Applying Deep Learning for Morphological Analysis in the Sinhala Language

Yasas Ekanayaka, Randil Pushpananda, Viraj Welgama, Chamila Liyanage

***Abstract*—This research was performed for analyzing morphology of the Sinhala language. Six different deep learning architectures, including RNN, LSTM, and GRU, with and without bidirectional processing was used in the study. Two different datasets, in both Sinhala and Roman scripts, were considered, with each dataset consisting of a total of 644k unique entries. The results were compared to identify the best-performing architecture. Among all the approaches, the model trained with the Sinhala script dataset using bidirectional Gated Recurrent Unit (BiGRU) as the deep learning architecture provided the highest accuracy (87.96%). Several other experiments, such as predicting morphemes and definitions separately, were also considered to assess the behavior of deep learning in morphological analysis in the Sinhala language. All these experiments yielded more than 88% accuracy. These positive results demonstrate the promising potential of deep learning approaches for morphological analysis in the Sinhala language. Using the best performing model, we developed an application for users who are interested in learning and analyzing Sinhala words and their morphology.**

***Keywords*—Morphological analysis, Sinhala morphology, Sinhala language, Deep learning, RNN, LSTM, GRU**

I. Introduction

Morphology is the study of the internal structure of words and the principles by which words are formed in a language . When applying formal linguistic knowledge to develop natural language applications for computers , morphological analysis is necessary to enable the computer to understand this knowledge . Sinhala , being an Indo -Aryan language , is reported to be morphologically rich . In terms of linguistic resources available for natural language processing (NLP), Sinhala is considered a low -resource language with a limited number of language resources available for NLP research and development [1]. Therefore, analyzing morphology of the Sinhala language would be beneficial, and thus, this paper discusses a study we carried out to analyze morphology of the Sinhala language.

Correspondence : Yasas D Ekanayaka (E-mail : ydilshan .ek@gmail .com)
Received:08.03.2023 Revised:19.04.2023 Accepted: 19.06.2023

Yasas D Ekanayaka, Randil Pushpananda, Viraj Welgamaare from University of Colombo School of Computing, Sri Lanka. (ydilshan.ek@gmail.com, rpn @ucsc .cmb .ac .lk , wvw @ucsc .cmb .ac .lk) Chamila Liyanage is from Language Technology Research Laboratory , University of Colombo School of Computing, Sri Lanka. (cml@ucsc.cmb.ac.lk)

DOI: https://doi.org/10.4038/icter.v16i2.7262

Morphological Analysis (MA) is useful in understating the structure of words. For instance, the English word *unreadable* consists of three morphemes: *un-* + *read* + *-able*. Among these, *read* is the root, while *un-* and *-able* are the prefix and suffix, respectively. Further, there are two types of morphology: inflectional morphology and derivational morphology. Inflectional morphology studies how components such as roots, prefixes, and suffixes combine within a word to modify it and showcase different grammatical categories. On the other hand, derivational morphology studies the creation of new lexemes from existing words. For example, adding suffixes like *-s* and *-ing* to the word *read* can create the respective forms of *reads* and *reading* to indicate grammatical meanings of singular and continuous, respectively. These are examples of inflectional morphology, as they do not change the Part-Of-Speech (POS) category of the root word after adding suffixes to it. However, when the suffix *-able* is added to the verb root *read*, it becomes *readable*, an adjective. Since the word has transformed into a different POS category, this can be identified as an example of derivational morphology.

Discussing the importance of analyzing the morphology of a particular language in relation to NLP highlights its primary use in tasks such as information retrieval, language modeling, and machine translation. Accordingly, morphological analysis is valuable for developing various NLP applications, including stemming, lemmatization, part-of-speech tagging, and named entity recognition. It is also useful in developing language-related applications like grammar checkers, text-to-speech systems and search engines. Moreover, a morphological analyzer will also be helpful for those who study words of a particular language.

The rest of this paper is organized as follows: The next subsection I-A discusses the morphological complexity of the Sinhala language. Section II summarizes related works for Sinhala and other languages. Section III describes the methodological approach used in the research. Here, we discuss the dataset, data preprocessing, the method of evaluation, and hyperparameter tuning. Section IV presents a detailed description of the research process, where we discuss various deep learning architectures, their performance with varying dataset sizes, dataset partitioning for model training, and error analysis. Finally, the paper concludes with a summary of the current research and a discussion of future work in section V.

A. Morphological complexity in Sinhala

Sinhala is an Indo-Aryan language spoken by approximately 70% of the population in Sri Lanka. It has been enriched by influences from Old and Middle Indo-Aryan languages, as well as several other languages, including Tamil, Portuguese, Dutch, and English. Sinhala is a diglossic language with two distinct varieties: Spoken Sinhala (also known as Colloquial Sinhala) and Written Sinhala (also known as Literary Sinhala) [2]. As a highly inflected language, nouns in Sinhala can be inflected for the grammatical features of number, gender, case, definiteness, and person, while verbs are conjugated for the grammatical features of tense, number, gender, person, and volition [3]. For instance බැලුවේය *bæluvēya*[1] (looked) is a verb form that indicates the grammatical features of past tense, singular, masculine, 3rd person, and volitive. Accordingly, a verb stem in Sinhala can be conjugated into more than 250 unique forms, demonstrating the morphological richness of the language.

In addition to the suffixes used to denote the grammatical features mentioned above, Sinhala morphology is further complicated by the suffixes derived from particles. For instance, the suffix -ත් *-t* in Sinhala is derived from the particle ද *da* and is used for several functions. In the example එළුවාත් *eḷuvāt* (goat also), it has been used as a conjunction. Furthermore, Sinhala uses *-i* as a predicative marker for non-verbal predicates. Therefore, Liyanage et al. [4] argues that it is a morphological feature and should be marked in morphological annotations. For instance, the word ගුරුවරයෙකි *guruvarayek-i* (a teacher), which appears with a predicative marker, differs from ගුරුවරයෙක් *guruvarayek* in which the rest of the morphological features are the same.

Another aspect that showcases the morphological richness of Sinhala is the process of creating multiwords by applying morpho-phonemic changes, referred to as Sandhi. Sinhala Sandhi is a complex system used to generate new forms and can be classified into two types named internal Sandhi and external Sandhi.

I. Internal Sandhi refers to morpho-phonemic changes that occur when prefixes/suffixes are attached to words.
Ex: පොත් *pot* (book) + එන් *en* (from) = පොතෙන් *poten* (From the book)

II. External Sandhi refers to morpho-phonemic changes that occur among words.
Ex: ආත්ම *ātma* (self) + අභිමාන *abhimāna* (pride) = ආත්මාභිමාන *ātmābhimāna* (self-righteous)

Thus, in this research, we are primarily focused on studying the inflectional changes that occur within words. Furthermore, from the above two sandhi categories, we have chosen to narrow our focus to the first category, known as internal Sandhi, as the second category of external Sandhi is a complex system that requires a separate treatment.

II. Literature review

As morphology is a formal linguistic phenomenon, Sinhala morphology has been discussed in many traditional grammar books, including Dharmarama Thero [5] and Karunatillake [6], and it has further been discussed in non-conventional grammar books like Kumarathunga [7],[8]. Furthermore, there are numerous formal linguistic studies reported, among which Abhayasinghe [9], Parawahera [10], Sugunasiri [11], Chandralal [12] and Stonham [13] are prominent examples.

However, MA as a field of study in NLP, there have only been a couple of reported studies on morphological analysis of Sinhala, that have employed both rule-based and data-driven approaches. Rule-based approaches involve using a set of rules that describe the behavior of morphemes, whereas data-driven approaches rely on datasets.

Among these studies, Viraj et al. [14], has defined the morpheme segmentation boundaries of Sinhala words and established standard definitions for Sinhala word morphology. The study also grouped Sinhala words into 43 sub-categories based on their part-of-speech (POS) categories and word endings. Fernando and Weerasinghe [15], on the other hand, has developed a parser capable of analyzing and generating Sinhala verbs. This is reported to be the first-ever parser developed for analyzing Sinhala verb morphology. The model is reported to be capable of analyzing verbs for 45 inflectional rules for each stem and includes data for 400 verb stems. As a related study to morphological analysis, Nandathilaka et al. [16] has conducted a research on rule-based lemmatization for the Sinhala language.

In addition, Premjith et al. [17], have used a deep learning approach for Malayalam morphological analysis at the character level with Roman script. Then they used Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU) as the approaches. For the results, RNN, LSTM and GRU have obtained accuracies of 98.08%, 97.88% and 98.16% respectively (using 1,54,277 total words). In Prasad et al. [18], have used a deep-learning-based character level approach for morphological inflection and generation in the Sanskrit language. This study has used RNN, LSTM, GRU, Bi-RNN, Bi-LSTM and Bi-GRU as the deep learning architectures. They have compared the results of each approach and have provided the best-performing model (Bi-GRU 98.42% using 101,674 total words). Further, Prabha et al. [19] have done a study about applying deep learning

[1] The transliteration of Sinhala words in the paper follows the ISO 15919 standard.

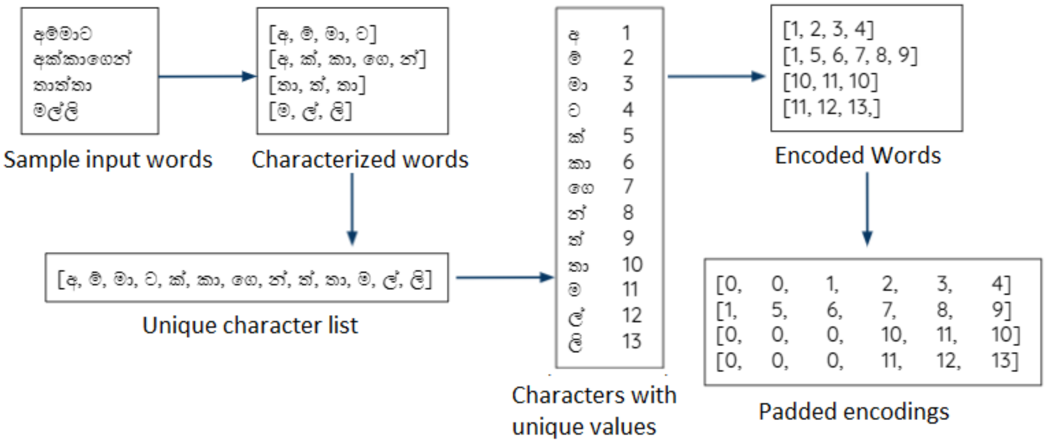


Fig. 1. Encoding process of input data

for Past-Of-Speech (POS) tagging in the Nepali language. In this study, the Bi-LSTM model has shown the highest accuracy (99.85% using 100,720 total words). In Makhambetov et al. [20], they have done a Data-Driven morphological analysis and disambiguation for the Kazakh language. To do the morphological analysis, they have used a two-step segmentation-ranking strategy. From this method, they have gathered 88.75% accuracy for the seen data (gathered from the dataset).

As mentioned above, most studies on morphological analysis for other languages have used RNN, LSTM, and GRU approaches, and many have demonstrated high performance. This suggests that deep learning approaches may also yield better results for morphological analysis in the Sinhala language as well.

Among the deep learning architectures, RNN is a class of artificial neural networks that are often used with sequential data. Connections between nodes represent a directed graph with a sequence of data. This allows it to display a strong temporal behavior. LSTM (Hochreiter and Schmidhuber [21]), is a type of RNN that is designed to address the issue of vanishing gradients in traditional RNNs. LSTMs incorporate a gating mechanism that permits the network to remember or ignore information for extended peri1ods. This mechanism comprises three gates: the input gate, the forget gate, and the output gate. Forget gate decides what information should be thrown or kept away. GRU (Cho et al. [22]), is a gating mechanism in RNNs. This is also similar to the LSTMs and it has fewer parameters than the LSTMs. GRUs use a hidden state to transfer information and it has two gates (reset gate and update gate). The update gate decides what is information that needs to keep, and the information can throw away. These are the three deep learning models used in this research, both with and without bidirectional processing.

III. Methodology and Design

A. The Dataset

The dataset used in the research was gathered from the Language Technology Research Laboratory - at the University of Colombo School of Computing (LTRL-UCSC). This dataset comprises about 736k unique words written in Sinhala script, and their corresponding morphemes along with morphological annotations. Although Sylak-Glassman [23] provides a morphological feature schema for annotation, this dataset uses different tags for certain morphological features, because the annotation was completed prior to the release of Unimorph.

Moreover, the dataset was also transliterated into the Roman script to experiment with the outcomes of the morphological analyzer. The transliteration was performed following the transliteration approach used in Viraj et al. [14].

B. Data pre-processing

Some words in the dataset lacked morphological labels. Although the number of these words was relatively small compared to the entire dataset, they should have either been annotated or removed. However, annotating them required the assistance of a linguist, who would have to study the annotation process of the dataset, making it a complicated task. As a result, we decided to remove the words with no morphological annotations. In addition, there were duplicate entries in the dataset, which were also removed. Consequently, the dataset consisted of a total of 644k unique entries, each with its corresponding morphological annotations.

Most previous studies on other languages have used character-level analysis for morphological analysis instead of word-level analysis. Accordingly, Premjith et al. [17] and Prasad et al. [18] have reported that using characterized words as input data provides better results than using individual words. It also helps create a relatively large number of words.

Since our approach involved supervised models, both the input and label were taken into consideration. Therefore, we adopted the approach of using characterized words as input data, as shown in Figure 1. Furthermore, we utilized morpheme-level labels as input labels, as demonstrated in Figure 2.

To encode the dataset, we employed two different methods for input words and labels. As the process of encoding the input words, we first created a list of unique words from the dataset and characterized each of them in order to create a list of characterized words. Using that list of characterized words, we then created a list of unique characters and assigned a unique value to each identified character. Next, we reconstructed the dataset using these values to generate the encoded unique word list. Since the encoded characterized words are of varying lengths, we had to pad these inputs to achieve inputs of equal length (with the maximum word length being the padding length). The process of converting input words into a set of vectors is depicted in Figure 1, and we utilized these values as inputs to the deep learning model.

The process of encoding labels differs from how input words are encoded. While we characterize words by encoding them for inputs, labels are identified at the morpheme level. A single word may consist of several morphemes, and similarly, a single label may also be associated with several morphemes. Therefore, we created a list of unique morphemes using those available in the dataset and assigned a unique value to each morpheme in the list. Using these values, we encoded the set of labels. Next, we converted each label into a 2D vector, similar to a one-hot-matrix, with the maximum unique value as the matrix width and the maximum label length as the matrix height. After this step, we obtained labels of the same size for the input words. Figure 2 depicts the process of encoding labels.

Given the vocabulary size of the data and labels, we obtained 78 unique characters from the words and 1075 unique morphemes from the labels (this value may be adjusted based on the dataset).

C. Deep learning model

Premjith et al. [17] proposed an algorithm for conducting morphological analysis in the Malayalam language. Their experiments using the algorithm yielded high results. Based on this algorithm, we developed our deep learning model.

The word embedding layer serves as the input layer of this model. It has an input dimension equivalent to the vocabulary size, which is the number of unique characters (78). The output dimension of the embedding layer is set to 256, as determined through hyper-parameter tuning. The following layer in the model is the deep learning architecture layer, which can consist of `RNN`, `LSTM`, `GRU`. `Keras` offers the capability to implement all of these deep learning architectures within a single model. Additionally, to incorporate bidirectional architecture, a bidirectional wrapper is available in `Keras` for these deep learning architectures.

The final layer of the model is a fully connected layer with an output dimension equivalent to the number of available morpheme tags in the dataset. The output layer uses the ‘Softmax’ activation function for multi-label classification. The model is trained using the ‘Adam’ optimizer, with a categorical cross-entropy loss.

D. Method of evaluation

As discussed in Section I-A, analyzing morphology in Sinhala is complex because a single form of a morpheme in Sinhala can be encoded for multiple grammatical meanings simultaneously. Therefore, identifying the correct morphological features of a word in Sinhala is challenging. For instance, Figure 3 represents the morphological ambiguity of the noun root අංක (aṁka), while Figure 4 exemplifies ambiguity of an inflected noun form අංකයෙන් (aṁkayen).

As shown in Figure 3, the form අංක (aṁka) has multiple morphological labels. However, as a root word in Sinhala, it does not indicate any structural unit in morphology. Nevertheless, it does indicate grammatical feature of number: `plural` (since noun roots in Sinhala are typically plural by default) and three cases: `nominative`, `accusative`, and `vocative`. Furthermore, in Figure 4, the form අංකයෙන් (aṁka-y-en) has two suffixes. The suffix *-y* indicates the grammatical features of `singular` and `definite`, while *-en* indicates the cases of `instrumental` and `ablative`.

In the initial stage of the research, each morpheme was treated as a distinct label, and the first entry that appeared in the dataset with a similar morpheme was chosen as the representation. However, it is crucial to consider all possible labels for the morphemes in a given word, since a single input may represent multiple labels, as illustrated in Table I.

TABLE I
Predict one vs Predict all

Task	**input**	**label**
Predict one	[ග, ව, ය, ◌ා, ත, ◌්]	[[ගව: `N+RT`], [යා: `+SG`],[∼:`+DF`], [∼:`+NOM`], [ත්:`+CJ`]]
Predict all	[ග, ව, ය, ◌ා, ත, ◌්]	[ගව:`N+RT`, යා:`+SG`, ∼:`+DF`, ∼:`+NOM`, ත්:`+CJ`], [ගව:`N+RT`, යා:`+SG`, ∼:`+DF`, ∼:`+ACC`, ත්:`+CJ`]

In the evaluation, two distinct approaches were considered, as shown in Table I: (i) predicting a single label, and (ii) predicting all possible labels. For the second approach, which involves predicting all labels, the morphological analyzer’s predictions were considered alongside the exact labels avail-

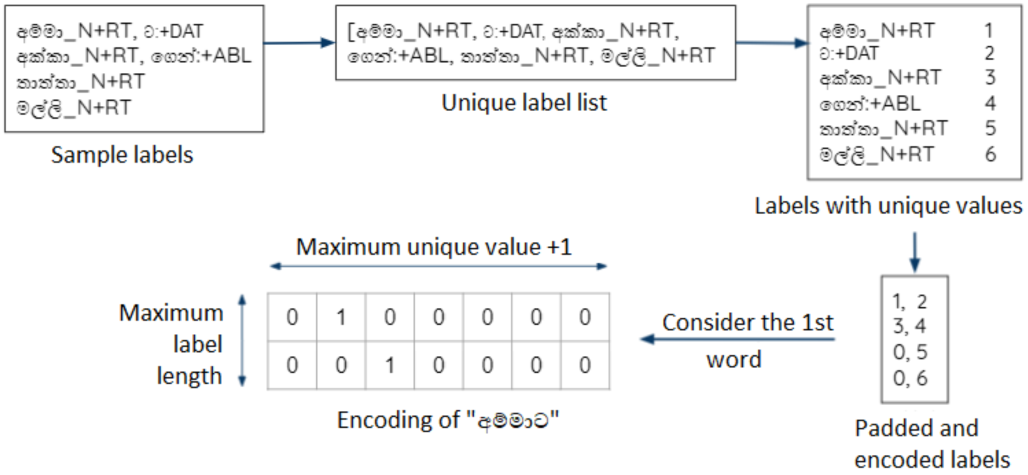


Fig. 2. Encoding process of labels

අංක aṁka (number)

අංක - N+RT
අංක - N+RT+PL+NOM
අංක - N+RT+PL+ACC
අංක - N+RT+PL+VOC

Fig. 3. Morphological ambiguity of a noun root

අංකයෙන් aṁka-y-en

අංක - N+RT+SG+DF+INT
අංක - N+RT+SG+DF+ABL

Fig. 4. Morphological ambiguity of inflected nouns

TABLE II
Initially selected values for Hyper-parameter

Hyper-parameter	Selected values
Activation function	relu, sigmoid, softmax, tanh
Optimizer	Adam, SGD, RMSprop
Number of neurons	32, 64, 128, 256, 512
Batch size	32, 64, 128, 256, 512
Embedding size	32, 64, 128, 256, 512

able in the data. However, for the first approach, only one prediction was generated by the morphological analyzer.

E. Hyper-parameter tuning

Building a deep learning model is an iterative process that involves creating a preliminary structure and then modifying it until a version is obtained that can be trained efficiently in terms of time and computational resources. These settings are referred to as hyperparameters, and the process of finding an optimal set of hyperparameters is known as hyperparameter tuning. In this study, we focused on tuning the following hyperparameters: the number of neurons, activation function, optimizer, batch size, embedding size, and epochs. Initially, we assigned specific values for each parameter, as outlined in Table II.

Then we considered all possible combinations that could be generated from these values. For this, we selected one value from each hyperparameter and created a list of combinations (e.g., ['relu', 'Adam', 32, 32, 32]). We used these values in our deep learning model and calculated the accuracy of the morphological analyzer. Based on the results of hyperparameter tuning, the highest accuracy was obtained when the morphological analyzer used the following hyperparameters: embedding_size = 256, units = 512, activation = softmax, optimizer = Adam, and batch_size = 32. Using these hyperparameters, we trained the model and analyzed the predictions.

IV. Experiments and Results

A. Input labels

Since a supervised learning model is used in this study, we need to consider the labels. The labels can be represented at either the morpheme level or character level. To assess the performance of deep learning models with these two types of labels, we trained a Bi-GRU model and compared the results. To identify morpheme boundaries in character level labels, we used a star (*) symbol, as was done in the study by Premjith et al. [17].

For this study, we utilized 100k words, dividing them into 80k for training and 20k for testing. The results presented in Table III revealed that using morpheme-level labels in deep learning produced better results in the Sinhala script than using character-level labels.

TABLE III
Morpheme level labels or character level labels

Level	Input	Label	Acc
Character	[ග, ව, ය, ා, ත, ්]	[ග, ව, *, යා, *, ත්]	91.18%
Morpheme	[ග, ව, ය, ා, ත, ්]	[ගව, යා, ත්]	95.60%

B. Different deep learning architectures

The study utilized datasets in both Sinhala and Roman scripts. We analyzed the performance of the morphological analyzer with both scripts using various deep learning architectures, and the results are presented in Table IV and Table V.

TABLE IV
Performance of the morphological analyzer with different deep learning architectures with Sinhala script

Archi	Total	Train	Test	Correct	Acc
RNN	150000	120000	30000	14207	47.36%
LSTM	150000	120000	30000	13297	44.32%
GRU	150000	120000	30000	20148	67.16%
Bi-RNN	150000	120000	30000	25216	84.05%
Bi-LSTM	150000	120000	30000	24351	81.17%
Bi-GRU	150000	120000	30000	26107	87.02%

TABLE V
Performance of the morphological analyzer with different deep learning architectures with Roman script

Archi	Total	Train	Test	Correct	Acc
RNN	150000	120000	30000	12006	40.02%
LSTM	150000	120000	30000	11577	38.59%
GRU	150000	120000	30000	14223	47.41%
Bi-RNN	150000	120000	30000	24834	82.78%
Bi-LSTM	150000	120000	30000	23742	79.14%
Bi-GRU	150000	120000	30000	25803	86.01%

As shown in Table IV and Table V, experiments were conducted using the same dataset size of 150k entries, with 120k entries for training and 30k for testing, for both Sinhala script and Roman script. Six experiments were performed for each dataset using three different deep learning architectures: `RNN`, `LSTM`, and `GRU`, both with and without `bidirectional processing`.

The results of the experiments with the Sinhala script dataset presented in Table IV show that the `LSTM` without bidirectional processing achieved the lowest accuracy, while the `bidirectional GRU` achieved the highest accuracy of 87.02%. Similarly, Table V presents the results of the experiments with the Roman script dataset. Even for this dataset, `LSTM` without bidirectional processing achieved the lowest accuracy (38.59%) among all the experiments, while `bidirectional GRU` achieved the highest accuracy.

Based on these findings, the `bidirectional GRU` deep learning architecture achieved the highest accuracy across all experiments. Therefore, this architecture was used to train various models for different data sizes in this research.

Sample word:	[පොතකින්]
MF prediction with noun root:	[[පොත්,+SG,+ID,+INT], [පොත්,+SG,+ID,+ABL]]
MF prediction:	[[N+RT,+SG,+ID,+INT], [N+RT,+SG,+ID,+ABL]]

Fig. 5. Predicting grammatical features of morphemes for a sample word

C. Performance with the dataset size

Although the full dataset is relatively a large number of annotated data (including 644k words), it was difficult to train the deep learning model with the full dataset due to limited computational power. Therefore, the models were trained and evaluated using the `bidirectional GRU` deep learning architecture across different dataset sizes. The initial model with Sinhala script data was trained on 50k words, with 40k for training and 10k for testing. The dataset was gradually increased up to 150k words, with 120k for training and 30k for testing. The initial model showed an accuracy of 84.80%, and this increased gradually with the size of the dataset. The final model with 150k words showing an accuracy of 87.96%. Table VI presents the performance of different models based on the size of the dataset.

TABLE VI
Performance of the morphological analyzer with different dataset sizes with Sinhala script

Total	Train	Test	Correct	Accuracy
50000	40000	10000	8480	84.80%
75000	60000	15000	12872	85.81%
100000	80000	20000	17281	86.41%
125000	100000	25000	21824	87.30%
150000	120000	30000	26388	87.96%

The results presented in Table VI were obtained using data in Sinhala script. While we carried out the same practice for the datasets in Roman script, the results were slightly lower, similar to those presented in Table V, and therefore, they will not be discussed further.

D. Predicting morphemes

A morphological analyzer is not only useful for analyzing and predicting the grammatical features of morphemes, but it may also predict how words are formed by combining a number of morphemes. Examples can be found in Figures 5 and 6.

Figure 5 shows an analysis of the word පොතකින් (*potakin*), including the noun root and its grammatical features, as well as all the morphological labels. While the most useful way of presenting the results of a morphological analyzer is

Sample word: [පොතකින්]
Morpheme prediction: [පොත + ක + ◌ින්]

Fig. 6. Predicting number of morphemes occupying in a sample word

Sample input: [ග, ව, ය, ◌ා, ත, ◌්]
Sample label: [ගව, යා, ත්]

Fig. 7. Training data for predicting morphemes

by providing grammatical features of morphemes, it's also important to identify the set of morphemes that make up a particular word as depicted in Figure 6.

Therefore, in this experiment, we applied deep learning to predict morphemes. Since we only considered morphemes in this experiment, we did not take into account the symbols or signs such as '∼' (which have been used to indicate additional definitions for a given word) appeared in the dataset. The experiment was done with a dataset consisting of 268,914 words, with 215,131 words for training and 53,783 for testing. The input for the experiment was the characterized words, while the corresponding morphemes were used as labels, as shown in Figure 7.

Compared to the other experiments, applying deep learning for the prediction of morphemes has yielded relatively higher results. Accordingly, this experiment yielded a 97.16% accuracy rate for correct predictions. The use of characterized words as input enabled the deep learning model to accurately identify the relationship between characters and morphemes.

Furthermore,this experiment and the resulting model is useful for both stemming and morpheme concatenation in the Sinhala language. For example, given the word ගවයාත් *gavayāt*, the system can recognize the stem ගව *gava* and identify the remaining parts as suffixes. Conversely, if the morphemes ගව + යා + ත් (*gava+yā+t*) are provided, the system can recognize the two words ගවයා *gavayā* and ගවයාත් *gavayāt*. Additionally, we found that this experiment can be conducted even without root morphemes in the training dataset, which can significantly reduce the required computational power.

E. Predicting definitions

A crucial aspect of morphological analysis is providing definitions for the identified morphemes, which describe their grammatical or morphological properties. In our approach, we used labeled words along with their corresponding morpheme definitions to establish the relationship between characters and morpheme definitions. To train the morphemes with definitions, we utilized all possible lists of definitions for particular morphemes or words, as illustrated in Figure 8. The dataset used in this experiment contained 268k entries, consisting of 215k entries for training and 53k entries for testing. In this experiment, we obtained 94.07% accuracy for correct predictions.

Sample input: [ග, ව, ය, ◌ා, ත, ◌්]
Sample label: [[N+RT, +SG, +DF, +NOM, +CJ],
[N+RT, +SG, +DF, +ACC, +CJ]]

Fig. 8. Training data for predicting definitions

F. Analysis of words from a newspaper

Although we have conducted several evaluations using a selected sample from the dataset, they cannot be considered true evaluations. Therefore, we aimed to perform a genuine evaluation using actual data extracted from a piece of text. To achieve this, we extracted a sample of text from a newspaper article. We chose the 'Divaina' newspaper, which is a mainstream newspaper published daily, and selected an article published on November 10, 2021. The article contained 597 words. Firstly, we removed the stop words and punctuations from the list of words, leaving 478 words. A sample of words from that list is shown in Figure 9. To analyze this list of words, we sought assistance from a linguist. According to the evaluation, it shows an accuracy rate of 76.39% for correct predictions.

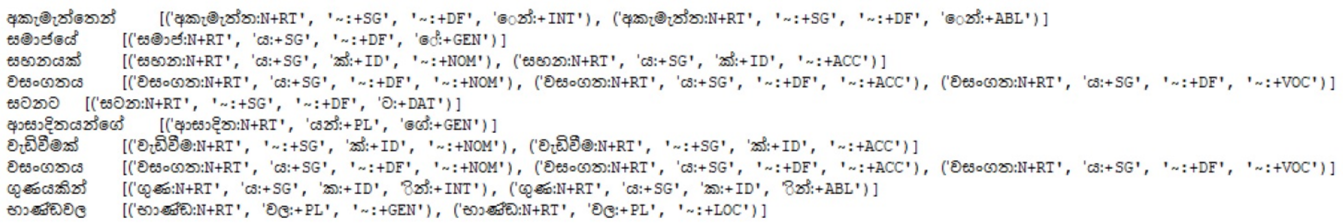
G. Partitioning the dataset and training the model

As discussed in Section IV-C, it was challenging to train a deep learning model with the full dataset due to limited computational power. Therefore, only a maximum of 150k words were used for training and testing. Despite having a relatively large dataset of 644k words, training a single model with the entire dataset was not feasible. As a result, the dataset was partitioned into seven subsets, and deep learning models were trained on each subset separately. The partitioning was done to ensure that each subset did not exceed the maximum limit of 150k words. Initially, the entire dataset was sorted according to the Sinhala alphabet and then partitioned according to the alphabetical order. Details regarding the training and testing data, as well as the results, are presented in Table VII.

TABLE VII
Performance of partitioned data-sets

Word range	Total	Train	Test	Correct	Acc
අ-ඔ	122000	109800	12200	11243	92.16%
ක-ඪ	106360	95724	10636	9603	90.29%
ද-න	63155	56839	6316	5532	87.59%
ප-භ	105685	95116	10569	9588	90.72%
ම-ළ	71698	64528	7170	6401	89.27%
ව-ෂ	70795	63715	7080	6459	91.23%
ස-ෆ	104370	93933	10437	9410	90.16%

Based on the results presented in Table VII, the lowest accuracy obtained was 87.59%. However, all the other six



```
අකැමැත්තෙන්     [('අකැමැත්ත:N+RT', '~:+SG', '~:+DF', 'ෙන්:+INT'), ('අකැමැත්ත:N+RT', '~:+SG', '~:+DF', 'ෙන්:+ABL')]
සමාජයේ     [('සමාජ:N+RT', 'ය:+SG', '~:+DF', 'ේ:+GEN')]
සහනයක්     [('සහන:N+RT', 'ය:+SG', 'ක්:+ID', '~:+NOM'), ('සහන:N+RT', 'ය:+SG', 'ක්:+ID', '~:+ACC')]
විසංගතය     [('විසංගත:N+RT', 'ය:+SG', '~:+DF', '~:+NOM'), ('විසංගත:N+RT', 'ය:+SG', '~:+DF', '~:+ACC'), ('විසංගත:N+RT', 'ය:+SG', '~:+DF', '~:+VOC')]
සටනට   [('සටන:N+RT', '~:+SG', '~:+DF', 'ට:+DAT')]
ආසාදිතයන්ගේ     [('ආසාදිත:N+RT', 'යන්:+PL', 'ගේ:+GEN')]
වැඩිවීමක්     [('වැඩිවීම:N+RT', '~:+SG', 'ක්:+ID', '~:+NOM'), ('වැඩිවීම:N+RT', '~:+SG', 'ක්:+ID', '~:+ACC')]
විසංගතය     [('විසංගත:N+RT', 'ය:+SG', '~:+DF', '~:+NOM'), ('විසංගත:N+RT', 'ය:+SG', '~:+DF', '~:+ACC'), ('විසංගත:N+RT', 'ය:+SG', '~:+DF', '~:+VOC')]
ගුණයකින්     [('ගුණ:N+RT', 'ය:+SG', 'ක:+ID', 'ින්:+INT'), ('ගුණ:N+RT', 'ය:+SG', 'ක:+ID', 'ින්:+ABL')]
භාණ්ඩවල     [('භාණ්ඩ:N+RT', 'වල:+PL', '~:+GEN'), ('භාණ්ඩ:N+RT', 'වල:+PL', '~:+LOC')]
```

Fig. 9. Predicting sample of words from a newspaper article

categories achieved an accuracy of around or over 90%, with the highest accuracy obtained being 92.16%.

H. Error analysis

After manual analysis of the results, we discovered that errors predominantly occurred with words other than nouns. Further analysis revealed that the 150k-word dataset randomly selected for training and testing the model consisted mostly of nouns, which was due to the abundance of nouns in the full dataset. Other categories, such as verbs and adjectives, were included in the dataset in small proportions. As a result, a different approach was required to resolve this issue.

We found that Welgama et al. [24] had provided results under several `POS` categories. To address the issue, we segmented the dataset based on the `POS` category of the root morphemes and tested the performance of the morphological analyzer separately on nouns and verbs. This experiment was conducted using the same deep learning architecture as the previous experiments, namely the `bidirectional GRU` architecture. The results of this experiment are presented in Table VIII.

TABLE VIII
Performance of the morphological analyzer with verbs and nouns separately

Total	Train	Test	noun/verb	Correct	Acc
50000	40000	10000	nouns	9272	92.72%
			verbs	9609	96.09%
100000	80000	20000	nouns	18858	94.28%
			verbs	19503	97.52%
150000	120000	30000	nouns	28456	94.85%
			verbs	29317	97.72%

Table VIII shows the results of three experiments conducted using datasets containing 50k, 100k, and 150k words, respectively. The experiments demonstrate that the datasets with verbs achieved the highest accuracy in all three cases, while the datasets with nouns achieved the lowest. Additionally, a similar experiment was conducted using a dataset in Roman script, which produced the same issues as the Sinhala script dataset. Similar to the previous experiments using Roman script datasets, the results were very low.

I. An application for the morphological analysis

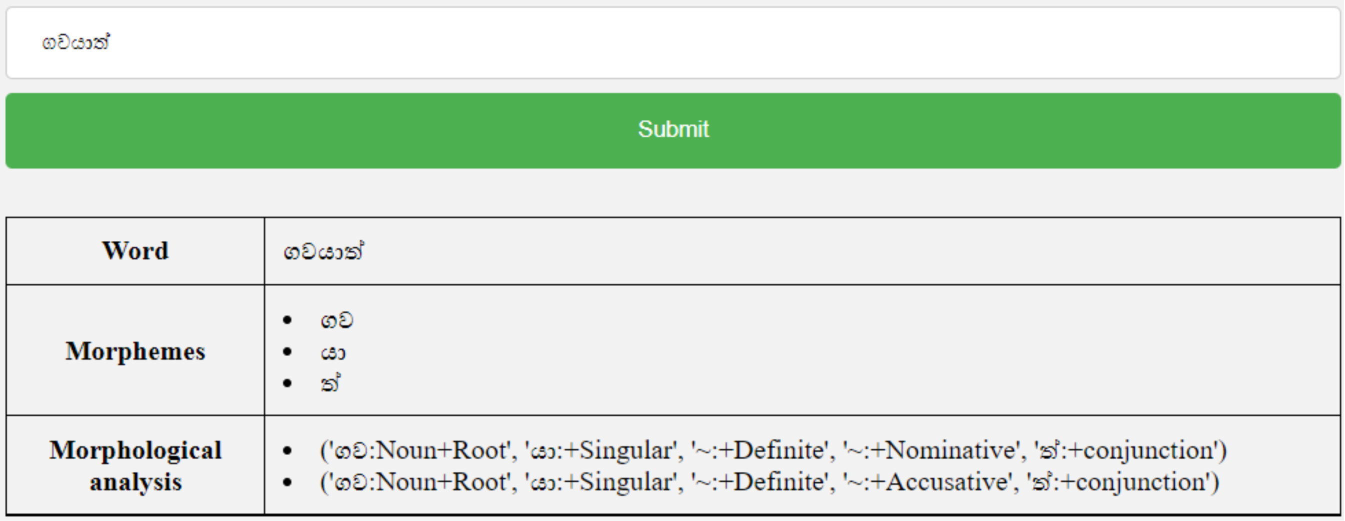
Our research has resulted in the development of a deep learning model for morphological analysis of the Sinhala language, which has been used to create an application for users interested in learning Sinhala words and their morphology. As shown in Figure 10, the application is capable of providing a list of possible morphemes and their definitions for a given word. This application[2] is currently available online and provides grammatical features mentioned in the training dataset. However, to improve its usability for local people, we plan to include Sinhala definitions for the grammatical features and also incorporate stem and suffix information. This will enhance the application's ability to provide a more comprehensive understanding of Sinhala morphology.

V. Conclusion

This paper discusses the process of analyzing the morphology of the Sinhala language. The data used for this analysis consisted of entries containing words and their definitions, represented in both Sinhala script and Roman script. Six different deep learning architectures were employed, including RNN, LSTM, and GRU, both with and without bidirectional processing. The bidirectional GRU architecture achieved the highest accuracy (87.96%) for morphological analysis in Sinhala using the dataset in which the main entries in Sinhala script. Furthermore, we conducted experiments by separating the morphemes and definitions, achieving 97.16% accuracy for morpheme prediction and 94.07% for definition prediction. Increasing the data size was found to improve the system's accuracy. This research demonstrates that deep learning is more effective in morphological analysis of the Sinhala language.

At this stage of the research, we only considered internal Sandhi words as morpho-phonemic changes, whereas such constructions should be analyzed in a morphological analyzer. In the future, we plan to extend the research by analyzing external Sandhi words in the Sinhala language as well. Furthermore, the required computing power for the research is dependent on the encoded matrix size (Figure 1 and Figure 2). Hence, future studies may focus on finding ways to reduce the encoded matrix size.

[2] Application: GitHub link to the application



Word	ගවයාත්
Morphemes	• ගව • යා • ත්
Morphological analysis	• ('ගව:Noun+Root', 'යා:+Singular', '~:+Definite', '~:+Nominative', 'ත්:+conjunction') • ('ගව:Noun+Root', 'යා:+Singular', '~:+Definite', '~:+Accusative', 'ත්:+conjunction')

Fig. 10. The application developed for Sinhala morphological analysis

References

[1] N. de Silva, “Survey on publicly available sinhala natural language processing tools and research,” *arXiv preprint arXiv:1906.02358*, 2019.

[2] J. W. Gair, “Sinhalese diglossia,” *Anthropological Linguistics*, pp. 1–15, 1968.

[3] C. Liyanage, R. Pushpananda, D. L. Herath, and R. Weerasinghe, “A computational grammar of sinhala,” in *Computational Linguistics and Intelligent Text Processing: 13th International Conference, CICLing 2012, New Delhi, India, March 11-17, 2012, Proceedings, Part I 13*, Springer, 2012, pp. 188–200.

[4] C. Liyanage, K. Sarveswaran, T. Nadungodage, and R. Pushpananda, “Sinhala dependency treebank (stb),” in *Proceedings of the Sixth Workshop on Universal Dependencies (UDW, GURT/SyntaxFest 2023)*, 2023, pp. 17–26.

[5] R. Dharmarama Thero, *sidath sangarawa (edition)*. Vidyalankara pirivena, Kelaniya, Sri Lanka, 2010.

[6] W. Karunatillake, *Sinhala bhasha vyakaranaya*. M. D. Gunasena Co. Ltd, Sri Lanka, 2009.

[7] M. Kumarathunga, *kriya viwaranaya*. M. D. Gunasena Co. Ltd, Sri Lanka, 1993.

[8] M. Kumarathunga, *vyakarana vivaranaya*. M. D. Gunasena Co. Ltd, Sri Lanka, 2000.

[9] A. A. Abhayasinghe, “A morphological study of sinhalese.,” Ph.D. dissertation, University of York, 1973.

[10] N. P. Parawahera, “Phonology and morphology of modern sinhala,” Ph.D. dissertation, 1990.

[11] S. H. Sugunasiri, “Morphological analysis of the finite verb in spoken sinhala,” MA dissertation, University of Pennsylvania, 1966.

[12] D. Chandralal, “Sinhala,” *Sinhala*, pp. 1–312, 2010.

[13] J. Stonham, “What morphology can tell us about syntax: An integrated approach to sinhala verb formation,” *Language Sciences*, vol. 19, no. 2, pp. 139–151, 1997.

[14] W. Viraj, W. Ruvan, and M. Niranjan, “Defining the gold standard definitions for the morphology of sinhala words,” *Research in Computing Science*, vol. 90, pp. 163–171, 2015.

[15] N. Fernando and R. Weerasinghe, “A morphological parser for sinhala verbs,” in *Proceedings of the International Conference on Advances in ICT for Emerging Regions*, 2013.

[16] M. Nandathilaka, S. Ahangama, and G. T. Weerasuriya, “A rule-based lemmatizing approach for sinhala language,” in *2018 3rd International Conference on Information Technology Research (ICITR)*, IEEE, 2018, pp. 1–5.

[17] B. Premjith, K. Soman, and M. A. Kumar, “A deep learning approach for malayalam morphological analysis at character level,” *Procedia computer science*, vol. 132, pp. 47–54, 2018.

[18] V. Prasad, B. Premjith, C. Chandran, K. Soman, and P. Poornachandran, “Deep learning based character-level approach for morphological inflection generation,” in *2019 International Conference on Intelligent Computing and Control Systems (ICCS)*, IEEE, 2019, pp. 1423–1427.

[19] G. Prabha, P. Jyothsna, K. Shahina, B. Premjith, and K. Soman, “A deep learning approach for part-of-speech tagging in nepali language,” in *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, IEEE, 2018, pp. 1132–1136.

[20] O. Makhambetov, A. Makazhanov, I. Sabyrgaliyev, and Z. Yessenbayev, “Data-driven morphological analysis and disambiguation for kazakh,” in *International Conference on Intelligent Text Processing and Computational Linguistics*, Springer, 2015, pp. 151–163.

[21] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[22] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, “On the properties of neural machine translation: Encoder-decoder approaches,” *arXiv preprint arXiv:1409.1259*, 2014.

[23] J. Sylak-Glassman, “The composition and use of the universal morphological feature schema (unimorph schema),” *Johns Hopkins University*, 2016.

[24] V. Welgama, R. Weerasinghe, and M. Niranjan, “Evaluating a machine learning approach to sinhala morphological analysis,” in *Proceedings of the 10th International Conference on Natural Language Processing, Noida, India*, 2013.