ENHANCING CONVERSATIONAL AI MODEL PERFORMANCE AND EXPLAINABILITY FOR SINHALA-ENGLISH BILINGUAL SPEAKERS

2022-056

Project Proposal Report

Jayasinghe D.T.
(Dissanayake D.M.I.M., Hameed M.S., Sakalasooriya S.A.H.A.)

B.Sc. (Hons) Degree in Information Technology Specialising in Data Science

Department of Information Technology

Sri Lanka Institute of Information Technology
Sri Lanka

January 2022

ENHANCING CONVERSATIONAL AI MODEL PERFORMANCE AND EXPLAINABILITY FOR SINHALA-ENGLISH BILINGUAL SPEAKERS

2022-056

Project Proposal Report

B.Sc. (Hons) Degree in Information Technology Specialising in Data Science

Department of Information Technology

Sri Lanka Institute of Information Technology
Sri Lanka

January 2022

DECLARATION, COPYRIGHT STATEMENT AND THE STATEMENT OF THE SUPERVISORS

We declare that this is our own work, and this proposal does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any other university or Institute of higher learning and to the best of our knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Name	**Student ID**	**Signature**
Dissanayake D.M.I.M.	IT19069432	
Hameed M.S.	IT19064932	
Jayasinghe D.T.	IT19075754	
Sakalasooriya S.A.H.A.	IT19051208	

The above candidates are carrying out research for the undergraduate Dissertation under my supervision.

Name of the supervisor: Dr. Lakmini Abeywardhana
Signature of the supervisor: Date: 11/02/2022

Name of the co-supervisor: Ms. Dinuka Wijendra
Signature of the co-supervisor: Date: 11/02/2022

ABSTRACT

Chatbots are a trending topic nowadays. Due to the recent advancement in chatbots, many industries and businesses have adopted chatbots to enhance customer reach and provide efficient assistance. Since chatbots have reached a stage where they utilize artificial intelligence and natural language processing (NLP), they have become more efficient in understanding human-generated text. However, with the rise of bilingual and monolingual speakers who speak one or more languages in addition to their mother tongue, the term code-switching/code-mixing has come into existence, where users mix two or more languages while speaking/writing. It has posed a new challenge to natural language processing, especially chatbots that use artificial intelligence and NLP. NLP researchers have introduced new solutions, such as monolingual machine learning models to understand the code-switched textual data. However, it has made the models considerably larger, making it harder to use these in a production environment since they are resource-heavy. Another issue in processing code-switched textual data is generating word vectors for more than one language. Word Embeddings models such as Word2Vec will encounter out-of-vocabulary (OOV) tokens more often in a code-switched dataset as the interchangeable words between languages depend on individual speakers. This research component addresses the issues encountered while processing Sinhala-English code-switched textual data through token mapping, a technique to map several representations of the same word given in different languages to a single language. It can handle the out-of-vocabulary tokens in Word2Vec word embeddings models and make them lightweight and production-ready.

Keywords**:** Natural Language Processing, Code-mixing, Word Embeddings, Token Mapping, OOV token handling

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Abbreviation	Description
AI	Artificial Intelligence
API	Application Programming Interface
CaaS	Conversational AI- as-a-Service
CBOW	Continuous Bag of Words
CDD	Conversational Driven Development
DOCX	Word Open XML Format Document
FAQ	Frequently Asked Questions
GUI	Graphical User Interface
GPU	Graphics Processing Unit
IDE	Integrated Development Environment
JS	Java Script
LKR	Sri Lankan Rupee
ML	Machine Learning
MT5	Multilingual Text-to-Text Transfer Transformer
NLP	Natural Language Processing
NLU	Natural Language Understanding
OOV	Out-Of-Vocabulary
PDF	Portable Document Format
RAM	Random Access Memory
SaaS	Software-as-a-Service
SEETM	Sinhala English Equivalent Token Mapper
SKLEARN	Scikit Learn
SLIIT	Sri Lanka Institute of Information Technology
SQL	Structured Query Language
TF-IDF	Term Frequency-Inverse Document Frequency
TPU	Tensor Processing Unit
USCS	University of Colombo School of Computing

VS	Visual Studio

1. INTRODUCTION

1.1 Background & Literature survey

Many industries and businesses utilize chatbots to assist their customers. Chatbots have been a trending topic since their early stages. They started from a rule-based form where every response had to be hard-coded. Chatbots with hard-coded rules were unable to process complex queries presented by users. Artificial intelligence-based chatbots were the solution found for the above issue. They have a better language understanding capability and generate advanced, human-like text responses. With machine learning models integrated, which can handle natural language processing (NLP), AI-based chatbots process more complex natural language queries generated by humans. These types of chatbots can mimic advanced conversations by understanding the context of the conversation better [1].

Languages such as English have a lot of NLP tools or are always in the realm of the NLP research interests. However, languages such as Sinhala have a considerably low research interest. Since there are more bilingual and monolingual speakers today, there is a need for bilingual and monolingual machine learning models that can handle more than a single language, including Sinhala. The terms code-mixing and code-switching can often be confusing, although they have distinct meanings. "code-mixing" is the practice of typing words of one language by incorporating the alphabet of another language, while mixed usage of more than one language is known as code-switching. Both terms are current research interests [11]. Researchers have developed monolingual models for various tasks that can handle code-switched text data, such as multilingual BERT [2], fast text word embeddings [3], and MT5. A couple of drawbacks of these multilingual models are the vast increment in vocabulary size and training time. However, low resource languages such as Sinhala have only a few existing monolingual, bilingual, and multilingual machine learning models for NLP [4].

With the rise of the Internet, social media, and online presence in general, people often use text-based approaches to communicate with others or retrieve information, and NLP is more required than ever to process these textual data. Online platforms have extended their language support beyond English, creating an immense vacancy for non-English NLP tools. Increment in mobile devices and apps such as Helakuru has caused the digital presence of Sinhala to emerge [5]. The ability to switch between keyboard interfaces quickly has become one of the reasons for code-switched language patterns to exist. In [11], it is described how social media has influenced people to use code-mixed language, and it discusses how people are using words such as "කෑම" using different representations like "Kema" or "Kama" using English alphabet characters. The vice-versa of the above fact is also true, as people use Sinhala characters to represent English words such as "Degree" as "ඩිග්‍රී" while typing.

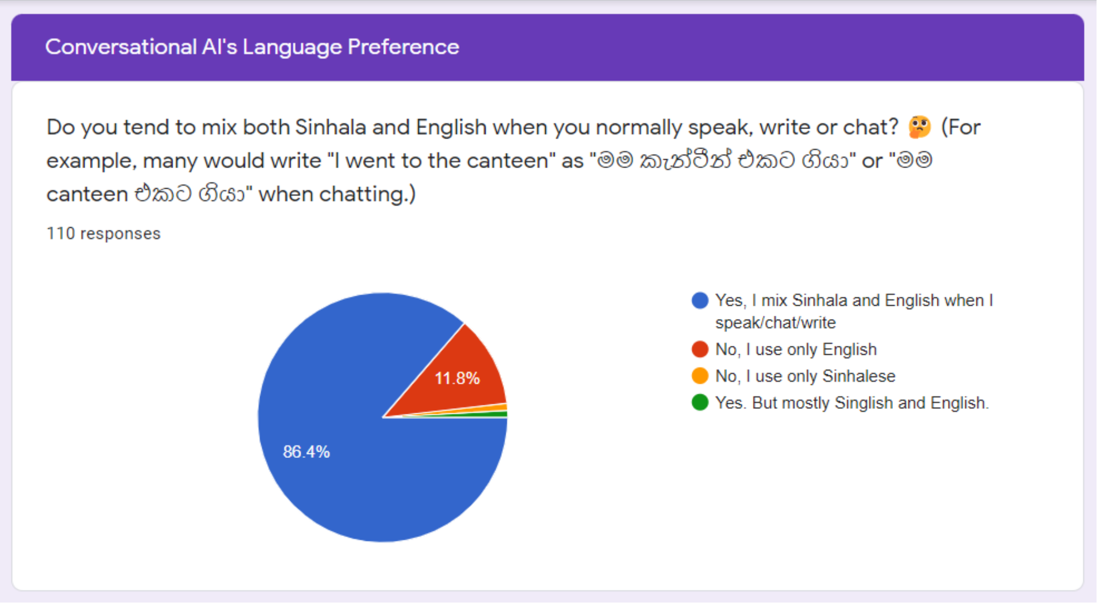
Table 1.1: The difference between code-mixing and code-switching for Sinhala and English

English version	Sinhala version	Code-mixed version	Code-switched version
What are the requirements needed to do an IT degree?	තොරතුරු තාක්ෂණ උපාධියක් කිරීමට තිබිය යුතු අවශ්‍යතා මොනවාද?	IT degree ekak karanna ona requirements monawada?	IT degree එකක් කරන්න ඕන requirements මොනවාද?
How to apply for Library Membership?	පුස්තකාල සාමාජිකත්වය සඳහා අයදුම් කරන්නේ කොහොමද?	Library Membership ekata apply karanne kohomada?	Library Membership එකට apply කරන්නේ කොහොමද?

In modern chatbot building frameworks such as Dialog Flow and Rasa [6], pre-trained machine-learning models such as language models and word embeddings can be attached to the natural language understanding (NLU) pipeline to generate dense features to train other machine learning components such as text classifiers and entity recognition models. These frameworks support languages beyond English. However, these frameworks usually prefer lightweight models since large models might cause high latency and are usually resource intensive. [7]

Word embeddings can be used to generate dense features by representing words in the vector space. Binary Encoding, TF Encoding, TF-IDF Encoding, Latent Semantic Analysis Encoding, and Word2Vec Embedding can be listed as a few out of the many word embeddings techniques. Word2Vec Embeddings [14] is still a widely used word embeddings technique, and there are two main approaches to train word embeddings models known as CBOW (Continuous Bag of Words) and Skip-Gram. Both are neural network architectures and can essentially learn to represent words in the vector space. By default, neural networks work best when there is more training data. [15]Thus, having a large enough text corpus is required to train a good word embeddings model. Not having a specific token can cause out-of-vocabulary (OOV) token issues. Researchers have introduced n-gram based word embeddings models such as fastText and byte-pair embeddings to address the OOV token issue. Modern NLP-based solutions such as artificial intelligence chatbots can rely on these word embeddings models to generate dense features to train their integrated deep learning models for better accuracy [6][8].

To get a clear idea about the language background, a recent survey was conducted for the overall research project including questions covering all four research components. The survey studied the current trends in text-based communication, machine learning model usage preference, and code-switching trends. The population of interest of the survey was 110 Undergraduates of the Faculty of Computing of Sri Lanka Institute of Information Technology, and it was decided based on the technical nature of the survey questions. Refer to Appendix A: Survey Form to observe the complete list of survey questions and responses.

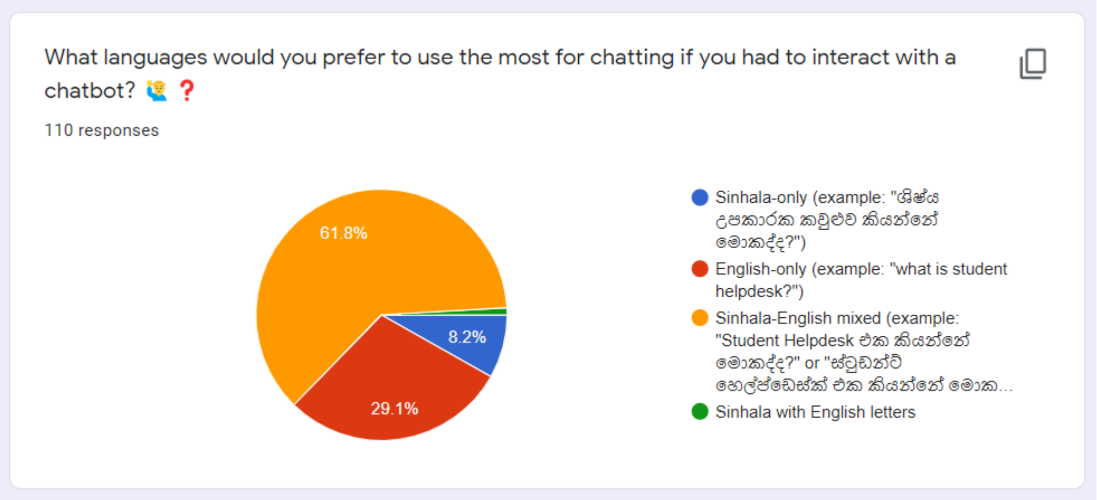


Conversational AI's Language Preference

Do you tend to mix both Sinhala and English when you normally speak, write or chat? 🤔 (For example, many would write "I went to the canteen" as "මම කැන්ටීන් එකට ගියා" or "මම canteen එකට ගියා" when chatting.)

110 responses

Figure 1.1: Summary of survey responses received for the question "Do you tend to mix both Sinhala and English when you normally speak, write, or chat?”

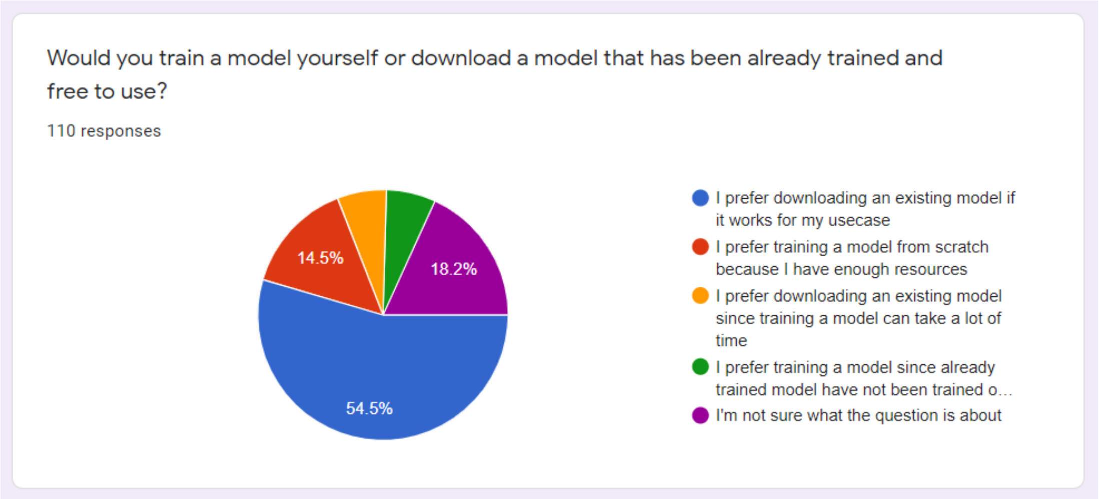


What languages would you prefer to use the most for chatting if you had to interact with a chatbot? 🙋 ❓

110 responses

Figure 1.2: Summary of survey responses received for the question "What languages would you prefer to use the most for chatting if you had to interact with a chatbot?"

Figure 1.1 and Figure 1.2 illustrates how participants of the survey prefer using a code-switched sentence structure rather than using a single language out of the given languages, Sinhala, and English.



Would you train a model yourself or download a model that has been already trained and free to use?

110 responses

Figure 1.3: Summary of survey responses received for the question "What languages would you prefer to use the most for chatting if you had to interact with a chatbot?"

Figure 1.3 illustrates how the survey results confirm that there is a higher probability to download existing machine learning models in which case a lower file size is preferred.

1.2 Research Gap

Although some research conducted around Sinhala has explored Sinhala-English code-switched text processing up to an extent, it is still considerably low. The survey mentioned in [4] demonstrates much of the research done based on Sinhala and Sinhala-English code-switched text processing, and it is a valuable resource in identifying the current research gap. Few research papers have focused on building Sinhala chatbots [8][9]. Especially the research paper mentioned in [8] discusses how word embeddings models such as fastText models can be used to increase the accuracy and is closer to the objectives of this research. Another research [9] has built a Sinhala chatbot claiming it is the first Sinhala chatbot, and it has incorporated NLP components such as morphological analyzers, Sinhala parsers, and knowledge bases. However, it lacks machine learning components such as word embeddings. The research in [10] embodies training a word2vec model on the USCS Sinhala News dataset using the continuous bag of words (CBOW) method. However, the research paper has mentioned only elementary text-preprocessing techniques such as stop words removal and lemmatization. In addition to that, none of the research papers mentioned above highlight text-preprocessing methods for Sinhala-English code-switched textual data.

Few research papers [11]-[13] have considered Sinhala-English code-switched text data processing. Although they have discussed word-level language detection, code-switching point detection, and pretraining machine learning models, they do not significantly mention proper text-preprocessing techniques that can be used on code-switched textual data. The paper in [11] has stated code-switching as a challenge in modern Sinhala text. The same research has introduced a technique called "dictionary mapping" to standardize the representation of Sinhala characters written in English characters. This technique is somewhat similar to the method proposed by this research to preprocess Sinhala-English code-switched text.

Table 1.2: Comparing and contrasting with previous related research work done

Research paper reference	**Languages used**	**Code-switched/ Code-mixed**	**Word Embeddings model type**	**Chatbot Framework**	**Special Data pre-processing techniques proposed**
[8] paper	Sinhala	-	fastText Embeddings	Rasa	Basic text preprocessing
[10] paper	Sinhala	-	Word2Vec	-	Basic text preprocessing
[11] paper	Sinhala (Main), English, Singlish	Code-mixed data	-	-	Dictionary Mapping (for characters)
This Research component	Sinhala (Main), English	Code-switched data	Word2Vec	Rasa	Equivalent Token Mapping and Character Mapping

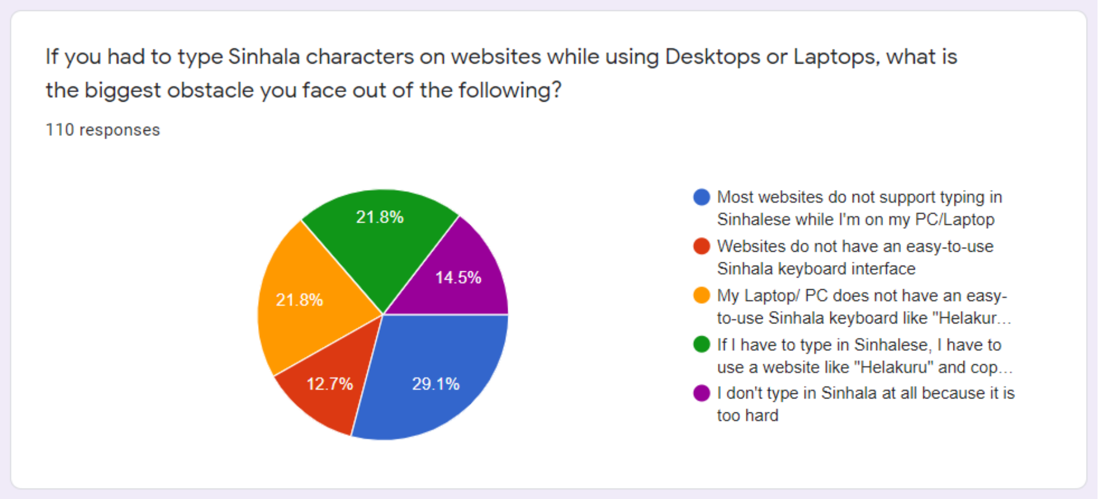
Although research papers [8] and [10] mention training word embeddings models for Sinhala, they have not considered Sinhala-English code-switched text processing. Although [8] mentions training fastText models for Rasa chatbots, no alterations have been done to the pre-processing steps for the data or the fastText models. The focus of this research is developing "token mapping": a rule-based approach to identify and map tokens present in English in the text corpus to their corresponding representation in Sinhala characters to achieve lightweight word embeddings models and handle out-of-vocabulary words efficiently. Thus, “token mapping” seems like a relatively novel approach considered by this research component, and there is a noticeable gap between the objectives of the related work done previously.

1.3 Research Problem

NLP (Natural Language Processing) tools and models for processing Sinhala and Sinhala-English code-switched textual data and feature engineering are considerably low. Developing deep learning-based models for NLP such as word embeddings, language models, and named entity recognizers from scratch requires a large amount of language-specific training data which can be rare for low resource languages. Although widely used and high resource languages such as English have many NLP tools or are in the realm of the NLP research interests, languages such as Sinhala have a considerably low research interest. In [4], it is demonstrated that there is a lot of focus on Sinhala-based research and datasets, however, it also explains that keeping the research finding locked away from public access is one of the key reasons why Sinhala is still considered as a low resource language.

Since there are more bilingual and multilingual speakers today, many researchers focus on building bilingual and multilingual machine learning models that can handle more than a single language. Building multilingual models that can handle hundreds of languages to address modern issues such as code-mixing and code-switching may not be suitable for attaching to a conversational AI since those models can be heavy in size and require a lot of training data not in one language but many.

Generally, users use a physical keyboard that has an English-specific layout when interacting with the computing devices. If they need to type in Sinhala or else code-switch between English and Sinhala, they will have to use a service like Helakuru to type in the Sinhala words they want and then copy and paste it into the conversational AI, which is a cumbersome task, and most users would not even bother to go through the trouble for it. Due to that reason, having a keyboard interface that can handle code-switching and code-mixing is a must. More precisely, the keyboard interface must be attachable and support most web applications.



If you had to type Sinhala characters on websites while using Desktops or Laptops, what is the biggest obstacle you face out of the following?

110 responses

Figure 1.4: Summary of survey responses received for the question related to typing issues exist for Sinhala in various devices

2. OBJECTIVES

2.1 Main Objectives

The main objective of this research component is to provide an effective data pre-processing technique to handle Sinhala-English code-switched textual data using a specific technique introduced known as “equivalent token mapping” to reduce the word embeddings model size and handle OOV tokens when training deep learning-based models for NLP.

This research component fits into the main objective of the overall research project through providing an effective data pre-processing technique to handle Sinhala-English code-switched textual data in conversational AI assistants.

2.2 Specific Objectives

Specific objectives of the research components are as follows.

1. Develop an algorithm that uses character-mapping to enable End-users to interact with the conversational AI assistant using code-switching and code-mixing (a rule-based keystroke mapper).

2. Develop an algorithm called “SEETM” using “equivalent token mapping” which can be used to handle out-of-vocabulary (OOV) tokens in code-mixed training data due to less amount of training data available.

3. Demonstrate how lightweight monolingual word embeddings models that need less amount of training data can handle code-mixed training data tokens.

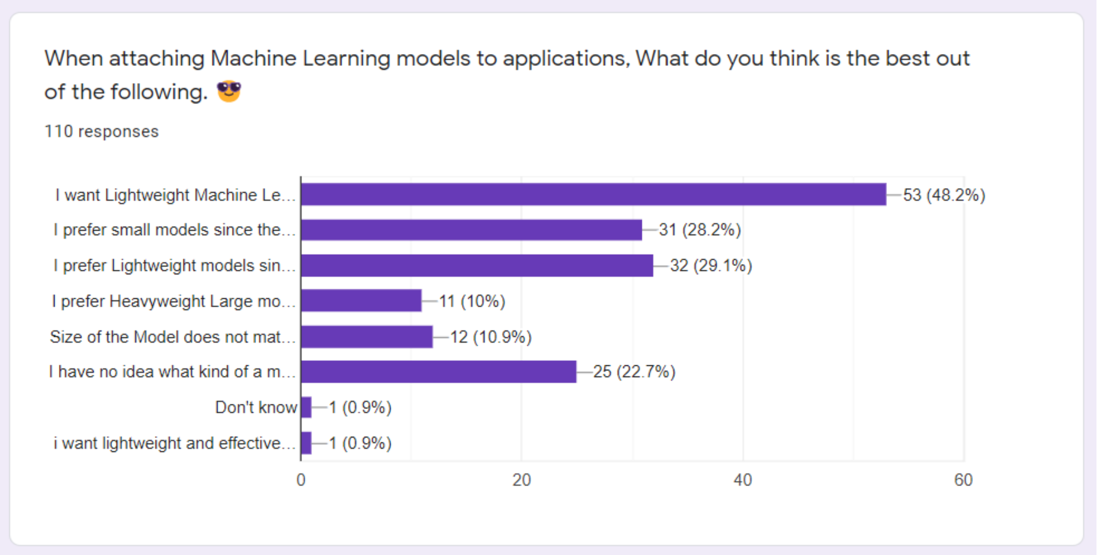
4. Develop an interactive approach to expose equivalent token mapping feature and present token mapping results for a given input sentence.

5. Integrate the SEETM algorithm and character mapping interface with Rasa conversational AI assistant developed for the overall research project to allow lightweight word embeddings training.

3. METHODOLOGY

3.1 Requirements Gathering and Analysis

The requirement gathering phase of this research component mainly concentrated on identifying existing limitations in handling Sinhala-English code-switched textual data in modern applications and machine learning models. Survey mentioned in subsection 1.2 aided in identifying many of the specific requirements of the research component. The need for a well-designed keyboard interface that can handle Sinhala-English code-switching and the trend in utilizing pre-trained machine learning models were the two of the key findings which confirmed the existing research gap and helped in deriving the following research component-specific requirements.



When attaching Machine Learning models to applications, What do you think is the best out of the following. 😎

110 responses

Figure 3.1: Survey responses which shows the increasing trend towards using lightweight models

All the specific requirements of the individual research component are as follows.

3.1.1 Functional requirements

The non-functional requirements of SEETM are as follows.

1. The proposed “equivalent token mapping” based algorithm, SEETM, should effectively map equivalent tokens to a single representation as a pre-processing technique.

2. The research component should demonstrate how SEETM can handle out of vocabulary words though single vector representation for all equivalent terms in a code-switched dataset.

3. The research component should demonstrate how SEETM can be used to achieve lightweight word embeddings models.

4. SEETM should have the ability to act as a standalone application with an easy-to-use interface to perform equivalent token mapping in a specified corpus or a text data instance while it should also be seamlessly integrated with the Sinhala-English code-switching Rasa conversational AI developed as the final product of the overall research.

5. The research component should provide a keyboard interface that can handle Sinhala-English code-switching and code-mixing using "character mapping" rule-based technique.

3.1.2 Non-functional requirements

The non-functional requirements of SEETM are as follows.

1. SEETM should be efficient.
2. SEETM should be efficient.
3. SEETM should be a modular which should be an integrable UI to websites.
4. SEETM should be scalable.
5. SEETM should be user friendly.

3.2 Feasibility Study

A feasibility study was carried out to investigate the technical, financial, legal, operational, and scheduling feasibility of the research component.

3.2.1 Technical feasibility

The research component requires having in-depth knowledge in Rasa framework, Rasa machine learning models, data science, and software development. There should be adequate computing resources to train Rasa machine learning models. Training machine learning models in Rasa do not require advanced computing resources such as GPUs, TPUs, or High RAM/CPU instances, although it is better if such resources are available. Hard drive space should be a foremost concern as the Rasa framework may need a reasonable amount of disk space for caching model files. There should be a correctly configured cloud infrastructure for the overall deployment of the final product.

3.2.2 Financial feasibility

The cloud infrastructure for training machine learning models or deployment should not be expensive. There can be minor expenses for domain name purchasing and cloud infrastructure resource usage. Then they are acceptable since the final product includes a market worth that can wrap the costs affected.

3.2.3 Legal feasibility

Datasets utilized in the research components should be publicly open datasets. Any scraped data from websites that are not public should have the approval of the actual owners and should have documented approval. The research component should not extract content as it is from earlier work done without the written permission of the original authors. Any python packages or other software used must be open source or legally purchased, and the original authors must be credited if requested.

3.2.4 Operational feasibility

This research component must not conflict with other research components of the same research project or other research projects. The research component should address the gap and should have maintained novelty in contrast to the previous work accomplished. The research component should be feasible to fulfil the scope stated by the research project requirements.

3.2.5 Scheduling feasibility

The research component tasks should be attached to the Gantt chart mentioned in section 5 of this proposal report and the pre-defined research project milestones.

3.3 Preparation of Datasets

Overall research requires two datasets in total. Both datasets are domain-specific datasets with Sinhala-English code-switched text data. However, subsections 3.3.1 and 3.3.2. clearly explain the notable differences between the datasets. Data augmentation techniques were used with the following objectives in mind

1. Overcome the low-resource nature of the collected Sinhala-English code-switched text data.
2. Capture as many code-switched tokens as possible.
3. Capture as many sentence structures and patterns as possible.

Four versions of the data samples will be generated for both datasets incorporating various code-switching patterns, and the duplicates will be removed to preserve the quality of the collected textual data. This research component utilizes both datasets combined to identify English tokens and English tokens written using Sinhala characters to understand how they can be mapped to a single representation.

3.3.1 General dataset for machine learning model training

The first domain-specific dataset for machine learning model training and NLP text pre-processing tool development contains scraped Sinhala-English code-switched textual data from websites related to SLIIT and from news articles. (https://support.sliit.lk/, https://sliitinternational.lk/, and https://www.sliit.lk/ are few such websites). In addition to that, publicly available documents, such as PDFs and Docx files taken from the above SLIIT-based websites since they are both public and official. Dataset will utilize data augmentation techniques to overcome the low-resource issue.

3.3.2 Domain specific dataset for conversational AI training

The second domain-specific dataset for training the Rasa-based conversational AI contains handcrafted Sinhala-English code-switched textual data. This dataset should be designed carefully as a training dataset for the intent classification task of conversational AI according to the guidelines provided by Rasa. As in the previous case, the second dataset also will utilize data augmentation techniques to overcome the low-resource issue. There will be around 78 intents (classes), and each class initially contains a minimum of 10 examples as a standard. Note that the number of training examples may increase in the future to increase the overall performance of conversational AI.

3.4 Individual Component Architecture

The high-level architecture of the individual research component and its sub-components has been illustrated in the Figure 3.2. The SEETM algorithm for equivalent token mapping will be developed as a standalone python package. It mainly consists of a Unicode-based character level language identifier module for English and Sinhala, and a Token Mapping module where all equivalent tokens will be given a single representation. The core standalone package will contain a locally runnable server that can be used as a GUI web application to map tokens in any given Sinhala-English code-switched text corpus, paragraph, or sentence. The SEETM algorithm will be utilized within a custom developed Rasa component to incorporate token mapping in conversational AI models. This component is proposed as a separate module that incorporates the core SEETM package.

The character mapping algorithm is another rule-based algorithm that will be developed under this research component which will be directly integrated with the main frontend web application of the final conversational AI developed by integrating all research components. This algorithm enables typing in Sinhala and English characters all using the same English keyboard interface available in devices. The above algorithms have been described in detail in subsection 3.5.

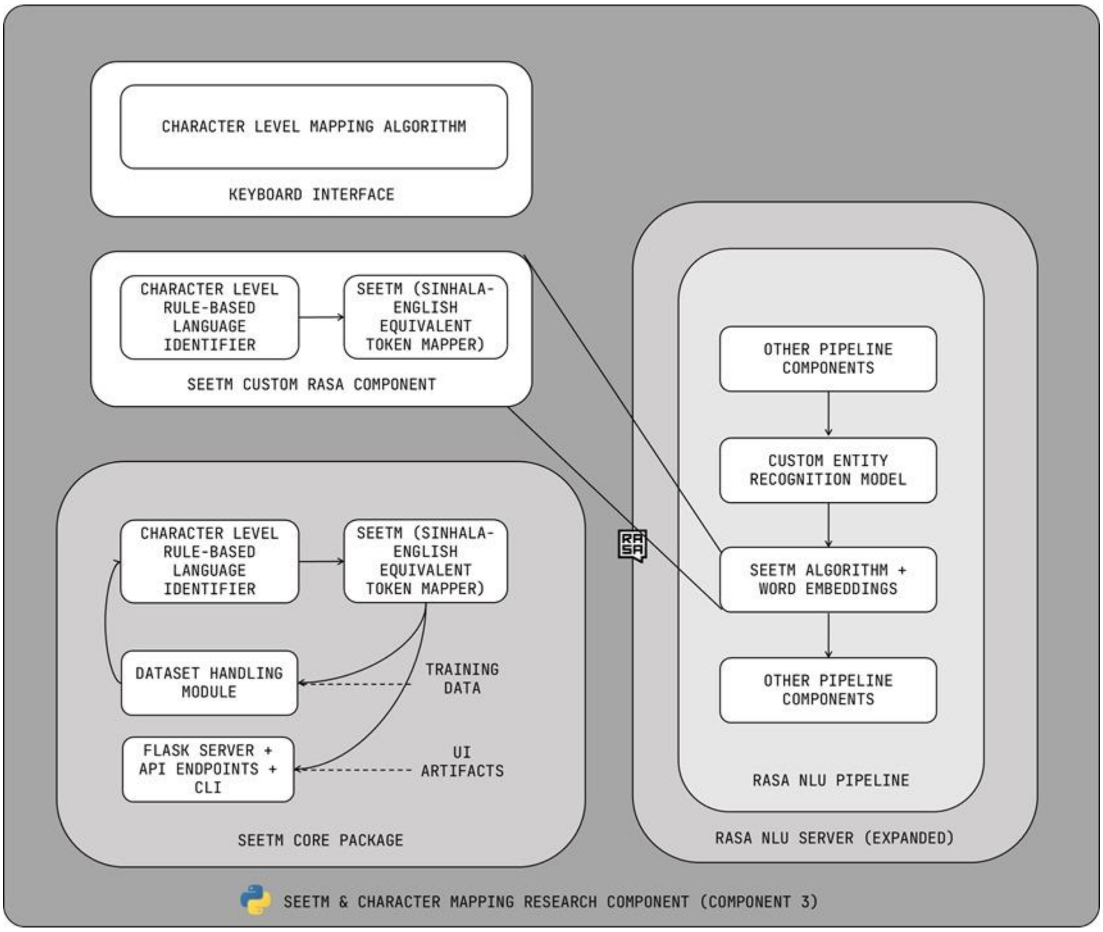

CHARACTER LEVEL MAPPING ALGORITHM
KEYBOARD INTERFACE
CHARACTER LEVEL RULE-BASED LANGUAGE IDENTIFIER
SEETM (SINHALA-ENGLISH EQUIVALENT TOKEN MAPPER)
SEETM CUSTOM RASA COMPONENT
CHARACTER LEVEL RULE-BASED LANGUAGE IDENTIFIER
SEETM (SINHALA-ENGLISH EQUIVALENT TOKEN MAPPER)
DATASET HANDLING MODULE
TRAINING DATA
FLASK SERVER + API ENDPOINTS + CLI
UI ARTIFACTS
SEETM CORE PACKAGE
OTHER PIPELINE COMPONENTS
CUSTOM ENTITY RECOGNITION MODEL
SEETM ALGORITHM + WORD EMBEDDINGS
OTHER PIPELINE COMPONENTS
RASA NLU PIPELINE
RASA NLU SERVER (EXPANDED)
RA SA
SEETM & CHARACTER MAPPING RESEARCH COMPONENT (COMPONENT 3)

Figure 3.2: High-level architectural diagram of the individual research component

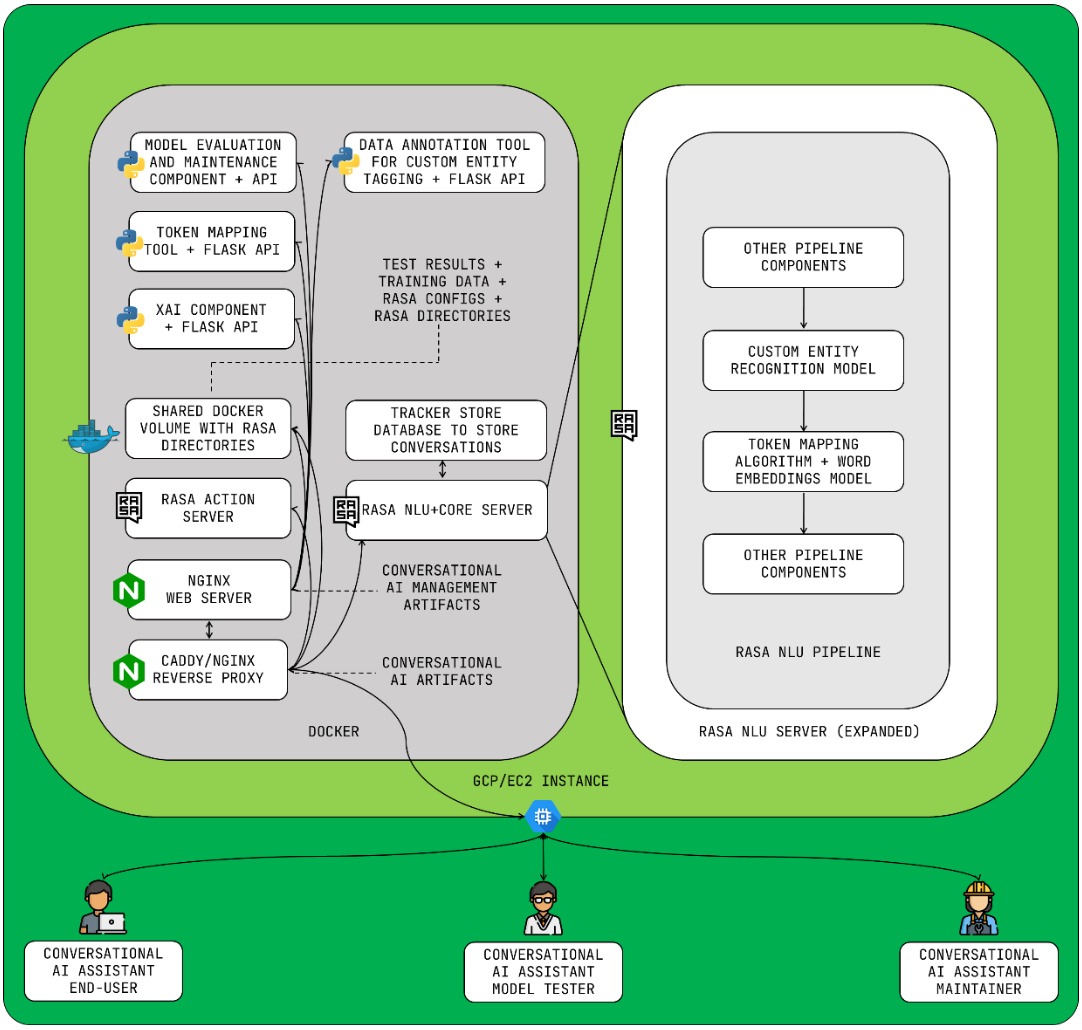

MODEL EVALUATION AND MAINTENANCE COMPONENT + API
DATA ANNOTATION TOOL FOR CUSTOM ENTITY TAGGING + FLASK API
TOKEN MAPPING TOOL + FLASK API
TEST RESULTS + TRAINING DATA + RASA CONFIGS + RASA DIRECTORIES
XAI COMPONENT + FLASK API
SHARED DOCKER VOLUME WITH RASA DIRECTORIES
TRACKER STORE DATABASE TO STORE CONVERSATIONS
RASA ACTION SERVER
RASA NLU+CORE SERVER
NGINX WEB SERVER
CONVERSATIONAL AI MANAGEMENT ARTIFACTS
CADDY/NGINX REVERSE PROXY
CONVERSATIONAL AI ARTIFACTS
DOCKER
OTHER PIPELINE COMPONENTS
CUSTOM ENTITY RECOGNITION MODEL
TOKEN MAPPING ALGORITHM + WORD EMBEDDINGS MODEL
OTHER PIPELINE COMPONENTS
RASA NLU PIPELINE
RASA NLU SERVER (EXPANDED)
GCP/EC2 INSTANCE
CONVERSATIONAL AI ASSISTANT END-USER
CONVERSATIONAL AI ASSISTANT MODEL TESTER
CONVERSATIONAL AI ASSISTANT MAINTAINER

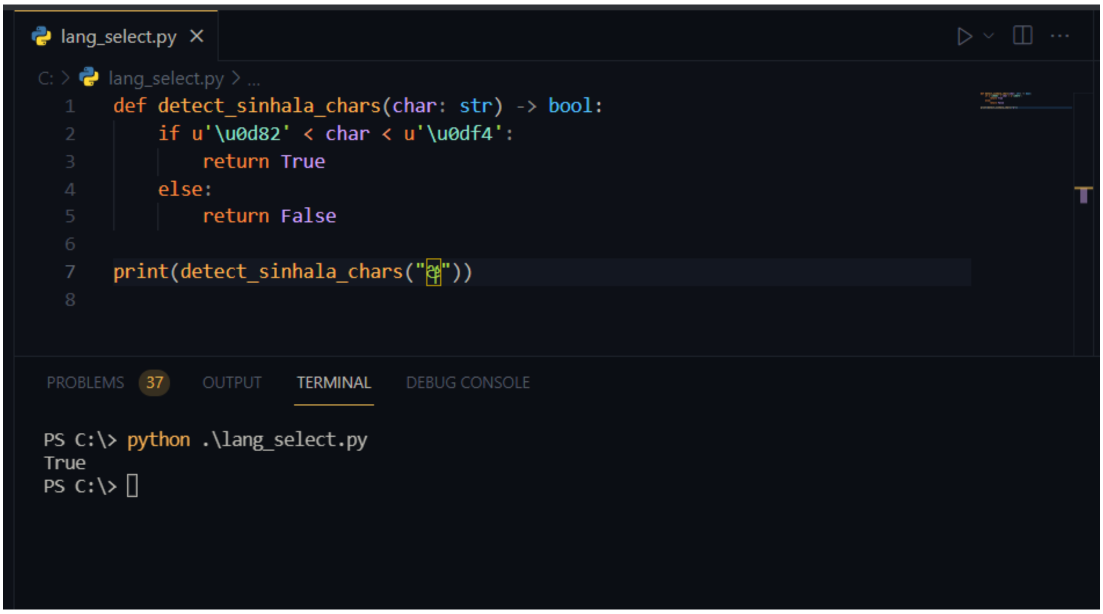
Figure 3.3: High-level architectural diagram of the purposed solution with all research components integrated and how Token Mapping (SEETM) fits in.

Figure 3.3 illustrates how token mapping research component (SEETM) is integrated in the overall system architecture after all research components are integrated.

3.5 Algorithm Development

3.5.1 SEETM algorithm using equivalent token mapping

The proposed SEETM algorithm utilizes an equivalent token mapping approach to reduce the number of tokens in the training data by mapping the same word in various languages to a single representation. For instance, a code-switched dataset might contain the tokens Degree, ඩිග්‍රී, and උපාධි which has the same meaning. Equivalent token mapping maps all these three words into a single representation which can be one form out of the three. The algorithm will map the English words found to its Sinhala representation or vice-versa, which will be an option that the users can change. Before the algorithm can map the tokens, it should tokenize the text using a white space tokenizer and correctly identify the word-level language based on the Unicode range. A simple python code that can detect Sinhala characters looks like the simple code given in the Figure 3.4. An advanced algorithm will be developed and optimized using a similar logic.



```python
def detect_sinhala_chars(char: str) -> bool:
    if u'\u0d82' < char < u'\u0df4':
        return True
    else:
        return False

print(detect_sinhala_chars("අ"))
```

```
PS C:\> python .\lang_select.py
True
PS C:\> 
```

Figure 3.4: A simple python script with a character level language detection for Sinhalese based on Unicode values

After the language identification step is done, all English tokens identified will be mapped into the Sinhala representation of the word using the character mapping algorithm to generate the code-mixed representation of the English token. For instance, the English word “Course” will be mapped into “කෝර්ස්”. The original word “Course” will then be replaced by the mapped representation in a given corpus. The completed algorithm including character mapping and SEETM algorithms will be developed as a Python package. The Python package will contain a locally executable server made with Flask with an interactive web application where users will be able to upload a Sinhala text corpus, document, or a sentence/question where code-switching is present to get the English tokens mapped into the corresponding Sinhala character representation. The major benefits of the equivalent token mapping approach are as follows.

1. Reduces the number of tokens which reduces the size of vocabulary in NLP models such as word embeddings. This will result in a slightly less model size after training.

2. Allows to receive the same vector representation to all equivalent tokens instead of entirely different or slightly different word representations.

3. Assists to handle out-of-vocabulary words when a word embedding model vocabulary have not seen the English version of the word during training.

The effectiveness of the SEETM, the equivalent token mapping approach, will be assessed by training a Word2Vec model with and without token mapping done to the training dataset. The same Python package will be used to develop a custom NLU pipeline component for Rasa conversational AI assistants to be able to get the advantage of already mapped tokens. A more detailed explanation of Word2Vec model training and integrating SEETM with Rasa conversational AI assistants can be found in subsection 3.6.

3.5.2 Sinhala-English code-switching using character mapping

The same character mapping technique described in the previous subsection 3.5.1 will be implemented in the frontend of the main conversational AI frontend to enable Sinhala-English code-switching supported keyboard interface based on the keystroke events. This allows users to interact with the conversational AI using code-switching typing pattern easily, especially on personal computers.

3.6 Machine Learning Model Training and Testing

Word2Vec word embeddings models will be trained using continuous bag of words approach and used to evaluate effectiveness of the SEETM algorithm. Several metrics will be considered for the evaluation as mentioned below.

1. Word Vectors: A single Word2Vec model will be trained on a token mapped Sinhala-English code-switched corpus and a validation set will be used with and without token mapping applied. The number of out-of-vocabulary terms found will be calculated.

2. Model size: Two Word2Vec models will be trained on Sinhala-English code-switched corpus, with and without token mapping. The size of the two models will be compared and the size difference will be noted.

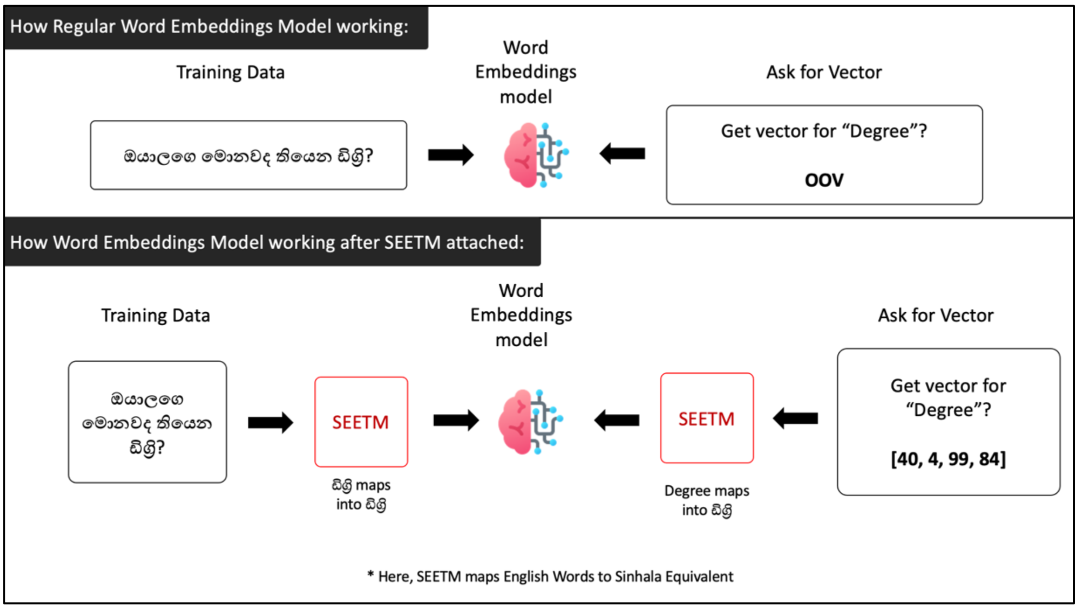
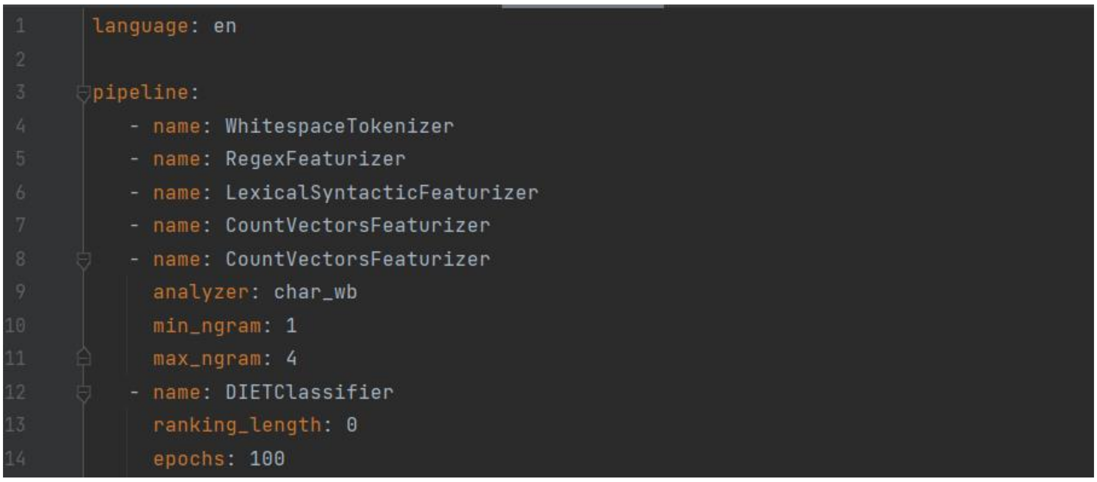

How Regular Word Embeddings Model working:
Training Data
Word Embeddings model
Ask for Vector
ඔයාලගෙ මොනවද තියෙන ඩිග්‍රී?
Get vector for “Degree”?
OOV
How Word Embeddings Model working after SEETM attached:
Training Data
Word Embeddings model
Ask for Vector
ඔයාලගෙ මොනවද තියෙන ඩිග්‍රී?
SEETM
ඩිග්‍රී maps into ඩිග්‍රී
SEETM
Degree maps into ඩිග්‍රී
Get vector for “Degree”?
[40, 4, 99, 84]
* Here, SEETM maps English Words to Sinhala Equivalent

Figure 3.5: How Word Embeddings model work without SEETM and With SEETM attached

The Word2Vec model trained on token mapped version of the previously mentioned datasets and a custom NLU pipeline component for mapping new text data will be attached to the pipeline of the Rasa conversational AI assistant that will be created by integrating all research components.



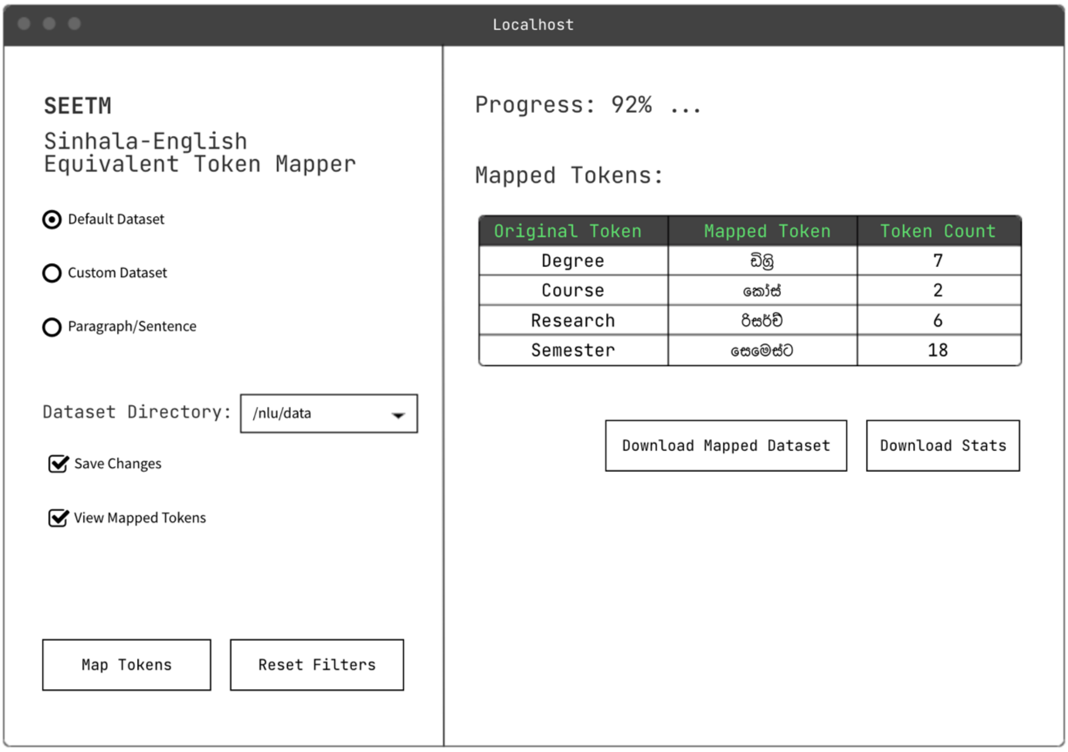
```
language: en

pipeline:
    - name: WhitespaceTokenizer
    - name: RegexFeaturizer
    - name: LexicalSyntacticFeaturizer
    - name: CountVectorsFeaturizer
    - name: CountVectorsFeaturizer
      analyzer: char_wb
      min_ngram: 1
      max_ngram: 4
    - name: DIETClassifier
      ranking_length: 0
      epochs: 100
```

Figure 3.6: The default Rasa NLU pipeline, which is configurable

3.7 User Interfaces Visualizations

Figure 3. is a wireframe designed for the individual research component. A less experienced user with the terminal will be able to use the proposed GUI to interact with the tool to use the SEETM algorithm to map English words in either the Rasa conversational AI assistant dataset or an external dataset of choice and get a list of token mappings done. As previously mentioned, the tool can be attached with Rasa framework and the GUI tool can be used to perform token mapping for a preferred external text corpus or any given text.



SEETM

Sinhala-English Equivalent Token Mapper

● Default Dataset

○ Custom Dataset

○ Paragraph/Sentence

Dataset Directory: /nlu/data

☑ Save Changes

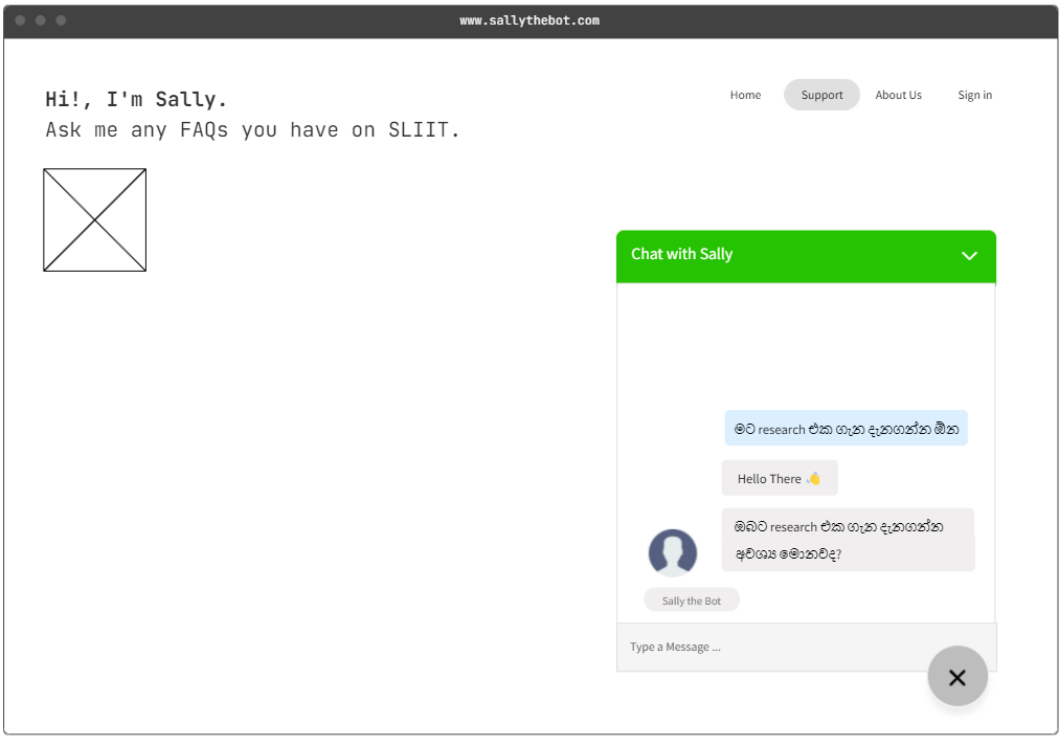
☑ View Mapped Tokens

Progress: 92% ...

Mapped Tokens:

Original Token	Mapped Token	Token Count
Degree	ඩිග්‍රි	7
Course	කෝස්	2
Research	රිසර්ච්	6
Semester	සෙමෙස්ට	18

Figure 3.7: Wireframe of the proposed user interface of SEETM local server for efficient token mapping



Hi!, I'm Sally.

Ask me any FAQs you have on SLIIT.

මට research එක ගැන දැනගන්න ඕන

Hello There 👋

ඔබට research එක ගැන දැනගන්න අවශ්‍ය මොනවද?

Figure 3.8: Wireframe for the main user interface where the Sinhala-English code-switching enabled keyboard interface will be attached

Figure 3. illustrate a basic wireframe of the proposed frontend for the overall research and how the character mapping algorithm will be utilized to type Sinhala and English code-switching text queries without having to use any external applications or services.

3.8 Tools and Technologies

The SEETM algorithm implementation and package development will be done using Python 3.8. However, packages such as NumPy, Pandas, Sklearn, NLTK will be used for the preparation and preprocessing of the datasets and text pipelines. Gensim will be used for the implementation and training of Word2Vec models which will then be attached to the Rasa NLU pipeline as a custom component for dense feature extraction. The Overall implementation of the complete research and model training and testing for the common Rasa conversational AI assistant will be using the specific Rasa version 2.8.12. PyCharm, VS code and Google CoLab will be used as the IDEs.

The complete list of tools and technologies used in this research component and the overall research are as follows.

Table 3.1: Task and Tools to be used

Task	Tools to be used
Algorithm Implementation and SEETM package development	Python 3.8, NumPy, Pandas, sklearn, NLTK, Gensim, Rasa 2.8.12, PyCharm, Visual Studio Code, Google CoLab
Character mapping and the keyboard interface development	JavaScript
Server development as a standalone frontend for the SEETM token mapper	Flask, JavaScript, Bootstrap, React JS (optional), Streamlit
NoSQL Database development (for the overall research solution)	MongoDB Atlas, MongoDB Compass
Cloud Infrastructure management (for the overall research solution)	One out of Google Cloud Platform or Amazon Web Services
Conversational AI development by Integrating all research components (Backend)	Rasa 2.8.12, MongoDB Atlas, Gensim, spaCy, Docker
Conversational AI frontend development	React JS, Bootstrap, socket.io, JavaScript
Overall system deployment	Caddy 2, NGINX, Git, Docker, docker-compose

4. DESCRIPTION OF PERSONAL AND FACILITIES

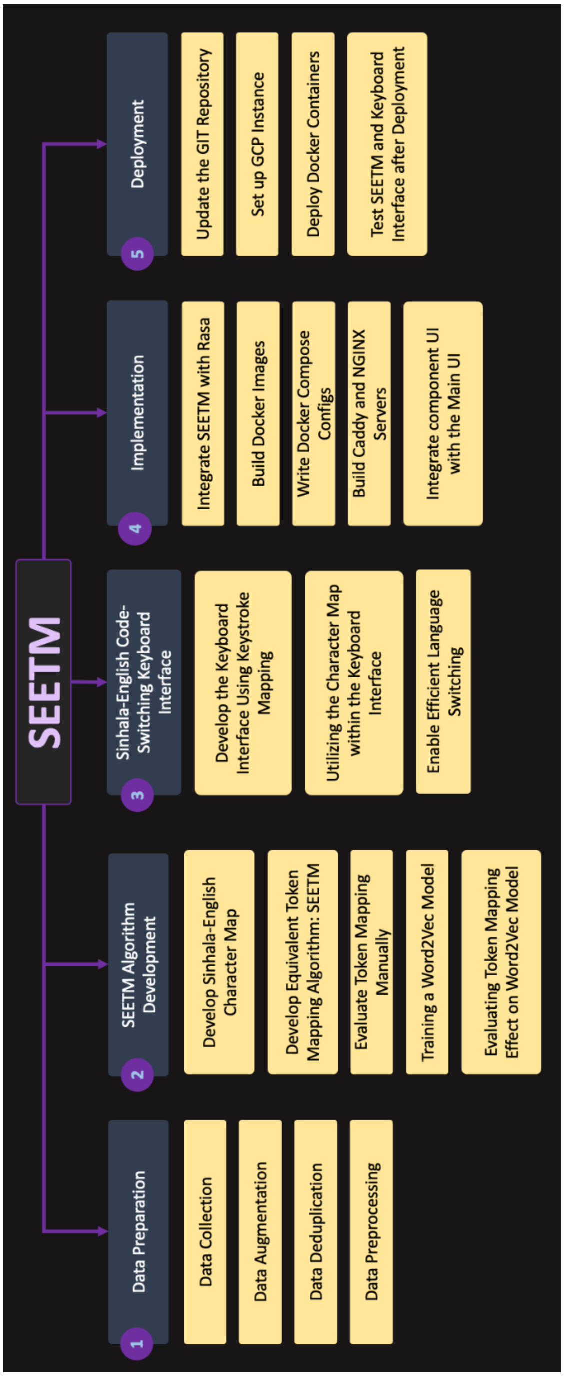
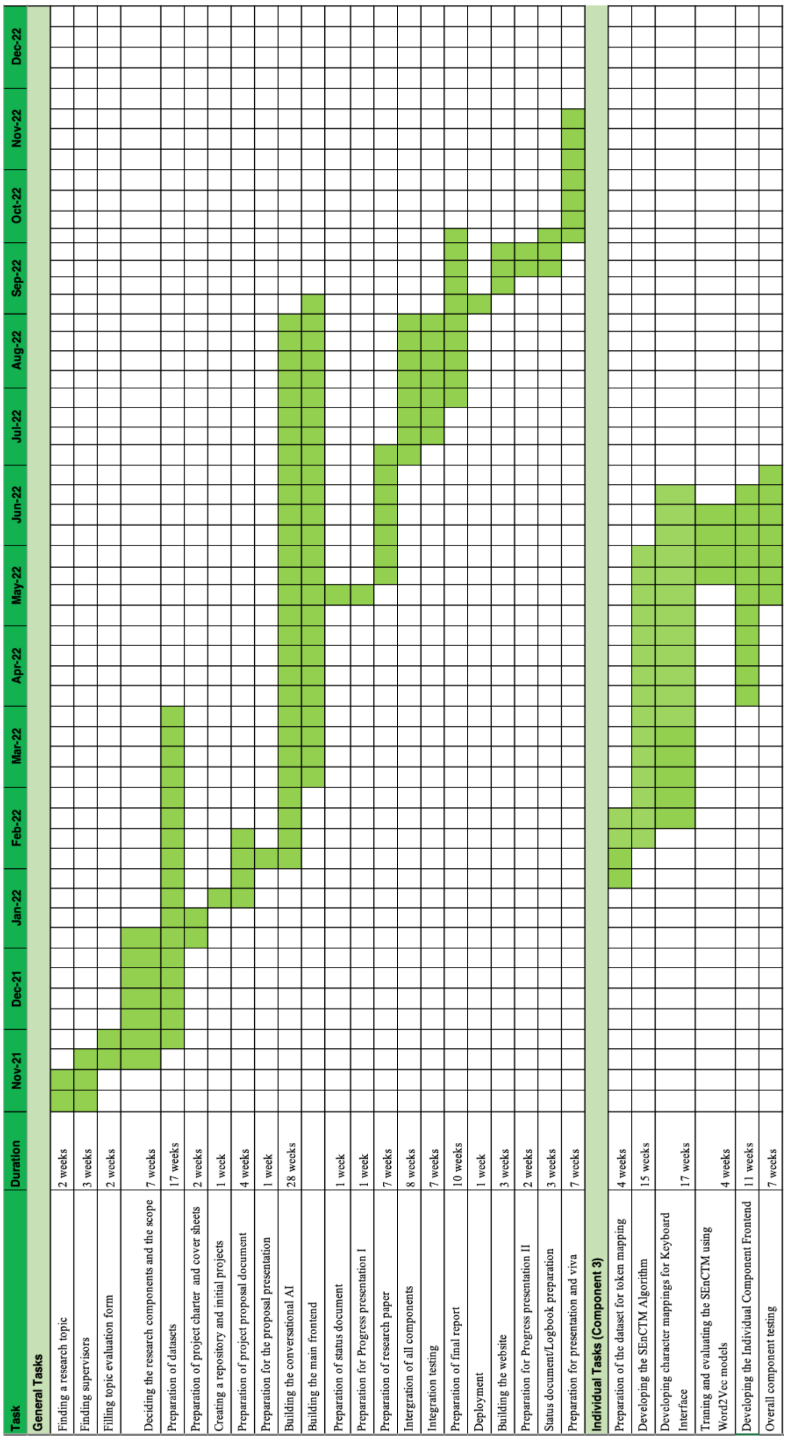

SEETM
1
Data Preparation
Data Collection
Data Augmentation
Data Deduplication
Data Preprocessing
2
SEETM Algorithm Development
Develop Sinhala-English Character Map
Develop Equivalent Token Mapping Algorithm: SEETM
Evaluate Token Mapping Manually
Training a Word2Vec Model
Evaluating Token Mapping Effect on Word2Vec Model
3
Sinhala-English Code-Switching Keyboard Interface
Develop the Keyboard Interface Using Keystroke Mapping
Utilizing the Character Map within the Keyboard Interface
Enable Efficient Language Switching
4
Implementation
Integrate SEETM with Rasa
Build Docker Images
Write Docker Compose Configs
Build Caddy and NGINX Servers
Integrate component UI with the Main UI
5
Deployment
Update the GIT Repository
Set up GCP Instance
Deploy Docker Containers
Test SEETM and Keyboard Interface after Deployment

Figure 4.1: Work breakdown Structure of the individual research component

Figure 4.1 all the tasks are breaks into 5 main tasks which are data preparation, SEETM algorithm development, Code-switching keyboard UI development, implementation and final deployment.

5. GANTT CHART



Task	Duration
General Tasks	
Finding a research topic	2 weeks
Finding supervisors	3 weeks
Filling topic evaluation form	2 weeks
Deciding the research components and the scope	7 weeks
Preparation of datasets	17 weeks
Preparation of project charter and cover sheets	2 weeks
Creating a repository and initial projects	1 week
Preparation of project proposal document	4 weeks
Preparation for the proposal presentation	1 week
Building the conversational AI	28 weeks
Building the main frontend	
Preparation of status document	1 week
Preparation for Progress presentation I	1 week
Preparation of research paper	7 weeks
Intergration of all components	8 weeks
Integration testing	7 weeks
Preparation of final report	10 weeks
Deployment	1 week
Building the website	3 weeks
Preparation for Progress presentation II	2 weeks
Status document/Logbook preparation	3 weeks
Preparation for presentation and viva	7 weeks
Individual Tasks (Component 3)	
Preparation of the dataset for token mapping	4 weeks
Developing the SEnCTM Algorithm	15 weeks
Developing character mappings for Keyboard Interface	17 weeks
Traning and evaluating the SEnCTM using Word2Vec models	4 weeks
Developing the Individual Component Frontend	11 weeks
Overall component testing	7 weeks

Figure 5.1: Gantt chart for overall project and the individual component

6. COMMERCIALISATION PLAN

Each research component of the overall research project provides applications for various Natural Language Processing and Machine Learning model evaluation-related tasks. Although the outputs of each research component can help design many products and services, they were all integrated to build a conversational AI that fully supports Sinhala-English code-switching. The main reasons for developing a conversational AI are as follows.

1. Conversational AIs are a trending topic, and there is an increasing demand for Sinhala-based conversational AIs. Many businesses are looking to increase their customer reach by using conversational AIs for a vast range of business tasks, including providing technical assistance, automating manual tasks such as booking tickets, and providing the information requested by the customers. Although that is the case, it is hard to find Sinhala-based conversational AIs, especially in Sri Lanka. Thus, developing Sinhala-based conversational AIs was identified as a potential business opportunity.

2. Although there are many conversational AI development frameworks, most of them lack evaluation tools to debug machine learning models. In frameworks like Rasa, model evaluations are highly technical, and the average developers without machine learning knowledge fail to identify problems and debug the machine learning models. Solving this issue by allowing non-technical users to maintain and evaluate machine learning models and conversational AIs is another business opportunity where evaluation tools can be built and released as add-on features.

3. Businesses are moving towards cloud-based solutions, especially SaaS products from traditional standalone applications, web applications, and on-premises solutions, where the maintenance cost is high. A cloud-based highly configurable conversational AI would be an appropriate solution for many businesses where the effort for maintenance is considerably low.

The end product of the overall research concentrates on designing a solution for the above potential business ideas and opportunities. The conversational AI developed as the research end-product is mainly a SaaS or simply a CaaS (conversational AI-as-a-service) product that a business can purchase with a set of add-on features, as illustrated in Figure 6.1. In addition to the CaaS packages, on-premises and demo cloud-based conversational AI packages are available for affordability and convenience. The final product of the overall research has the following archivable user benefits.

1. Businesses can purchase a CaaS package and eliminate conversational AI maintenance efforts.

2. Developers can use code-less maintenance tools to maintain conversational AIs they purchase without having in-depth knowledge about the backend deployment and the conversational AI development framework.

3. Businesses can purchase evaluation tools as add-ons and generate model explanations and evaluate machine learning models themselves to avoid extra maintenance costs. Here, the generated evaluations are easy to understand by non-technical users, which is not a feature of any existing chatbot framework.

4. Users of conversational AI can easily use the Sinhala-English code-switchable keyboard interface, and businesses can attract more users from having this feature as it eliminates the need to use third-party Sinhala typing services.

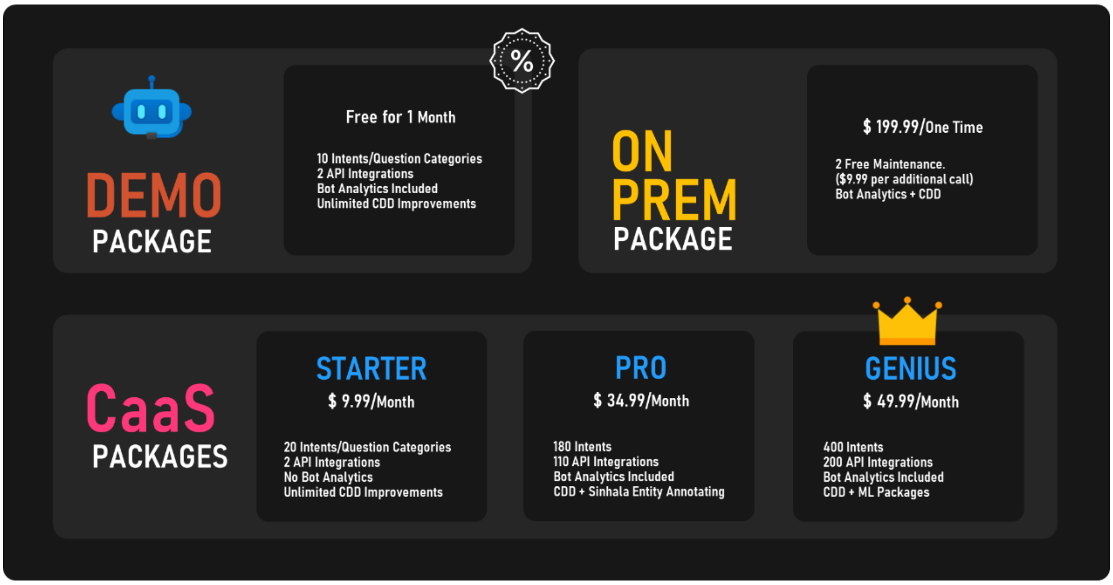
5. Businesses have the freedom to purchase either the CaaS packages or on-premises packages as per their need, while anyone can test the demo conversational AI for a period of 1 month before deciding to purchase any of the other packages that have a cost assigned.

6. Businesses can reach a vast customer range through Sinhala-English code-switching-based conversational AIs, especially within the Sri Lankan market, and it is possible to eliminate the need for having a dedicated customer care staff. It will dramatically lower the expenses of the business.

The proposed commercialisation plan of the final product of the overall research component contains a set of convenient packages and the offered features of each package differ based on the cost attached to it. The distribution of the features and the package cost were carefully planned and designed by analyzing the existing purchasable packages of similar SaaS products and conversational AIs. Table 6.1 clearly illustrates the feature variation and the cost difference of the packages offered by the proposed commercialisation plan.

Table 6.1: Feature variations and cost difference of packages proposed by the commercialisation plan

Feature	**Packages**				
	Demo	**On-Prem**	**CaaS**		
			Starter	**Pro**	**Genius**
Intents	10	Unlimited	20	180	400
API Integrations	2	Unlimited	2	110	200
Bot Analytics	✓	✓	-	✓	✓
CDD	✓	✓	✓	✓	✓
Sinhala Entity Annotating	✓	-	-	✓	✓
ML Evaluation Tools	-	-	-	-	✓
Maintenance Fee	-	2 Free + $9.99 per additional call	-	-	-
Trial Duration	1 Month	-	-	-	-
Package Price	Free	$199.99	$9.99	$34.99	$49.99



DEMO
PACKAGE

Free for 1 Month

10 Intents/Question Categories
2 API Integrations
Bot Analytics Included
Unlimited CDD Improvements

ON
PREM
PACKAGE

$ 199.99/One Time

2 Free Maintenance.
($9.99 per additional call)
Bot Analytics + CDD

CaaS
PACKAGES

STARTER
$ 9.99/Month

20 Intents/Question Categories
2 API Integrations
No Bot Analytics
Unlimited CDD Improvements

PRO
$ 34.99/Month

180 Intents
110 API Integrations
Bot Analytics Included
CDD + Sinhala Entity Annotating

GENIUS
$ 49.99/Month

400 Intents
200 API Integrations
Bot Analytics Included
CDD + ML Packages

Figure 6.1: Proposed Commercialization Plan

7. BUDGET AND BUDGET JUSTIFICATION

The budget justification for all four research components is given in **Error! Reference source not found.**.

Table 7.1: Budget justification for the overall research project

Component Name	Individual Item Price (LKR)	Number of Items	Duration	Total Item Price (LKR)
Domain Name	2148.43/year	1	1 year	2148.43
GCP Instance	10683.84/month	1	6 months	64103.06
Reference Book: Basaka Mahima by J.B. Dissanayake	1250.00	1	-	1250.00
Research Paper Publication	25000.00	1	-	25000.00
Grand Total	-	-	-	92,501.49

REFERENCES

[1]. S. J. a. K. M. Guendalina Caldarini, "A Literature Survey of Recent Advances in Chatbots," 17 Jan 2022. [Online]. Available: https://arxiv.org/pdf/2201.06657.pdf

[2]. J. Devlin, M.-W. Chang, K. Lee, en K. Toutanova, “BERT: Pre-training of deep bidirectional Transformers for language understanding”, 2018. [Online]. Available: https://arxiv.org/abs/1810.04805

[3]. P. Bojanowski, E. Grave, A. Joulin, en T. Mikolov, “Enriching word vectors with subword information”, 2016. [Online]. Available: https://arxiv.org/abs/1607.04606

[4]. N. de Silva, “Survey on publicly available Sinhala Natural Language Processing tools and research,” Jun. 2019. [Online]. Available: https://arxiv.org/abs/1906.02358

[5]. Bhasha Lanka, “හෙළකුරු Keyboard,” [Online]. Available: https://www.helakuru.lk/keyboard

[6]. T. Bocklisch, J. Faulkner, N. Pawlowski, A. Nichol, “Rasa: Open-Source Language Understanding and Dialogue Management,” 2017. [Online]. Available: https://arxiv.org/abs/1712.05181

[7]. T. Bunk, D. Varshneya, V. Vlasov, A. Nichol, “DIET: Lightweight Language Understanding for Dialogue Systems,” 2020. [Online]. Available: https://arxiv.org/abs/2004.09936

[8]. B. Gamage, R. Pushpananda, and R. Weerasinghe, “The impact of using pre-trained word embeddings in sinhala chatbots,” in *2020 20th*

International Conference on Advances in ICT for Emerging Regions (ICTer). IEEE, 2020, pp. 161–165.

[9]. “First sinhala chatbot in action,” Proceedings of the 3rd Annual Sessions of Sri Lanka Association for Artificial Intelligence (SLAAI), University of Moratuwa, 2006. https://slaai.lk/proc/2006/budditha.pdf

[10]. T. KasthuriArachchi and E. Y. A. Charles, "Deep Learning Approach to Detect Plagiarism in Sinhala Text," *2019 14th Conference on Industrial and Information Systems (ICIIS)*, 2019, pp. 314-319, doi: 10.1109/ICIIS47346.2019.9063299.

[11]. A. Kugathasan and S. Sumathipala, "Standardizing Sinhala Code-Mixed Text using Dictionary based Approach," *2020 International Conference on Image Processing and Robotics (ICIP)*, 2020, pp. 1-6, doi: 10.1109/ICIP48927.2020.9367353.

[12]. Smith and U. Thayasivam, "Sinhala-English Code-Mixed Data Analysis: A Review on Data Collection Process," *2019 19th International Conference on Advances in ICT for Emerging Regions (ICTer)*, 2019, pp. 1-6, doi: 10.1109/ICTer48817.2019.9023739.

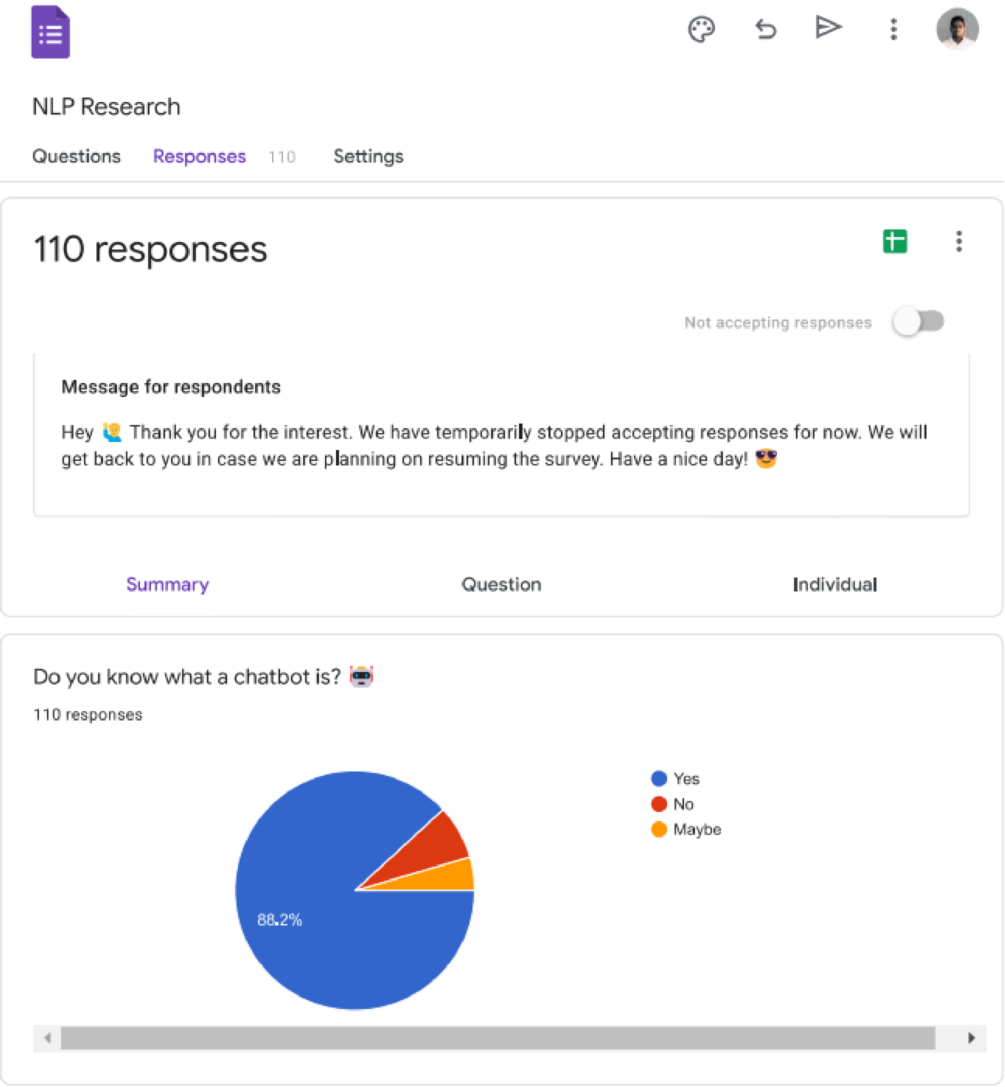
[13]. I. Smith and U. Thayasivam, "Language Detection in Sinhala-English Code-mixed Data," *2019 International Conference on Asian Language Processing (IALP)*, 2019, pp. 228-233, doi: 10.1109/IALP48816.2019.9037680.

[14]. T. Mikolov, K. Chen, G. Corrado, en J. Dean, “Efficient estimation of word representations in vector space”, [Online]. Available: https://arxiv.org/abs/1301.3781

[15]. S. S. Y. S. H. W. K. W. Fangyu Gu, "NLP: Word Embedding Techniques for Text Analysis," 4 Feb 2020. [Online]. Available: https://medium.com/sfu-cspmp/nlp-word-embedding-techniques-for-text-analysis-ec4e91bb886f

APPENDICES

Appendix A: Survey Form



NLP Research

110 responses

Not accepting responses

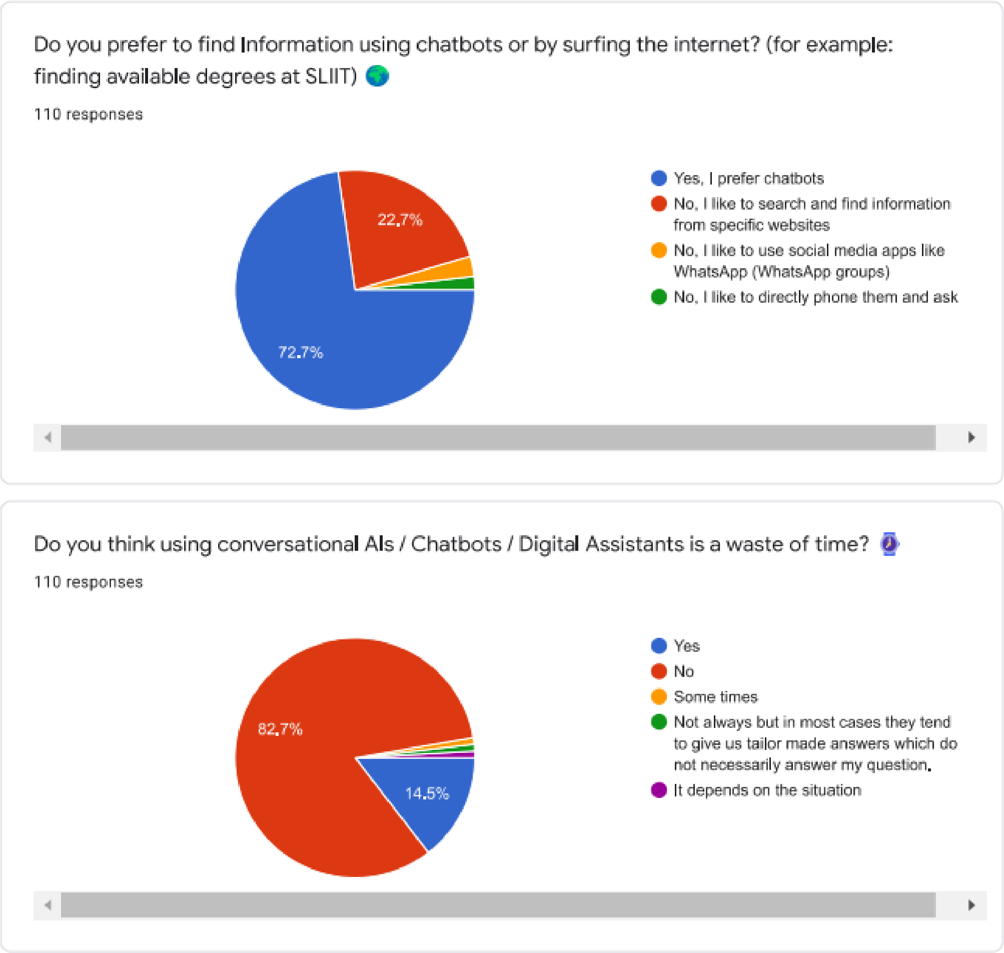
Message for respondents

Hey 🙋 Thank you for the interest. We have temporarily stopped accepting responses for now. We will get back to you in case we are planning on resuming the survey. Have a nice day! 😎

Do you know what a chatbot is? 🤖

110 responses

Figure A.1: Complete survey form questions and responses – part 1



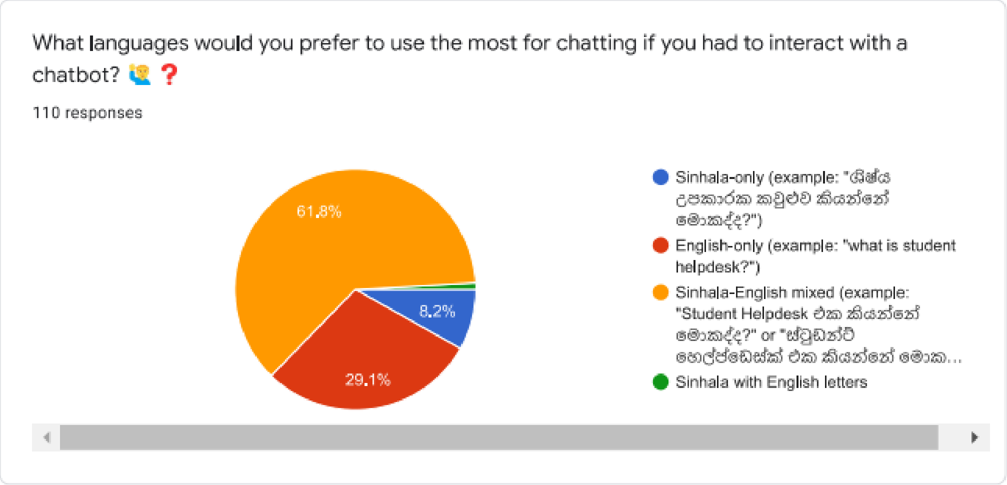
Do you prefer to find Information using chatbots or by surfing the internet? (for example: finding available degrees at SLIIT) 🌎

110 responses

Do you think using conversational AIs / Chatbots / Digital Assistants is a waste of time? ⏱

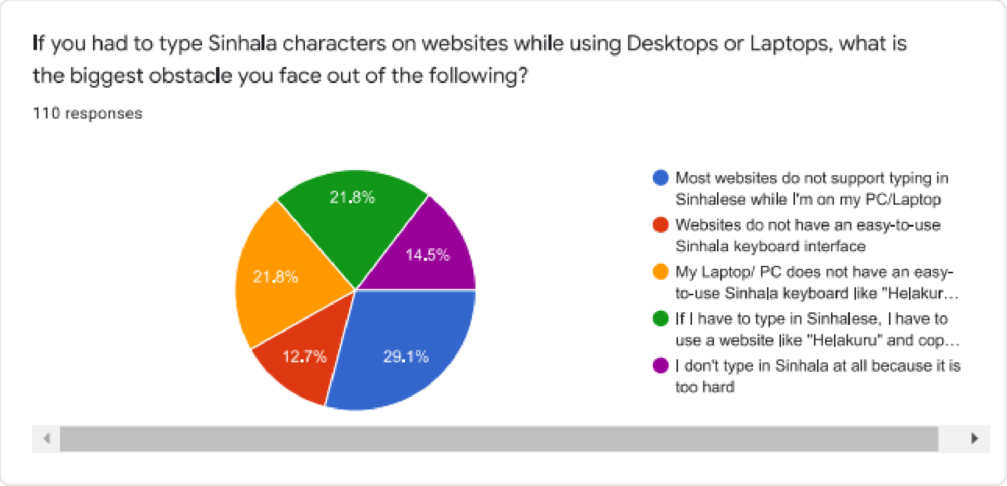
110 responses

Figure A.2: Complete survey form questions and responses – part 2



What languages would you prefer to use the most for chatting if you had to interact with a chatbot? 🙋 ❓

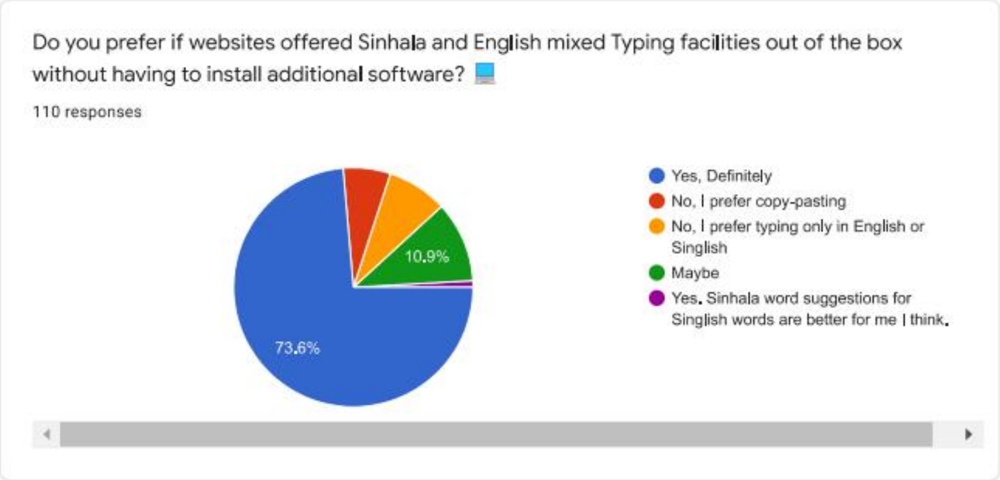
110 responses



If you had to type Sinhala characters on websites while using Desktops or Laptops, what is the biggest obstacle you face out of the following?

110 responses

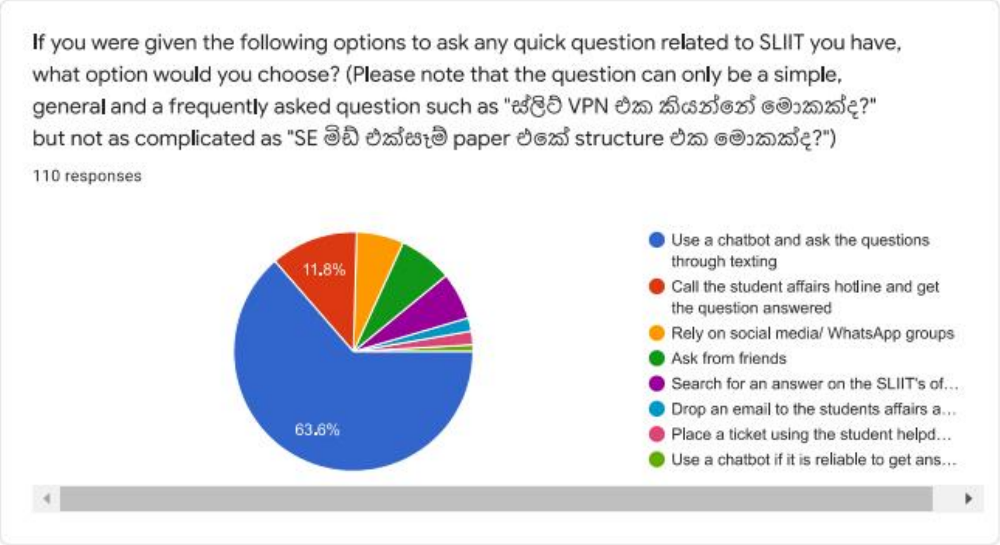
Figure A.3: Complete survey form questions and responses – part 3



Do you prefer if websites offered Sinhala and English mixed Typing facilities out of the box without having to install additional software? 💻

110 responses

10.9%
73.6%
● Yes, Definitely
● No, I prefer copy-pasting
● No, I prefer typing only in English or Singlish
● Maybe
● Yes. Sinhala word suggestions for Singlish words are better for me I think.

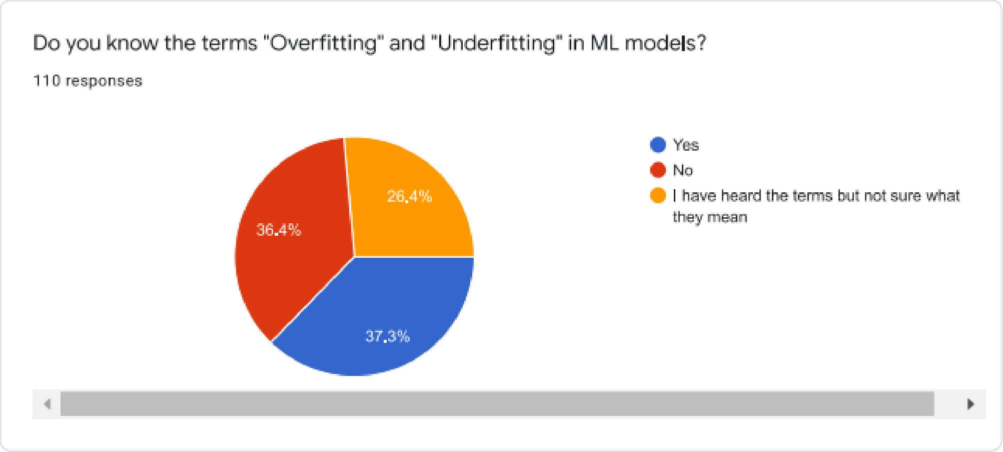


If you were given the following options to ask any quick question related to SLIIT you have, what option would you choose? (Please note that the question can only be a simple, general and a frequently asked question such as "ස්ලිට් VPN එක කියන්නේ මොකක්ද?" but not as complicated as "SE මිඩ් එක්සෑම් paper එකේ structure එක මොකක්ද?")

110 responses

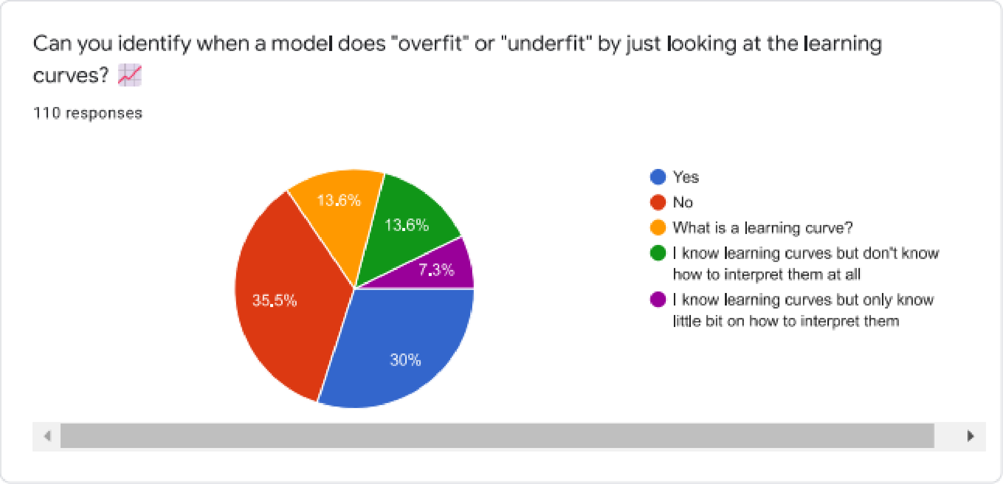
11.8%
63.6%
● Use a chatbot and ask the questions through texting
● Call the student affairs hotline and get the question answered
● Rely on social media/ WhatsApp groups
● Ask from friends
● Search for an answer on the SLIIT's of…
● Drop an email to the students affairs a…
● Place a ticket using the student helpd…
● Use a chatbot if it is reliable to get ans…

Figure A.4: Complete survey form questions and responses – part 4



Do you know the terms "Overfitting" and "Underfitting" in ML models?

110 responses

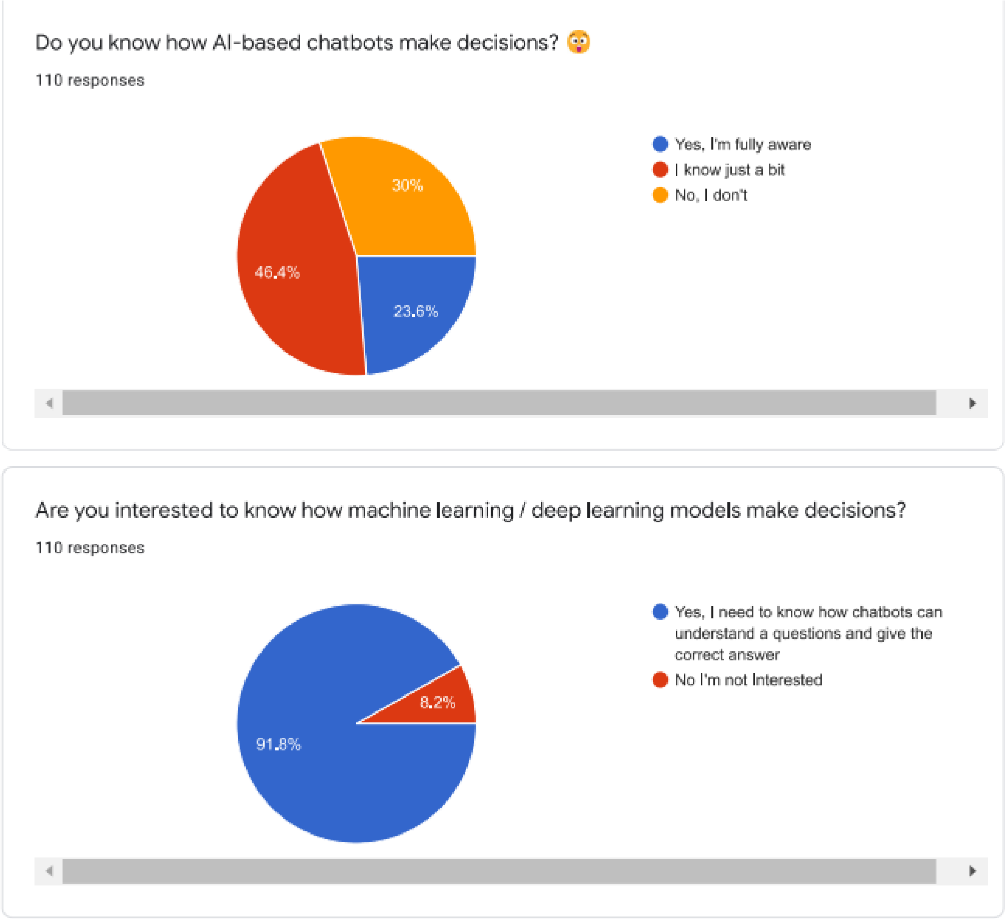


Can you identify when a model does "overfit" or "underfit" by just looking at the learning curves? 📈

110 responses

13.6%
13.6%
7.3%
35.5%
30%
● Yes
● No
● What is a learning curve?
● I know learning curves but don't know how to interpret them at all
● I know learning curves but only know little bit on how to interpret them

Figure A. 5: Complete survey form questions and responses – part 5



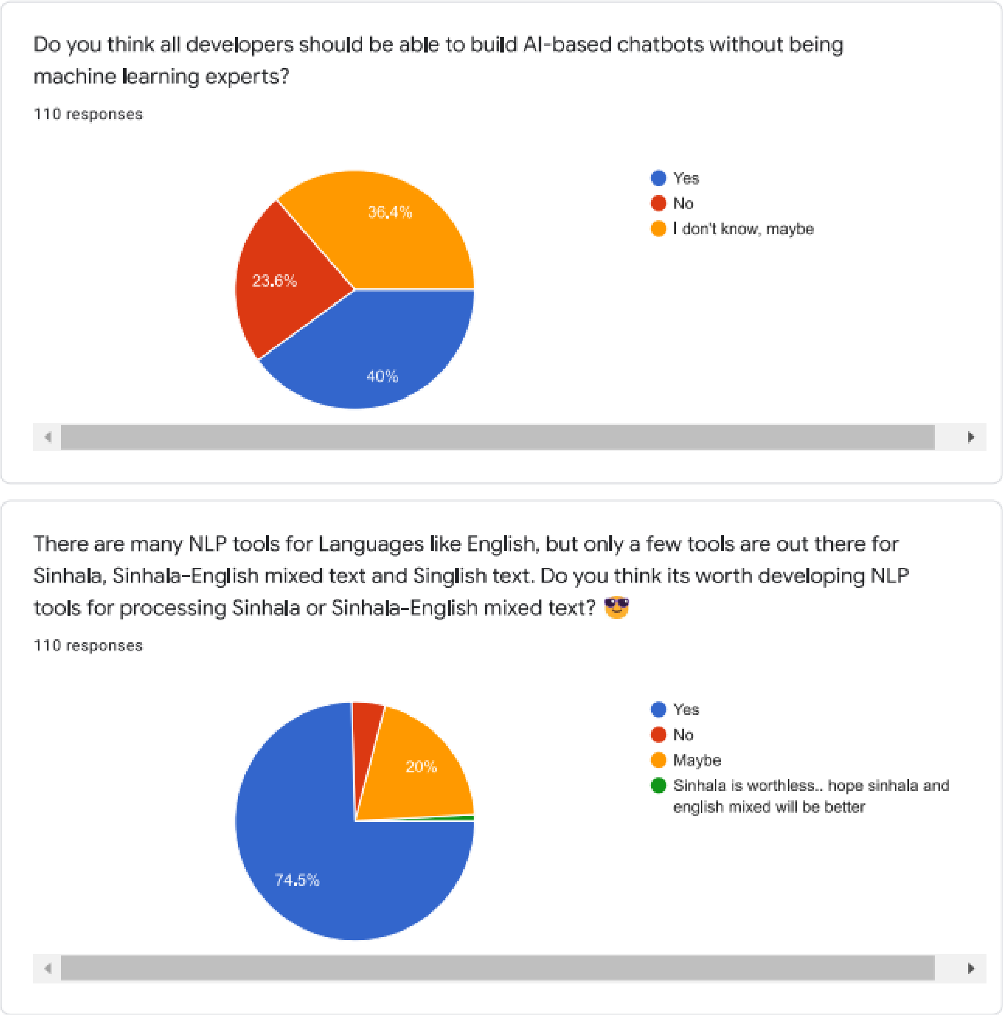
Do you know how AI-based chatbots make decisions? 😳

110 responses

Are you interested to know how machine learning / deep learning models make decisions?

110 responses

Figure A.6: Complete survey form questions and responses – part 6



Do you think all developers should be able to build AI-based chatbots without being machine learning experts?

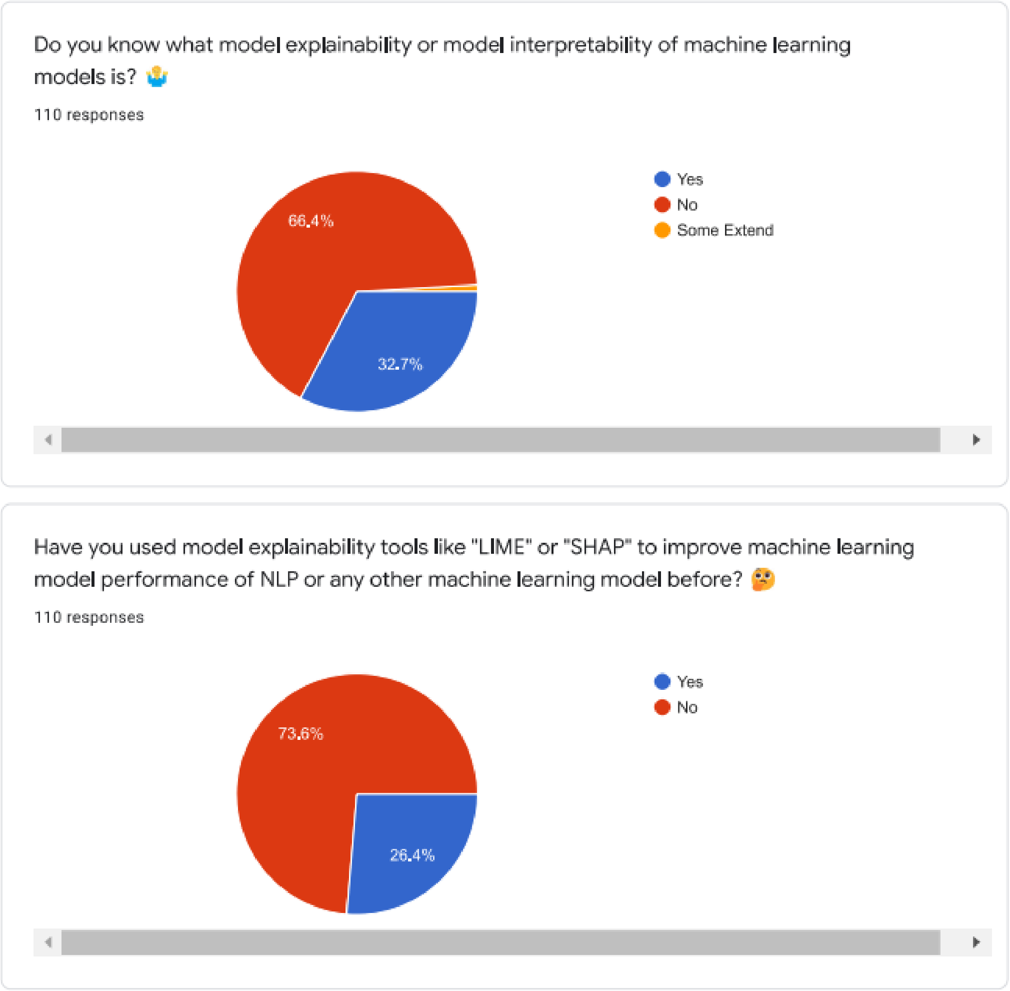
110 responses

There are many NLP tools for Languages like English, but only a few tools are out there for Sinhala, Sinhala-English mixed text and Singlish text. Do you think its worth developing NLP tools for processing Sinhala or Sinhala-English mixed text? 😎

110 responses

20%
74.5%
● Yes
● No
● Maybe
● Sinhala is worthless.. hope sinhala and english mixed will be better

Figure A.7: Complete survey form questions and responses – part 7



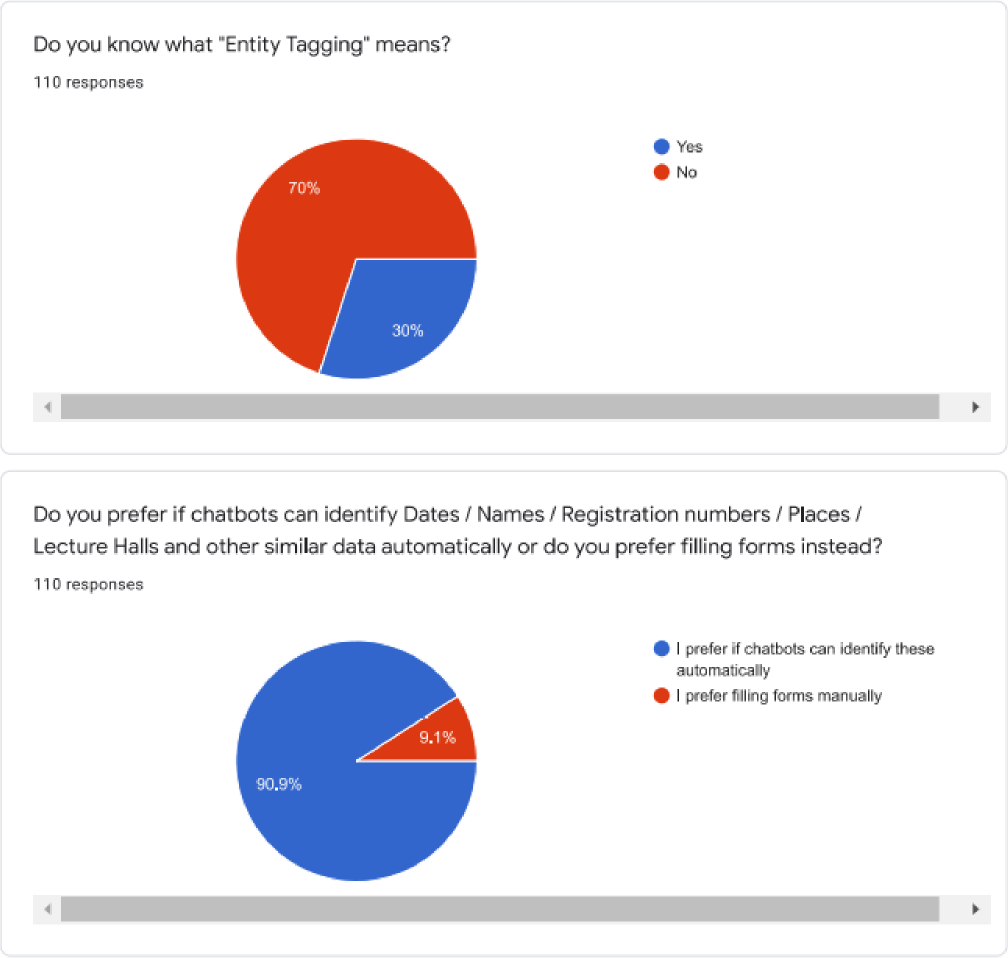
Do you know what model explainability or model interpretability of machine learning models is? 🤷

110 responses

Have you used model explainability tools like "LIME" or "SHAP" to improve machine learning model performance of NLP or any other machine learning model before? 🤔

110 responses

Figure A.8: Complete survey form questions and responses – part 8



Do you know what "Entity Tagging" means?

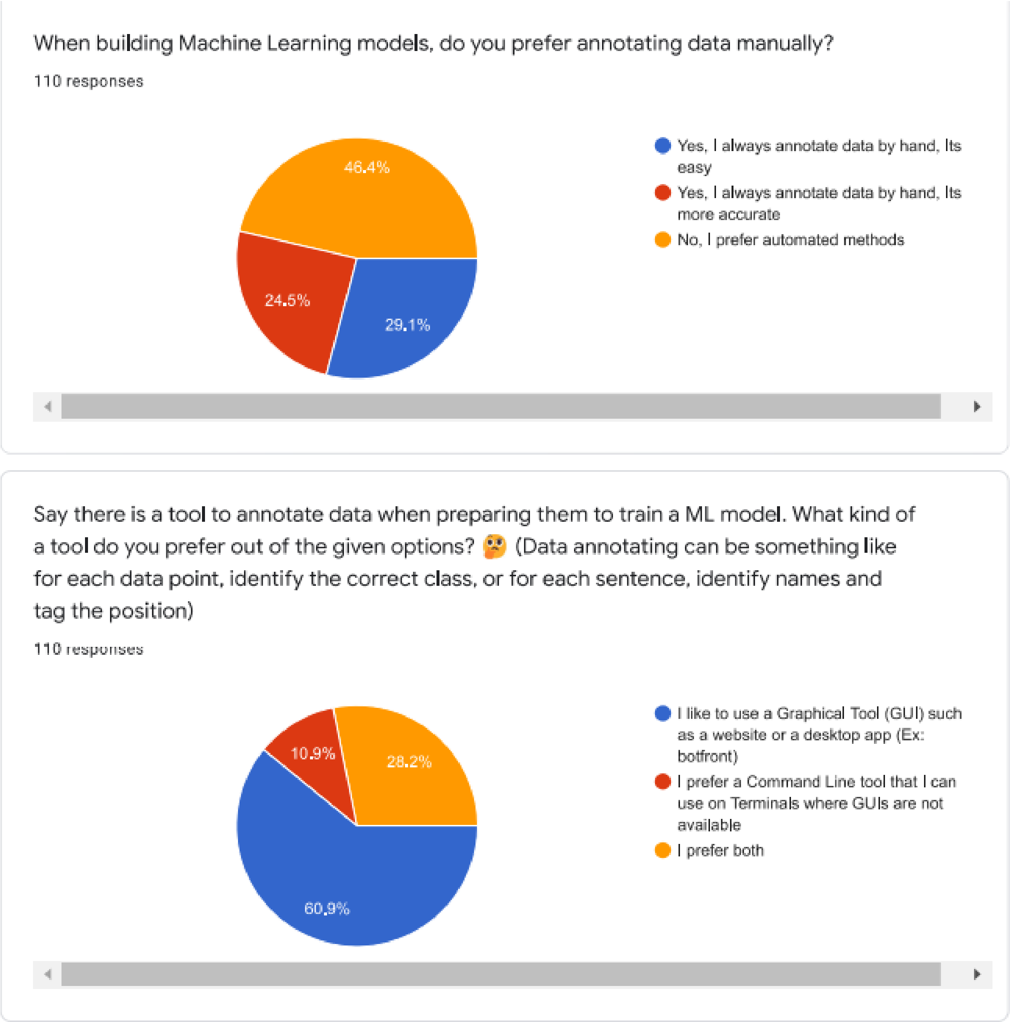
110 responses

Do you prefer if chatbots can identify Dates / Names / Registration numbers / Places / Lecture Halls and other similar data automatically or do you prefer filling forms instead?

110 responses

● I prefer if chatbots can identify these automatically
● I prefer filling forms manually
9.1%
90.9%

Figure A.9: Complete survey form questions and responses – part 9



When building Machine Learning models, do you prefer annotating data manually?

110 responses

● Yes, I always annotate data by hand, Its easy

● Yes, I always annotate data by hand, Its more accurate

● No, I prefer automated methods

Say there is a tool to annotate data when preparing them to train a ML model. What kind of a tool do you prefer out of the given options? 🤔 (Data annotating can be something like for each data point, identify the correct class, or for each sentence, identify names and tag the position)

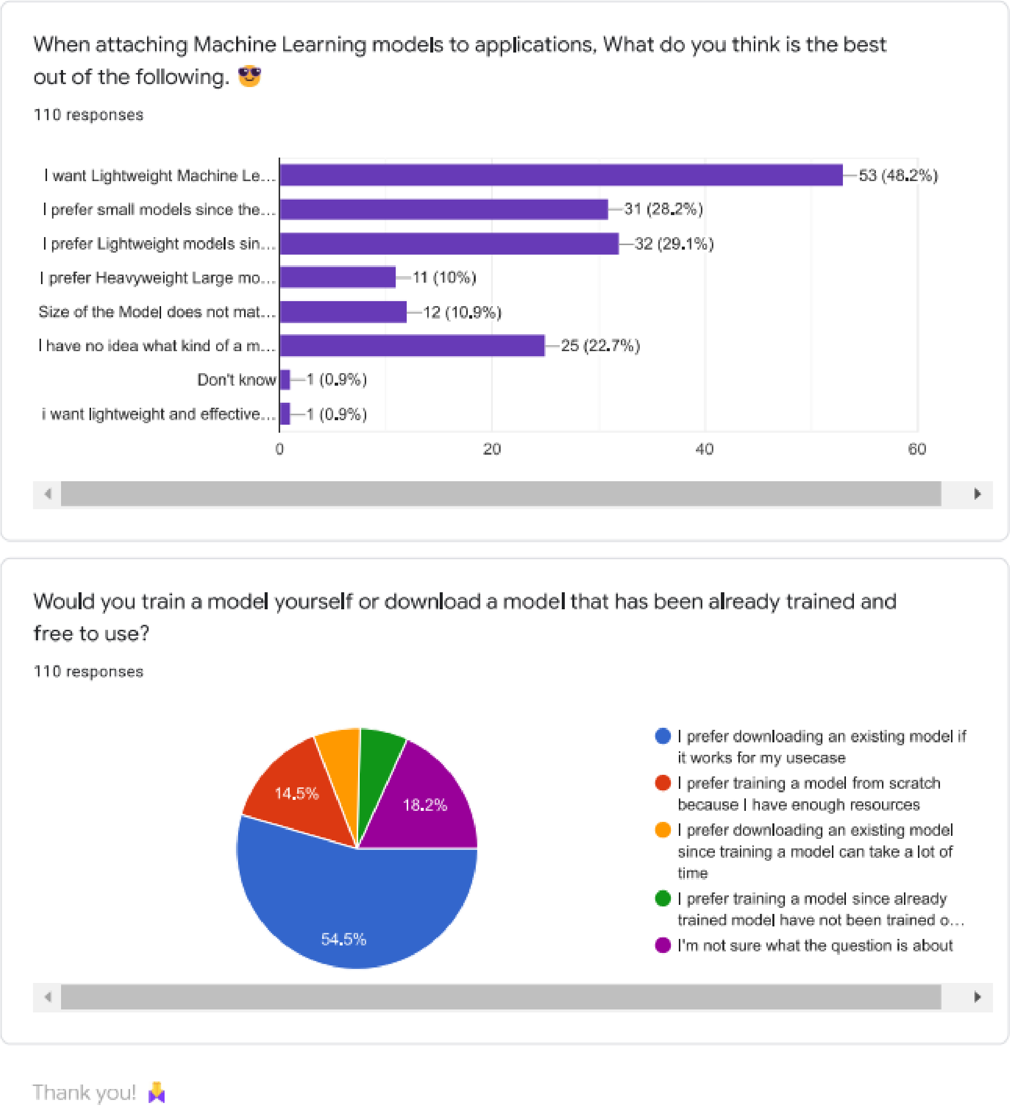
110 responses

● I like to use a Graphical Tool (GUI) such as a website or a desktop app (Ex: botfront)

● I prefer a Command Line tool that I can use on Terminals where GUIs are not available

● I prefer both

Figure A.10: Complete survey form questions and responses – part 10



When attaching Machine Learning models to applications, What do you think is the best out of the following. 😎

110 responses

I want Lightweight Machine Le…
I prefer small models since the…
I prefer Lightweight models sin…
I prefer Heavyweight Large mo…
Size of the Model does not mat…
I have no idea what kind of a m…
Don't know
i want lightweight and effective…
53 (48.2%)
31 (28.2%)
32 (29.1%)
11 (10%)
12 (10.9%)
25 (22.7%)
1 (0.9%)
1 (0.9%)
0
20
40
60

Would you train a model yourself or download a model that has been already trained and free to use?

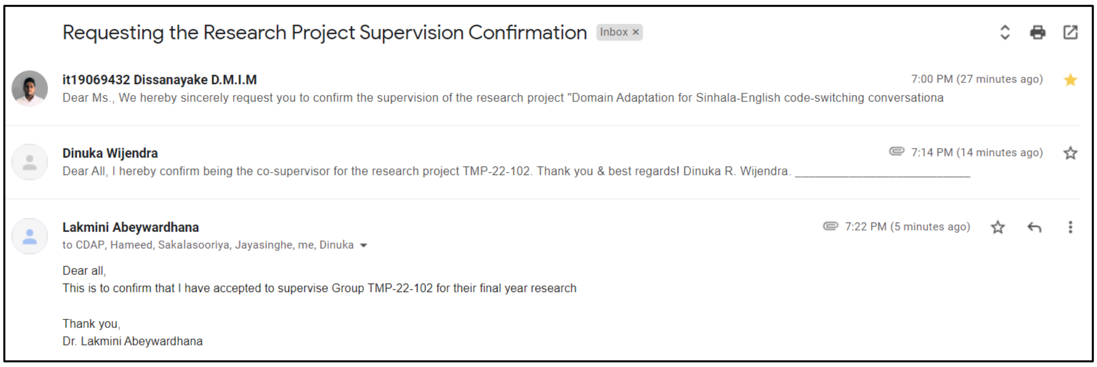
110 responses

14.5%
18.2%
54.5%
● I prefer downloading an existing model if it works for my usecase
● I prefer training a model from scratch because I have enough resources
● I prefer downloading an existing model since training a model can take a lot of time
● I prefer training a model since already trained model have not been trained o…
● I'm not sure what the question is about

Thank you! 🙌

Figure A.11: Complete survey form questions and responses – part 11

Appendix B: Supervision Confirmation Emails



Requesting the Research Project Supervision Confirmation

it19069432 Dissanayake D.M.I.M 7:00 PM (27 minutes ago)

Dear Ms., We hereby sincerely request you to confirm the supervision of the research project "Domain Adaptation for Sinhala-English code-switching conversationa

Dinuka Wijendra 7:14 PM (14 minutes ago)

Dear All, I hereby confirm being the co-supervisor for the research project TMP-22-102. Thank you & best regards! Dinuka R. Wijendra. ______________________________

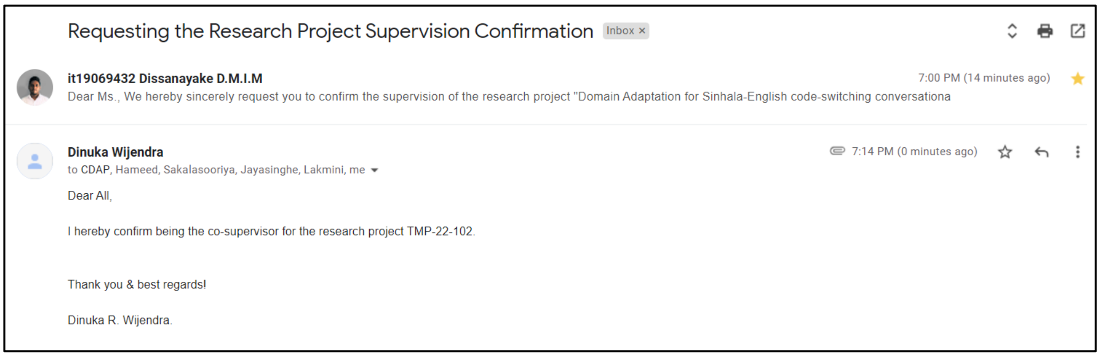
Lakmini Abeywardhana 7:22 PM (5 minutes ago)

to CDAP, Hameed, Sakalasooriya, Jayasinghe, me, Dinuka

Dear all,
This is to confirm that I have accepted to supervise Group TMP-22-102 for their final year research

Thank you,
Dr. Lakmini Abeywardhana

Figure B.1: Research project supervision confirmation email



Requesting the Research Project Supervision Confirmation

it19069432 Dissanayake D.M.I.M 7:00 PM (14 minutes ago)

Dear Ms., We hereby sincerely request you to confirm the supervision of the research project "Domain Adaptation for Sinhala-English code-switching conversationa

Dinuka Wijendra 7:14 PM (0 minutes ago)

to CDAP, Hameed, Sakalasooriya, Jayasinghe, Lakmini, me

Dear All,

I hereby confirm being the co-supervisor for the research project TMP-22-102.

Thank you & best regards!

Dinuka R. Wijendra.

Figure B. 2: Research project co-supervision confirmation email