S	
E1	
E2	
For Office Use Only	

Masters Project Final Report (MCS) 2019

Project Title	Singlish to Sinhala Converter using Machine Learning
Student Name	Anjali Diluni de Silva
Registration No. & Index No.	2016/MCS/025 16440254
Supervisor's Name	Dr. A.R. Weerasinghe

For Office Use ONLY

Singlish to Sinhala Converter using Machine Learning

A dissertation submitted for the Degree of Master of Computer Science

Ms. A.D. de Silva
University of Colombo School of Computing

2020

Declaration

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

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

Student Name: Ms. A. D. de Silva
Registration Number: 2016/MCS/025
Index Number: 16440254

____
Signature: Date: 21/06/2020

This is to certify that this thesis is based on the work of
Mr./Ms. A.D. de Silva
under my supervision. The thesis has been prepared according to the format stipulated and is of acceptable standard.

Certified by:
Supervisor Name: Dr. A. R. Weerasinghe

____
Signature: Date:

Abstract

In the modern world it is hard to successfully cope with each other throughout the entire system, without adopting modern technology. With the enhancement of the technology artificial intelligence play a crucial role in the society. Today most of the activities which involves with the human beings have been learnt by the machines and perform it as human brains perform them.

Machine transliteration is a process of converting a Romanized script into another language without considering the meaning of the word. It's a conversion between two types of alphabets. Even though English is considered to be a universal language, most of the people are not fluent in the English language. But still they know how to use the English alphabet. So people preferred to do the communication, using their native language. Even though Unicode characters are available for most of the language, people use English characters to communicate with each other. But not communicate in English. Typing the wordings using English characters but the meaning is from their native language. This process is very common among the today's world.

When it comes to Sri Lanka, most of the people are chatting by using Romanized Sinhala through social media. For example: "oyata kohomada?". There are lots of existing applications which converts Singlish characters to Sinhala fonts. But there are some applications which needs to perform analysis based on the data which collect through social media such as Hate Speech Detection. So in such circumstances it is required to convert an entire Romanized Sinhala script to Sinhala fonts. So for that purpose it is really beneficial to have Singlish to Sinhala converter which has been developed by training a model with a large number of Singlish and Sinhala phrase pairs.

So through this project, it has been achieved. Singlish to Sinhala converter has been developed by training a model using Long Short-Term Memory(LSTM) algorithm. It has been used six thousand Singlish and Sinhala pairs to train the model. The model's accuracy has been evaluated using BLEU score and it is around 40%.

Since the corpus consists of little number of data, the accuracy has been decreased. To have a better accuracy it is required to increase the number of phrases and train the model.

However, this particular converter will be beneficial for the community who are performing some analysis using Romanized Sinhala scripts. They do not want to spend time on perform the conversion manually. They can directly input the document which contains Romanized Sinhala and get the output as a document which has been converted the entire content to the Sinhala font.

Acknowledgement

I am using this opportunity to express my inmost gratitude and special thanks to Dr. A. R. Weerasinghe Senior Lecturer, University of Colombo School of Computing(UCSC) who in spite of being extremely busy with his duties, took time out to hear, guide and keep me on the correct path and allowing me to carry out my project successfully.

It is my radiant sentiment to place on record my best regards, deepest sense of gratitude to all the other lecturers and staff members of the UCSC for their careful and precious guidance which were extremely valuable for my study both theoretically and practically.

Also this is a great opportunity to thank all my friends for giving me immense encouragement and for all their suggestions throughout my individual project.

Nevertheless, I grant my gratitude towards my family for their kind co-operation and encouragement which helped me in completion of this project.

Table of Contents

Table of Figures

Chapter 01:

INTRODUCTION

1.1 Problem Statement

In the modern world it is hard to successfully cope with each other throughout the entire system, without adopting modern technology. Modern technology has become so vital that it has made the entire life system so ease that it can be done by the click of a button. Over the past decade, technological innovations facilitate the collection of consequential amounts of subjectively detectable data about essentially anybody who accesses material online.

Sri Lanka is a country which has a multiracial society. The people who lives Sri Lanka mainly use three languages. i.e Sinhala, Tamil and English. Most of the time people are using Sinhala and Tamil languages to communicate with each other. But today almost everything deals with the English language. Since English is a language which is considered to be a universal language, most of the countries are using English language. Therefore, most of the operations, communication systems and exchanging information are performing in English.

In Sri Lanka most of the people are communicating using Sinhala language. When they are communicating via social media such as Facebook, Whatsapp, Viber etc. they use Sinhala as the communication medium. Even though now already exists the applications which allows to type words directly using Sinhala fonts, most of the time people are typing the Sinhala words using English letters which is referred to as Romanized Sinhala or simply known as Singlish. In Sri Lanka most of the people are communicating via social media using Singlish. So when analyzing data which collected through online resources for some applications such as Hate Speech Detection, it is required to convert the Singlish wordings to the words written in Sinhala fonts, to gain an accurate output.

Therefore, to conduct a better analysis for some circumstances, it is required to convert the Romanized Sinhala scripts to the scripts which consists of pure Sinhala font.

1.2 Motivation

With the enhancement of the technology there are lots of systems which provides the translations for different languages which are widely used among the global community. There exist some applications which required human conversations gathered from the social media, as data for the development of those applications such as Hate Speech Detection. When considering the data gathered through social media as mentioned above, may require some sort of conversion, because in Sri Lanka most of the people are using Romanized Sinhala (Singlish) for chatting with the others. So, for the development of some applications (Hate Speech Detection etc.), it may require to convert Singlish into Sinhala fonts to gain more accuracy and make the lives easy for the developers of those applications.

And also this project’s outcome is beneficial for the people who doesn’t know to read or write Sinhala letters but understand and speak Sinhala language. If they know to type the Sinhala word using Singlish, then through this trained model he/she can get the corresponding Sinhala word using Sinhala fonts.

Therefore, in order to allow to gain the benefit of the technology by each and every individual, thought of this particular research to be carried out and propose a solution for the identified problem.

1.3 Overview of the Project

The solution which is named as ‘A Singlish to Sinhala Converter’ allows the users to give a conversation which has taken place using Singlish and then convert it to the exact Sinhala words using Sinhala fonts. This research will be carried out with the research areas of Natural Language Processing and Machine Learning.

The proof of the concept of the conversion of Singlish to Sinhala words will be the final outcome of this research project.

The proposed solution will be really beneficial for the people who are conducting some sort of analysis using Singlish words as well as for the applications such as Hate Speech Detection.

1.4 Objectives of the Project

- To establish a proof to convert Singlish typed 'anyway by anyone' in its natural form to Sinhala using machine learning approach.
- To identify the most appropriate Sinhala word expresses by the Romanized Sinhala(Singlish) words typed by using different spellings.
- To provide user friendly environment by adopting with new technologies.
- To gain reputation of being equipped with latest techniques in machine learning.

1.5 Scope of the Project

The deliverable of this project would be a proof of concept which converts Romanized Sinhala phrases into Sinhala phrases. The outcome of this product is really beneficial for the researchers who are collecting data through social media to conduct researches with respect to the Sinhala language. Because most of the people are using Romanized Sinhala for the communication via social media. Therefore, it would be beneficial if it is possible to convert the entire files containing the chats in Romanized Sinhala into Sinhala language. Then the converted files can be directly used for the researches with respect to the Sinhala language. But there’s a challenge with respect to the conversions. i.e. there can be some circumstances as in the same Sinhala word can be written by using different spellings in English. The proposed model is mapping the Singlish word directly to Sinhala word. With that mechanism, tried to obtain the most appropriate Sinhala word, expresses with different spellings in English.

1.6 Thesis Outline

The content of the thesis is organized as follows.

Chapter 01: Introduction

Provides an introduction to the topic of the thesis describing the problems and the motivations that are connected to the research area.

Chapter 02: Literature Review

Interprets the research background that has been referred throughout the entire development process of the proposed model.

Chapter 03: Methodology

Presents the development process of the proposed solution for the identified problem.

Chapter 04: Evaluation

Presents the steps followed on evaluating the built solution for a better accuracy.

Chapter 05: Discussion and Conclusion

Presents the achievement of the project objectives, discuss the final state of the built solution and the future enhancement for the project.

Chapter 02:

LITERATURE REVIEW

2.1 Overview of the Chapter

With the enhancement of the technology, most of the people are using online resources for the communication such as Whatsapp, Viber, Messenger, Skype etc. Since English is the universal language, the aforementioned social media use mainly English language. Even though English is the universal language, most of the people in Sri Lanka are not grammatically familiar with that language. But still they are using English letters to type Sinhala words which is known as Singlish. Even though Sinhala font is available now most of the people in Sri Lanka using Singlish for the communication. So when it is required to extract the exact Sinhala meaning of the word which is written by using Singlish, it is beneficial to have a converter which can be given the output in Sinhala when the Singlish word is given. These data extraction is really required for some applications such as Hate Speech Detection.

Since the technology is enhancing very significantly, it is really beneficial to adopt for the new technologies in order to become a part of the modern world and to make the people's lives easy. Today the people have started to make new inventions by identifying the problems with these language barriers. The entire world has become a part of that new inventions related to different languages.

With the aforementioned requirement most of the people in different countries have started on developing applications in order to cater with their native languages. Currently there exists lots of applications which provide solutions for the language barriers occurred when dealing with the communication around the world. Since this research project also based on some sort of similar thought, had to study some of the existing systems or applications on translating to different languages especially English-Sinhala translation applications available among the world. Since the main focus is based on Singlish-Sinhala translation, mainly conducted the literature review on the transliteration from one language to another.

The literature review based on the language transliteration applications, was really beneficial for the implementation of the concept behind "Singlish to Sinhala Converter". It was able to identify and understand well the requirements of a real language transliterating application which is to be developed using machine learning techniques. It was helpful to identify the further improvements, needed to be done for the proposed solution.

2.2 Transliteration

Transliteration has been defined as follows:

“Process of expressing the sound of how a word is pronounced in the source language in the alphabet of the target language”. [1]

“A type of conversion of a text from one script to another that involves swapping letters in predictable ways.” [2]

Translation and Transliteration are two different processes. In translation it changes the source script to a completely different language based on the meaning of the source script. But when compared to the transliteration, it is not bothered about the meaning of the romanized word. It considers only about the pronounciation. It converts a text from one script to the other based on the way it sounds.

There are mainly four basic models for transliteration.

1. Grapheme based transliteration model

   It maps source language graphemes or characters with the target language graphemes or characters directly. It’s not considering about any phonetic knowledge of the source lnguage words. It is known as orthographic process. [3]

2. Phoneme based transliteratin model

   It has an intermediary step to be followed when performing the transliteration. Firstly the source language graphemes or characters are mapped with the source phoneme/phonetic and then the source phoneme/phonetic is mapped with the target language grapheme or characters. It is mainly focusing on the pronounciation rather than the spellings of the source language. [3]

3. Hybrid based transliteration model

   It is a combination of grapheme based transliteration probability and phoneme based transliteration probability through linear interpolation. [3]

4. Combined/Correspondence based transliteration model

   It combines either any number of grapheme based models or any number of phoneme based models. But it is not combining both of the types together. [3]

Since the ‘Singlish to Sinhala converter’ is totally based on the concept of transliteration, it is really important to identify and understand the role of the transliteration. Based on the fndings of the literature review on transliteration, the proposed solution was followed the grapheme based model on implementation. Its directly converting source grapheme or characters into the target grapheme or characters.

2.3 Natural Language Processing (NLP)

Natural Language Processing (NLP) can be recognized as a field of artificial intelligence. It will provide all the access to identify and understand the human language. Usually computers will identify only the machine language. But according to the requirements of today’s world it is really important to understand the human language and process further according to it. Due to this NLP human will be able to interact with their computers by using their natural conversations without focusing on programming languages such as Java, C, C++ etc.

Main steps of NLP

1. Understanding the natural language received by the computer.

   Computer will convert the natural language into programming language by performing a speech recognition routine. This task will be achieved by using a statistical model. The first task can be known as speech to text process.
2. Part-of-Speech tagging (POS) / Word-category disambiguation.

   It will identify the words according to their grammar. i.e nouns, verbs, adjectives etc. It will use lexicon rules in order to code it to the computer.
3. Text-to-speech Conversion

   It converts the programming language into an audible format or text format.

Natural Language Processing will give the atmosphere for the users as they are interacting with another human not with a computer. Even though they actually interact with a computer it will not be felt by the users.[27]

Since this research project is based on the research area of Natural Language Processing, it is really important to have an idea on what is all about this NLP. Language transliteration falls on the area of NLP. [4] [5]

2.4 Machine Learning

Machine Learning can be identified as a field of study which gives the capability for a computer to learn without being programmed explicitly. Today most of the applications which are connected with the humans’ activities are developed by using machine learning. Lots of Machine Learning algorithms are available today such as Naïve Bayes, K-Means, Random Forest, Long Short-Term Memory etc. [26]

2.4.1 Long Short-Term Memory (LSTM)

LSTM is a kind of Recurrent Neural Networks(RNN). This contains a track of previous events. It’s having a loop. The same network copies again and again until it reaches to the successor. Following diagram, Figure 1 depicts the loop of the recurrent neural network. [28]

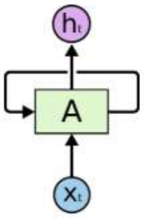


Figure 1: Recurrent Neural Network loop

In the above diagram, *A* refers to as a chunk of neural network. x_t is the input and h_t is the output. x_t goes through *A* over and over again until it reaches h_t successfully. The following diagram, Figure 2 illustrates how it looks like when unfold the loop.

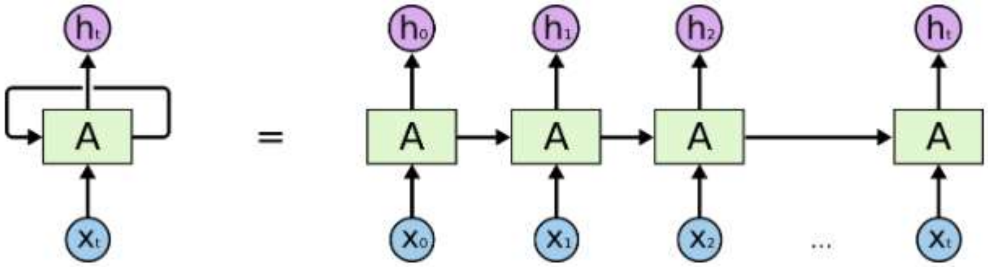


Figure 2: An unfolded Recurrent Neural Network

When look at the above diagram, it can be seen like it consists of recurrent neural networks as sequences and lists. But RNNs are not that much accurate for the long term dependencies.

But when comes to the LSTM, it’s a special type of RNN. It is skilled of learning long term dependencies. The repeating module in an LSTM comprises four interacting layers, while the repeating module in a standard RNN comprises only a single layer. The following two figures, Figure 3 and Figure 4 shows the difference.

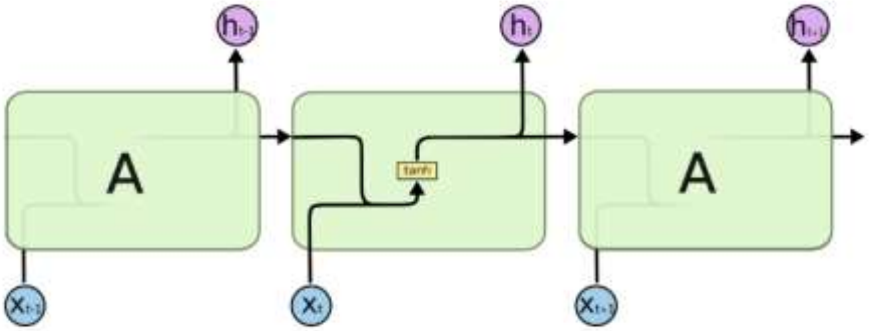


Figure 3: Repeating module in a Standard RNN

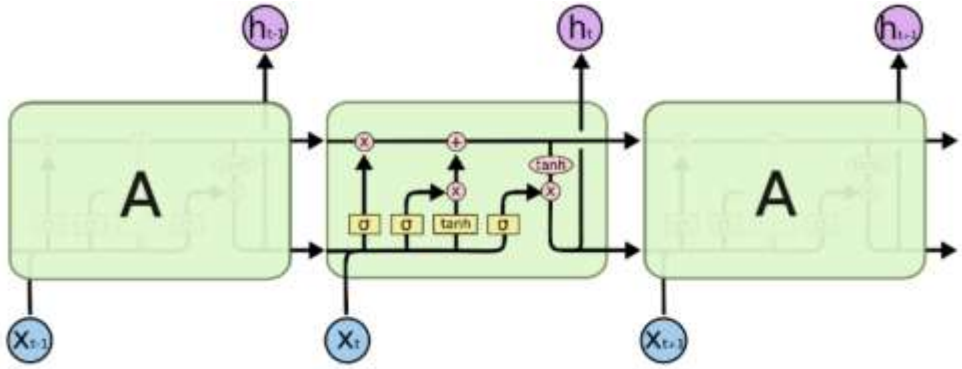


Figure 4: Repeating module in a LSTM

The symbols of the above two figures are denoted in the following Figure 5.

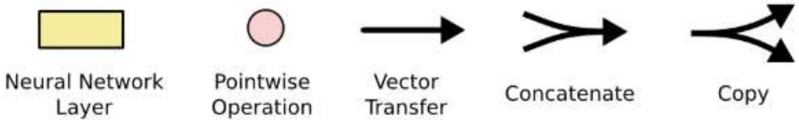


Figure 5: Meaning of the symbols in the RNN and LSTM diagrams

The above diagrams illustrate that each and every line transmits an entire vector from the output node to the inputs of the others. Steps followed in the LSTM model can be described as follows.

1. Decide regarding the information that throws away from the cell state. This decision is taken by the sigmoid layer which is known as *forget gate layer*. It considers about the h_{t-1} and give an output between 0 and 1.

   0 → completely get rid of this
   1 → completely keep this

2. Decide on what are the new information that have to be stored in the cell state. It consists of two parts.

   i) Sigmoid layer which is branded to be as *input gate layer* takes the decision on which values to be modernised.
   ii) *tanh* layer will create a vector for new candidate values which is required to be added to the state.

3. Update the old cell state into the new cell state.

4. Decide on what is to be taken as output. First need to run the sigmoid layer by deciding the parts of the cell state to be taken as the output. Then send the cell state through *tanh*. After that multiply it by the output of the sigmoid gate. Then it will give the output as it is required.

Since this project is based on machine learning which means, training the model to predict the future conversions of the Singlish words to Sinhala fonts, thought of choosing the LSTM model to train the system. The proposed solution is highly depending on the previous mappings of the Singlish words to Sinhala fonts. Since LSTM is capable of long term dependencies, thought of selecting LSTM to train the model. [6] [7]

2.5 Python Libraries

The entire model was built on Python language. To complete successfully had to use some important libraries in Python. A brief description on the used libraries as follows.

2.5.1 NumPy library

NumPy library is used when dealing with the arrays. It stands for Numerical Python. Even though python has lists, it is very slow when processing. But NumPy fifty times faster than the Python lists. [8]

2.5.2 Codecs registry

Codecs registry is used for encoding and decoding purposes. Codecs module together with the encodings package, supports the developers to overcome through the different language barriers. Codecs library was used to open and read files which contains Unicode Sinhala characters in this project. [9] [10]

2.5.3 Tensorflow and Keras libraries

Since the proposed solution is based on machine learning, it is totally depending on the data. In order to have an accurate output the data has to be preprocessed. For data preparation and tokenized the text, used Tensorflow and Keras libraries. [11][29][30]

2.5.4 Natural Language Toolkit (NLTK)

NLTK is a platform to build Python programs in order to work with the human language data. It allows to make machine understand the human language and give responses accordingly. [12]

2.6 BLEU score

Bilingual Evaluation Understudy(BLEU) score was used to evaluate the performance or the accuracy of the proposed solution. It is a score for comparing a candidate translation of text to one or more reference translations. The value of the score ranges between the 0 and 1. If the system has a perfect matching or accuracy, then the score will be 1.0. If the system does not have any single match, then the score will be 0.0. NLTK provides the *sentence_bleu()* function to evaluate a candidate sentence against one or more reference sentences. *Corpus_bleu()* function used to evaluate multiple sentences. i.e. paragraph or a document. So in this research the *Corpus_bleu()* function has been used to evaluate the system. [13]

2.7 Related Work

There have been conducted many researches based on these language transliterations and translations, since it is a crucial requirement in the current world. As a basement for this research it is really important and beneficial on focusing regarding the applications which have been developed based on the language transliterations as well as language translations.

Before move onto the transliterations, thought of get an idea regarding language translation systems. Because it was really beneficial to understand the concepts behind the translations performed on Sinhala language. The proposed solution deals with the Sinhala Unicode characters. Therefore, studied regarding the following existing applications based on **translations** on Sinhala language.

2.7.1 A Translator from Sinhala to English and English to Sinhala

It is a rule based machine translation system which allows to convert English to Sinhala and Sinhala to English. Main features included in this application are Sinhalese font translator, English grammar and spell checker. Sinhalese font translator enables to interpret Sinhalese word written in English letters which is known as Singlish. When translating Sinhala text to English, the user input value ought to be in Singlish form. It is not allowed to type directly in Sinhala on the interface. When translating English texts in to Sinhala then the user input should be given as an English statement.

Main objective of the application

Enable a smooth flow translation of words in order to eliminate the language barrier which can be occurred between the users.

Limitations of the system

In order to use this particular system anyhow it is required to know to play with the English characters. If it is required to translate Sinhala statements into English then it is required to give the user input using Singlish. That means should interpret the Sinhalese word written in English characters. Even for that the user should have a significant knowledge on using the English characters. [14]

2.7.2 Example Based Machine Translation for English-Sinhala Translations

It is an application which is used for English-Sinhala translations and has been created specially focusing on government domain. It uses bilingual corpus of English Sinhala and it is performing the translations mainly on sentence level. It will perform an intra-language matching. i.e when the sentence is given the system will retrieve the English sentence and give the corresponding Sinhala sentence as the output. Then the system performs a scoring algorithm on the Sinhala sentence in order to discover out the most occurring phrases in the sentence. And that phrase can be identified as the finest candidate translation for the phrase. When dealing with the translations it has found some of the problems. The system will give the most suitable translation for a given statement. The system allows even a text file to be entered and translate the entire text file by considering the one sentence at a time. The system has another feature. i.e. it allows to learn from past translations which have been done previously.

Limitations of the application

It provides only English to Sinhala translation. It’s not focusing on translating Sinhala to English. [15]

2.7.3 A Rule Based Syllabification Algorithm for Sinhala

It’s a study of Sinhala syllable structure and developed an algorithm for classifying syllables in Sinhala words. The algorithm has been tested for 30,000 distinct words in Sinhala. And the accuracy of it has been given as 99.95%.

Relevance to the research

Identifying the syllables of the Sinhala corpus is beneficial to the proposed solution. It's completely depending on the Sinhala words entered by the users using Singlish. Hope that the developed algorithm will be helpful to the proposed solution. [16]

Following existing applications are based on the **machine transliteration**.

2.7.4 Machine Learning based English-to-Korean Transliteration using Grapheme and Phoneme infomration.

It's grapheme and phoneme based transliteration model. When developing the model, it has considered about orthographical as well as phonetic converting process. It has compared with the previous grapheme based and phoneme based models using several machine learning techniques. This model has showed about 13~78% performance improvement. [17]

Relevance to the proposed solution:

Since the Singlish to Sinhala converter was developed using machine learning research area, this particular research paper gave an idea regarding grapheme based and phoneme based models. In the proposed solution, it has followed the grapheme based model.

2.7.5 Sinhala Grapheme-to-Phoneme Conversion and Rules for Schwa Epenthesis

This paper illustrates an architecture to convert Sinhala Unicode text into phonemic specification of pronunciation. This particular study was mainly focused on confusions schwa and /a/ vowel epenthesis for consonants. They have proposed some rules to overcome the disambiguates and it was tested using 30000 distinct words. The accuracy of the Grapheme-to-Phoneme conversion model is 98%. [18]

Relevance to the proposed solution:

This research paper provided an idea regarding the graphemes and phonemes and the conversion happens from grapheme to phoneme. Such concepts were applied in the proposed solution.

2.7.6 Rule based approach for transliteration of English to Tigrigna

The purpose of this article is to transliterate from English to Tigrigna language. It has discussed rule based approach for the transliteration. It has defined a collection of rules on grammar, lexicon. It has discussed regarding the issues on rule based approach as well. The developed system has been evaluated using word accuracy rate and it has performed as 90.9% accuracy. [19]

Relevance to the proposed solution:

It has provided rule based approach for the transliterations. The proposed solution also based on the transliteration concept. Therefore, the knowledge was gained regarding the transliteration with a different approach other than machine learning techniques.

2.7.7 A Deep Learning Approach to Machine Transliteration

In this paper it has been presented a novel transliteration technique which is based on deep belief networks. It has been shown some properties which can be used for transliteration and also for translation. The proposed system does not depend on word alignments and beam-search decoding. It consists of interesting properties regarding the reordering of sequences. It states that adding DBN-based transliterations are far behind the other approaches, it improves the overall results by 1%. [20]

Research gap between the existing solutions and the proposed solution:

There are lots of existing transliteration systems have been developed by using various methodologies or approaches. All of those approaches have been performed with differnet accuracies. Some of them are having really high accuracy and some of them have performed with lower accuracies. So among them still there are some methodologies or approaches which have to be tested whether they are suitable to perform transliterations such as Convolutional Neural Networks, Recurrent Neural Networks, Long-Short Term Memory(LSTM) etc. Therefore just thought of develop a model using LSTM technology to perform this transliteration. Further, according to the carried out literature review, it was identified that machine learning approach have not been used for the Romanized Sinhala transliteration. So according to the identified research gaps, thought of conducting the research on ‘A Singlish to Sinhala converter using Machine Learning’.

Chapter 03:

METHODOLOGY

This system has been developed by considering the grapheme based transliteration model. That means it directly converts the source language graphemes or characters to the target language graphemes or characters. The developed system consists of two main parts. i.e. data preparation and machine transliteration. The overall system architecture can be illustrated by the following diagram: Figure 6.

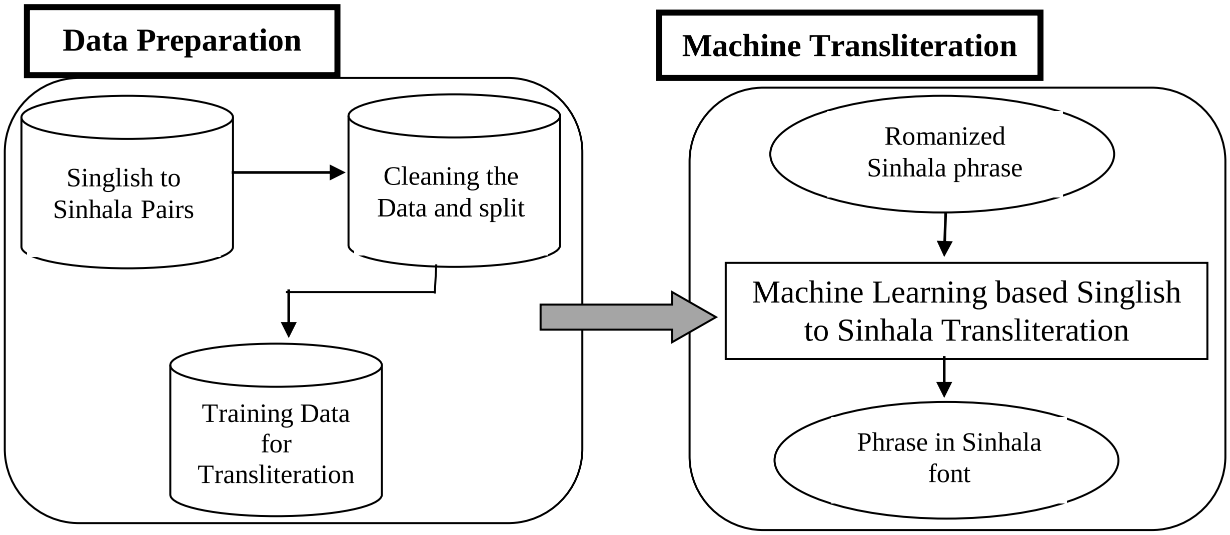

Data Preparation
Singlish to Sinhala Pairs
Cleaning the Data and split
Training Data for Transliteration
Machine Transliteration
Romanized Sinhala phrase
Machine Learning based Singlish to Sinhala Transliteration
Phrase in Sinhala font

Figure 6: Overall System Architecture

3.1 Data Preparation

Since this system is based on machine learning, it’s highly depending on the data which has been collected. In order to make the predictions more accurate, it is required to have more and more words written in Romanized Sinhala.

3.1.1 Gathering Data

For this project, the required data collected from a secondary source. Those data were gathered to develop an application on hate speech detection. Those data have been gathered through the conversations taken place through social media such as Whatsapp, Viber, Messenger etc. It consists of two thousand and five hundred entire sentences. But it was a combination of Singlish sentences as well as the sentences which has been written using Sinhala characters. Since the objective of the project was to convert Singlish words to Sinhala characters, the sequences which has been written using Sinhala fonts had to remove from the data set. After removing all the pure Sinhala characters, it was ended up with six thousand Singlish words and that particular data set was chosen to train the model.

3.1.2 Clean the Data

According to the proposed solution it was required to map each Singlish word with the word written in Sinhala characters. This task was done by annotating each and every Romanized Sinhala word to Sinhala font manually. With that, the corpus was built with the pairs of Singlish word and Sinhala word.

The data set has to be loaded in order to clean it. The data was loaded by using the *load_doc()* in python and the file was loaded as a blob of text. The file consists of Unicode characters. Therefore, when opening the file, utf-8 encoding type was used. In the loaded file it consists of lines with a pair of phrases as Singlish and then Sinhala, separated with a tab character. It was required to fragment the loaded text firstly by line and then by phrase. To achieve it *to_pairs()* function was used.

The chosen data consists of not only the letters, but it consists of lots of unwanted characters such as question marks, delimiters, full stops etc. Therefore, it was required to remove those unwanted characters from the data set. That particular task was accomplished through the functions *maketrans()* and *translate()* as mentioned below.

```
 table = str.maketrans('', '', string.punctuation)
# remove punctuation from each token
 line = [word.translate(table) for word in line]
```

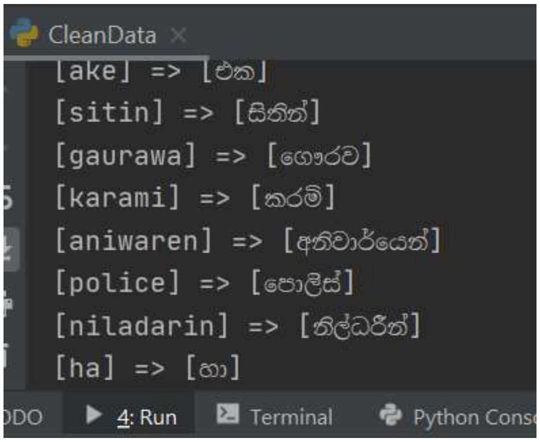
Further the data set consists of the words with uppercase as well as lowercase together. Therefore, it was required to transform the entire corpus to either lowercase or uppercase. So in this case, it was transformed to the lowercase by using the *lower()* function. By using the *split()* function each and every line of the data set was tokenized on white space.

Since it involves with the Unicode characters, it was required to normalize all Unicode characters, and it was done by using the following code segment.

```
line = normalize('NFD', line).encode('utf-8')
line = line.decode('UTF-8')
```

Performing the aforementioned operations on the data set, it was cleaned as required. The clean data set was saved in a separate file. To save the clean text to a file it was used the pickle API. That means clean text was saved as a byte stream in a new file.

Sample output of the cleaned data, illustrates in the following Figure 7.



```
[ake] => [එක]
[sitin] => [සිතින්]
[gaurawa] => [ගෞරව]
[karami] => [කරමි]
[aniwaren] => [අනිවාර්යෙන්]
[police] => [පොලිස්]
[niladarin] => [නිල්ධරීන්]
[ha] => [හා]
```

Figure 7: Cleaned text sample

3.1.3 Split Text

The cleaned data required to divide into two categories as training set and the testing set. The model has to be trained using training data set and the model should be tested by using the testing data set. The six thousand phrase pairs were separated as five thousand eight hundred for training and the remaining for the testing. The data set was shuffled before it was splitting. The splitting process was achieved as follows.

```
train, test = dataset[:5800], dataset[5800:]
```

Once the splitting was successfully done, the training data set and testing data set were saved separately in two files. Those two files also saved as pickle files.

3.1.4 Train the model

In order to train the model, it is required to load each and every data set which has been saved separately as training and testing data set. Since it is required to map words with the integers for modelling, used separate tokenizer for the Singlish sequence and the Sinhala sequence. To create the tokenizers, used Keras Tokenize class.

```
def create_tokenizer(lines):
    tokenizer = Tokenizer()
    tokenizer.fit_on_texts(lines)
    return tokenizer
```

When preparing the tokenizers, it considered the maximum length of the phrases and the vocabulary size of the phrases.

Since input and output sequence should be encoded to integers and padded to the maximum phrase length, used a word embedding for the input sequences and one hot encode the output sequences. Encode and pad sequences task were achieved by the following code segment.

```
def encode_sequences(tokenizer, length, lines):
    # integer encode sequences
    X = tokenizer.texts_to_sequences(lines)
    # pad sequences with 0 values
    X = pad_sequences(X, maxlen=length, padding='post')
    return X
```

The model required to predict the probability of each word in the vocabulary as output. For this purpose, the output sequence needs to be one-hot encoded. This task was achieved through the following code.

```
def encode_output(sequences, vocab_size):
    ylist = list()
    for sequence in sequences:
        encoded = to_categorical(sequence, num_classes=vocab_size)
        ylist.append(encoded)
    y = array(ylist)
    y = y.reshape(sequences.shape[0], sequences.shape[1], vocab_size)
    return y
```

By using the aforementioned two functions, prepared the train and test dataset to train the model.

3.2 Machine Transliteration

For this new proposed solution as the model, used encoder-decoder Long Short Term Memory(LSTM) model. Since it is a kind of Recurrent Neural Network, it allows to learn order dependence in sequence prediction problems. Machine transliteration is heavily based on the previous mappings of the words, LSTM model is used to achieve this task. In this model, the front-end model encodes the input sequence and the backend model decodes word by word. The function *define_model()* was used to configure the model. In this model, it has been used the Adam optimization algorithm. Adam optimizer has been used with the intention of updating network weights in training data based on iterations. It minimizes the categorical loss function. Code segment to define the model as follows.

```
def define_model(src_vocab, tar_vocab, src_timesteps, tar_timesteps, n_units):
        model = Sequential()
        model.add(Embedding(src_vocab, n_units, input_length=src_timesteps,
                        mask_zero=True))
        model.add(LSTM(n_units))
        model.add(RepeatVector(tar_timesteps))
        model.add(LSTM(n_units, return_sequences=True))
        model.add(TimeDistributed(Dense(tar_vocab, activation='softmax')))
        return model
```

The model has been trained for 30 epochs and a batch size of 64. To make sure that the model skill is improving on each time, it has used *ModelCheckpoint()* function.

The plot of the model can be illustrated by the following Figure 8. That particular figure is created through the *plot_model()* function.

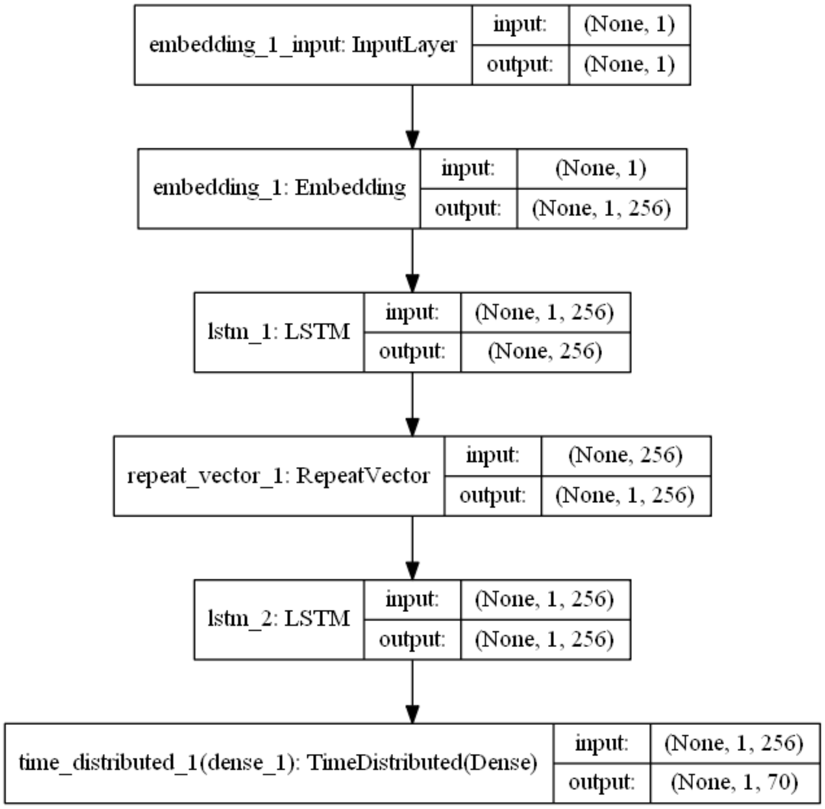


Figure 8: Plot of the model

Chapter 04:

EVALUATION

Evaluation on this project has been conducted based on the experiments. Experiments were carried out based on a secondary dataset which was collected by a researcher for a hate speech detection application. The cleaned data set was used for the experiments. It contains Romanized Sinhala word and its corresponding standard Sinhala word. Test set is composed of two Singlish Sinhala transliteration pairs.

Evaluation process was done in several phases throughout this project life cycle. i.e. evaluation was carried out from the data collection until the implementation of this project. It can be categorized under the following phases:

1. Evaluation carried out with respect to the initial data collection.

   As mentioned previously in the Methodology chapter, data was gathered as a secondary data set. At the initial data set it was contained only the phrases written in Romanized Sinhala (Singlish). The entire phrases were converted manually to Sinhala. Since they were converted manually, there was a high chance to contain mismatching phrases with its corresponding conversion. Therefore, to evaluate it, the converted content was checked by five people, one after the other.

2. Evaluation carried out on the data set used in model training.

   Initially the model was tried to train on the entire sentences available in the data set. But it was unable to get the output as required. Therefore, just tried to get the predicted value word by word, by considering the transliteration on word by word. So to train the model, it was required to organize the dataset as word by word transliteration. So based on that requirement, the newly prepared data set was to be evaluated again. That particular data set also was verified by another different five people, one after the other. Then used the output gained by the fifth person was considered as the data set to train the model.

3. Evaluation carried out on literature review.

   Since this project was totally based on machine learning techniques, it was required to select a suitable machine learning technique to train the developed model. Therefore, the methodologies used in the existing applications were critically evaluated by reading the articles thoroughly and selected the Long-Short Term Memory(LSTM) technique to train the model. Similarly, all the other technologies which have been used in the implementation of this proposed solution, were identified by conducting a critical evaluation on the existing researches carried out with respect to the machine transliteration and translation.

4. Evaluation carried out on the final outcome.

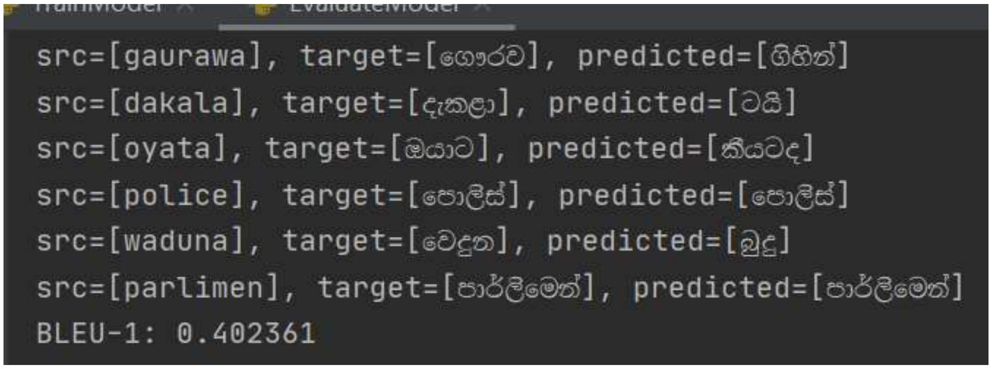
   The model predicts the entire output sequence using the following code segment.

   ```
   translation = predict_sequence(model, sin_tokenizer, source)
   ```

It will give a sequence of integers which allows to enumerate and use the tokenizer to map back to words. The abovementioned reverse mapping can be performed using the following code segment.

```
def word_for_id(integer, tokenizer):
    for word, index in tokenizer.word_index.items():
        if index == integer:
            return word
    return None
```

The trained model shows the source word and the target word as well as the predicted word. The output of the system received is depicted in the following Figure 9.



```
src=[gaurawa], target=[ගෞරව], predicted=[ගිහින්]
src=[dakala], target=[දැකළා], predicted=[ටයි]
src=[oyata], target=[ඔයාට], predicted=[කීයටද]
src=[police], target=[පොලිස්], predicted=[පොලිස්]
src=[waduna], target=[වෙදුන], predicted=[බුදු]
src=[parlimen], target=[පාර්ලිමෙන්ත්], predicted=[පාර්ලිමෙන්ත්]
BLEU-1: 0.402361
```

Figure 9: Output of the proposed solution

Evaluation is performed by Bilingual Evaluation Understudy(BLEU) score. It is a score to compare a candidate translation text to one or more reference translations. Through the Natural Language Toolkit(NLTK) library in python, used BLEU score to evaluate the outcome quantitatively. In this research *corpus_bleu()* function used to calculate the BLEU score. This sore has been used, because it can be used to evaluate multiple sentences such as a paragraph or a document. It was used as follows.

```
print('BLEU-1: %f' % corpus_bleu(actual, predicted, weights=(1.0, 0, 0, 0)))
```

Here it has used 1-gram score which means matches single words. It can be specified by using the *weight* argument with the *corpus_bleu()* function.

The value obtained for the BLEU score is 0.402361. Which means actual values of the corpus matches with the predicted values 0.402361 around 40%. This particular value can be considered as the accuracy of the developed solution for the identified problem.

Chapter 05:

DISCUSSION AND CONCLUSION

5.1 Overview of the chapter

This chapter includes a critical discussion about the project which was carried out throughout the year. The report consists of five chapters and within those each and every single chapter it covered the entire critical points which were related for this “A Singlish to Sinhala Converter using Machine Learning” development.

By considering all those factors this chapter will carry out a discussion as well as it will give a conclusion by mentioning the future enhancements that can be undertaken in future to make this product more efficient and accurate.

5.2 Results and Discussion

This project has been developed as a solution for the identified problem which has been stated in the chapter 01. When consider about the results of the evaluation conducted on this system, it shows that the prediction value is not that much accurate when compared to the actual value. As shown in the previous chapter, the BLEU score value obtained was around 40%. That particular value cannot be considered as a good value for this system. This value means that the words in the testing corpus gives the accuracy of only about 40%.

To train this transliteration model, it has been used six thousand phrase pairs. Based on the BLEU score, it can say that it is required to have more number of data to train the model. Current number of phrases are not enough to train this model to give the output accurately. Therefore, if the number of phrases included in the corpus increase, then the value of the BLEU score can be increased. That means the accuracy can be increased with the increment of the number of phrases included in the corpus.

Once the number of phrases increase in the corpus, then the model will be able to train more and more with the mappings of the Singlish words and the Sinhala words. Since this model has been trained based on the encoder-decoder LSTM model, the outcome of one layer is depending on the previous layer’s outcome. The model is training layer by layer. In order to have a better outcome this training has to be performed more and more. Therefore, to perform a better training it is required to have more data.

When considering about the objectives of this research, it has covered the objectives into some extent. Even though the same Sinhala word typed in English letters with different spellings, it was tried to give the correct Sinhala word as the output. Even this objective can be achieved, if the count of the data set is increased.

5.3 Challenges faced

When developing this project, had to face for some challenges due to various reasons.

5.3.1 Parallel corpora size

The major bottleneck for this research was the number of phrases used. To have a better accuracy, it is required to have around millions of phrases. In this project it has been used only six thousand phrases to train the model. Since it is a low count to train a model, it is not giving that much of accuracy on this developed model.

5.3.2 Different spellings for the same Sinhala word

As the input it considered the Romanized Sinhala phrases which means, the Sinhala words have been written by using the English characters. So the same Sinhala word can be written in different ways by different users. So the model had to train for such circumstances as well. It was achieved for some extent, but still it is required to have more such to train the model for such situation. Due to the lack of data the outcome was not that much accurate.

5.3.3 Memory requirement

When increasing the number of data to be trained, it was very time consuming. It took a long time to train the model. To run the evaluation also it took more time.

5.4 Future Enhancements

However, there is further work to be done for this “Singlish to Sinhala converter using Machine learning” model.

- Since the accuracy of the model is low, need to carry out more training on the data to increase the accuracy. Therefore, it is required to gather and clean more Romanized Sinhala phrases through social media.

- Should develop a front end to this model. It should allow the users to get the benefits out of this trained model. To achieve it, hoping to develop a web browser plugin, which allows the users to enter their entire file which consists of Romanized Sinhala phrases and give the output as a file in Sinhala fonts.

5.5 Conclusion

“A Singlish to Sinhala converter” has been developed mainly focusing on the users who are performing analysis by using Singlish phrases for some other applications. Currently there exists some Singlish to Sinhala converters built by using different methodologies. But this converter has been developed by using machine learning techniques. It has been trained a model and predict the values later on based on the input values.

This system would be beneficial for the users who wants to convert an entire file which consists of the content in Romanized Sinhala to Sinhala font. Then their time would be saved on performing different types of analysis. Otherwise, they have to convert word by word to Sinhala font. It’s really time consuming. So it will contribute a positive impact for the public who are willing to build different types of applications using Sinhala language.

By considering the above mentioned future enhancements, if it is able to perform those enhancements on this model, then it would be a great product for the community. But there’s a concern regarding the accuracy of the system because of the number of data that has been used to train the model. If the size of the data set is increased, then it might increase the accuracy rate of the model.

This entire thesis consists of the areas which were carried out throughout the development process of the “A Singlish to Sinhala converter using machine Learning” in detail. Therefore, whoever continues on working towards for the future enhancements the details that are included within this thesis will become really beneficial.

Finally, all the tasks are covered in a sequence and a methodical manner and it can be said that this document consists of all the necessary information to implement the “Singlish to Sinhala converter” successfully.

References

[1] “What is transliteration and how is it different to translation?,” 10 November 2014. [Online]. Available: https://www.londontranslations.co.uk/faq/what-is-a-transliteration-and-how-is-it-different-to-a-translation/. [Accessed 12 January 2020].

[2] “Transliteration,” [Online]. Available: https://en.wikipedia.org/wiki/Transliteration. [Accessed 12 January 2020].

[3] M. L. Dhore, R. M. Dhore and P. H. Rathod, “Survey on Machine Transliteration and Machine Learning Models,” *International Journal on Natural Language Computing,* vol. IV, no. 2, 2015.

[4] “NAtural Language Processing,” 7 March 2018. [Online]. Available: https://www.investopedia.com/terms/n/natural-language-processing-nlp.asp. [Accessed 2 August 2019].

[5] J. Brownlee, 22 September 2017. [Online]. Available: https://machinelearningmastery.com/natural-language-processing/. [Accessed 15 August 2019].

[6] C. blog, “Understanding LSTM networks,” 27 August 2015. [Online]. Available: https://colah.github.io/posts/2015-08-Understanding-LSTMs/. [Accessed 31 January 2020].

[7] J. Brownlee, “A gentle introduction to Long Short-Term Memory Networks by the Experts,” 24 May 2017. [Online]. Available: https://machinelearningmastery.com/gentle-introduction-long-short-term-memory-networks-experts/. [Accessed 1 February 2020].

[8] “NumPy Introduction,” [Online]. Available: https://www.w3schools.com/python/numpy_intro.asp. [Accessed 5 January 2020].

[9] “Unicode,” [Online]. Available: https://docstore.mik.ua/orelly/other/python/0596001886_pythonian-chp-9-sect-6.html. [Accessed 10 January 2020].

[10] “Codec registry and base classes,” [Online]. Available: https://docs.python.org/2/library/codecs.html. [Accessed 20 January 2020].

[11] “Tokenization and Text Data Preparation with Tensorflow & Keras,” [Online]. Available: https://www.kdnuggets.com/2020/03/tensorflow-keras-tokenization-text-data-prep.html. [Accessed 5 February 2020].

[12] “NLTK tutorial in python,” [Online]. Available: https://www.guru99.com/nltk-tutorial.html. [Accessed 10 February 2020].

[13] J. Brownlee, “A gentle introduction to calculating the BLEU score for text in Python,” 20 November 2017. [Online]. Available: https://machinelearningmastery.com/calculate-bleu-score-for-text-python/. [Accessed 5 Febrary 2020].

[14] L. Wijerathna and S. L. K. W L S L Somaweera, “A Translator from Sinhala to English and English to Sinhala,” in *International Conferences on Advances in ICT for Emerging Regions (ICTer2012),* Colombo, 2012.

[15] A. Silva and R. Weerasinghe, “Example Based Machine Translation for English-Sinhala Translations,” in *International Conferences on Advances in ICT for Emerging Regios(ICTer2016),* Colombo, 2016.

[16] R. Weerasinghe, A. Wasala and K. Gamage, “A Rule Based Syllabification Algorithm for Sinhala,” in *IJCNLP'05 Proceedings of the Second International joint conference on Natural Language*

Processing, Korea, Springer-Verlag Berlin, Heidelberg, 2005, pp. 438-449.

[17] O. Jonh-Hoon and C. Key-sun, “Machine Learning based English-to-Korean Transliteration using Grapheme and Phoneme information,” in *IEICE Transactions on Information and Systems*, 2005.

[18] A. Wasala, R. Weerasinghe and K. Gamage, “Sinhala Grapheme-to-Phoneme Conversion and Rules for Schwa,” in *21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, Australia, 2006.

[19] G. Josan and G. Hailu, “RULE BASED APPROACH FOR TRANSLITERATION OF ENGLISH TO TIGRIGNA,” 2017.

[20] T. Deselaers, S. Hasan and O. Bender, “A Deep Learning Approach to Machine Transliteration,” in *StatMT '09: Proceedings of the Fourth Workshop on Statistical Machine*, 2009.

[21] L. Katragadda, P. Deshpande, A. Dutta and N. Arora, “MAchine Learning for Transliteration,” in *United States Patent Application Publication*, Unites States, 2008.

[22] “The Python Tutorial,” [Online]. Available: https://docs.python.org/3/tutorial/. [Accessed 29 February 2020].

[23] “Python Tutorial,” [Online]. Available: https://www.w3schools.com/python/. [Accessed 28 February 2020].

[24] “Python Tutorial,” [Online]. Available: https://www.tutorialspoint.com/python/index.htm. [Accessed 25 February 2020].

[25] “4 Machine Learning Techniques You Should Recognize,” 16 November 2017. [Online]. Available: https://blogs.oracle.com/bigdata/machine-learning-techniques. [Accessed 20 December 2020].

[26] “Techniques of Machine Learning,” [Online]. Available: https://www.simplilearn.com/techniques-of-machine-learning-tutorial. [Accessed 18 December 2020].

[27] R. Shaikh, “Gentle start to Natural Language Processing using Python,” 20 October 2018. [Online]. Available: https://towardsdatascience.com/gentle-start-to-natural-language-processing-using-python-6e46c07addf3. [Accessed 5 December 2020].

[28] R. Silipo, “Neural Machine Translation with Sequence to Sequence RNN,” 15 February 2019. [Online]. Available: https://www.dataversity.net/neural-machine-translation-with-sequence-to-sequence-rnn/. [Accessed 28 January 2020].

[29] “Tensorflow,” [Online]. Available: https://en.wikipedia.org/wiki/TensorFlow. [Accessed 2 March 2020].

[30] “Keras,” [Online]. Available: https://www.tensorflow.org/guide/keras. [Accessed 5 March 2020].