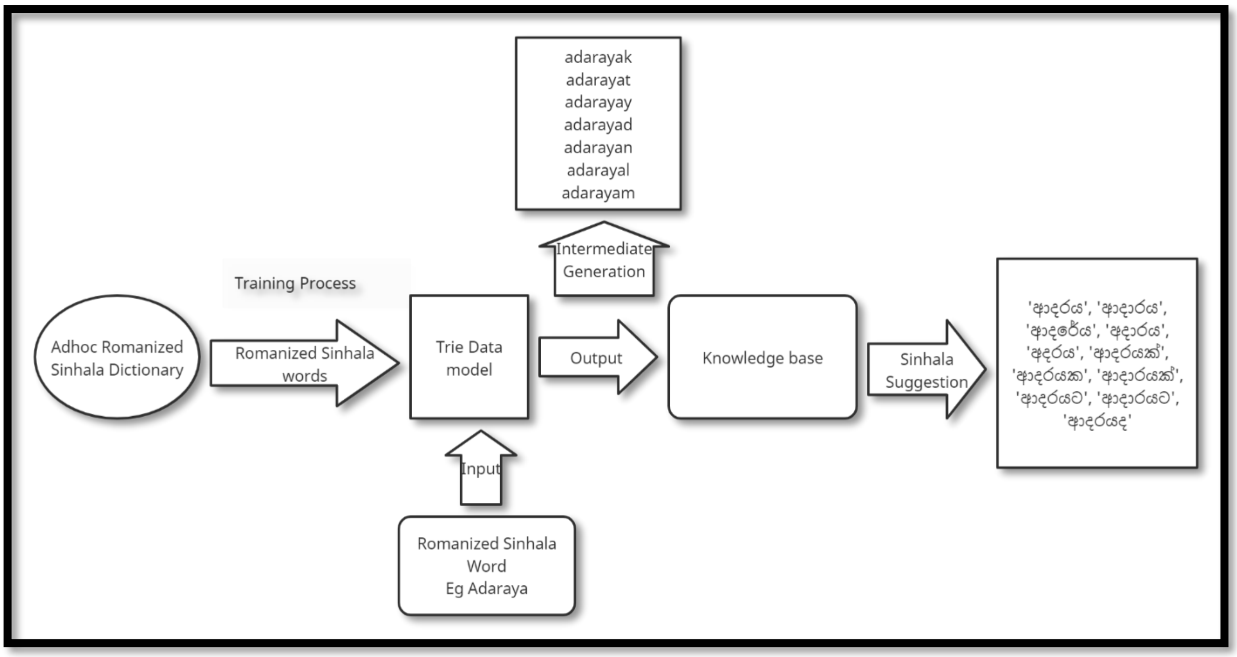


Figure 4.1 Trie Model training

1.6.2 Best Word Suggestion

The trained Trie can return all the Romanized Sinhala word suggestions of a Romanized Sinhala character set. The Suggested words are compared with the knowledge base which contains its Sinhala representation to return the best outcome to the users. The best word suggestion algorithm mainly uses a knowledge-based to return the best outcome. The above diagram discusses the exact data flow of the proposed solution.

```
def suggestionsRec(self, node, word):

    # Method to recursively traverse the trie
    # and return a whole word.
    if node.last:
        sin_indexes = [n for n, x in enumerate(eData) if x == word]
        for i in sin_indexes:
            y = int(i)
            if sData[y] not in lstword:
                # print(sData[y])
                lstword.append(sData[y])
```

1.7 GUI Implementation

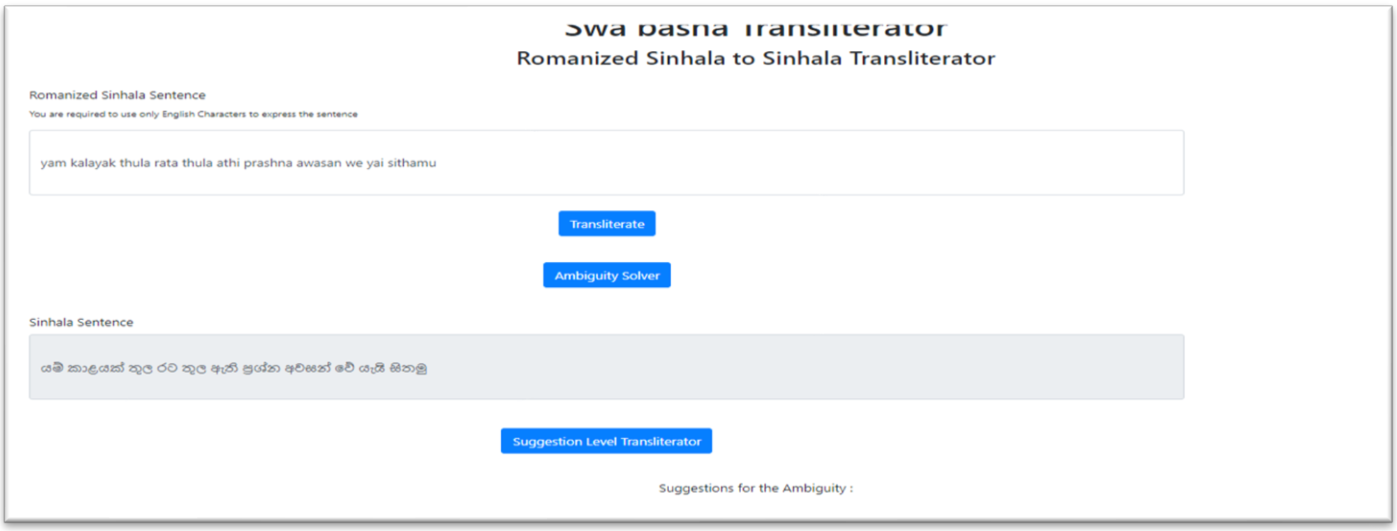
GUI implementation is done using python flask, HTML, CSS, and JavaScript. The main routing file of the program is as follows.

```
import singlishTag
import BreakData


app= Flask(__name__)

@app.route("/")
def index():
   return render_template("index.html")


@app.route('/', methods =["GET", "POST"])
def gfg():
    if request.method == "POST":

       text = request.form.get("source")
       result = singlishTag.triGramTranslate(text)
       return render_template("index.html",source=text,result=result)
    return render_template("index.html")

@app.route('/suggest', methods =["GET", "POST"])
```

```
def funsug():
    if request.method == "POST":
        s=request.form.get("s")
        r=request.form.get("r")
        text = request.form.get("stext")
        result = BreakData.translate(text)
        val=""

        if result == -1 :
            val = "No suggestions"
        elif isinstance(result,int):
            val = "No suggestions"
        else:
            for j in result:
                val+=j
                val+='\n'

        return render_template("suggest.html",source=r,result=s,suggest=val)
    return render_template("suggest.html")

@app.route('/form', methods =["GET", "POST"])
def form():
    if request.method == "POST":
        text = request.form.get("source")
        result = BreakData.translate(text)
        strfinal=""
        for i in result:
            strfinal+=i
            strfinal+=" "
        return render_template("form.html",source=text,result=strfinal)

    return render_template("form.html")

if __name__ =="__main__":
    app.run(host="127.0.0.1", port=8000, debug=True)
```

1.8 GUI

The index page is as follows. The user can insert the Romanized Sinhala sentence into the top section and the button transliterate will generate the expected output in the section Sinhala sentence. By ambiguity solver, the user can request word suggestions as indicated in the Figure 4.3 of the section.



Swa basha Transliterator

Romanized Sinhala to Sinhala Transliterator

Romanized Sinhala Sentence

You are required to use only English Characters to express the sentence

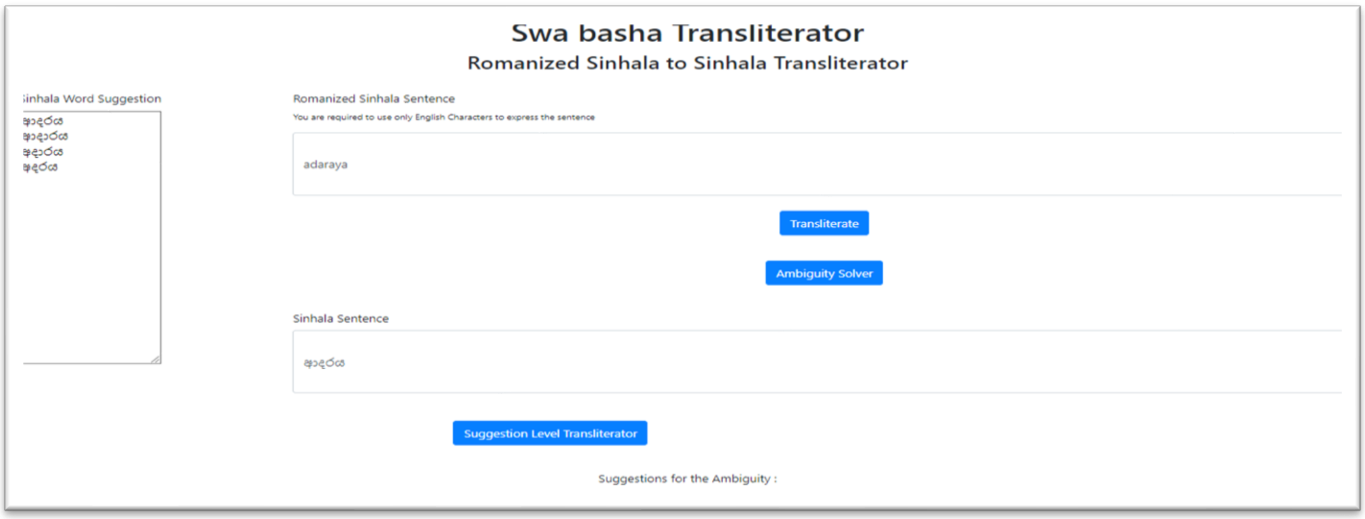
yam kalayak thula rata thula athi prashna awasan we yai sithamu

Sinhala Sentence

යම් කාළයක් තුල රට තුල ඇති ප්‍රශ්න අවසන් වේ යැයි සිතමු

Suggestions for the Ambiguity :

Figure 4.2 GUI for Transliterator home page



Swa basha Transliterator

Romanized Sinhala to Sinhala Transliterator

inhala Word Suggestion

ආදරය
ආදාරය
අදාරය
අදරය

Romanized Sinhala Sentence

You are required to use only English Characters to express the sentence

adaraya

Sinhala Sentence

ආදරය

Suggestions for the Ambiguity :

Figure 4.3 GUI for suggestion level Transliterator

1.9 Chapter Summary

In this chapter the implementation of the system, the Technology stack, used algorithms and the UI components are discussed.

CHAPTER 5

EVALUATION AND RESULTS

5.1 Chapter Overview

This chapter provides an overview of Quantitative evaluations and Qualitative evaluations based on feedback collected from domain experts, technical experts, academic experts, and researchers for different aspects of the research and prototype, such as the novelty of the concept, quality of the final product etc. The author's evaluation of different criteria is also to be presented. Functional and non-functional requirements will be evaluated to ensure that system meets the expected requirement.

5.2 Evaluation Methodology and Approach

The Evaluation is to measure the success of a project. Feedback to be collected regarding the research problem, design, implementation, testing and other major phases of a project. The Romanized Sinhala to Sinhala Transliterator is a research project that mainly focuses on proposing a novel architecture for Reverse transliterating. Therefore, the author has chosen qualitative and quantitative approaches to evaluate the success of the project. To analyze the qualitative data, deductive thematic analysis is to be used. Despite the qualitative evaluations, quantitative evaluation feedback will be gathered during the evaluation phase and will be analyzed and presented in the quantitative evaluation chapter. Questionnaires will be distributed, and interviews to be conducted with domain experts to gather feedback. The benchmarking evaluation will be performed against the existing system against different datasets.

5.3 Evaluation Criteria

5.3.1 Self-evaluation

Criteria	Authors Evaluation	Discussion
Concept of the research	Authors' view on the Concept of the research to be provided.	The main aim of the research is to build a reverse transliterator for Romanized Sinhala to Sinhala which can respond to Adhoc transliterals
Scope and depth of the project	Authors' view on the Scope and depth of the project to be provided.	The Scope and the depth have been discussed under section no one.
System design	Author's view on the System design to be provided.	The proposed hybrid architecture which is muchly credible to perform the transliteration process has been discussed. Ngram model, Rule based with Suggestion level algorithm enhanced the accuracy of the system.
Solution and the prototype	Authors' view on the Solution and the prototype to be provided.	The web-based application with user-friendly UI has been provided. Prediction and suggestion level interfaces provide a better sight to the user.
Limitations and future enhancements	Authors' view on the Limitations and future enhancements to be provided.	The system is limited to Sinhala where code mixed language can be used in the future to predict the better output.

5.3.2 Qualitative Evaluation

The following evaluation criteria were defined according to the principles identified for the evaluation. The Qualitative analysis will be performed based on the below criteria and the form uses to collect qualitative feedback is attached in the appendix.

Evaluation Criteria

Criteria	Purpose
Concept / Novelty / Difficulty / Scope	
The novelty and concept	To evaluate the novelty of the proposed concept.
The scope of the project	To evaluate the project scope by experts.
Proposed architecture	The proposed solution architecture was evaluated.
Technical difficulty	Evaluate and get feedback on the technical difficulty.
Design / Architecture / Implementation	
Overall Solution	Evaluate whether the system provides a solution to the identified problem
System architecture	Evaluate the system architecture.
Technology stack	Expert feedback on the used technology stack
Limitations	limitations on design
Future work/features suggestions	Improvements/ suggestions for the design or architecture.
Quality of the GUI	Expert feedback on the GUI quality.

Selection of Evaluators

The selection of evaluators will be based on the following Requirements and Expertise.

Category	Evaluator	Requirement	Expected Expertise
C1	Domain and Technical Experts (Area of NLP, Data Science and Machine learning)	Evaluating Novelty, Scope, Architecture, Design, and implementation.	• Experience in the area. • Graduate
C2	Technical Experts	Evaluating the Technical Stack, Usability and GUI Components.	• Experience in the area. • Graduate

C3	Beginner Researchers	Evaluating the overall Solution and the technical limitations.	• Undergraduate
C4	General Community	Evaluating the overall solution.	• Using English – Sinhala Keyboards • Technical Fluency to use an application

Qualitative Evaluation Results

The below table demonstrates the evaluators' details who evaluated the system.

EID	Position	Affiliation	Academic	Category
1	Research Head	Orel-IT	PhD	C1, C2
2	System Engineer and Senior Lecturer	Jabil, New York Florida International University	PhD	C2
3	Senior Lecturer	University of Kelaniya	PhD	C1, C2
4	PhD Candidate	Robert Gordon University	PhD (Reading)	C1, C2
5	PhD Candidate	University of Westminster	PhD (Reading)	C1, C2
6	Senior Lecturer	Informatics Institute of Technology	MSc	C1
7	Lecturer	Informatics Institute of Technology	MSc	C1, C2
8	Lecturer	Informatics Institute of Technology	MSc	C1, C2
9	Quality Assurance Engineer	Matific	MSc (Reading)	C2
10	Research Assistant	Edith Cowan University	MSc (Reading)	C2
11	Lecturer	Sri Lanka Institute of Information Technology (SLIIT)	MSc	C1, C2
12	Software Engineer	Intervest Software Technologies	MSc	C2
13	Research Assistant	University of Chemistry and Technology, Prague	MSc (Reading)	C4
14	Undergraduate Student	Informatics Institute of Technology	BSc (Reading)	C3
15	Undergraduate Student	Informatics Institute of Technology	BSc (Reading)	C3
16	Lecturer	Informatics Institute of Technology	BSc	C4

The general feedback by the evaluators are as follows.

EID	Feedback
1	The candidate has done a reasonable attempt to make transliteration of Romanized Sinhala to Sinhala using a hybrid approach. The ad-hoc transliteration algorithm used in the system has enhanced the experience of the transliteration. System need be integrated with Transliteration frameworks. Overall solution is very Good.
2	The candidate has put significant effort and time towards the project and developed a usable transliteration tool. Data were acquired in the form of research beforehand. He has considered most of the practical aspects and well-defined the project's scope, and achieved them. The same person may spell the same word in different letters based on their mood, and the candidate has considered that in the development. Great presentation and excellent work.
3	This Romanized sinhala to sinhala translator is technically an ensemble model containing different algorithms resulting good accuracy. It's usability and practicability is high. Usability is further improved with "Ambiguity Solver" and "Suggestions" and better to make this available as a software/service.
4	The research appears to be challenging, with the prototype showcasing the end result in converting variations of a Romanized Sinhala word to the corresponding possible Sinhala word or alternatives. The GUI could be improved but it is satisfactory in showing the essence of the research. Could suggest to do more random inputs from social media to get a real-feel of a accuracy rating.
5	A good research problem and solution indeed. The approach and the methodology are satisfying, and the result is also promising. Hope this would be improved into a pluggable solution to the current systems.
6	This work is focused on Romanized sinhala to sinhala translation by a novel model that contains a combination of statistical, rule based and a knowledge-based approaches. This has contributed towards improving the accuracy of this type of translations. Also the researchers have considered different types romanizations to improve the practicality of the product. A simple and user friendly UI improves the usability of the application. Overall, from both the technical contribution and practicability point of view, this application this is a very good work.
7	Ambiguity solver words cannot identify the quickly. try to show those words separate area. This is a good project more than currently existing system. That suggestion words are more effective. Then user can get clear idea about what it is.
8	The depth of the project is good but try to apply a neural network model to improve the accuracy further.
9	Transliteration has been taken into the next step through this research project where it can be further improved into a better version in the business perspective.
10	very good scope
11	Good attempt in terms of the scope of the research and the usability of the application.
12	Satisfied with the output demonstrated using the prototype. UI could be further improved.
13	Excellent work. This would make the work more efficient.
14	The system architecture, implementation, technologies and tools are all excellent
15	Accuracy, Fluency, and clarity of the local language are acceptable. Appropriate tone and style are used.
16	Some Sinhala words do not consider the actual Sinhala grammar. This would be fine if the user is only interested in learning pronunciation but would be a problem if the user is also interested in learning writing as well.

5.3.3 Quantitative Evaluation

The Quantitative analysis of this research work describes the credibility of the model and system based on the numerical data. The author of this work will use two quantitative evaluations as follows.

A descriptive analysis of Quantitative data

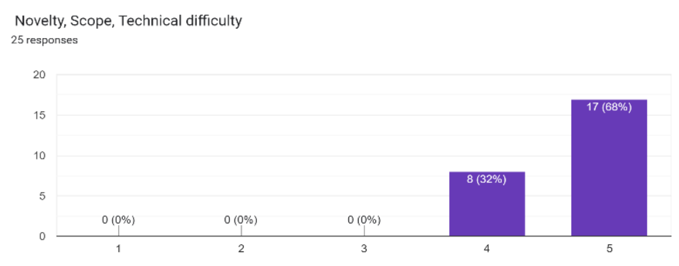
The quantitative analysis will be carried out according to the below 5 main sections. The results captured will be analyzed using the descriptive analysis method.

Table 5.1 Evaluation Criteria for Quantitative data in Survey

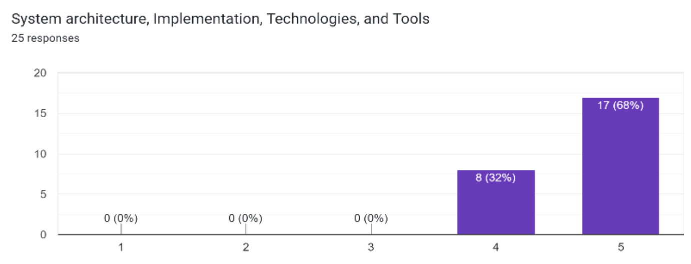
Criteria	Evaluating Section	Description	Evaluation Criteria
1	Novelty, scope, technical difficulty	Linear scale to be used.	0 -5 scale 0 - poor 5 - Excellent
2	System architecture, implementation, Technologies, and Tools	Linear scale to be used.	
3	Overall solution	Linear scale to be used.	
4	Testing, Accuracy of the prototype	Linear scale to be used.	
5	Overall feedback on the GUI	Linear scale to be used.	

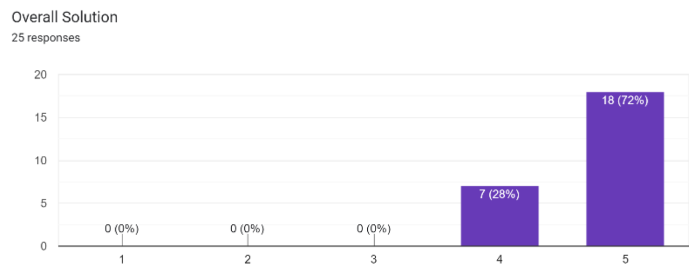
Quantitative Evaluation Results based on the evaluation.

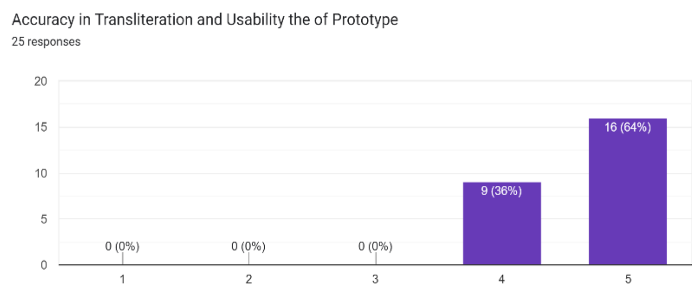
- 25 evaluators have responded to the evaluation form. The summary of the results is as follows.
- The discussion is based on the criteria given in table 5.1.

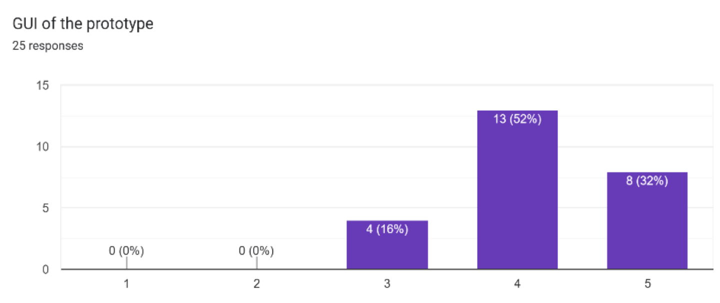
Criteria: **Novelty, Scope, Technical difficulty**	
Result 	Discussion: 68% of the evaluators say that the novelty, scope and technical difficulty of the project is very high, and 32% percentage agree on the same with stating high.

Author’s review: This idea of transliteration using Adhoc transliteration is a novel idea where currently any existing systems are not capable to transliterate using vowel-free typing.

Criteria: **System architecture, Implementation, Technologies, and Tools**	
Result 	Discussion 68% of the evaluators say that the System architecture, Implementation, Technologies, and Tools of the project is highly appropriate and relevant, and the rest 32% percentage agree on the same.
Author’s review: The system uses ngram, rule and knowledge based approaches together to perform the transliteration. The hybrid model has done justice to the Romanized Sinhala to Sinhala Transliteration.	

Criteria: **Overall Solution**	
Result 	Discussion 72% of the evaluators say that the Overall Solution of the project is highly appropriate and can be pluggable into the current system, and the rest 28% percentage agree on the same with a rating as 4.
Author’s review: The overall proposed solution is capable of Transliterating Romanized Sinhala to Sinhala. Adhoc Transliterals are taken into consideration when personalized typing is captured through the work.	

Criteria: **Accuracy in Transliteration and Usability the of Prototype**	
Result 	Discussion 64% of the evaluators say that the Accuracy of the transliteration of the project is excellent while rest 36% percentage agree on the same with rating as 4.
Author’s review: The system accuracy is 84% in the suggestion level. The system's accuracy can be enhanced by using more data to train the model.	
Criteria: **GUI of the prototype**	

Result	Discussion
	32% of the evaluators say that the GUI component of the project is excellent and rest 52% agree with the status as good. 16% say its moderate.
Author’s review: The GUI need to be enhanced to improve the user-friendliness. The system can be implemented such that it can be a pluggable module to all the current system	

5.3.4 Numerical Analysis

BLEU (Bilingual Evaluation Understudy) is a metric for automatically evaluating machine-translated text. The BLEU score is a number between zero (0) and one (1) that measures the similarity of the machine-translated text to a set of high-quality reference translations. (Osborne and Koehn, 2006) The proposed system will be evaluated with the below datasets to generate the BLEU score.

The Dataset	Reference
Hate Speech Dataset for Romanized Sinhala by Liwera.	(Liwera and Ranathunga, 2020)

Results

The corpus_bleu of NLTK was used to identify the result. The following Bleu matrixes were used.

```
print('BLEU-1: %f' % corpus_bleu(a, p, weights=(1.0, 0, 0, 0)))
print('BLEU-2: %f' % corpus_bleu(a, p, weights=(0.5, 0.5, 0, 0)))
print('BLEU-3: %f' % corpus_bleu(a, p, weights=(0.3, 0.3, 0.3, 0)))
print('BLEU-4: %f' % corpus_bleu(a, p, weights=(0.25, 0.25, 0.25, 0.25)))
```

The Results are as follows.

BLEU	Score
BLEU-1	0.820000
BLEU-2	0.804462
BLEU-3	0.777854
BLEU-4	0.711657

Furthermore different systems available in the literature, and commercial products will be evaluated based on the test data of the Dakshina dataset. (Roark et al., 2020). The word level Accuracy is to be measured using the accuracy score based on the below criteria.

Criteria		Dataset to be used
Word Level Accuracy	Predicting level	Dakshina Dataset (Roark et al., 2020)
	Suggestion level	Dakshina Dataset ((Roark et al., 2020)

The benchmark of the system compared to the current literature are as follows. The models were tested with 200 unique data records. 100 data record was randomly selected from liwera's (Liwera and Ranathunga, 2020) dataset, which was much towards the general Sinhala words for communication and 100 unique data tuples were taken from the Dakshina dataset which is from the documentation. The sentences were fed to the model and word-based suggestion from the Trie model was evaluated.

The accuracy was generated using the below equations:

Accuracy = (Correctly suggested words / Total words) * 100%

Results

Transliterator	Technology Used	Accuracy (word Level)
(Liwera and Ranathunga, 2020)	N gram + rule-based Algorithm	0.62
(Athukorala and Sumanathilaka, 2021)	Rule based and Fuzzy logic	0.74
Proposed Novel Swabasha Transliterator	Ngram + Rule base and Trie with a knowledge base	0.84

5.3.5 Evaluation of Functional Requirements.

The functional and non-functional requirements are to be evaluated based on the below criteria.

FRID	**Requirement ID**	**Priority Level**	**Status**
FR1	The user should be able to type Romanized Sinhala words via the keyboard as input.	Highly Important	Completed.
FR2	The system should generate the native Sinhala words by considering the entered Romanized Sinhala words.	Highly Important	Completed.
FR3	The system should predict and suggest appropriate Sinhala word and display it to the user.	Highly Important	Completed.
FR4	The system should be able to use the predicted word for the output.	Important	Completed.
Functional Requirement completion rate			**100%**

5.3.6 Evaluation of the Research Questions

- **RQ1: How to create a valid dataset to perform back transliteration.**
  The dataset creation has been conducted based on the rule identified through the survey which has been done over the general community. The typing patterns were identified and build an algorithm to generate all the typing possibilities.

- **RQ2: How to identify the Sinhala words and their typing patterns which are frequently used by people when texting.**
  The Social media data has been extracted using an open-source tool called face pager and the data was analyzed to identify the typing patterns.

- **RQ3: How to build the optimal algorithm to back-transliterate Romanized Sinhala to Sinhala.**
  A hybrid algorithm which contains an n-gram model, Rule-based and Knowledge-based has been used to perform the back-transliteration process. The algorithm will be evaluated with multiple datasets for its performance. Benchmarking will be performed concerning the existing products.

- **RQ4: What kind of preprocessing steps and learning methods should be carried out for the core component**.
  Multiple NLP-based preprocessing steps have been used to perform the preprocessing of the data required to be fed to the model. Detailed information has been provided in the methodology chapter.

5.4 Benchmarking the functionalities

Criteria	Swabasha Transliterator (Proposed Solution)	(Liwera and Ranathunga, 2020)	(Silva and Ahangama, 2021)	Helakuru (Bhasha, 2011)
Transliteration of Romanized Sinhala to Sinhala.	✓	✓	✓	✓
Short typing (Adhoc Transliteration) for Transliteration	✓	✓	✗	✗
Word Suggestion for ambiguity-based words.	✓	✗	✗	✗
Web application for the tool.	✓	✗	✗	✓
Next characters suggestion for Romanized Sinhala based on input.	✓	✗	✗	✓

5.5 Chapter Summary

This chapter focused on evaluating the model using Quantitative and Qualitative techniques. A detailed evaluation plan is discussed in the above chapter.

CHAPTER 6

CONCLUSION AND FUTURE WORKS

6.1 Chapter overview

This chapter presents the project's final remarks. It was analyzed whether the project aims, and objectives were met successfully, the challenges encountered and the limitation with future works. The research contributions to the domain and computer science are presented.

6.2 Conclusion

In a conclusion, the proposed novel transliterator can Reverse transliterate the Romanized Sinhala to Sinhala which is written using shorthand typing. The aim of the project *which is to analyze, design, develop and evaluate a generalized system that back transliterates Romanized Sinhala to Sinhala using word level N-gram and rule-based module with handling ambiguity in the transliteration process by suggestion algorithm using a Trie* has been successfully achieved.

And Adhoc Romanized Sinhala to Sinhala Dataset has been created using a novel algorithm based on the survey data. This novel transliterator which uses a hybrid approach can escalate the Native language usage among Sri Lankans on digital platforms.

6.3 Limitations of the Research

- Code mixed Romanized Sinhala is not considered in the project.
- The annotation algorithm can be enhanced by analyzing the survey data.
- Special characters are not handled by transliterators.

6.4 Future work of the Research

6.4.1 Research Components

- The Neural-based approach which uses the seq2seq model can cooperate with the model.
- The code-mixed language which uses both Romanized Sinhala and English can cooperate with the algorithm.
- A general framework for extracting the rules from the survey data can be introduced.
- Short-net acronyms can be introduced with the system.
- The proposed architecture can be tested with south Indian languages.

6.4.2 Product Component

- The system can be made available as a keyboard to the mobile application, such that users can use the software as a 3rd party application to type Sinhala.
- The mobile compatibility of the web tool can be enhanced.

6.5 Chapter Summary

This chapter discussed the conclusion, limitations and future works that can be cooperated in the project.

REFERENCES

De Silva, D. et al. (2008) 'Sinhala to English Language Translator', in 2008 4th International Conference on Information and Automation for Sustainability. 2008 4th International Conference on Information and Automation for Sustainability (ICIAFS), Colombo, Sri Lanka: IEEE, pp. 419–424. doi:10.1109/ICIAFS.2008.4783983.Athukorala, M.U., Sumanathilaka, D.K., 2021. Swa Bhasha: Message-Based Singlish to Sinhala Transliteration. Int. Conf. Innov. Info-Bus. Technol. ICIIT2022 Inform. Inst. Technol. Colombo 006 Sri Lanka 6.

Bhasha, D.P., 2011. Helakuru Transliterator.

de Silva, N., 2021. Survey on Publicly Available Sinhala Natural Language Processing Tools and Research. ArXiv190602358 Cs.

Hettiarachchi, N., Weerasinghe, R., Pushpanda, R., 2020. Detecting Hate Speech in Social Media Articles in Romanized Sinhala, in: 2020 20th International Conference on Advances in ICT for Emerging Regions (ICTer). Presented at the 2020 20th International Conference on Advances in ICT for Emerging Regions (ICTer), IEEE, Colombo, Sri Lanka, pp. 250–255. https://doi.org/10.1109/ICTer51097.2020.9325465

Jakob Jünger, T.K., 2014. Facepager 3.6.

Joshi, A., Mehta, K., Gupta, N., Valloli, V.K., 2018. Indian Language Transliteration Using Deep Learning, in: 2018 IEEE Recent Advances in Intelligent Computational Systems (RAICS). Presented at the 2018 IEEE Recent Advances in Intelligent Computational Systems (RAICS), IEEE, Thiruvananthapuram, India, pp. 103–107. https://doi.org/10.1109/RAICS.2018.8635077

Liwera, W.M.P., Ranathunga, L., 2020. Combination of Trigram and Rule-based Model for Singlish to Sinhala Transliteration by Focusing Social Media Text, in: 2020 From Innovation to Impact (FITI). Presented at the 2020 From Innovation to Impact (FITI), IEEE, Colombo, Sri Lanka, pp. 1–5. https://doi.org/10.1109/FITI52050.2020.9424880

Nair, J., Krishnan, K.A., Deetha, R., 2016. An efficient English to Hindi machine translation system using hybrid mechanism, in: 2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI). Presented at the 2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI), IEEE, Jaipur, India, pp. 2109–2113. https://doi.org/10.1109/ICACCI.2016.7732363

Osborne, C.C.-B.M., Koehn, P., 2006. Re-evaluating the Role of BLEU in Machine Translation Research 8.

Roark, B., Wolf-Sonkin, L., Kirov, C., Mielke, S.J., Johny, C., Demirsahin, I., Hall, K., 2020. Processing South Asian Languages Written in the Latin Script: the Dakshina Dataset 11.

Sandaruwan, D., Sumathipala, S., Fernando, S., 2021. Neural Machine Translation Approach for Singlish to English Translation. Int. J. Adv. ICT Emerg. Reg. ICTer 14, 36. https://doi.org/10.4038/icter.v14i3.7230

Silva, L. de, Ahangama, S., 2021. Singlish to Sinhala Transliteration using Rule-based Approach, in: 2021 IEEE 16th International Conference on Industrial and Information Systems (ICIIS). Presented at the 2021 IEEE 16th International Conference on Industrial and Information Systems (ICIIS), IEEE, Kandy, Sri Lanka, pp. 162–167. https://doi.org/10.1109/ICIIS53135.2021.9660744

Vidanaralage, A.J., Illangakoon, A.U., Sumanaweera, S.Y., Pavithra, C., Thelijjagoda, S., 2018. Sinhala Language Decoder, in: 2018 National Information Technology Conference (NITC). Presented at the 2018 National Information Technology Conference (NITC), IEEE, Colombo, pp. 1–5. https://doi.org/10.1109/NITC.2018.8550074

Wijerathna, L., Somaweera, W.L.S.L., Kaduruwana, S.L., Wijesinghe, Y.V., De Silva, D.I., Pulasinghe, K., Thellijjagoda, S., 2012. A Translator from Sinhala to English and English to Sinhala (SEES), in: International Conference on Advances in ICT for Emerging Regions (ICTer2012). Presented at the 2012 International Conference on Advances in ICT for Emerging Regions (ICTer), IEEE, Colombo, Western, Sri Lanka, pp. 14–18. https://doi.org/10.1109/ICTer.2012.6421408

Dhariya, O., Malviya, S. and Tiwary, U.S. (2017) ‘A hybrid approach for Hindi-English machine translation’, in *2017 International Conference on Information Networking (ICOIN). 2017 International Conference on Information Networking (ICOIN)*, Da Nang, Vietnam: IEEE, pp. 389–394. doi:10.1109/ICOIN.2017.7899465.

Hettiarachchi, N., Weerasinghe, R. and Pushpanda, R. (2020) ‘Detecting Hate Speech in Social Media Articles in Romanized Sinhala’, in *2020 20th International Conference on Advances in ICT for Emerging Regions (ICTer). 2020 20th International Conference on Advances in ICT for Emerging Regions (ICTer)*, Colombo, Sri Lanka: IEEE, pp. 250–255. doi:10.1109/ICTer51097.2020.9325465.

Joshi, A. *et al.* (2018) ‘Indian Language Transliteration Using Deep Learning’, in *2018 IEEE Recent Advances in Intelligent Computational Systems (RAICS). 2018 IEEE Recent Advances in Intelligent Computational Systems (RAICS)*, Thiruvananthapuram, India: IEEE, pp. 103–107. doi:10.1109/RAICS.2018.8635077.

Liwera, W.M.P. and Ranathunga, L. (2020) ‘Combination of Trigram and Rule-based Model for Singlish to Sinhala Transliteration by Focusing Social Media Text’, in *2020 From Innovation to Impact (FITI). 2020 From Innovation to Impact (FITI)*, Colombo, Sri Lanka: IEEE, pp. 1–5. doi:10.1109/FITI52050.2020.9424880.

Nair, J., Krishnan, K.A. and Deetha, R. (2016) ‘An efficient English to Hindi machine translation system using hybrid mechanism’, in *2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI). 2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, Jaipur, India: IEEE, pp. 2109–2113. doi:10.1109/ICACCI.2016.7732363.

P.J., A., V.P., A. and K.P., S. (2010) ‘Kernel Method for English to Kannada Transliteration’, in *2010 International Conference on Recent Trends in Information, Telecommunication and Computing. 2010 International Conference on Recent Trends in Information, Telecommunication and Computing (ITC 2010)*, Kochi, Kerala: IEEE, pp. 336–338. doi:10.1109/ITC.2010.85.

Ranathunga, S. *et al.* (2018) ‘Si-Ta: Machine Translation of Sinhala and Tamil Official Documents’, in *2018 National Information Technology Conference (NITC). 2018 National Information Technology Conference (NITC)*, Colombo: IEEE, pp. 1–6. doi:10.1109/NITC.2018.8550069.

Sandaruwan, D., Sumathipala, S. and Fernando, S. (2021) ‘Neural Machine Translation Approach for Singlish to English Translation’, *International Journal on Advances in ICT for Emerging Regions (ICTer)*, 14(3), p. 36. doi:10.4038/icter.v14i3.7230.

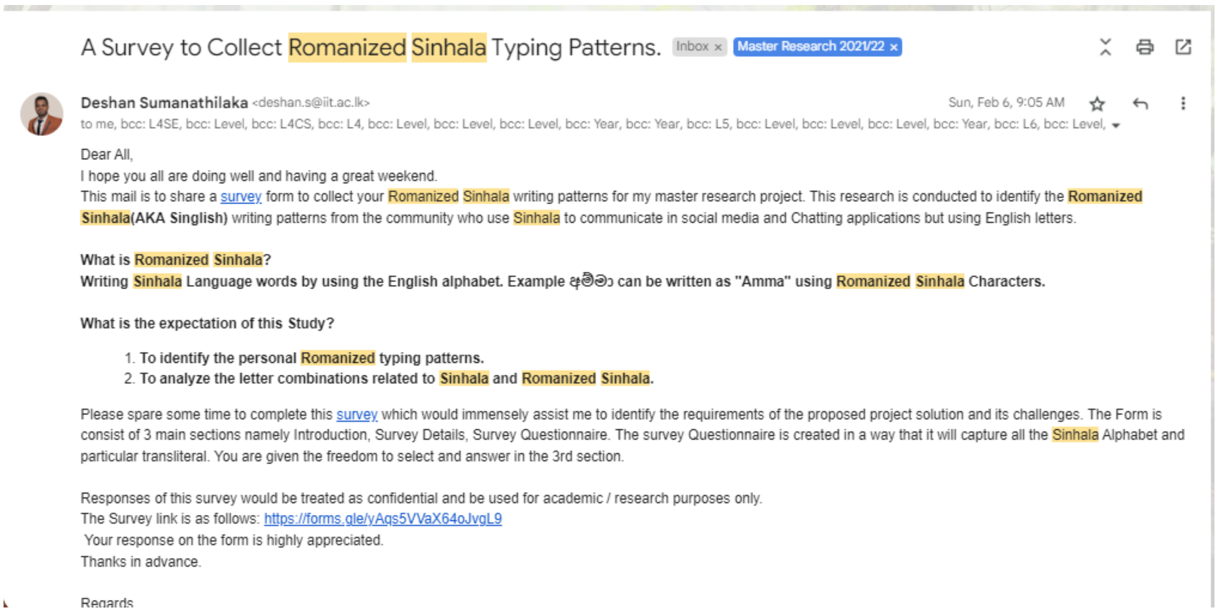
de Silva, N. (2021) ‘Survey on Publicly Available Sinhala Natural Language Processing Tools and Research’, *arXiv:1906.02358 [cs]* [Preprint]. Available at: http://arxiv.org/abs/1906.02358 (Accessed: 13 July 2021).

Vidanaralage, A.J. *et al.* (2018) ‘Sinhala Language Decoder’, in *2018 National Information Technology Conference (NITC). 2018 National Information Technology Conference (NITC)*, Colombo: IEEE, pp. 1–5. doi:10.1109/NITC.2018.8550074.

Wijerathna, L. *et al.* (2012) ‘A Translator from Sinhala to English and English to Sinhala (SEES)’, in *International Conference on Advances in ICT for Emerging Regions (ICTer2012). 2012 International Conference on Advances in ICT for Emerging Regions (ICTer)*, Colombo, Western, Sri Lanka: IEEE, pp. 14–18. doi:10.1109/ICTer.2012.6421408

Appendix

1. Form to collect the Romanized Sinhala \ Sinhala Transliterals



A Survey to Collect Romanized Sinhala Typing Patterns. Inbox × Master Research 2021/22 ×

Deshan Sumanathilaka <deshan.s@iit.ac.lk> Sun, Feb 6, 9:05 AM

to me, bcc: L4SE, bcc: Level, bcc: L4CS, bcc: L4, bcc: Level, bcc: Level, bcc: Level, bcc: Year, bcc: Year, bcc: L5, bcc: Level, bcc: Level, bcc: Level, bcc: Year, bcc: L6, bcc: Level,

Dear All,
I hope you all are doing well and having a great weekend.
This mail is to share a survey form to collect your Romanized Sinhala writing patterns for my master research project. This research is conducted to identify the **Romanized Sinhala(AKA Singlish)** writing patterns from the community who use Sinhala to communicate in social media and Chatting applications but using English letters.

What is Romanized Sinhala?
Writing Sinhala Language words by using the English alphabet. Example අම්මා can be written as "Amma" using Romanized Sinhala Characters.

What is the expectation of this Study?

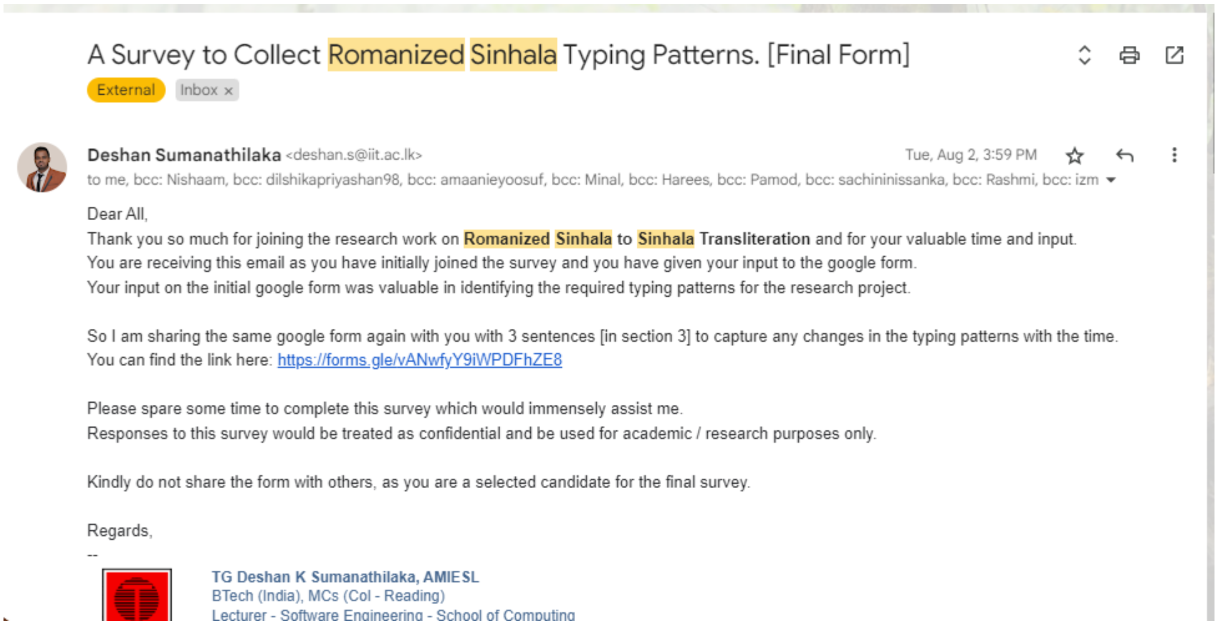
1. **To identify the personal Romanized typing patterns.**
2. **To analyze the letter combinations related to Sinhala and Romanized Sinhala.**

Please spare some time to complete this survey which would immensely assist me to identify the requirements of the proposed project solution and its challenges. The Form is consist of 3 main sections namely Introduction, Survey Details, Survey Questionnaire. The survey Questionnaire is created in a way that it will capture all the Sinhala Alphabet and particular transliteral. You are given the freedom to select and answer in the 3rd section.

Responses of this survey would be treated as confidential and be used for academic / research purposes only.
The Survey link is as follows: https://forms.gle/yAqs5VVaX64oJvgL9
Your response on the form is highly appreciated.
Thanks in advance.

Regards

2. Final form shared among the selected group



A Survey to Collect Romanized Sinhala Typing Patterns. [Final Form]

External Inbox ×

Deshan Sumanathilaka <deshan.s@iit.ac.lk> Tue, Aug 2, 3:59 PM

to me, bcc: Nishaam, bcc: dilshikapriyashan98, bcc: amaanieyoosuf, bcc: Minal, bcc: Harees, bcc: Pamod, bcc: sachininissanka, bcc: Rashmi, bcc: izm

Dear All,
Thank you so much for joining the research work on **Romanized Sinhala to Sinhala Transliteration** and for your valuable time and input.
You are receiving this email as you have initially joined the survey and you have given your input to the google form.
Your input on the initial google form was valuable in identifying the required typing patterns for the research project.

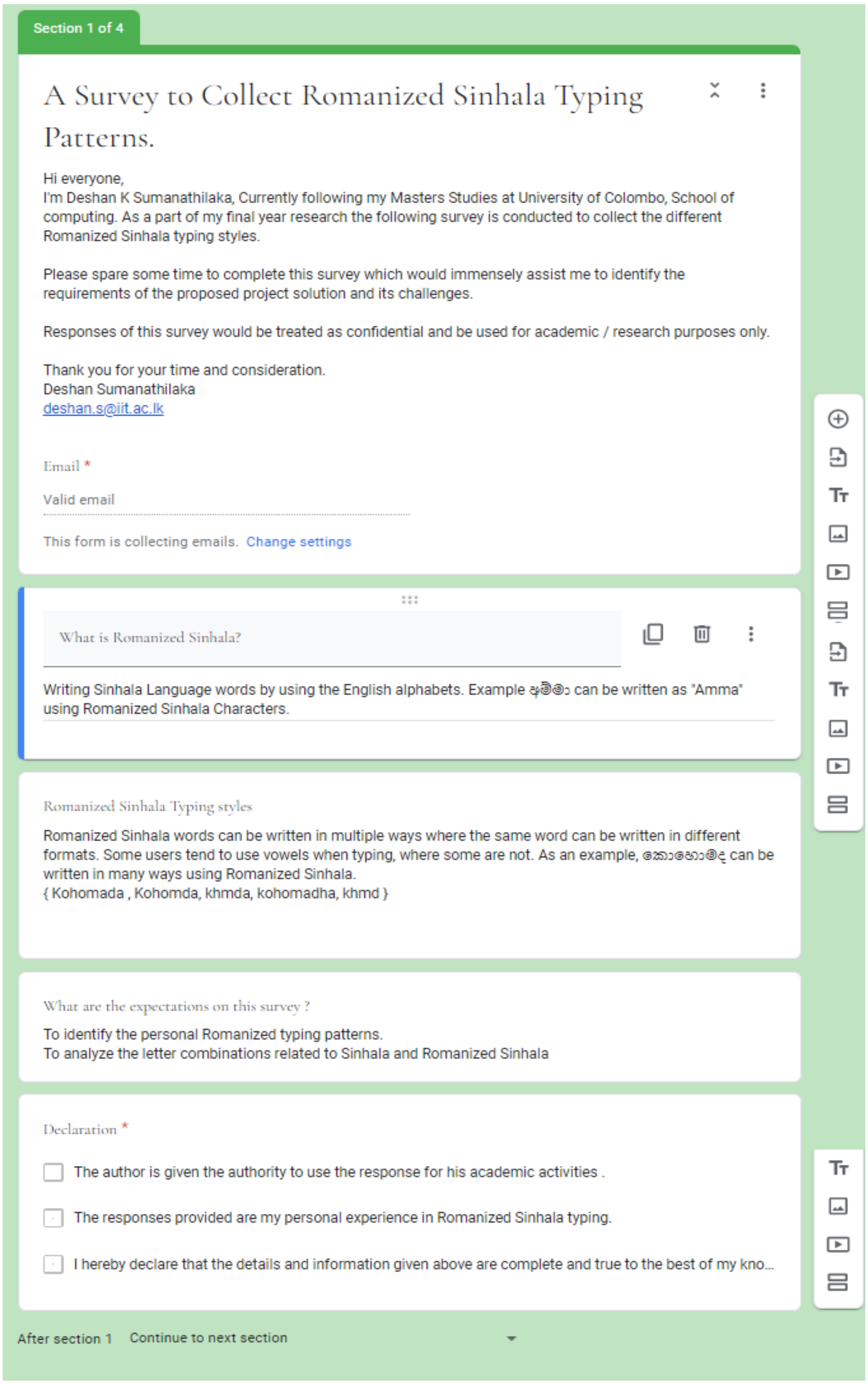
So I am sharing the same google form again with you with 3 sentences [in section 3] to capture any changes in the typing patterns with the time.
You can find the link here: https://forms.gle/vANwfyY9iWPDFhZE8

Please spare some time to complete this survey which would immensely assist me.
Responses to this survey would be treated as confidential and be used for academic / research purposes only.

Kindly do not share the form with others, as you are a selected candidate for the final survey.

Regards,
--
TG Deshan K Sumanathilaka, AMIESL
BTech (India), MCs (Col - Reading)
Lecturer - Software Engineering - School of Computing

3. Google form shared with the community



Section 1 of 4

A Survey to Collect Romanized Sinhala Typing Patterns.

Hi everyone,
I'm Deshan K Sumanathilaka, Currently following my Masters Studies at University of Colombo, School of computing. As a part of my final year research the following survey is conducted to collect the different Romanized Sinhala typing styles.

Please spare some time to complete this survey which would immensely assist me to identify the requirements of the proposed project solution and its challenges.

Responses of this survey would be treated as confidential and be used for academic / research purposes only.

Thank you for your time and consideration.
Deshan Sumanathilaka
deshan.s@iit.ac.lk

Email *

Valid email

This form is collecting emails. Change settings

What is Romanized Sinhala?

Writing Sinhala Language words by using the English alphabets. Example අම්මා can be written as "Amma" using Romanized Sinhala Characters.

Romanized Sinhala Typing styles

Romanized Sinhala words can be written in multiple ways where the same word can be written in different formats. Some users tend to use vowels when typing, where some are not. As an example, කොහොමද can be written in many ways using Romanized Sinhala.
{ Kohomada , Kohomda, khmda, kohomadha, khmd }

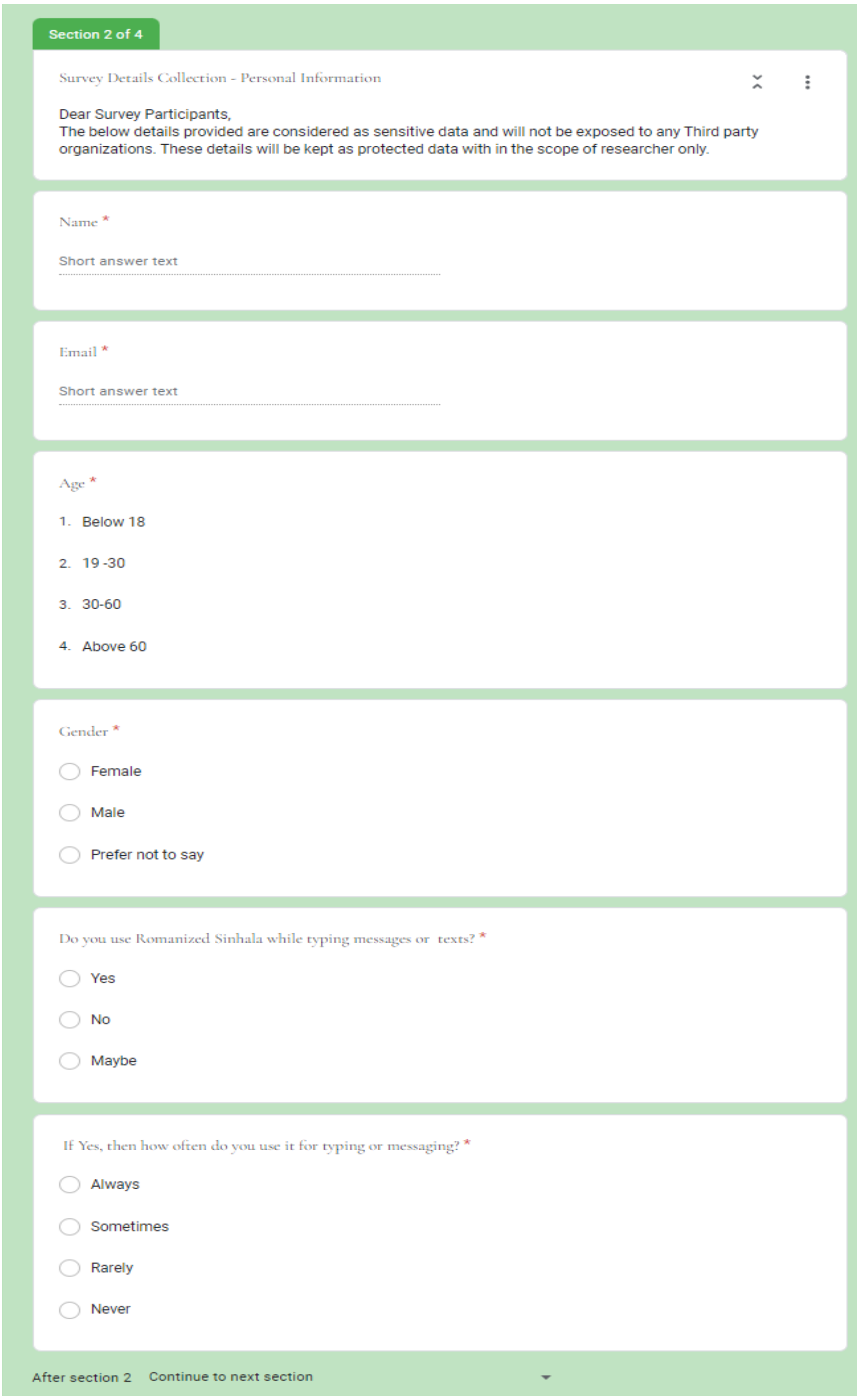
What are the expectations on this survey ?

To identify the personal Romanized typing patterns.
To analyze the letter combinations related to Sinhala and Romanized Sinhala

Declaration *

- [ ] The author is given the authority to use the response for his academic activities .
- [ ] The responses provided are my personal experience in Romanized Sinhala typing.
- [ ] I hereby declare that the details and information given above are complete and true to the best of my kno...

After section 1 Continue to next section


Section 2 of 4

Survey Details Collection - Personal Information

Dear Survey Participants,
The below details provided are considered as sensitive data and will not be exposed to any Third party organizations. These details will be kept as protected data with in the scope of researcher only.

Name *

Short answer text

Email *

Short answer text

Age *

1. Below 18
2. 19 -30
3. 30-60
4. Above 60

Gender *

◯ Female

◯ Male

◯ Prefer not to say

Do you use Romanized Sinhala while typing messages or texts? *

◯ Yes

◯ No

◯ Maybe

If Yes, then how often do you use it for typing or messaging? *

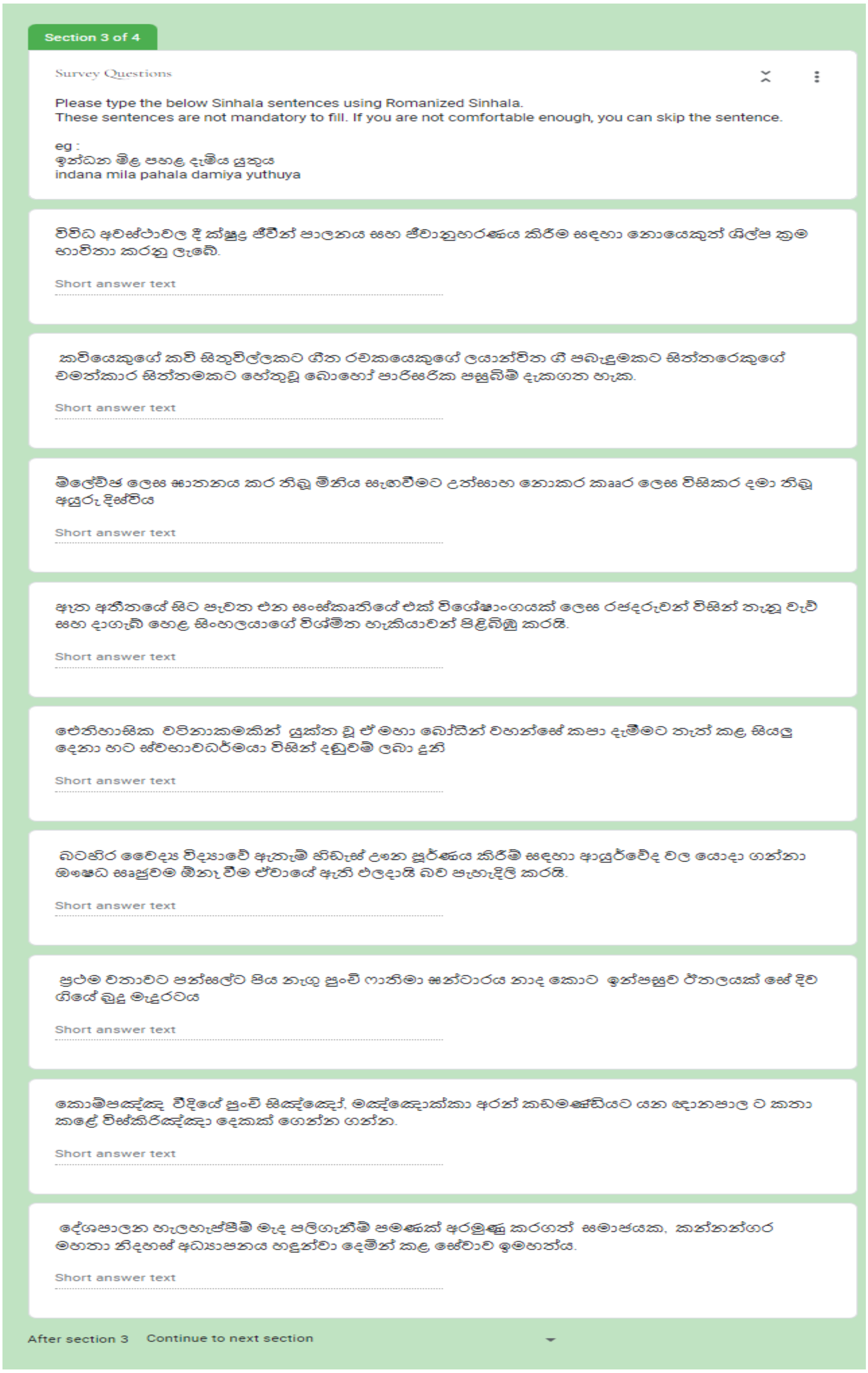
◯ Always

◯ Sometimes

◯ Rarely

◯ Never

After section 2 Continue to next section



Section 3 of 4

Survey Questions

Please type the below Sinhala sentences using Romanized Sinhala.
These sentences are not mandatory to fill. If you are not comfortable enough, you can skip the sentence.

eg :
ඉන්ධන මිළ පහළ දැමිය යුතුය
indana mila pahala damiya yuthuya

විවිධ අවස්ථාවල දී ක්ෂුද්‍ර ජීවීන් පාලනය සහ ජීවානුහරණය කිරීම සඳහා නොයෙකුත් ශිල්ප ක්‍රම භාවිතා කරනු ලැබේ.

Short answer text

කවියෙකුගේ කවි සිතුවිල්ලකට ගීත රචකයෙකුගේ ලයාන්විත ගී පබැඳුමකට සිත්තරෙකුගේ චමත්කාර සිත්තමකට හේතුවූ බොහෝ පාරිසරික පසුබිම් දැකගත හැක.

Short answer text

මිලේච්ඡ ලෙස ඝාතනය කර තිබූ මිනිය සැඟවීමට උත්සාහ නොකර කෲර ලෙස විසිකර දමා තිබූ අයුරු දිස්විය

Short answer text

ඈත අතීතයේ සිට පැවත එන සංස්කෘතියේ එක් විශේෂාංගයක් ලෙස රජදරුවන් විසින් තැනූ වැව් සහ දාගැබ් හෙළ සිංහලයාගේ විශ්මිත හැකියාවන් පිළිබිඹු කරයි.

Short answer text

ඓතිහාසික වටිනාකමකින් යුක්ත වූ ඒ මහා බෝධීන් වහන්සේ කපා දැමීමට තැත් කළ සියලු දෙනා හට ස්වභාවධර්මයා විසින් දඬුවම් ලබා දුනි

Short answer text

බටහිර වෛද්‍ය විද්‍යාවේ ඇතැම් හිඩැස් උගන පූර්ණය කිරීම් සඳහා ආයුර්වේද වල යොදා ගන්නා ඖෂධ ඍජුවම ඕනෑ වීම ඒවායේ ඇති ඵලදායී බව පැහැදිලි කරයි.

Short answer text

ප්‍රථම වතාවට පන්සල්ට පිය නැගූ පුංචි ෆාතිමා ඝන්ටාරය නාද කොට ඉන්පසුව ඊතලයක් සේ දිව ගියේ බුදු මැදුරටය

Short answer text

කොම්පඤ්ඤ වීදියේ පුංචි සිඤ්ඤෝ, මඤ්ඤොක්කා අරන් කඩමණ්ඩියට යන ඥානපාල ට කතා කළේ විස්කිරිඤ්ඤා දෙකක් ගෙන්න ගන්න.

Short answer text

දේශපාලන හැලහැප්පීම් මැද පලිගැනීම් පමණක් අරමුණු කරගත් සමාජයක, කන්නන්ගර මහතා නිදහස් අධ්‍යාපනය හඳුන්වා දෙමින් කළ සේවාව ඉමහත්ය.

Short answer text

After section 3 Continue to next section

Extended Abstract: Romanized Sinhala to Sinhala Transliteration using a Hybrid Approach

Sumanathilaka T.G.D.K.
Informatics Institute of Technology, Ramakrishna Road, Colombo 006, Sri Lanka
deshan.s@iit.ac.lk

Ruvan Weerasinghe
University of Colombo School of Computing, 35, Reid Avenue, Colombo 007, Sri Lanka
arw@ucsc.cmb.ac.lk

Priyadarshana YHPP
Informatics Institute of Technology, Ramakrishna Road, Colombo 006, Sri Lanka
prasan.y@iit.ac.lk

***Abstract*—The advent of Web 2.0 with the introduction of social media platforms and instant messaging coupled with support for native languages has caused a global communication revolution. With the commencement of multi-language capability in the digital world, Sri Lankans switched to using both native Sinhala and Romanized Sinhala for their day-to-day typing. However, typing "Romanized Sinhala" using ad hoc transliterations and shortened net acronyms has resulted in its own set of problems. Chief among them is the ambiguity posed by such transliteration which results in less accurate and time-consuming deciphering. This study aims to introduce a novel reverse transliterator that can back transliterate Romanized Sinhala to Sinhala using Statistical and Neural Based approaches.**

Keywords—Romanized Sinhala, Transliteration, Neural Approach, Statistical approach, Sinhala, Typing

I. Introduction

According to the Oxford Dictionary, the word transliteration can be defined as "*the act of* ***writing words or letters using letters of a different alphabet or language.*** [6] The word transliteration comes from the Latin word "*transliteratus*" which explains "*trans*" as across and "*Littera*" *as* a letter. Transliteration helps people to pronounce words and names in foreign languages. As an example, the Sinhala word "*අම්මා*" is represented using English as "Amma". Unlike the translation, which tells the user about the meaning of a particular word, the transliteration only provides an idea of how the word is pronounced by using a familiar language. The reverse transliteration schema can transliterate a word or a sentence of a particular language expressed in a different language to the original language schema. As an example, the Sinhala word *කොහොමද*" which is expressed as Roman English "kohomdha" can be back transliterated to the original word "*කොහොමද*" using the Reverse transliteration techniques. In this study, the reverse transliteration schema will be modeled in a way that it can be used for different typing patterns where a Romanized Sinhala word expressed in short typing aka without vowels can be reverse transliterated to the original Sinhala word. The Sinhala word "*කොහොමද*" can be expressed as "*kohomadha, kohomada, kohomda, khmda, khmd*" where different patterns of interpreting the same word are captured in the study before transliterating it back to the original language schema. As Sinhala is a low resource language[1], collecting the transliteral required for the study is challenging. So, the study will mainly focus on social media for collecting transliterals and code-mixed words will not be handled. Collected transliterals will be used to train the model required for the reverse transliteration process which mainly focuses on Statistical and Neural Approaches. According to the existing literature, Liwera and Ranathunga[2] used the trigram model along with the ruled-based approach to build a model for reverse transliteration, where the model is mainly trained on social media hate speech. According to the study of Liwera, the prediction process uses Trigram, Bi-gram, and Uni-gram taggers and missing transliterals will be transliterated using a rule-based approach. But according to the study ambiguity of different typing schema is not handled in the study where 0.62-word level and 0.77 letter level accuracy has been achieved. The Neural language translator for Singlish to English modeled by Sandaruwan [3] has used seq2seq neural machine transliteration which is purely based on attention mechanism. This study is mainly focused on deep multi-layer RNN which consists of bidirectional LSTM as recurrent units. The encoder RNN consumes the input resource words whereas the decoder RNN process the target word. The above-proposed architecture of [3] has achieved a 24.13 BLEU score on Singlish to English by seeing approximately 0.26M parallel sequence pairs with 50 K+ word vocabulary. The Native Sinhala studies have been performed in past few years by many researchers to enhance the Rule-based approach which is used for transliteration process but still, these proposed models are unable to handle different typing patterns of Romanized Sinhala where the reverse transliteration schema is not accurate. So, Considering the above limitation, the authors of this work will propose a novel model which can bridge the existing gap in the next chapter.

II. Methodology

A. Dataset Collection.

The transliterals required for the study to train the models are collected through different social media platforms. Mainly the YouTube comments in Romanized Sinhala were extracted and transliterated to respective Sinhala Words. The Transliterals used in the study of Liwera[2] and the Dakshina dataset [5] are combinedly used to create an unbiased dataset for the study.

B. *Identifiyng the ad hoc Transliteration schemas*

A survey was conducted to capture the different typing patterns of Romanized Sinhala. Diverse communities were selected based on gender and age groups for the survey where the inputs collected were analyzed to identify writing patterns. The identified writing styles and short typing styles will be used in the preprocessing steps before feeding them to the neural model.

C. *Pre Processing*

The initial processing of the dataset has been performed to build the transliterals required for the study. The data tuples are formatted as follows.

TABLE I ROMANIZED SINHALA AND SINHALA DATA TUPLE

Romanized Sinhala	*mama dakapu thawath nihathamani soduru minisek !*
Sinhala	*මම දැකපු තවත් නිහතමානී සොඳුරු මිනිසෙක් !*

The formatted data tuples were examined and punctuation, emojis, and non- Romanized words were removed. The cleaned dataset was used to build the transliterals. The build transliterals from the above sentence are as follows.

TABLE Ii PROCESSED TRANSLITERALS

mama/මම	dakapu/දැකපු	*thawath/තවත්*
nihathamani/නිහතමානී	*soduru/සොඳුරු*	*minisek/මිනිසෙක්*

Once the transliterates are generated from the dataset duplicate removal and changing to lowercase was performed on the generated transliterals to use in the N-gram tagger.

D. *Reverse Transliteration Model*

The reverse transliteral model is consisting of 3 phases.

1) The Statistical approach with N-gram tagger.

The Romanized Sinhala sentence entered by the user will be tokenized. Each token will be checked for Trigram, Bigram, and Unigrm taggers where if transliteration is available it will be appended with the final list where the reverse transliterated Sinhala sentence will be maintained. If the expected transliteral is not found the word "NONE" will be appended with the sentence.

2) The Neural Approach

The tokens which consist of the word "NONE" after the initial process will be examined in this chapter. Each Romanized Sinhala word related to the token will be preprocessed using the pattern identified in the ad hoc transliteration schema and fed to the encoder of the Neural model. The expected output will be Sinhala word(s) related to the Romanized Sinhala token from the decoder. If the Neural approach suggests more than one transliteration related to the Inputted Romanized Sinhala token, the suggestions will be analyzed in the finalizing phase.

3) The Transliterating Finalizing Phase

In this phase, if the neural model predicts more than one transliteral related to the transliteration, the respective words will be checked for the best suitability. The selection will be based on a knowledge-based approach where the most suitable word will be suggested to the user in the suitability order. This phase will tackle the ambiguity-related issues in the transliteration process.

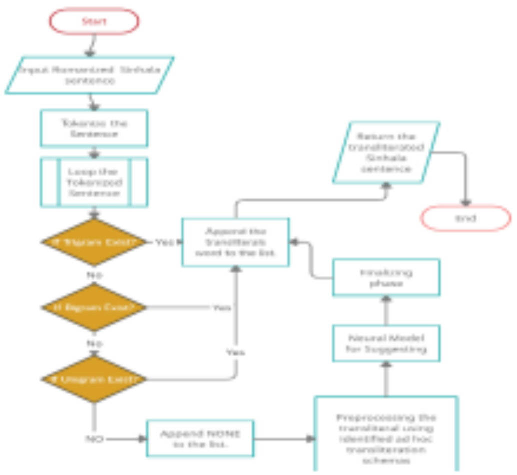

Start
Input Romanized Sinhala sentence
Tokenize the Sentence
Loop the Tokenized Sentence
If Trigram Exist?
Yes
No
If Bigram Exist?
Yes
No
If Unigram Exist?
Yes
NO
Append the transliterals word to the list
Append NONE to the list.
Preprocessing the transliteral using identified ad hoc transliteration schemas
Neural Model for Suggesting
Finalizing phase
Return the transliterated Sinhala sentence
End

FIGURE I PROPOSED ARCHITECTURE OF THE SYSTEM

The finalized sentence which is in Sinhala will be output to the user. A simple UI will be used for the Data input and output process. So, the above Novel approach can be used in the Reverse transliteration process of Romanized Sinhala to Sinhala where it can be adapted with a future keyboard.

E. *Evaluation*

The model will be evaluated using both qualitative and quantitative approaches.

1) Word Level and Sentence level accuracy

The model will be evaluated using word-level accuracy for predicting and suggesting. The Sentence level accuracy will be checked for predicting.

2) Qualitative Evaluation

The evaluation of the system will be carried out with the users who are unfamiliar with the Sinhala keyboard. The users will be given access to the system where different statements will be evaluated based on different Romanized typing patterns.

III. CONCLUSION AND FUTURE WORKS

The proposed architecture can enhance the experience of Romanized Sinhala in texting. The ambiguity issue on the transliteration process will be addressed in the finalizing phase of the model where the model can adapt to a Romanized Sinhala – Sinhala Keyboard as a future implementation.

REFERENCES

[1] A. J. Vidanaralage, A. U. Illangakoon, S. Y. Sumanaweera, C. Pavithra, and S. Thelijjagoda, "Sinhala Language Decoder," in *2018 National Information Technology Conference (NITC)*, Colombo, Oct. 2018, pp. 1–5. doi: 10.1109/NITC.2018.8550074.

[2] W. M. P. Liwera and L. Ranathunga, "Combination of Trigram and Rule-based Model for Singlish to Sinhala Transliteration by Focusing Social Media Text," in *2020 From Innovation to Impact (FITI)*, Colombo, Sri Lanka, Dec. 2020, pp. 1–5. doi: 10.1109/FITI52050.2020.9424880.

5th

Ad-hoc Reverse Transliteration schema for Romanized Sinhala to Sinhala

Sumanathilaka TGDK[1] Priyadarshana YHPP[2] Ruvan Weerasinghe[3]

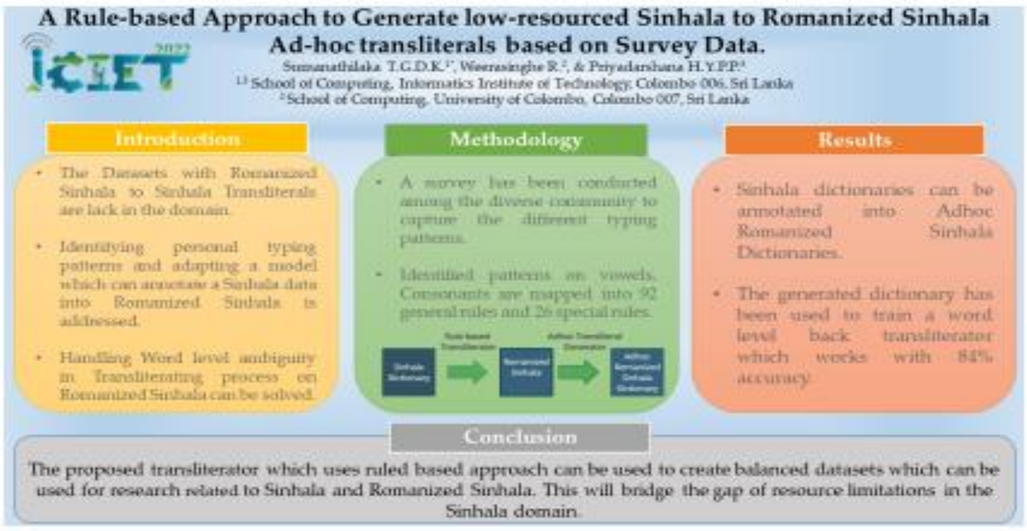
[1] Informatics Institute of Technology, Colombo 6, Sri Lanka, deshan.s@iit.ac.lk
[2] Informatics Institute of Technology, Colombo 6, Sri Lanka, prasan.y@iit.ac.lk
[3] University of Colombo School of Computing, Colombo 7, Sri Lanka, arw@ucsc.cmb.ac.lk

Keywords: Romanized Sinhala, Transliteration, Neural Approach, Statistical approach, Sinhala, Typing

With the revolution of social technology, the introduction of social media platforms and instant messages strengthen the native language used for communication in electronic media. With the commencement of multi-language compatibility in the digital arena both native Sinhala and Romanized Sinhala became prominent among the general community. Machine transliteration provides the ability to transliterate the alphabet of one language to another using computational approaches. The informal shorthand language that uses in texting known as "Singlish" makes texting easier as the words in Sinhala can be interpreted using English letters with different typing patterns. But typing "Romanized Sinhala" using ad hoc transliterations and short net acronyms and getting the expected output in native Sinhala is less accurate. The current transliterators with a rule-based approach use a letter-level transliteration with a defined rule for the transliteration schema. But Singlish via shortened hand-based typing is not compatible with the current system. The proposed ad-hoc schema uses multiple computational approaches Aka Hybrid Approach to accomplish the requirement of ambiguity-free transliteration. The statistical approach used in the first phase uses an N-gram tagger where the tokens are fed to Trigram, Bigram, and Unigram taggers respectively. The unknown token from the initial phase is fed to the second phase with the Neural model which will predict respective words. The neural model is based on the pattern identified through the survey which was conducted over the diverse community. The third phase which is the finalizing phase uses a knowledge-based model to find the most suitable word from the predicted words pool. This phase will solve the ambiguity of a word selection. Therefore, the proposed novel transliterator which can back transliterate Romanized Sinhala to Sinhala using the Hybrid approach can use to enhance the reverse transliteration schema which will escalate the usage of Native Sinhala for the communication.

Title: A Rule-based Approach to Generate low-resourced Sinhala to Romanized Sinhala Ad-hoc Transliterals based on Survey Data.

Sumanathilaka T.G.D.K.[1*], Weerasinghe R.[2], & Priyadarshana H.Y.P.P.[3]
[1,3] School of Computing, Informatics Institute of Technology, Colombo 006, Sri Lanka
[2] School of Computing, University of Colombo, Colombo 007, Sri Lanka
Email correspondence: deshan.s@iit.ac.lk



A Rule-based Approach to Generate low-resourced Sinhala to Romanized Sinhala Ad-hoc transliterals based on Survey Data.

Sumanathilaka T.G.D.K.[1*], Weerasinghe R.[2], & Priyadarshana H.Y.P.P.[3]
[1,3] School of Computing, Informatics Institute of Technology, Colombo 006, Sri Lanka
[2] School of Computing, University of Colombo, Colombo 007, Sri Lanka

Introduction

- The Datasets with Romanized Sinhala to Sinhala Transliterals are lack in the domain.
- Identifying personal typing patterns and adapting a model which can annotate a Sinhala data into Romanized Sinhala is addressed.
- Handling Word level ambiguity in Transliterating process on Romanized Sinhala can be solved.

Methodology

- A survey has been conducted among the diverse community to capture the different typing patterns.
- Identified patterns on vowels, Consonants are mapped into 92 general rules and 26 special rules.

Sinhala Dictionary
Rule based Transliterator
Romanized Sinhala
Adhoc Transliteral Generator
Adhoc Romanized Sinhala Dictionary

Results

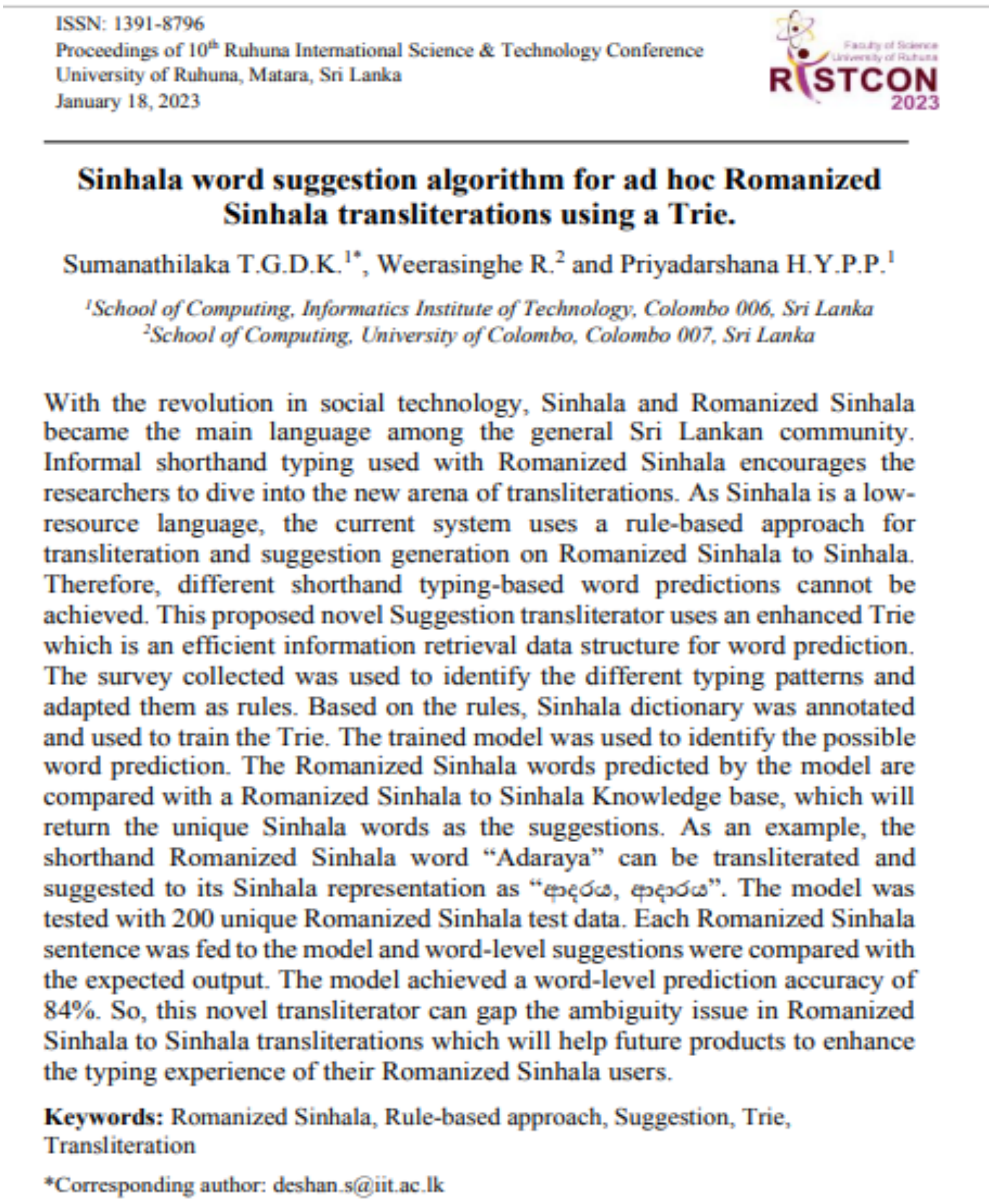
- Sinhala dictionaries can be annotated into Adhoc Romanized Sinhala Dictionaries.
- The generated dictionary has been used to train a word level back transliterator which works with 84% accuracy.

Conclusion

The proposed transliterator which uses ruled based approach can be used to create balanced datasets which can be used for research related to Sinhala and Romanized Sinhala. This will bridge the gap of resource limitations in the Sinhala domain.

With the social and technological revolution, the usage of social media platforms and instant message services strengthens native language compatibility.The Sinhala and the Romanized Sinhala became prominent among typing languages among the general Sri Lankan community which informal short-hand-based typing and short net acronyms were used for easier typing. But the availability of resources for the Sinhala language and Romanized Sinhala for research purposes is comparatively less and it a low-resourced language. To gap the resource limitation issue, the proposed Transliterator which uses a rule-based approach can be used to annotate Sinhala words in dictionaries to Romanized Sinhala words containing Ad hoc typing patterns. The survey was conducted among diverse communities based on the stratified sampling method which focuses on age, gender, and language fluency. An online survey with 12 Sinhala sentences were given to users, which will capture the transliteration schemas required for the annotation process. Based on the input received, the Adhoc Romanized Sinhala typing patterns were identified and the 92 general rules and 26 special rules required for the transliteration were generated. The vowel patterns, consonants with vowels and character-based transliteration patterns were extracted from the rules. Based on the identified rules, the Sinhala dictionaries were annotated to the Romanized Sinhala Dictionary which contains different Romanized Sinhala transliterals. As an example, the Sinhala word **කොහොමද** was transliterated to "**khomada, kohmada, kohomda, kohomad, khmada, khomda, khomad, kohmda, kohmad, kohomd, khmda, khmad, khomd, kohmd, khmd**" using the identified rules. The annotated dictionary was tested with back Transliterator which works with 84% accuracy. Therefore, this proposed novel Transliteration annotator can gap the resource limitation issue in Sinhala to Romanized Sinhala transliterals which will help future researchers in the Sinhala domain to escalate the efficiency and the accuracy of their research works in the native language.

Keywords: Romanized Sinhala, Rule-based approach, Survey, Transliterals



ISSN: 1391-8796
Proceedings of 10th Ruhuna International Science & Technology Conference
University of Ruhuna, Matara, Sri Lanka
January 18, 2023

Sinhala word suggestion algorithm for ad hoc Romanized Sinhala transliterations using a Trie.

Sumanathilaka T.G.D.K.[1*], Weerasinghe R.[2] and Priyadarshana H.Y.P.P.[1]

[1]School of Computing, Informatics Institute of Technology, Colombo 006, Sri Lanka
[2]School of Computing, University of Colombo, Colombo 007, Sri Lanka

With the revolution in social technology, Sinhala and Romanized Sinhala became the main language among the general Sri Lankan community. Informal shorthand typing used with Romanized Sinhala encourages the researchers to dive into the new arena of transliterations. As Sinhala is a low-resource language, the current system uses a rule-based approach for transliteration and suggestion generation on Romanized Sinhala to Sinhala. Therefore, different shorthand typing-based word predictions cannot be achieved. This proposed novel Suggestion transliterator uses an enhanced Trie which is an efficient information retrieval data structure for word prediction. The survey collected was used to identify the different typing patterns and adapted them as rules. Based on the rules, Sinhala dictionary was annotated and used to train the Trie. The trained model was used to identify the possible word prediction. The Romanized Sinhala words predicted by the model are compared with a Romanized Sinhala to Sinhala Knowledge base, which will return the unique Sinhala words as the suggestions. As an example, the shorthand Romanized Sinhala word "Adaraya" can be transliterated and suggested to its Sinhala representation as "ආදරය, ආදාරය". The model was tested with 200 unique Romanized Sinhala test data. Each Romanized Sinhala sentence was fed to the model and word-level suggestions were compared with the expected output. The model achieved a word-level prediction accuracy of 84%. So, this novel transliterator can gap the ambiguity issue in Romanized Sinhala to Sinhala transliterations which will help future products to enhance the typing experience of their Romanized Sinhala users.

Keywords: Romanized Sinhala, Rule-based approach, Suggestion, Trie, Transliteration

*Corresponding author: deshan.s@iit.ac.lk

Swa-Bhasha: Romanized Sinhala to Sinhala Reverse Transliteration using a Hybrid Approach

T.G.D.K. Sumanathilaka
School of Computing,
Informatics Institute of Technology
Colombo 006, Sri Lanka
deshankoshala@gmail.com

Ruvan Weerasinghe
School of Computing,
University of Colombo
Colombo 007, Sri Lanka
arw@ucsc.cmb.ac.lk

Y.H.P.P. Priyadarshana
School of Computing,
Informatics Institute of Technology
Colombo 006, Sri Lanka
toprasanyapa@gmail.com

***Abstract*— With the social and technological revolution, the usage of social media platforms and instant message services strengthens native language compatibility in the digital arena. The Sinhala and the Romanized Sinhala became the prominent typing languages among the general Sri Lankan community. Informal short-hand-based typing and short net acronyms were used for easier Sinhala typing. But Typing Romanized Sinhala using ad-hoc transliterations and getting the expected output in native Sinhala is less accurate and time-consuming. Therefore, this study aims to introduce a novel reverse transliterator which can back transliterate and suggest Romanized Sinhala to Sinhala words. The Transliterator has been modelled using the Statistical approach with Trigram and Rule-based model for back transliteration purposes and Knowledge-based with a Trie data structure for suggesting purposes. The proposed solution is capable of transliterating both formal and informal shorthand Romanized Sinhala. This hybrid model used in the study is capable of efficient transliteration with the word level accuracy of 0.84. This proposed model can be used in digital platforms to enhance the usability of native Sinhala communication in a much more efficient way.**

Keywords—Romanized Sinhala, Transliteration, Tri-gram, Rule-based, Prediction, Suggestion

Introduction

According to the Oxford Dictionary, the word transliteration [1] means "*the act of* ***writing words or letters using letters of a different alphabet or language.*** The word transliteration comes from the Latin word "*transliteratus"* which explains "*trans"* as across and "*Littera" as* a letter. The transliterations help non-native speakers to pronounce a native word by its Transliteration. As an example, the Sinhala word *"අම්මා"* is represented using English (Latin) as "Amma". Unlike translation, which tells the user about the meaning of a particular word, the transliteration only provides an idea of how the word is pronounced by using a familiar language. The reverse transliteration schema can transliterate a word, or a sentence of a particular language expressed in a different language to the original language schema[2]. As an example, the Sinhala word "*කොහොමද"* which expressed as Roman English "kohomda" can be back transliterated to the original word "*කොහොමද"* using the Reverse transliteration techniques. In this study, the reverse transliteration schema will be modelled in a way that it can be used for different typing patterns where a Romanized Sinhala word expressed in short typing aka without vowels can be reverse transliterated to the original Sinhala word. The Sinhala word "*කොහොමද"* can be expressed as "*kohomada, kohomada, kohomda, khmda, khmd"* where different patterns of interpreting the same word are captured in the study before transliterating it back to the original language schema. As Sinhala is a low resource language [3], collecting the transliteral required for the study is challenging. So, the study was mainly focused on the publicly available dataset and social media comments for collecting transliterals. The different typing patterns were captured through the online survey and the results were analyzed to identify the different transliterations schemas. Collected transliterals were used to train the hybrid model required for the reverse transliteration process which mainly focused on statistical, Rule based and knowledge base. The Trie structure, a well-known architecture for efficient information retrieval has been used with the knowledge base to perform the suggestion-level Transliteration to handle the ambiguity in the transliteration process.

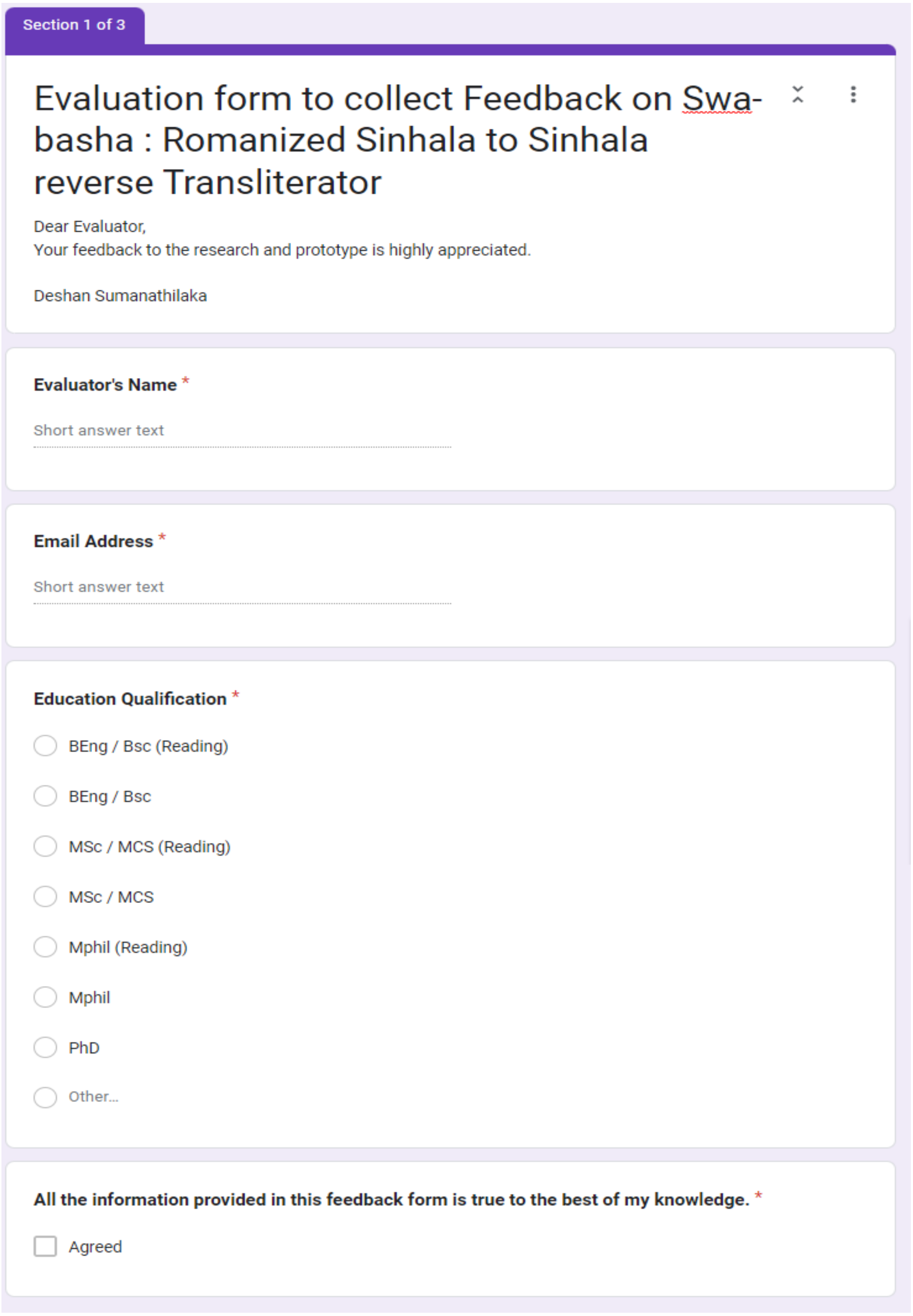
Related Works

Many countries have adapted machine translation algorithms to their languages. Many of the translation and transliteration processes have taken place in the literature due to the language barrier between native speakers and non-native speakers. The below section will discuss the Sinhala-based machine Translation and non-Sinhala translation techniques.

Rule-based Approach

This is a common machine translation approach used in machine transliteration. The main base of this approach is to implement a system under well-defined rules that are related to the source language and the related language. This approach can be classified based on three schemas namely Direct machine transliteration, transfer based and interlingual Machine translation. Currently, many commercial ....

Evaluation form:



Section 1 of 3

Evaluation form to collect Feedback on Swa-basha : Romanized Sinhala to Sinhala reverse Transliterator

Dear Evaluator,
Your feedback to the research and prototype is highly appreciated.

Deshan Sumanathilaka

Evaluator's Name *

Short answer text

Email Address *

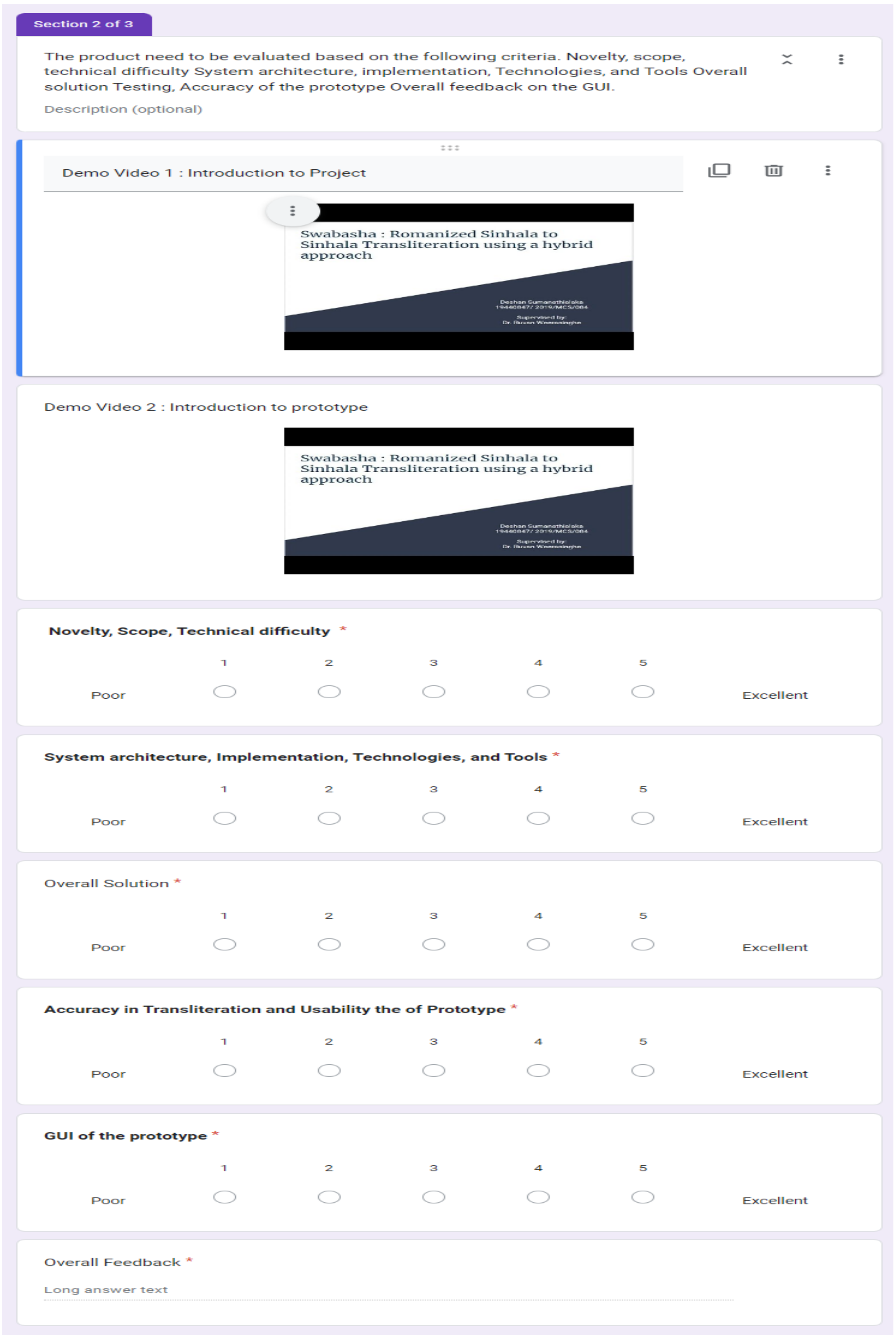
Short answer text

Education Qualification *

- ◯ BEng / Bsc (Reading)
- ◯ BEng / Bsc
- ◯ MSc / MCS (Reading)
- ◯ MSc / MCS
- ◯ Mphil (Reading)
- ◯ Mphil
- ◯ PhD
- ◯ Other...

All the information provided in this feedback form is true to the best of my knowledge. *

☐ Agreed



Section 2 of 3

The product need to be evaluated based on the following criteria. Novelty, scope, technical difficulty System architecture, implementation, Technologies, and Tools Overall solution Testing, Accuracy of the prototype Overall feedback on the GUI.

Description (optional)

Demo Video 1 : Introduction to Project

Swabasha : Romanized Sinhala to Sinhala Transliteration using a hybrid approach

Deshan Sumanathilaka
19440847/ 2019/MCS/084
Supervised by:
Dr. Ruvan Weerasinghe

Demo Video 2 : Introduction to prototype

Swabasha : Romanized Sinhala to Sinhala Transliteration using a hybrid approach

Deshan Sumanathilaka
19440847/ 2019/MCS/084
Supervised by:
Dr. Ruvan Weerasinghe

Novelty, Scope, Technical difficulty *

	1	2	3	4	5	
Poor	◯	◯	◯	◯	◯	Excellent

System architecture, Implementation, Technologies, and Tools *

	1	2	3	4	5	
Poor	◯	◯	◯	◯	◯	Excellent

Overall Solution *

	1	2	3	4	5	
Poor	◯	◯	◯	◯	◯	Excellent

Accuracy in Transliteration and Usability the of Prototype *

	1	2	3	4	5	
Poor	◯	◯	◯	◯	◯	Excellent

GUI of the prototype *

	1	2	3	4	5	
Poor	◯	◯	◯	◯	◯	Excellent

Overall Feedback *

Long answer text

4. Rule based rule

```
def initializeVar():
    vowelsUni.append("ඌ")
    vowels.append("oo")
    vowelModifiersUni.append("ූ")
    vowelsUni.append("ඕ")
    vowels.append("o\\)")
    vowelModifiersUni.append("ෝ)
    vowelsUni.append("ඕ")
    vowels.append("oe")
    vowelModifiersUni.append("ෝ)
    vowelsUni.append("ආ")
    vowels.append("aa")
    vowelModifiersUni.append("ා)
    vowelsUni.append("ආ")
    vowels.append("a\\)")
    vowelModifiersUni.append("ා)
    vowelsUni.append("ඈ")
    vowels.append("Aa")
    vowelModifiersUni.append("ෑ)
    vowelsUni.append("ඈ")
    vowels.append("A\\)")
    vowelModifiersUni.append("ෑ)
    vowelsUni.append("ඈ")
    vowels.append("ae")
    vowelModifiersUni.append("ෑ)
    vowelsUni.append("ඊ")
    vowels.append("ii")
    vowelModifiersUni.append("ී")
    vowelsUni.append("ඊ")
    vowels.append("i\\)")
    vowelModifiersUni.append("ී")
    vowelsUni.append("ඊ")
    vowels.append("ie")
    vowelModifiersUni.append("ී")
    vowelsUni.append("ඊ")
    vowels.append("ee")
    vowelModifiersUni.append("ී")
    vowelsUni.append("ඒ")
    vowels.append("ea")
    vowelModifiersUni.append("ේ")
    vowelsUni.append("ඒ")
    vowels.append("e\\)")
    vowelModifiersUni.append("ේ")
    vowelsUni.append("ඒ")
    vowels.append("ei")
    vowelModifiersUni.append("ේ")
    vowelsUni.append("ඌ")
    vowels.append("uu")
    vowelModifiersUni.append("ූ")
    vowelsUni.append("ඌ")
    vowels.append("u\\)")
    vowelModifiersUni.append("ූ")
    vowelsUni.append("ඖ")
    vowels.append("au")
    vowelModifiersUni.append("ෞ)
    vowelsUni.append("ඇ")
    vowels.append("\\a")
    vowelModifiersUni.append("ැ)
    vowelsUni.append("අ")
    vowels.append("a")
    vowelModifiersUni.append("")
    vowelsUni.append("ඇ")
    vowels.append("A")
    vowelModifiersUni.append("ැ)
    vowelsUni.append("ඉ")
    vowels.append("i")
    vowelModifiersUni.append("ි")
    vowelsUni.append("එ")
    vowels.append("e")
    vowelModifiersUni.append("ෙ")
    vowelsUni.append("උ")
    vowels.append("u")
    vowelModifiersUni.append("ු")
    vowelsUni.append("ඔ")
    vowels.append("o")
    vowelModifiersUni.append("ො)
    vowelsUni.append("ඓ")
    vowels.append("I")
    vowelModifiersUni.append("ෛ")
    specialConsonantsUni.append("ං")
    specialConsonants.append("\\n")
```

```
specialConsonantsUni.append("ඃ")
specialConsonants.append("\\h")
specialConsonantsUni.append("ඞ")
specialConsonants.append("\\N")
specialConsonantsUni.append("ඍ")
specialConsonants.append("\\R")
# special character Repaya
specialConsonantsUni.append("ර්‍")
specialConsonants.append("R")
specialConsonantsUni.append("ර්‍")
specialConsonants.append("\\r")
consonantsUni.append("ඬ")
consonants.append("nnd")
consonantsUni.append("ඳ")
consonants.append("nndh")
consonantsUni.append("ඟ")
consonants.append("nng")
consonantsUni.append("ථ")
consonants.append("th")
consonantsUni.append("ධ")
consonants.append("dh")
consonantsUni.append("ඝ")
consonants.append("gh")
consonantsUni.append("ඡ")
consonants.append("ch")
consonantsUni.append("ඵ")
consonants.append("ph")
consonantsUni.append("භ")
consonants.append("bh")
consonantsUni.append("ඣ")
consonants.append("jh")
consonantsUni.append("ෂ")
consonants.append("sh")
consonantsUni.append("ඥ")
consonants.append("GN")
consonantsUni.append("ඤ")
consonants.append("KN")
consonantsUni.append("ළු")
consonants.append("Lu")
consonantsUni.append("ඛ")
consonants.append("kh")
consonantsUni.append("ඨ")
consonants.append("Th")
consonantsUni.append("ඪ")
consonants.append("Dh")
consonantsUni.append("ශ")
consonants.append("S")
consonantsUni.append("ද")
consonants.append("d")
consonantsUni.append("ච")
consonants.append("c")
consonantsUni.append("ත")
consonants.append("th")
consonantsUni.append("ට")
consonants.append("t")
consonantsUni.append("ක")
consonants.append("k")
consonantsUni.append("ඩ")
consonants.append("D")
consonantsUni.append("න")
consonants.append("n")
consonantsUni.append("ප")
consonants.append("p")
consonantsUni.append("බ")
consonants.append("b")
consonantsUni.append("ම")
consonants.append("m")
consonantsUni.append("ය")
consonants.append("Y")
consonantsUni.append("ය")
consonants.append("y")
consonantsUni.append("ජ")
consonants.append("j")
consonantsUni.append("ල")
consonants.append("l")
consonantsUni.append("ව")
consonants.append("v")
consonantsUni.append("ව")
consonants.append("w")
consonantsUni.append("ස")
consonants.append("s")
consonantsUni.append("හ")
consonants.append("h")
consonantsUni.append("ණ")
consonants.append("N")
consonantsUni.append("ළ")
```

```
consonants.append("L")
consonantsUni.append("ළ")
consonants.append("K")
consonantsUni.append("ඣ")
consonants.append("G")
consonantsUni.append("ඝ")
consonants.append("P")
consonantsUni.append("ඵ")
consonants.append("B")
consonantsUni.append("ඹ")
consonants.append("f")
consonantsUni.append("ෆ")
consonants.append("g")
consonantsUni.append("ග")
consonants.append("r")
specialCharUni.append("◌ෲ)
specialChar.append("ruu")
specialCharUni.append("◌ෘ)
specialChar.append("ru")
```