IMPROVIING THE ACCURACY OF THE OUTPUT OF SINHALA OCR BY USING A DICTIONARY

Dineesha N Ediriweera

098256

Dissertation submitted in partial fulfillment of the requirements for the degree Master of Science

Department of Computer Science & Engineering

University of Moratuwa

Sri Lanka

December 2012

I declare that this is my own work and this dissertation does not incorporate without acknowledgement any material previously submitted for a Degree or Diploma in any other University or institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Also, I hereby grant to University of Moratuwa the non-exclusive right to reproduce and distribute my thesis/dissertation, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as articles or books).

Signature: Date:

The above candidate has carried out research for the Masters dissertation under my supervision.

Signature of the supervisor: Date

Acknowledgement

First and foremost I wish to express my gratitude to the supervisor of this project, Prof. Gihan Dias, Senior Lecturer, Department of Computer Science and Engineering, University of Moratuwa for his valuable guidance and advice given without hesitation throughout a long period of time. My sincere thanks deserve all the lecturers of the MSc course, and the coordinator, for their guidance and assistance. Specially, I wish to thank Dr. Chandana Gamage, Head of Department, CSE for the support given in completing the project. I would like show my greatest appreciation to Mr. Harsha Wijewardena, Snr. Lecturer, University of Colombo, School of computing, for his tremendous support and help. Besides that, I would like to appreciate Dr. H. L. Premaratne and Dr Ruwan Weerasinghe for the help given. My Special thanks go to Navoda, UCSC for the endless help given in getting sample data. I would like to thank all the academic and non academic staff of CSE for their support in various ways in past years. I like to take this opportunity to thank all my MSC class mates for their kind cooperation throughout the study period. This would not have been possible without the kind support and help of Associated Newspapers of Ceylon Ltd. and many individuals in the organization. I would like to extend my sincere thanks to all of them. My heart felt gratitude goes to my colleagues at ANCL, for their understanding and the assistance given in various ways. Finally, I wish to pay my gratitude to my family members, relations and friends for their understanding and support given during my studie period. This study would not have been successful without the understanding and dedication of my loving son.

Abstract

This research proposes a system to improve the accuracy of Sinhala OCR by post-processing techniques using a dictionary. Several methods are integrated into the system to get the best output in 3 steps. A word found in the dictionary, at any step is considered as correct. A word which can not be validated as correct is left for the following step. In each step word hypothesis net is generated with probable candidate words considering the similarity measures and word statistics. The selection is based on the best matching word in the hypothesis, which has the maximum score. Assuming that frequent words are more likely to be appeared and being correct, the candidate words with the highest score enable correcting the word. The score is estimated by multiplying the frequency of the word and character similarity measures. Manual error correction with the samples proved the accuracy of this phenomenon. Hence the error detecting and correcting is based on this principle. Confusion Character Pairs, word prefixes, suffixes, stems and Confusion Character Groups are lookups for them. In addition a linguistic feature which is also proved by statistics in Sinhala language is also utilized. One such feature is word formation with prefix root and suffix components. A few syntactical rules in Sinhala language has also incorporated into the system. Majority of the errors present in the OCR output are single character errors, and the system is capable of correcting multiple errors up to 5 characters. The result shows that the system improves word accuracy from 59.8% to 92.6%.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF GRAPHS

LIST OF APPENDICES

1. INTRODUCTION

Optical character recognition (OCR) refers taking an image of scanned text from paper, either printed or hand written and converting the image into a sequence of corresponding characters in machine-readable form. Those characters may or may not be correctly recognized by the OCR. The post processing is being used to ensure the output sequence of OCR to be as same as the original document. If a particular word could not be verified, a replacement or a suggestion is made to form a sensible word. There can be many reasons which may affect the accuracy of the OCR output text. Some of them would be the noise in the source document scanned, structural similarity of some of the characters, and complexity of the script which may occur due to the components in different directions and character combinations.

Sometimes, the recognizing process of the characters would not be accurate enough due to various reasons, for example when the original document is in degraded form. As a result, the readability of OCR output becomes too poor. A possible remedial action to improve the result would be to use of validating techniques after the recognition process.

The requirement of an OCR system for Sinhala script becomes important in reproducing the documents of the National Archives, Sri Lanka, archived Newspapers, and old books of which the source documents may be in degraded form.

Apart from a few scripts in south Asia, such as Devanagari, Gurmuki, Sindhi, Tamil, Thai and Teligu, a little advancement has been achieved, in research for the development of OCR for Brahmi descended scripts, compared to those for Latin, Chinese, Korean and Japanese scripts [20]. Most of the Indian scripts do not have any robust commercial OCRs [25]. For Sinhala Language, there are only very few efforts have been made [14] and OCR software for Sinhala language is not available as a commercial product. But, OCR is being training for Sinhala [48] at University of Colombo School of Computing and it is the Open Source Google Tesseract OCR Engine [47] of which output is satisfactory at a character level. There might be

possibilities of developing Sinhala OCR up to a commercial level by using the similar techniques implemented on Indic script.

For last two decades, there has been an improvement in the strategies behind the OCR. But, still there are problems remaining in recognizing the correct character. Nevertheless OCR knowledge base has widely employed in recognition stage to improve the accuracy of recognition, errors are still being detected leaving them to rectify after recognition. There are various techniques employed to correct those such as using a lexicon, Language models, Grammar Rules, Statistical information of n-grams, Syntax Analysis, However there may be complex words or character groups which will be difficult to recognize by above techniques. In such cases it must be transferred to human beings for amending.

1.1 Motivation and Objectives

Sinhala character recognition is at a research level and to have a widely used OCR for Sinhala script is far behind compared to the OCRs for Latin script. There are preserved documents decaying with time. Many old publications are yet to be reproduced. Those necessities would be encouraged to develop a commercial level Sinhala OCR. With localization of Information Technology during last few years, has exaggerated the necessity further. This motivates me to do a research on this area.

The main objective of this research is to improve the accuracy of Sinhala OCR output by post processing techniques using a dictionary. The proposed system improves the accuracy of the recognized Sinhala text at word level. Training of the Sinhala words is being done by University of Colombo School of Computing [47]. The output accuracy of the Sinhala OCR is at a satisfactory level at a character level. But accuracy of words recognized is not in a satisfactory level. Hence, this research proposes several methods to improve the word level accuracy of the final output sequence of the OCR.

1.2 Outline of the Thesis

This report is structured as follows: Chapter 2 contains an overview of character recognition, different strategies employed on character recognition, methods detecting and correcting OCR errors. Chapter 3 Describes Sinhala language and OCR related language features and language rules. Chapter 4 introduces techniques used for correcting OCR errors by exploiting the nature of OCR post processors in earlier researches. Chapter 5 explains the detection and correction strategy used in our proposal for misspelled words. In Chapter 6, the proposed strategy is evaluated on the data. Finally, the conclusions are presented in Chapter 7 and give an outlook on future work.

2. LITERATURE REVIEW

Character recognition is a part of pattern recognition in which images of characters are recognized and related character codes are returned. Character recognition is further classified into two types based on the input method [33]. They are On-line character recognition and Off-line character Recognition.

Online Character Recognition is real time character recognition. It recognizes the dynamic motion during writing. Offline Character Recognition is a process that recognizes already printed or written document. It allows hard copies of written or printed text to be rendered into editable, soft copy versions. Offline Character Recognition can be further categorized into two. They are Magnetic ink Character Recognition (MICR) and Optical Character Recognition (OCR).

In Magnetic ink Character Recognition (MICR) text has been printed in special fonts with magnetic ink usually containing iron oxide to be magnetized when reading from the machine. The system has been in efficient use for a long time in banks around the world for processing of cheques [33].

Optical Character Recognition (OCR) system uses of an optical input device, usually, a scanner to capture images and feed those into the recognition system. The image can be handled as a whole and text cannot be manipulated separately in an application [26]. OCRs are of two types, as OCRs for recognizing printed characters and OCRs for recognizing hand-written text.

OCRs meant for printed text recognition are generally more accurate and reliable compared to the OCRs for hand written text, because of the reason that the font is standard for printing and variety of writing styles are exist for hand written text [26].

Before a century, the requirement for creating a reading device for blind persons initiated the necessity for OCRs [26]. Nowadays, OCRs are widely used for various requirements such as form reading, storing data, reproduction of old documents and

books and processing of text for various purposes like translation, transliteration or converting text to speech etc. Data in electronically manipulatable formats facilitate displaying, searching and transportation and OCRs play a major role in getting the available data into electronically processable format.

The formats accepted in OCR software are JEG, TIFF, GIF, PDF whereas output formats are text, Microsoft Word, RTF, PDF. Widely used OCRs are Abbyy FineReader, Adobe Acrobat Professional and Google Tesseract OCR (Open source). There are software used to build the OCRs [21]. In addition, commercial off-the-shelf OCR (COTS) software packages like Tesseract have become tools in these applications. As a result of that, OCR software has become openly available.

OCR systems based on Latin script were developed first and those are very successful in commercial use. Accuracy of commercial OCR software varied from 71% to 98% [26]. Frequent words are tended to be more correct, but accuracy in recognizing domain specific text or Special names and uncommon general words are still a problem. Accuracy of OCRs depends on the sharpness of the scanned image, the nature of the original document and the OCR software [26]. The text converted by using OCRs can be automatically translated into other languages and/or spoken form during the process.

Even in Latin script, English letter simple L (l) and number 1 has the highest misrecognition. Misrecognition of “r” and “n” as “m” is another. q and g, B and 8, O and 0, S and 5, 1 and i and Z and 2 are also frequent misrecognized characters. In addition, words at the end of line with or without hyphens are used to be recognized inaccurately. Types of OCR error are listed [26] as in Table 2.1.

Manual OCR error correction is too expensive to verify every single OCR character and is very cumbersome. OCR errors often look correct as the recognized word is correct in spellings, but in the sense and the context it may be incorrect. OCR error rates are highly variable, based on the quality of the images, font types, etc. These errors are primarily caused by noise either inherent in the document or introduced by the digitizing process. Today OCRs are widely applied to paper-intensive industry,

with complicated backgrounds, degraded-images, heavy-noise, paper skew, picture distortion, low-resolution images, disturbed grid and lines and text image consisting of special fonts, symbols, glossary words etc. Therefore, it still needs better accuracy and reliability [33].

Table 2.1: Types of Errors

Error class	recognized word	correct word
Segmentation (missing space)	thisis	this is
Segmentation (split word)	depa rtme nt	department
Hyphenation error	de- partment	department
Character misrecognition	souiid	sound
Number substitution	0pporunity	Opportunity
Special char insertion	electi'on	election
Changed word meaning	mad	sad
Case sensitive	BrItaIn	Britain
Punctuation	this.is	this is
Destruction	NI.I II I	Minister
Currencies	?20	$20

OCR error correction is typically based on verifying characters that have been flagged as "suspicious" by the OCR engine. Automated and manual OCR error correction can only find and fix a fraction of the errors that are created in OCR [26].

One of the common ways of correcting these errors is to make use of a word dictionary for the language of the text to check if the OCR output words are valid words in a dictionary [40]. A higher level of linguistic knowledge [42] in grammar rules, syntactical [27] and semantic [44], probabilistic approaches [29] [18], may also be employed to improve the accuracy of the output of the OCR.

2.1 OCR Technology

The process of current OCR systems involves 3 stages, Pre-processing, Recognition and Post processing. Layout Analysis is used to understand the structure of the

document before applying the above stages. Pre-processing prepares the images with optimal quality for recognizing to be very effective. Recognition converts the binary images into electronic representation of characters. Post-processing enhance the accuracy of the recognized text by detecting and correcting errors at recognized stage.

2.1.1 Pre-processing

There are four steps for pre-processing such as Image acquisition, Transformation, Segmentation and Feature Extraction [1].

Image acquisition

Converting document to a numerical representation is image acquisition. It acquires the image of a document in colour, grey-levels, and in binary format. The image is scanned first. Resolution depends on the purpose of the application and the nature of the material. Then the scanned image is sampled and quantified into number of grey levels. Coding techniques are used to reduce the size of data representing. The quantized data is represented in run length or entropy coding. Usually two quantised levels of data are kept to proceed with sub tasks, which require detailed information retrieval.

Transformation

Transformation of image to image is characterized by input-output relationship. It involves enhancing the data in the representation image in several methods, Geometrical transformation, Filtering, Background Separation, Object Boundary Detection, and Structural Representation [1].

Geometric transformation corrects the distorted image and normalized in the relevant area. Skew distortion and optical distortions (barrel and pin cushion) are removed [1]. For skew detection Connected Component Analysis or Projection Analysis is used, whereas for Skew Correction Rotation algorithms are used [2]. Orientation corrections

are done on the image to improve the quality of the segmentation candidates. Estimating the slant angle by surrounded lines for an object, Shear transformation is applied to correct the same [2]. At last the ideal undistorted image is re-constructed, which is called Normalizing. Aspect ratio adaptive normalisation uses Forward Mapping or Backward Mapping. There are three popular normalization methods, linear normalization, moment normalization and nonlinear normalization, based on line density equalization [2].

Then filtering is done to improve the figure in two cases smoothing and noise removal [2]. Different types of linear filters are used for enhancement, edge and line detection etc. Non-linear filters like Rank order filters have operations for erosion, dilation, contour detection and median [1]. The median filters are popular as they remove noise. Morphological filters extract features of the image by using a structural element and morphological operators such as dilation, erosion, opening and closing [1]. Polynomial filters can also be used. Unless the average number of filters are applied, the feature extraction process would yield nothing [2].

For background detection of the text or figure two techniques are employed depending on the background, whether it is uniform or texture. For uniform backgrounds, Grey-levels threshold is used. Binerization improves the recognition rate, since that helps the algorithm to differentiate the background and the foreground by inspecting the pixel intensity [24]. The Noise removal is done using a Gaussian Filter [24]. To remove the noise introduced in binarization Global threshold and Local threshold is used. In global threshold the image is filtered out, before binarization, using linear or non-linear (mean) filters. The linear filters thicken the structures. Local thresholds filter the image after binarization. Mean filter is best suited as it fills the small holes in addition. For texture backgrounds such as highlighted text, background elimination is tackled by Morphological filters after binarization using erosion and dilation [1], [2].
To apply the segmentation object boundaries inner and outer contours have to be identified. Then the object boundary is specified by x y coordinates, in freeman chain code, which is a reduced set or in Fourier series [1].

Segmentation

Next is the extraction of layout information such as lines, words, characters, number of lines and number of characters, etc [24]. Segmentation divides the image into regions having objects of the same type. The four common algorithms are used. Connected component labelling is the widely used one. It labels the same region with the same value. X-Y tree decomposition used horizontal and vertical projections of the characters and character positions for easy identification of bands of text. For Line detection, horizontal projection is used. It scans lines from left to right [24]. After that, line separation of the object structural analysis called Thinning is used to extract the features of the object and place them in a form of a graph. Thinning has two types of pixels regular for lines, and irregular for ends and conjunctions [1]. These algorithms named Hilditch thinning algorithm and Zhang-Suen thinning algorithm are easy to implement [2].

Character recognition at breaks of Horizontal projection profile uses small projections to identify middle zone. To build the binary image the Run length smearing algorithm is applied in line by line and column by column and "AND"ed both matrices. It converts short white runs to black. Hough transformation is a better way in identifying the regions, especially when different objects are connected together. It identifies a curve for the object so that it is robust to noise.

Segmentation is also two types; Explicit and Implicit. In explicit segmentation a character boundaries are separated making a candidate list whereas in implicit segmentation characters are divided into equal frames of windows for feature extraction to match with templates. Any classifier can be used with the first, whereas the latter is often used by HMM model.

Feature extraction

Feature extraction helps classifying the symbols into classes. There are 4 categories of features, Geometric, Moments, Orthogonal Transforms and Relational Descriptors [1].

Simple geometric features like x-y direction, aspect ratio, area, perimeter, Euler number are used to broadly categorise the objects into classes. For that Gaussian Kernels or KFC Filters can also be used. Objects described by a set of moments and Zernike moment are very useful as its rotation invariant property. Feature Space Transformation methods use Linear Transformation or Kernels of the input image. Karhunen Loeve Transformation is one of the popular orthogonal transformations. Relational Descriptors provide structural information on document image and Layout structure [1]. Feature extraction captures the distinctive characteristics of the digitized characters for recognition.

2.1.2 Character/ Word Recognition

Recognition involves sensing, feature selection and creation, pattern recognition, decision making, and system performance evaluation [1]. For Character separation, vertical projection is used and it scans a line from top to bottom is used [24].

Feature Selection and Creation

Feature Selection is very important as it reduces the Sample Complexity, Computational Cost and Performance Issues. There are three approaches. Filter approach, filter out some features before the classifier is applied. Wrapper approach, wraps the feature selection algorithms with computational cost but unbiased classifier. A Hybrid model fits the sub set of features and the accuracy of matching to a classifier [2]. There are evaluation techniques of the features selection for recognizing. To build systems with character recognizing similar to human being, is a novel area of feature creation evolved [2].

Pattern Recognition

Pattern recognition assigns the given pattern to one of the known classes. A number of commercial pattern recognition systems exist for different applications. There are commonly used two methods; Template matching and Classification on Feature Space.

Template matching

Template matching compares the pattern with stored models of known patterns and selects the best match [7]. As match is expensive for different sizes and orientations, limited changes are done on the template, by stretching and deforming. Sometimes template is considered as consisting of smaller sub templates. Therefore different weights can be assigned to different special relationships. In general, template matching is suitable, where number of classes and variability within a class is small [1]. When an image is given for recognition, it is compressed until its average line height is equal to the height of the templates. Having compressed the image, the templates are matched against each character in the image [24].

Pattern classification based on feature space

In this method, features are summarised and classified accordingly. The main approaches are statistical methods, syntactic methods, neural networks and combinations of the above methods [2]. The accuracy of recognition is higher in feature extraction method than to pattern matching [11]. Using the natural language processing techniques, such as syntax analysis, semantic analysis, collation analysis, grammar rules, lexicons, contextual information and statistical language models would improve the accuracy further [11][14][15] [9].

Statistical pattern recognition methods are based on the Bayes decision theory, parametric and nonparametric methods. Bayes theory uses priory and conditional probability density functions and the probability for belonging to a predefined class is defined. Parametric classification methods assume class conditional density functions and estimate the parameters by maximum likelihood (ML), whereas nonparametric methods can estimate arbitrary distributions adaptable to training samples [2].

Support Vector Machines are based on the statistical learning theory of Vapnik. They analyse data and recognise patterns making it a non probabilistic method. It uses a binary linear classifier [2].

The neural network structures and learning algorithms are used in Artificial Neural Networks method for classification [2]. A neural network is composed of a number of interconnected neurons. The manner of interconnection differentiates the network models into feed forward networks, recurrent networks, self-organizing networks, and so on. In neural networks, a neuron is also called a unit or a node. A neuronal model has a set of connecting weights, a summing unit, and an activation function. The simplest technique used is the nearest neighbour approach, in which input pattern is analysed in several stages analogy to the human visual system. A decision function uses a threshold for deciding and this is called as learning.

In structural method the character pattern is represented as a feature vector of fixed dimensionality. The structural representation records the stroke sequence or topological shape of the character pattern, and hence resembles well to the mechanism of human perception and the input pattern is matched with the templates of minimum distance or maximum similarity [2]. It is often used with syntactical pattern recognition comprising linguistic and grammar. This was so difficult that it did not become popular. Two other techniques used in the structural methods, are attributed string matching and attributed graph matching [2]. The shape of the word is matched with a lexicon and the context of the same lexicon is used for improving the matching of the image to a word in the lexicon. To reduce the set of words in lexicon filters can be used [4].

Combining multiple classifiers is the usual method for decision making because, different classifiers vary in performance, vary classification accuracy and speed, and with different errors on concrete patterns. Structural recognition methods have some advantages over statistical methods and neural networks: They interpret the structure of characters, store less parameters, and are sometimes more accurate. Neural networks are considered to be pragmatic and obsolete compared to SVMs, but, they

yield competitive performance at much lower complexity. Potentially higher accuracies can be obtained by SVMs [17] and multiple classifier methods. There are many combination methods. They can be categorized into two; Parallel and Sequential. Parallel combination is more often adopted for improving the classification accuracy, whereas sequential combination is mainly used for accelerating the classification of large category set [2]. In most cases for accuracy of recognition, multi-classifiers of the same category are to be applied [5]. For example, different feature vectors of the image of the word, segmented individual characters and isolated characters can be processed with different classifiers of the same type [4].

Decision Making

Classifiers are combined and voter is used to get better result. The individual classifier output is ranked and the Borda count or Logistic Regression methods is used to make decisions. Non-parametric procedures, measure of confidence, statistical approaches, and formulations based on Bayesian analysis, Dempster-Shafer theories of evidence, neural networks, fussy theory and polynomial classifiers are also used in combination decisions [1]. Abstract level classifier majority vote is the simplest method and it is suitable for classifiers with one class. Majority vote technique uses simple majority vote or weighted votes. Classifiers are assigned unequal numbers according to their performance and weights are based on optimizing the value of objective function through a genetic algorithm [1].

Word Recognition

Depending on whether the characters are segmented or not, the word recognition methods can be categorized into two major groups: analytical (segmentation based) and holistic (segmentation-free). For modelling character classes, dynamic programming is used for word template matching and classification is based on vector representation of global features.

Classification-based recognition methods are primarily used for explicit segmentation. HMM-based methods can also be used for explicit segmentation, implicit segmentation, or holistic recognition which is used for non-segmented strings.

One or more recognition methods can be applied to make the decision on the word image. For noisy or degraded documents, recognizing the characters/ words becomes more difficult. In that case, relationship between the words in the document can be utilized [13].

Classification based recognition

String recognition is to classify the string image to a string class. A string class is a sequence of character classes assigned to the segmented character patterns. Character segmentation, character recognition, and linguistic processing can be formulated into a unified string classification model [18]. A string image can be partitioned into character patterns in many ways, and each segmented pattern can be assigned to different character classes. Classifiers should be trained at character level and string level to produce candidate pattern recognition. Candidates are identified by their minimum cost or maximum score on matching and there may be many candidates for a word image. There are two methods for searching the candidates; Exhaustive search and Heuristic search [5].

The context of Language can be defined by a set of legal words, called a lexicon, or a statistical form such as n-gram in an open vocabulary. The lexicon search strategy is also in two methods: matching with one entry and matching with all.

Segmentation-Recognition Candidates can be represented by a hypothesis network, called segmentation candidate lattice [2], [13]. Constraints are imposed on the candidates and pattern- class pairs are constructed to construct the Segmentation recognition lattice.

This can either be done before string recognition or dynamically during string recognition by using a lexicon. With the latter, single character/ character pair compatibility and linguistic bi-gram can be used.

Path search and lexicon organization strategies affect the time efficiency of string recognition. Single word/string recognition method is not sufficient as that a

combination of different methods is used [5]. HMM and Holistic methods can be useful in combining the methods.

Classification-based methods are particularly useful to applications where the linguistic context is weak, like numeral string recognition, or the number of character classes is large, like Chinese/Japanese character string recognition or in applications where the shapes of single characters are not discernable, like cursive word recognition [2].

Hidden Markov Model based recognition

This is the main technique for recognizing machine-print or online and offline handwriting data [2]. HMMs are extensions of Markov chains, which believe events occur according to an output probabilistic function; hence their description is a double stochastic process. HMM assumes observations do not depend on the previous ones but only on the current state.

There are two approaches: Implicit and Explicit Segmentation. In both methods, segmentation of the words into letters is done in the recognizing, which produces accurate results. In the first, data are sampled into a sequence of tiny frames whereas for the latter, text is cut into more meaningful units or graphemes, which are larger than the frames [2].

Holistic word recognition

It recognizes the word as a whole. In Holistic recognition a lexicon is used and features of the word shapes are compared with the lexicon [5], [8]. Holistic methods are useful for small and static lexicons. For large lexicons, they can be used with lexicon reduction. For dynamic lexicons the recognition system must have the reference models for all the words in the union of lexicons. Holistic recognition methods are reviewed from the perspectives of application domain, the nature of lexicon, the level and data structure of feature representation, and so on [2]. In this method, the three layers are extracted from the word image. The detected white area is reduced to one pixel wide top and base lines are estimated by using histograms of horizontally smeared image. For similar word shapes, feature templates detected by

convolution and threshold are used [5]. Word length, number and positions of ascenders/ descenders, holes, loops, near loops, number and direction of strokes, information on upper and lower contours of the word profile, end points and cross point are commonly captured global features [8]. The feature vector extracted for the word image was matched with a lexicon. It is useful in recognizing degraded documents and documents with wide range of qualities and different types of fonts. This method was successful in reducing the errors in premature recognition.

2.1.3 Post Processing

Human eye is able to read most texts irrespective of the fonts, styles, broken or missing, or with any distortion. But character recognition produces poor accuracy in those cases. In order to improve the accuracy of the OCR output, post processing is done. The objective of post-processing is to correct errors or resolve ambiguities in OCR results by using contextual information at the character level, word level, at the sentence level and at the level of semantics. There are many methods discussed on research papers.

The errors are introducing in recognizing characters introduced in segmentation and classification stages mainly due to the low quality images. Hence OCR output is facilitated including data validation and syntax analysis.

Lexical post processing is used to verify the OCR results using lexical knowledge and it is considered probabilities of letter transitions or extracts representation of all legal words. There are three approaches [3]. The first is a bottom up approach having three methods. In the first, binary probability or character probability of letter transitions is considered [3]. In the second, n-gram Markov models are employed and in the last, a combination of both is considered. For this approach, Viterbi algorithm is used [3]. The next approach is top down. It considers exact dictionary look up or error correction models for character sequences [4]. The exact dictionary looks up is implemented using tries or hashing. For error correction models, Levenshtein distance method or probabilistic models are used. The other approach is a hybrid of the two

[3]. The other methods used in research area are applying a lexicon look up for individual characters, which are reliably segmented in a word [4].

A lexicon lookup may be done in several forms. They are matching the recognized characters with a word in lexicon, comparing the features extracted by the recognized characters with the features extracted from words in a lexicon and comparing the features of the word scanned with features at a word level.

Lexical post-processing generates word hypotheses net for a word. For the selection from the word net statistical approaches such as frequency of words and word combinations (bi-grams), and synthetic parsing techniques such as linguistic structures and grammar rules are used.

Character level contextual post processing is mainly based on two types, statistical methods and lexicon methods [38]. In Statistical method, conditional probabilities of n-grams are gathered with training data to apply them to the testing data. In the other dictionary is used for correcting the errors in the recognized characters. Syntactical methods like grammar rules can also be incorporated to check for illegal character combinations. Some of such grammar rules are presence of two consecutive vowels or a word starting with a forbidden consonant or vowel [27].

The most common post-processing technique which operates at the word level is the dictionary look-up method [27]. Techniques based on statistical information about the language are also widely used [27]. In statistical method, letter n-grams are used to filter out unacceptable candidate words from the recognizer. An n-gram is a letter string of size n [27]. The probability of n-gram appears in a word is considered for each candidate word for the selection. In this case conditional probabilities in forward and backward directions are considered. Widely used n-grams are bi-grams and tri-grams.

There are countless post-processing approaches and algorithms proposing attempts to detect and correct OCR errors. In summary, those can be broadly broken down into

three major categories: manual error correction, dictionary-based error correction, and context based error correction [41].

3. SINHALA SCRIPT AND SCRIPTING

Sinhala Script is descended from the Brahmi script first documented in the edicts of Emperor Asoka of the third century BCE [23] and prevalent in the Indian subcontinent. Brahmi script is belonged to “abugida” or alpha-syllabary writing system which consists of vowels and consonants and a letter consists of a consonant, and a vowel notation [33]. A vowel can take the place of a consonant. Letters are written as a linear sequence, left to right. No difference exists in the size of the letters in a sentence as Capital or Simple.

Majority of Indic scripts based on Brahmi Script and have same features. They can be divided into three horizontal zones. Indic script identification is based on recognizing the vowels and consonants (basic characters) which constitute 94–96% of the text. Indic scripts - Devanagari, Tamil, Gurmukhi, Thai, Telugu, Kannada, Gujarati, Oriya, Bengali, Malayalam Urdu and Sinhala differ by varying degrees in their visual characteristics, but share some important similarities. Indic scripts present some challenges for OCR that are different from those faced with Latin oriental scripts as the complexity in the script, larger number of characters, characters are topologically connected, and being an inflectional language [40]. The speciality in Sinhala script is having more vocal sounds. It has rounded shaped glyph and has space between individual characters.

Although there are citations of research publications towards the OCR for these scripts, they are yet to achieve the commercial OCR products. Very little published research has been observed in the recognition of the Sinhala script [14].

3.1 Sinhala Language

Sinhala consists of 18 vowels, 41 consonants and 2 semi consonants totaling to 61 letters [32] as shown in Figure 3.1. Semi consonants are used to write vocal strokes

with speech sounds. There is a strong relation between the speech sound and the consonant when compared to English [35].

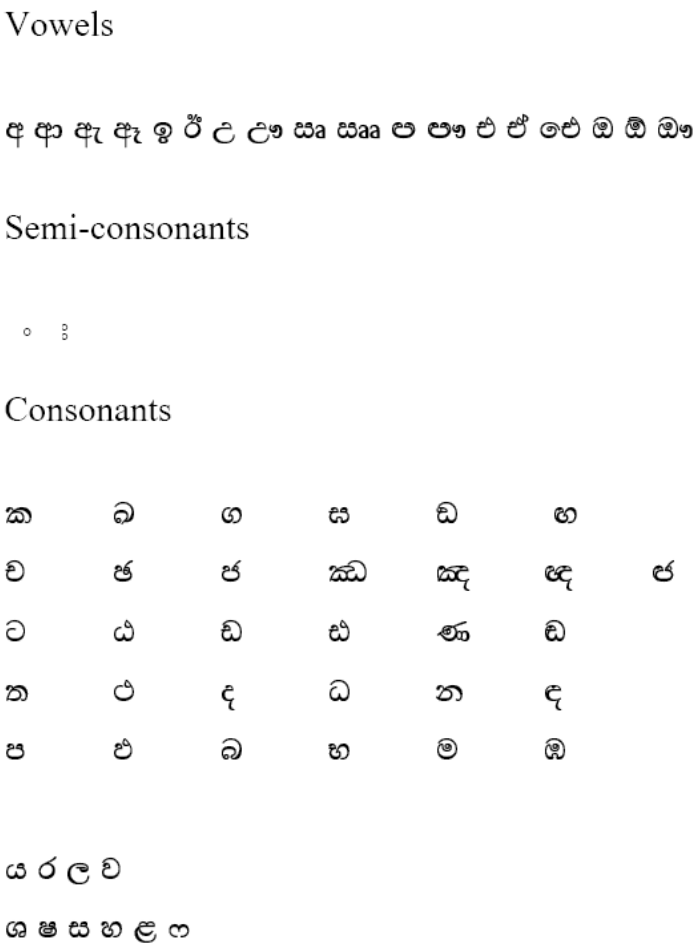


Figure 3.1 Sinhala Alphabet

All vowel forms are written in addition to the character. Each vowel has two forms; independent and dependent. The dependent forms are called modifiers and shown by special symbols. They are followed with consonant to make a composite character [32]. The modifiers may be one or many [20] in direction(s) in top, bottom, left or right to the consonant. A composite character is a combination of a consonant and a vowel as shown in Table 3.1.

Table 3.1: Different Consonant Modifier Combinations (with constant ‘ka’)

consonant	vowel	composite	Vowel form	sequence
ක	ආ	කා	ා	ක ා
ක	ඇ	කැ	ැ	ක ැ
ක	ඈ	කෑ	ෑ	ක ෑ
ක	ඉ	කි	ි	ක ි
ක	ඊ	කී	ී	ක ී
ක	උ	කු	ු	ක ු
ක	ඌ	කූ	ූ	ක ූ
ක	ඍ	කර	ර	ක ර
ක	ඎ	කෘ	ෘ	ක ෘ
ක	එ	කෙ	ෙ	ෙ ක
ක	ඒ	කේ	ේ	ෙ ක ේ
ක	ඓ	කෛ	ෛ	ෙ ෙ ක
ක	ඔ	කො	ො	ෙ ක ා
ක	ඕ	කෝ	ෝ	ෙ ක ා ේ
ක	ඖ	කෞ	ෞ	ෙ ක ෞ

More often, the composite characters have a different shape to its base (core) character but its shape is a combination of the consonant and the modifier both together. (Figure 3.2a) Consonant has an inherent vowel ‘a’ sound and its pure form is obtained by removing that using ‘al-lakuna’ (්). Sometimes, the composite characters have totally different shapes compared to the base character [20]. (Figure 3.2b) Some modifiers figures out different shapes for different base characters. (Figure 3.2c) This is valid for ‘al-lakuna’, ‘papilla’ and ‘diga papilla’. For ‘Al-lakuna’ forms are named as ‘kodiya’ and ‘raehaena’ whereas for papilla they are called ‘wak papilla’ and ‘kon papilla’ [35]. Even for the similar shaped composite characters as in Figure 3.2a, the modifier may be differing in size, orientation and appearance. (Figure 3.2d) Some modifiers have totally different shapes for different base characters too. (Figure 3.2e). Any vowel, consonant or composite character may be preceded to a semi-consonant. Hence, Indic

and South Asian Scripts are of much complexity and are more difficult for recognition than to Latin script.

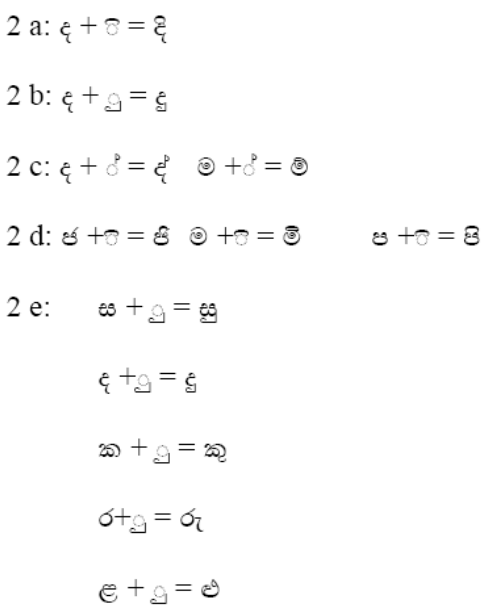


Figure 3.2 Different Consonant Modifier Combinations

The beauty of the script is consonants up to 3 can be combined to form ligatures (Figure 3.3 a). Of the ligatures, only the last consonant may contain a vowel form and even the inherent vowel sound is removed in other consonants in the ligature. Frequent conjoined forms occur with a consonants followed by speech sounds of '්‍ර' and '්‍ය' and they are called rakaransaya and yansaya respectively. Even consonants preceded by '්‍ර' makes a conjoined form and it is called repaya [32]. For these, special symbols are used. (Figure 3.3b). Any other consonant except '්‍ර' and '්‍ය' conjoins with the forms 'rakaransaya' and 'yansaya'. But, with repaya only '්‍ර' is not conjoined. The figure for 'repaya' with '්‍ය' has a different shape to normal repaya. (Figure 3.3c)

3 a: න් + ද = ඳ and

න් + ද් + ර = න්ද්‍ර

3 b: ක් + ර = ක්‍ර

ක් + ය = ක්‍ය

ර් + ක = ර්ක

3 c: ර් + ය = ර්‍ය and ර් + ය + ය = ර්‍ය්‍ය

Figure 3.3: Ligature Combination

Some consonants conjoined with each other to form conjunct characters. There are 12 conjuncts [31] (Figure 3.4). Almost all the consonants except '**ර** 'and '**ය** 'can make touching pairs. In those two cases, the inherent vowel of the preceding consonants has to be removed from the vocal sound as aforesaid. Touching pairs are very common in Buddhist and Old writings whereas conjunct characters are frequently used in Sanskrit writings, and Sinhala writings even in today's context.

ක්ෂ = ක් ෂ	ඬ = න් ධ	ත්ථ = ත් ථ	ධ = ද් ධ
ක්ව = ක් ව	න්ව =න් ව	ත්ව = ත් ව	ව = ද් ව
ඳ = න් ද	ත්ථ = ත් ථ	ඟ = ග් ධ	ඨ = ට් ඨ

Figure 3.4 Conjunct Characters

Any composite or conjoined character may be written with or without successive vowel form and with or without semi-consonant. (Figure 3.5)

ක	කං
කෝ	කෝං

Figure 3.5 : Semi Consonants

Similar to many Indic Scripts, Sinhala characters are also written in three strips. Sinhala characters can be classified into three non overlapping groups based on their relative heights in the 3-zone frame

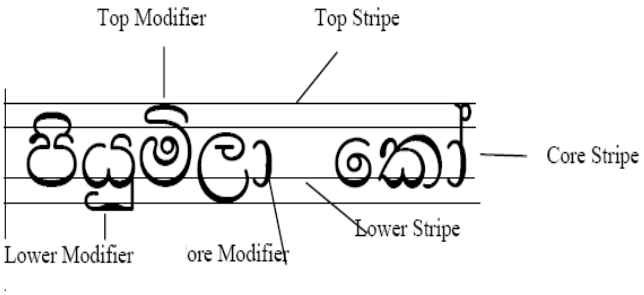


Figure 3.6 Three Zones in Sinhala Script

Some letters are written inside the core strip whereas others either move to top or lower strip [20]. In addition, modifiers are written in one or many sides. The first symbol takes the upper layer space; the second take the lower layer space while the fourth takes space in the middle layer after the consonant. The latter takes components in two directions. This is a unique characteristic belong to Indic scripts [19]. Any base character occupy in the middle, top and middle or bottom and middle layers [22]. Some modifiers use middle layer, and the others use middle and either top or bottom layers [20], [14], [15]. But composite characters belongs to 4 categories, according to the layers they occupy, middle layer only, middle and top layers only, middle and lower layers only, and middle, top and lower layers [24].

For Sinhala, separation needs in vertical direction to contain modifiers, which can not separate by spacing from the consonant, and in horizontal to break the conjunct or touching characters. Hence, multistage segmentation is used. In first stage, the vertical space is used as delimiter to extract the character images. In the second stage, based on the relative heights of image boxes, the tall image boxes are segmented horizontally for extraction of the lower modifiers. Finally, based on the relative width of image boxes, the wide image boxes are segmented vertically for extraction of the constituent characters of the conjuncts [33].

There are many syntactical rules in Sinhala Script which can be used in improving the accuracy in OCRs. In the higher level, sentence level grammar rules are related to subject and verb. Secondly in word level a dictionary can be used against the recognized word. Thirdly relation in-between individual characters confirm the recognition [32] [34].

Some of the syntactical rules are as follows [32], [34], [35].

1. The most beautiful 2 characters in the Sinhala alphabet are ඓ ඖ are currently not in use.
2. In addition ඝ is also very rarely in use.
3. When a letter can not be pronounced itself, it is pronounced with 1st letter අ. Hence, semi consonants and consonants sound with අ and make අං අඃ
4. Neither modifiers nor strokes go with Semi consonants.
5. No modifiers are used with ඞ too.
6. A word can not start with a pure consonant or a semi-consonant.
7. Usually a vowel does not come in the middle of the word. For that dependent vowel form is used. [35][20]
8. ඞ can be replaced with ◌ං but not the other way round (used in Pali or Sanskrit)
9. ◌a ◌aa are also used with words came from Pali or Sanskrit
10. ශ්‍ර ශ්‍ර pronounced similarly and the former is used in Sanskrit words. But the new words come from other languages like ක්‍රකර් has the normal sound of the adapilla.
11. Only one word starts with ණ and it is ණය
12. The 2nd and the 4th columns are named as ‘Gosha’ (more sound) and for them pure form (with al-lakuna to drop the inherent vowel) do not exist.
13. To display long ee sound, for the characters drawn up to the top layer, the base character preceded by 'kombuwa’ accompanied by ‘udupilla’ is used in top direction, instead of ‘kodiya’, which is the usual format for other characters.

14. For those consonants (Gosha) ෑ ෑ ෛ ෛෟ sounds do not exist.
15. Kundaliya comes at the end of a stanza
16. There are special Ligature forms for some composite characters :ර්ය ර්ව ළු ෆු
17. Before ට and ඩ, ණ is generally used instead of න, but not for new the words came from foreign languages
18. After ර ෂ ර්, for 'න' sound in most times ණ is used.
19. When a consonant in 3rd column comes after a vowel in the same row usually the 6th column character is written
20. The 5th column is called as 'sangaka' [20]. For those consonants pure consonant does not exist, and even some vowel forms (ෑ ෑ ෛ ෛෟ) do not exist as well.
21. These consonants (sangaka) does not use in the very first letter in a word without valid modifiers.
22. 'ර' does not follow by Yansaya & 'ශ' does not follow by rakaransaya and Repaya does not follow by 'ර'

The common feature for each character in Sinhala language is its inherent rounded shape. Some characters are different by adding additional feature. Eg: එ ඒ or ප ඹ or ට ච . This makes recognition bit difficult. In addition to that some modifiers placed on the characters make them confusing in recognition too. Eg: ම ම් මී. Rakaransaya and yansaya are common occurring components in Sinhala. They make rather complications in characters recognition. Adding components in any direction makes the language complex and that too reflex for recognition. Existence of conjunct characters and touching characters adds an extra challenge to meet for the effort.

3.2 Formation of Sinhala Words

In Sinhala language, it is identified that a root word is used to generate many number of word forms [20]. Root word is the basic and the smallest word invoking its meaning. Inflectional root words are stems and they are formed by the root word.

The same word stem is able to generate several numbers of nouns, adjectives, adverbs or verbs, considering tense, number, person and purpose etc. This enables a word in Sinhala language to be separated into prefix, stem, and suffix triples. Prefix is a leading common part used at the beginning of a word to alter the meaning. They are called "උපසර්ග" and 20 in number. Suffix is a trailing common part used at the ending of a word to form the exact representation in the meaning and many in number. A stem word is used with one or few prefixes or with many suffixes as well as depicted in the Figure 3.7.

Stem	frequency
ප්‍රකාශ-ය	63
ප්‍රකාශ-යක	2
ප්‍රකාශ-යකට	1
ප්‍රකාශ-යක්	49
ප්‍රකාශ-යකි	4
ප්‍රකාශ-යකින්	4
ප්‍රකාශ-යට	104
ප්‍රකාශ-යටපත්	1
ප්‍රකාශ-යන්	3
ප්‍රකාශ-යෙන්	6
ප්‍රකාශ-යේ	4
ප්‍රකාශ-යේදී	1
ප්‍රකාශ-වන්නේ	1
ප්‍රකාශ-වලින්	3
ප්‍රකාශ-වී	1
ප්‍රකාශ-වීමෙන්	2
ප්‍රකාශ-වේ	1
අ-ප්‍රකාශ	

Figure3.7: Words formed by the same stem

Suffixes affixed to a particular stem word can be grouped together. There are several number of word stems forming words with the same group of suffixes. Considering

this phenomena the stem groups and suffixes groups are identified as in Figure 3.8. This behavior is common for prefixes as well. Hence, groups can be identified for prefix/stem combinations and for prefix/stem/suffix combinations.

ළං	හැඩගැන්
හේතු	හැකි
හෙළි	හසුනො
හිර	හසු
හිමී	හමු
හැර	සෙල
හැඳින්	

Figure 3.8a: Some of the word stems for group1 suffixes

වනට	වන්න
වන	වන්නේ
වන්ටත්	වන්
වනතුරු	වන්ගෙන්
වන්ට	වනු
වන්නට	

Figure 3.8b: Suffixes in group 1

Therefore, there are four ways a stem word combines with prefix and suffix as in Figure 3.9.

Prefix	stem	suffix
අ	කරුණාවන්ත	වන

4 ways of forming words

කරුණාවන්ත

කරුණාවන්ත + වන

අ + කරුණාවන්ත

අ + කරුණාවන්ත + වන

Figure 3.9: 4 Combinations for a stem word

In addition, there are famous ten sets of grammar rules defined for the language in formation of words, but we do not go into details of them.

3.3 Statistics in Sinhala Language

The Statistics for the language by using UCSC lexicon [50] as the data store is as follows

Some of the statistics of the Sinhala lexicon are as below.

Number of words	= 6,57,131
Number of Unique words	= 70,142
Shortest word length	=1 (ද)
Longest word length (Unicode)	=24 (ජ්‍යෝතිශ්ශාස්ත්‍රඥයින්ගේ)

We used the characters which are isolated from their neighborhood for testing for the accuracy. Therefore, statistics were collected for those components as well.

For unique words,

Total # of segmented character components	=4,65,460
Mean component length	=4,65,460/70,142=6.635967038

For all words,

Total segmented character components read	=32,30,428
Average component length	=32,30,428/6,57,131=4.915957397
Longest word length	=15

(ජ්‍යෝතිශ්ශාස්ත්‍රඥයින්ගේ)

Hence for our purpose average word length could be considered as 5.

When we sorted the words according to the frequencies, the top 200 words has the frequency of 1,95,322 out of 6,57,131 total frequency of words. That is those 200 words out of 70142, or 0.285 % of words appear in 29.72% of the word space.

Mean word length of 200 words in unique words	=3.035
Average word length of 200 words in all words	=2.59

This shows us, most frequent words are very few and they are very short in length as in Table 3.2. See the Appendix I for the total list.

Table 3.2: List of top 20 words with frequency and component length

Word	Frequency	Length	Word	Frequency	Length
ද	6386	1	කළ	2819	2
ෙ ම	5236	2	අ ක ර	2725	3
ය	4844	1	ෙ ම ම	2690	3
ේ	4029	1	ස ඳ හ ා	2599	4
ම	3679	1	අ ැ ක	2585	3
අ ැ ති	3603	3	ක ර	2425	2
බ ව	3300	2	ව	2415	1
දී	3145	1	ෙ ල ස	2313	3
ව න	3064	2	ස හ	2271	2
වූ	2842	1	ව ය	2248	2

In contrast with that, the next 1600 most frequency words cumulate to another 30% of word space. Conjunct and touching letters are out of the scope in this research as the word list does not contain any of those.

4. POST PROCESSING TECHNIQUES FOR ERROR HANDLING

The objective of post-processing is to correct errors or resolve ambiguities in OCR results by using contextual information at the character level, word level, at the sentence level and at the level of semantics.

Character level contextual post processing is mainly of two types Statistical methods and using a Lexicon [38]. The both methods involve in detecting and correcting of one or more errors. In Statistical method conditional probability of n-grams are gathered with training data to apply them to the testing data. If all the n-grams for the word existed, the word is considered as correct. In the other method, dictionary is used. If the word is found in the dictionary it is assumed that all its characters have been correctly recognized. Otherwise the same dictionary is used for correcting the errors in the recognised characters.

In addition, syntactical methods like grammar rules can also be incorporated to check for illegal character combinations. Some of such grammar rules are presence of two consecutive vowels or a word starting with a forbidden consonant or vowel [27].

Word Level Error Detection and Correction

Contextual word recognition in post processing is performed on the OCR data stream at one level above character recognition, called the word level. By working at the word level, certain interferences and error rectifications are possible, which would not be feasible at the character level.

The most common post-processing technique operates at the word level is the dictionary look-up method [27]. Techniques based on statistical information about the language are also used as well [27]. In statistical method, an n-gram, a letter string of size n [27] is used to filter out unacceptable candidates, on which sub-strings of n-grams can not be generated, from the recognizer.

Use of N-gram

Riseman, E.M. and Hanson [39] used contextual processing based on positional binary n-gram statistics. The information differs from the usual n-gram letter statistics in that the probabilities are position-dependent and each is quantized to 1 or 0, depending upon whether or not it is nonzero (present).

Use of lexicon

Recognized words from the OCR engine are able to legitimate by using a lexicon. Hence, the input words are tested against a dictionary. In case the input word is found in the dictionary, the word is assumed to be correct and no more processing is done [19][20]. Otherwise the most similar dictionary entries are retrieved and considered as candidates [37]. Then candidate strings are generated by substituting the character in error by its confusion characters that are collected during the training phase [40].

Dictionary is not only used in post-processing, but also used in recognition stage. Especially in segment free method for word image matching either a dictionary or probabilities of statistics of words are used to match with them.

Lebourgeois et al. [12] described the general structure of automated document analysis for printed material. The character pre classification stage was used to reduce the number of patterns to recognize. Contextual processing introduced beyond the word spell correction after recognition. To distinguish the confused characters a tree optimization algorithm was used. Character prototype recognition was involved in 2 stages. They were extracting structural features with a horizontal Line Adjacency Graph and extracting statistical features by histograms the character image projection along four directions in horizontal, vertical and the two diagonal directions. The right hypothesis was selected by Contextual processing based on a Dictionary Viterbi algorithm using the substitution and transition probabilities.

Srihari et al. [4] proposed a method to recognize the characters with word shape analysis. It consisted of serial filters and parallel classifiers and the decision was made by combining a lexicon for best match case. The lexicon was used in two purposes: to match with the word shape in the image and to improve the matching the context knowledge of the words in a lexicon was used. The proposed system consisted of filters, classifiers and decision making mechanism. Filters were used to reduce input lexicon into a small set, classifiers take the filtered lexicon for ranking and decision making mechanism combines the results of the classifiers to produce the final ranking with the confidence score. The said 3 different approaches were combined in the classifier stage. Global feature extraction in the filtering process was used to estimate the word length and the word case. The set of classes were reduced by utilizing the maximum amount of reliability for input in degraded word images, character segmentation and recognize of any character in the word. If all the classifiers had agreed that could have been taken as the word. Regular expressions were used in constructing the constraints of the character positions. Then the word was graded by the number of constraints they match. Then the word was ranked and filtered out unlikely words. If there was no reliable decision the whole word was passed for classifier. They had used a 3 Classifiers approach: character based recognition, segmented word based recognition and word shape based recognition. In the character based method characters were isolated to be recognized and to be post processed with a dictionary to correct the recognition errors. It was believed better where segmentation of characters were not deformed especially for shorter words. Segmentation based word recognition method was applied and where characters could not be isolated the features extracted were matched with a lexicon. Word shape based method considered features extracted from the whole word to calculate a group of words in a dictionary to match the input with. Using all three methods together at different levels would give a better result. Character recognition used a fuzzy template matcher to identify individual characters and used heuristic string matching algorithm for post-processing to construct the set of possible strings. Then they are ranked against a lexicon considering the word length range and word case to take the character recognition decision. Segmentation based word recognition was used feature extraction from individual character image into feature

vectors to match them with the feature vectors of the lexicon. Then distance measures of the vectors were used to rank the words with a consideration of segmentation errors. Word shape analysis method extracted the details of the words and put them into a feature vectors. Then they are ranked by matching with feature vector of a lexicon. Two sets of feature extractions used were template defined and stroke direction distribution. Combination algorithm was designed to use the ranking of classifier output and computes the confidence score. Three confidence functions were used. They were highest rank method, Borda Count and Weighed Borda Count.

Error Correction by Using Confusion Characters

Misrecognition in OCR output are mainly due to similar shape characters which are called confusion characters. When confusion characters for each glyph position is considered, there may be few or many alternate words listed for candidates [12], [19]. Making all the possible words contributing to the word hypothesis would not be economical. Hence, a threshold has to be set to limit the candidates. Character ranking error in recognition is a better alternative for that. Top 3 ranked recognized characters for a glyph are taken into consideration [19]. For each position in the input word, there may be replacements making a huge word hypothesis. Within those suggestion words the selection is done by considering the likelihood score which is computed by statistical methods such as similarity measures and word frequencies [19]. The majority of the non word errors will be solved by this.

Chaudhuri and Pala [40] in their Bangla OCR use a simple strategy for post-processing dealing only with single character error in a word. The dictionary was used to look for an exact match. In case an exact match was not found, the candidate strings were generated by substituting the character in error by its confusions that were collected during the training phase. A dictionary-based error-correction scheme had been used where separate dictionaries were compiled for root word and suffixes that contain morpho-syntactic information as well.

Sinhala Language has a larger number of confusion characters [20] and there are groups of base characters which are similar in its shape [14], [15]. Those groups are identified and penalties can be assigned according to the level of confusion.

Use of Dictionaries as an Alternative Source for the Statistical Information

Takahashi et al. [28] proposed a spelling correction method using string matching between the input word and a set of candidate words selected from the lexicon. They classified and multi-indexed an input word according to a constant number of characters selected from the input word ignoring the relative position for selecting the candidate words from the dictionary. As a result, inappropriate words were selected as candidate words. The selected words were matched with the given word by approximate string matching and a penalty was assigned for the mismatch in the length, for mismatch in the position of the characters being matched and for mismatch between characters. They used two types of penalties one for a mismatch and the other for addition/deletion.

Integrated Multiple Sources of Knowledge

Xiang Tong and Evans [29] Included letter n-grams, character confusion probabilities, and word-bigram probabilities. Letter n-grams were used to index the words in the lexicon. Given a sentence to be corrected, the system decomposes each string in the sentence into letter n-grams and retrieves word candidates from the lexicon by comparing string n-grams with lexicon-entry n-grams. The retrieved candidates were ranked by the conditional probability of matches with the string, given character confusion probabilities. Finally, the word bigram models with Viterbi algorithm were used to determine the best scoring word sequence for the sentence. The system was able to correct non-word errors as well as real-word errors.

Veena Bansal and Sinha [18] proposed a system with various knowledge sources integrated in hierarchical manner. The knowledge sources were in statistical and lexical forms and a transient source was built while processing. The language structure, grammatical rules and Geographical features were used in recognition. The recognition system was based on segmentation and classification. The character

classification was based on a hybrid approach. The segmentation was based on statistical analysis of height and width with a suitable threshold. Then three layers were identified for text and were further divided according to the visual features of the script. Feature vectors for character classification, horizontal zero crossing for the image, two dimensional moments and 9 zone pixel density were taken and kept as transient knowledge. Using the structural properties of the Devanagari script a nine zone primitives were defined to filter out the candidates. The distance from the prototype to the matching character was taken as the confidence. The confusion matrix was built for characters with difficult in identifying. This confusion matrix was used in correction the output with a dictionary. The composition rules identified the sequence of symbols and assured that they were syntactically correct. A word envelop feature comprising the number of different features characters, script specific characteristics and corresponding vectors, upper and lower modifiers, were used to select the candidate words from the dictionary. The word dictionary was based on the certain features extracted during the process. In addition for mostly resembling character pairs rules were defined to take the decision. The binary image was segmented into characters and symbols for linearization. Various statistical information was produced on that to identify the characters. Then different knowledge sources were invoked to filter out some of the candidates and the confidence figure was considered with the remaining candidate hypotheses which were composed into words after verification and correction.

Jason et al. [11] proposed a new technique for Chinese OCR post processing and post-editing based on natural language processing for noisy documents. Contextual knowledge required for processing complex, confusing, huge character set is provided by the Language model. The system consisted of Error detection and Correction units. Error detection unit was used to re-consider the confidence of the error count using a statistical model of the image using the distribution of the relation between correct candidate against the error count whereas the Correction units which consists of noisy channel and language model to suggest possible corrections. To re-confirm the detection a dictionary was referred. For long words of 2 or more characters (Chinese) word segmentation were used to correct the errors while for

single character words bi-gram model was used. The noisy channel model was useful in correcting where the character is missing from the candidate list and they said was very effective in correcting errors.

Another example for multi knowledge use was in Thomas et al [10] proposed system. It had string matching algorithms with five types to use with a limited Vocabulary for OCR output correction.

Structural Properties of Characters/ Words in Recognition Process

Structural properties of the script are used for deciding the penalty of a mismatch, which makes the recognition task easier. For instance, the core characters are divided into three classes based on the region of the core strip covered [19]. Confusion similarity measures are decided on that regional or whole character information.

Sharma et al **[**42] designed a method based on shape similarities of the characters. Similar character subsets were made and numbered. Subset number for each OCR output word was computed. Words with same subset number were added to the same node. Codes for each word were implemented on an AVL tree and breath first search was stored in a dictionary file. When dictionary was looked up for the code of the input word, if the exact match was found, the word was considered correct. Even though the word was not in the list but it was grammatically correct, that too was considered as correct. Otherwise the words exist for the same codes were suggested as candidates. In other case if the code was not found but word was correct according to grammar rules it was added to the dictionary.

Structural features were used by Lehal and Singh [27] with robust font- and character-size independent for identification of visually similar words. Structural properties in 3 horizontal strips were analysed by Veena Bansal and Sinha [19] in assigning a penalty for mismatch in the lexicon.

Use of Linguistic Features/ Grammar rules in Recognition

Use of Linguistic features/ grammar rules has major role in recognition of characters. In Sinhala language, it is observable as in Figure 3.7; that a root verb produces 15

verb forms [20], several number of nouns and few adjectives and adverbs. Therefore, to implement all the words in all forms, in a lexicon would be very cumbersome and computationally costly. Further a dictionary can not be completed at whatever size, because of the new coming words. More frequent word parts can be taken as prefixes or suffixes depending on its relative position in the word. Then Root Stems are sought in Root List for identified suffixes and a root suffix pair together makes a legitimate word [30].

Chaudhuri and Pal [30] had developed an OCR error detection and correction technique for Bangla. They had used two separate lexicons of root words and suffixes, candidate root-suffix pairs of each input word were detected, their grammatical agreements were tested and the root/suffix part in which the error had occurred was noted. The correction was made on the corresponding error part of the input string by a fast dictionary access technique and alternative strings were generated for the erroneous word. Among the alternative strings, those of which satisfying grammatical agreement in root-suffix and also having the smallest Levenstein distance were finally chosen as correct ones.

In addition, separate dictionaries had been used by Chaudhuri and Pala [40] for their dictionary-based error-correction scheme. It had separate dictionaries compiled for root word and suffixes that contain morpho-syntactic information as well. Lehal and Singh [27] too used Punjabi grammar rules to eliminate illegal character combinations in corpora look-up in Panjabi OCR.

Edit Distance Method

A string recognized may differ from the original string due to three reasons. They are fragmenting one character into two, combining two characters into one or recognizing it as a different character. The minimum number of character positions needs to change the recognized characters to the original is called as the Edit distance between the two.

R.A. Wagner and Fischer [45] developed an algorithm to find the minimal sequence of edit operations for changing the given string to another. The length of the edit sequence was considered as the Levenshtein distance between the two strings.
The concept of Edit distance was further developed by H. Bunke [38] in his research based on classical approach. He assumed that one string was known apriori which was in a dictionary. The dictionary words were converted into deterministic finite states and stored. The finite states of the input word was also obtained and edit distance is computed between the input word and each entry in the dictionary.

Edit distance algorithm with a different method was proposed by Thomas et al [10]. It used Bayesian probability matching method to get the probability of match with the dictionary word. The frequency of occurrence of the dictionary word and the probabilities and frequencies of all the other dictionary words were considered to calculate the ranking score. A process called thinning was done to reduce the dictionary to high-probability candidate words before processing with the Bayesian function. For that the confidence values were taken considering the bi-grams of the words and the dictionary with the selected candidates was build.

Partitioning the Dictionary

A word, at least apart may be correct. Hence, the dictionary was partitioned by the proposed a method of Veena Bansal and Sinha [19] **.**The research concerned the issues of incorrect recognition of Devanagari character specially caused by fusion and fragmentation of characters by using a Hindi dictionary. The dictionary was partitioned into two as short words and remaining. The advantage of the partitioning of the dictionary was it reduces the search space as well as prevents forced matches. Next short words were further partitioned based on word envelop, character combination and presence of modifier symbols and word length information. The remaining was partitioned using a tag of fixed length string associated with the partition. After recognition exact match was sought from the partitioned dictionary words based on the input. A word may have an entry in one or more partitions.

Xiang Tong and Evans [29] too used letter n-grams to index the words in the lexicon. The (OCR) string also parsed into letter n-grams, and treated as a query over the database of lexicon entries including the 'beginning' and 'end' spaces surrounding the string. The “term frequency” of n-grams was observed for input word and the dictionary word. A threshold was set for candidates and the Viterbi algorithm is used to get the best word sequence for the strings.

The corpus used by Lehal and Singh [27] to develop a post processor for Gurmukhi was partitioned into two levels. For first level the corpus was split into seven disjoint subsets based on the word length and at second level, shape of word was used to further segment the subsets into a list of visually similar words. He generated a dynamic list of structures from each of the sub-list. Those dynamic lists were based on visually similar characters.

Word Hypothesis Net

Each error correction process involves in generating a word hypothesis net or a word lattice. Considering all the possible candidates would be not computationally economical. Hence limits are introduced for each measure to allow the most probable set of candidates to proceed.

Xiang Tong and Evans [29] had to rank the retrieved candidates in the lexicon matched with the string by the conditional probability in order. Ideally, each word in the lexicon should be compared to a given OCR string, to compute the conditional probability. However, this approach would be computationally too expensive. Instead, the system was put to operate in two steps, first to generate the candidates and then to specify the maximal number of candidates, N, to be considered for the correction of an OCR string.

Inter Word Relation within a Document

Same word is more probable to appear on the same document for several times. Those words are featured with similar characteristics. This inter-word relationship facilitates identifying the words efficiently. If two word images had been equivalent,

their recognition results should have been the same. Word image equivalence was only one of several visual inter-word relations.

Tao Hong and Hull [43] proposed a novel OCR post processing method based on word image equivalence having at least one other image in an input document. There were usually many occurrences of the same words. It was first located clusters of equivalent words in a document. The visual equivalence among the word images was computed by word image clustering and majority voting method.

A novel method was suggested by J. Hull and Hong [13] for using the visual relationships between word images in a document to improve the recognition. Conventional OCR systems isolate the characters and post processes the decisions with a lexicon. If the images are noisy, poor in identifying and the image of the text is not further considered. Taking the equivalent word images previously occurred in the document and special arrangements of them in various ways can be used to identify common set of words use in the document. In this process the recognition overcomes the noise in the document. The accuracy of OCR was further improved by employing post processing algorithms. In their research they used character based clustering and deciphering algorithms with modifications and concept of self teaching OCR system in classifiers. The relationships were defined as six types as equivalent, sub images, left part of and right part of and sub images of one word from sub-image of another for left and right parts. The sub images occur due to frequent words, prefixes and suffixes. Those relationships of the equivalent images were analyzed for post-processing. Six relationships are detected in steps by using clustering algorithms.

Prototypes were used to generate a quality image for the noisy images. Post processing algorithm was used to locate the word decision with high confidence. Then these words were used to learn images correspond to characters and character sequences. The learnt words were used to decompose the remaining images. The process is defines in 4 steps as voting, font learning, verification and re-recognition. The first 3 steps produced a lattice of overlapping sub-images and candidates for output by applying suitable thresholds.

Massive Dictionary

Solutions consist of using a lookup dictionary to search for misspelled words and correcting them suitably. While this technique tries to solve the actual problem, it in fact introduces another problem, the integration of a vast dictionary of massive terms that covers almost every single word in the target language.

Additionally, this dictionary should encompass proper nouns, names of countries and locations, scientific terminologies, and technical keywords. To end with, the content of this dictionary should be constantly updated so as to include new emerging words in the language [41]. It is almost impossible to compile such a wide-ranging dictionary.

Christian et al. [9] proposed a method to overcome the shortcomings of using a general dictionary, as there was a high probability to miss a word in a particular area of the document was based on and because of that the frequencies of words occur may also inaccurate resulting invalid candidates. Even a large dictionary or a dictionary with frequent words might be thematic. Therefore for the concept of ideal dictionary, effective dictionary was exploring with three categories static large dictionaries, dynamic dictionaries retrieved from web pages and use of a mixed of the two for lexical coverage and for accuracy. The optimal results are taken from the combined approach.

Youssef Bassil and Alwani [41] proposed a system perform on the series of cybernetic operations addressing those secondary problems arising in the contextual post-processing methods. His algorithm considers detecting and correcting of OCR non-word and real-word errors. Since in practice it is almost impossible to compile a wide-ranging dictionary, it would be wise using a web of online text corpuses containing all possible words, terms, expressions, jargon, and terminologies that have ever occurred in the language. This web of words can be seamlessly provided by Google search engine. Words chunks of 5 are fed into Google search and observe hit or suggestion. Google predicts next probable word using n-grams in words using web pages. The actual correction consists of replacing the original block in the OCR

output text by the Google's alternative suggested correction. His research addresses the following limitations dictionary-based approach requires a wide-ranging dictionary that covers every single word in the language, regular dictionaries normally target a single specific language and thus they cannot support multiple languages simultaneously, conventional dictionaries do not support proper and personal names, names of countries, regions, geographical locations and historical sites, technical keywords, domain specific terms, and acronyms, the content of a standard dictionary is static in a way that it is not constantly updated with new emerging words unless manually edited, and thus, it cannot keep pace with the immense dynamic breeding of new words and terms.

Real Word Problem

In Dictionary look up method either it accepts the word if it is exists in the dictionary, or rejects otherwise. Some words are rejected not because they are incorrect but because they are not included into the dictionary. Those are called False Negatives. Some words are accepted by the dictionary test but those are not the word in the OCR input. Those are called False Positives. This has been defined as non word error and real word error [29] respectively. False negatives become false positives if the dictionary is comprehensive [20]. But major part of false negatives is occurred due to misrecognized characters [29]. False Positives are not the real words even though they are dictionary words and they can only be corrected by taking context into account [29].

Most traditional word-correction techniques concentrate on non-word error correction and do not consider the context in which the error appears, in other words real word errors [29]. A non-word error occurs when a word in a source text is interpreted as a string that does not correspond to any valid word in a given word list or dictionary. A real-word error occurs when a source-text word is interpreted as a string that actually does occur in the dictionary, but is not identical with the source-text word [29]. Statistical language models have used to correct false positives [29].

Tong and Evans [29] had statistical language model with word bigram table to correct real-word problems. He used Viterbi algorithm to select the best word sequence from the candidate pairs of words.

Church and Hanks [44] had an alternative approach addressing the real word error. It was word collocation evaluation implemented at OCR output to perform post-processing. He employed mutual information to extract pairs of words that tend to occur within a fixed-size window (normally 5 words). Word collocation tables express the probability of two words being found in the text in the given order.

Work on Sinhala Script

There were several researches on Sinhala OCR systems during the last decade. Premaratna and Bigun [14] suggested a system for Sinhala OCR for the first and the foremost time. A segmentation free algorithm was used to recognize characters in Sinhala script as modifier based scripts are more difficult to segmentation. The algorithm was based on feature extracted for orientation. The theory used in the recognition process was the orientation field tensor, local neighbourhood characterized by the grey value changes in one direction, local orientation denoted as linear symmetry in a vector containing the orientation angle and the certainty measure. The edge detection algorithm using linear symmetry recognises vertical modifiers. The linear symmetry principle was also used to determine the skew angle. He had used a syntactical post-processing technique to distinguish confusion characters from the group members. Later, it was identified that feature extraction methods were not effective in word recognizing.

Premaratna et al. [15] extended his research to recognize characters using direction features apply for scripts consisting of large number of characters. The direction features was further used in the separation of confusion characters, detection of skew angle, segmentation of script and graphic object in addition to the features extracted in [14]. LS Tensor was taken by Local space filters and Gaussian derivative filters and a 3D vector was kept. Skew correction was done using moments and low pass filtering was used to separates the text area from the graphics area. Then horizontal

projection was done to adjust the sizes of the image to match with the templates. The system was trained for template matching. For recognizing characters Orientation field Tensor was used. Correlation of the LS tensor for the confusion characters within the same group was high as that a secondary filtering was done. The LS Tensor output was taken by suppressing non-maxima within a 3x3 neighbourhood and stored as row and column number and arranged into the words. A lexicon was used to detect the missing characters. Further enhancements to the recognised script can be achieved by using the HMM. The scope of the research considered the Ethiopic script as well.

Premaratna et al.[20] proposed another method using HMM. He identified that the complexity of the script confuses the feature based recognition and as that ANN was also in vain. The context based method would improve the word level accuracy. In this method missing characters were detected by comparing the word to be compared with a lexicon. The structural rules define by the language had motivated the HMM. It was a novel method to use HMMs for recognition with confusion characters. For HMM 5 tuples were used for finite set, output, probability of being in state, probability of next state given that the current state, and output probability matrix. The Viterbi algorithm calculates the probabilities recursively. It provides the optimal path by evaluating the probabilities. As a verb stem in Sinhala may form 15 different verb forms the lexicon used would be limited. To identify the characters of confusion groups a robust mechanism based on lexicon was used. It was in two stages recognition and optimization. For optimization Viterbi algorithm in HMMs was used to produce the most likely chain of characters. Grammar rule were also incorporated into it. Missing character widths were also detected and characters are suggested to fill the gaps using a lexicon. If that process fails, either State Transition Matrix or LS tensor method is used. Confusion Matrix was build by using probability of confusion between each and every character.

Another approach to identify Sinhala, Tamil and English scripts from a single document page was proposed by Umapada et al.[17] That too based on feature extraction. The water reservoir principle was applied in extracting features. Extracted

features were reservoirs in left, right and bottom for right convexity, top reservoir, and geographical features like height, distance from left, size dissimilarity, position and size of a dot and vertical strokes. He used a SVM with Gaussian Kernel based classifier. Two stage classifications were done first to identify Sinhala and then English and Tamil with different set of kernel parameters. In this research line and word segmentation had been done. Histogram based approach was used to convert the image into two tone to be solved by the kernel.

Weerasinghe et al.[16] proposed a method based on syllabification of Sinhala language. Syllabification algorithms are mainly used in text-to-speech. Rules defining the syllable boundaries of words were based on theories of Maximum Onset Principle and the Sonority Profile. The theories were different from language to language. The rules were sensitive to the sequence and a syllable was converted into speech. As this has no relation to our purposed, further details were omitted from the report.

The Google open source Tesseract OCR engine supports many languages [48] and it has become more accurate compared to other off-the-shelf OCR engines [48]. It's an open source OCR engine and claims to have higher accuracy than the commonly available commercial OCRs [48]. It can view images and translate in many languages. It started at HP labs as a research project and improved by Google. It has two parts the OCR engine and training data for a language [49]. The system has to be extensively trained for a particular language. Training of the Sinhala words is being done by University of Colombo School of Computing [47]. The output accuracy of the Sinhala OCR is at a satisfactory level at a character level. But accuracy of words recognized is not in a satisfactory level. That was the base for our system to run on.

A Sinhala corpus of 10,000,000 words has also been accumulated by UCSC under PAN localization project [50]. In addition to that there are text to speech, translation and transliteration software available for Sinhala script. [50]

5. A DICTIONARY BASED METHODOLOGY FOR SINHALA SCRIPT ERROR HANDLING

In the OCR output, some words recognized are correct and some are incorrect. If the words in the output is mismatching to the words in the original document, by single character or multiple characters, the word is incorrect. The method we implement to the system is using a dictionary and a word in the output checked for a hit in the dictionary. If it has a hit, we consider the word correct and otherwise, it is considered as incorrect. In this research dictionary is used to detect the errors of OCR output words, and to correct them in 3 stages.

The UCSC lexicon was selected as primary data source for the word store in our research [37]. The lexicon comprises statistical information of frequency for each word. The Google Tesseract OCR, trained by UCSC [47] for Sinhala language was selected as the OCR engine to get the OCRed text [36]. Sample size of 10 [Sample1] of tiff formatted files which contained 2765 number of words was selected from the newspaper cuttings to train the system. For testing the system, 115 articles with 30240 words, was selected from a daily newspaper publishing in Sinhala language. The samples selected, are only with body text. Otherwise, the OCRed text becomes unreadable even by human eye. The samples were scanned at 300 dpi and gray levels were adjusted to cope up the off-white background noise contained in the newspapers and transparency on the print of the over leaf.

Then text was OCRed and corrected manually to identify the OCR errors and the identified error list is available in Appendix A. Errors found on the samples were mainly in three types, replacing English letters by different English letters, replacing English letters for Sinhala glyphs and Sinhala glyph not being recognized accurately. Apparently, there were very few insertions and deletions errors caused by OCR engine in the output text of the samples. There were very few errors, caused by word fragmentation or combination. But, there were many hidden characters which were

not in the actual text. Therefore, a filtration module was introduced to run before the proposed system to improve the accuracy of the OCR output.

Preliminary Process

A normal user can obtain OCR text output for an imaged document by using any OCR engine available. For enabling them to use this software with any Sinhala OCR engine available, to improve the accuracy of the output the software was developed in such a way that it is able to run on OCRed text output, which is in correct Unicode sequence [6][31]. But, the OCRed text we obtained was in an intermediate stage and they were in Unicode but in the glyph order as those appear on the image. Therefore, the text was converted into true Unicode sequence before applying the module on the samples. The code for that conversion was combined into the earlier said filtering module and the output is as shown in Figure 5.1..

Some misrecognized characters were also rectified within the same process. One of that were line ends and paragraph ends appearing as short breaks i.e. 000A. As we consider flowing of text smoothly we replaced those with space characters. However, line ends and paragraph ends can not be identified separately, to separate paragraphs in the final formatted output. The next one, treated in the same, was unwanted Zero Width Joiners (ZWJ – 200D) appeared in between Sinhala characters. Those were also filtered by using the syntax logic; ZWJ appears in the places where if and only if it is preceded by a consonant character followed by ‘hal lakuna’ and it is in succession with another consonant character. The second consonant may or may not have a vowel form. Of course, more constraints in linguistics rules can also be applied there in depth as follows. If the second consonant character had been either ‘්ර’, 0DBB or ‘්ය’, 0DBA any other consonant would have been the first. In the other hand, if the first consonant character had been ‘්ර’, 0DBB, any other consonant would have been the second consonant character. In all other cases, the first consonant should be one of ක, න, ත, ද or ට and the second consonant would be in the respective group ෂ/ච, ඨ/ච/ධ, ච/ඨ, ච/ධ, ඨ. In Pali writings ZWJ would be preceded by consonant character and followed by ‘hal lakuna’ and same or another consonant

character should be in succession. In our research, general news was considered. Hence, we did not focus on Pali writings.

The next most frequent error was '෧', 0DB8 preceded by @ sign. @ was removed after observation of the samples, even though the actual character sequence could have contain the same sequence. But in our case, no @ sign was appeared before '෧ ' in the original text. Other errors found for filtering were 'ර ' recognized as 6 and the rightmost glyph for long O sound recognized as 5. Those were so rare cases that can be neglected. Hyphen sign is used to indicate continuation of the word. Instead of hyphen character, some other character (:) was appearing. Those were just dropped to make the word continue without splitting into two. Almost all the remaining errors, for which an action was not taken and the majority of the total errors visible to human eye, were misrecognized Sinhala characters.

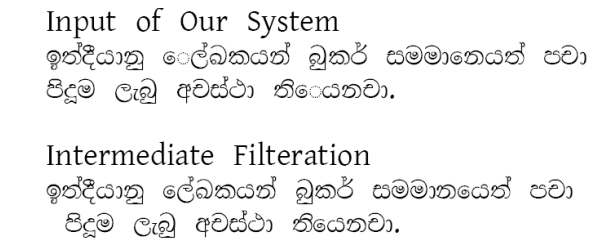


Figure 5.1: Intermediate output of the system

Stage1: Confusion Vector Pair List

At the first stage, the errors had to be detected, and at later stages corrections were dropped on the detected errors. Hence, we first checked the words against a dictionary to detect the errors. If the word is a HIT in the dictionary, it is assumed to be correctly recognized words. No further processing is done on those words. Otherwise, it is left for error correcting process at later stages.

When we checked, each recognized word, in the dictionary, some of them were found and some were not found. Some of the hit words were also different to its original word. Few were found, but, with no meaning, which were not valid words.

This happens because of the unclean data in the dictionary. Both these error types belonged to False Positives. On the other hand, some of the words were not found in the dictionary, even though it should have been, because the dictionary does not contain that word. They are called False Negatives.

False negatives are a subset of words which are not marked as correct. They remained same unless those words are inserted in to the dictionary. But, they will not change by any error corrections, as those make no hits in the dictionary. False positives are the words marked as correct, but it is not the original word. Hence, by correcting word errors the number of false positives may be increased.

There are many words in the OCRed text, which are marked as incorrect because of the word makes no hit in a dictionary due to an error present in the output word. Our next step was to correct those detected erroneous words. Human eye does this at a glance by substituting misrecognized characters by the meant characters or by similar characters, to get the meaning of the context.

First we tried to correct single words errors, which are the majority of the word errors. Substituting those errors by its original real character solves is a simple and efficient method. Statistics for the manually corrected errors shown as in Figure 5.2 can be used for this purpose.

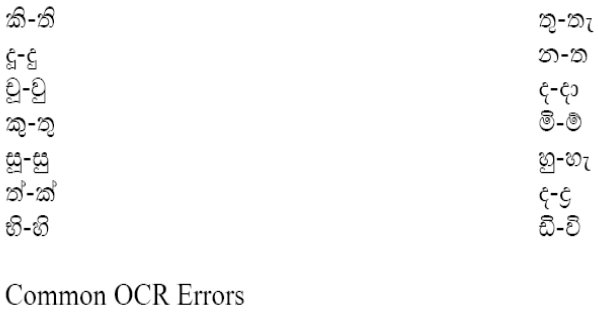


Figure 5.2: Common Errors; Recognized and Real character Pairs

Many of the pairs in the list are comprised with vertically separated single glyph components as in Figure 5.3. The number of errors present for each recognized and

the original pair is different. This is shown in the manually corrected list in Appendix A. At the same time, frequency of the individual vertically separated glyph component appears in general text is also different. To collect those statistics, the dictionary itself was used by component separation as depicted in Figure 5.3.

Figure 5.3: Component Separation of Words

The list of manually corrected errors shows us a single recognized character may be misrecognition of one or several number of original characters as in Figure 5.4. Each pair has a different probability to be occurred. Hence, it should be given a different measure for each pair according to the statistics obtained for component frequency and probable error count. We call it the similarity measure or confusion level. In our case, similarity measure is assigned for each pair in proportional to errors found in the samples and the similarity between the two.

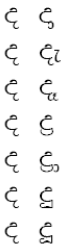


Figure 5.4: Same Misrecognized Character for Different Real Characters

In addition, many of those errors occurred due to a partly misrecognition character. For example ◌ි and ◌ී can be interchangeable in many strings. Therefore, instead of having all the confusion pairs in the list, for each consonant with the same common error, only that part which is the common of the glyph was considered for replacement in a mismatch. Hence the errors were short listed by grouping similar

errors for all characters. The pairs are consisted of confusion characters and they have a direction, the first character is to be matched with misrecognised character and in case of a match, it is to be replaced by the second character. Hence, we named it as Confusion Vector Pair List. The Confusion Vector Pair List was constructed by using the short listed common errors found by manually and available in Appendix B. The size of the individual character(s) in the Confusion Vector Pair may or may not be of same length in its code representation, but, at most times both are same in glyph size.

Confusion Vector Pair List is a list of two separate strings of characters with a similarity measure. Of the two, the first is the matching sub string to the recognized string part and the second is the probable candidate sub string. A hypothetical word is constructed by replacing the recognized string by the probable string if the matching sub string is the same. Then that hypothetical word is searched in the dictionary.

There may be some confusion character pairs, which were not in the training samples, but can occur through the OCR process for general text. After all, not each and every error found is considered for Confusion Vector. Less frequent errors are omitted in the Confusion Vector Pair List. This may leave some errors without correcting. But it will be useful in order not to scarify the computational time and to omit forced matches. We can get them corrected at a later stage in multiple error correction. Figure 5.5 shows the state and Appendix C contains the proposed list.

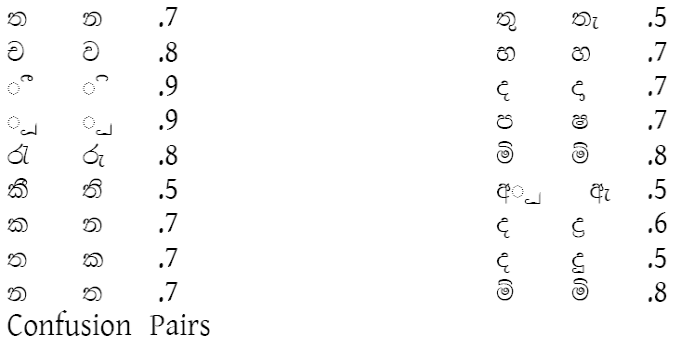


Figure 5.5: Confusion Pairs with Similarity Measurement

Since the software checks all the possible confusion vector pairs given in the list, the order of the vector is irrelevant. But having the list in a sorted order would be advantageous as by using binary search methods the system run time would be shorten for those set of vector sequence.

For an incorrectly recognized word, all the possible words with single error correction are considered. A word hypothesis is generated by replacing those confusion characters taking one at a time and by testing that with the dictionary for a hit. All the words candidates generated by Confusion Vector Pairs embed the word's frequency as well as aforesaid confusion or/and similarity measure. The likelihood of the word is measured by multiplication of the two scales, similarity and frequency and the word with the highest score becomes the selection as depicted in Figure 5.6. Further post processing is continued provided HIT is not found for the word in the Dictionary. The majority of words remaining are of multiple errors. But there are few false negatives and single error words, which were not corrected.

Word	candidates	measure		word-frequency	likelihood
විය	විය	0.8	x	2248	1798.4
	ීය	0.8	x	204	1.92
	විස	0.8	x	0	0
	විෂ	0.5	x	0	0

:. Selection = විය

Figure 5.6: Scores for likelihood

Stage 2: Using Structural Features; Prefix, Stem and Suffix

The next stage is introduced to facilitate a limited dictionary size. In Sinhala language, there are many words with frequently occurring strings, at starting and ending positions. Those are called Prefixes and Suffixes respectively. The remaining word part is called a Stem. If a suffix and/or a prefix exist for a word, the stem is to be checked against the dictionary. There are 4 combinations for a stem to be combined with prefix or a suffix as in Figure 3.9.

It can be observed that as in Figure 5.7, prefix and suffix lists increase the number of false positives by forced matches in the dictionary, if only one stem group is considered (default). To avoid that, separate stem lists are supposed to build for each suffix or prefix. Since all the stem words appear in a prefix list must be appeared in the relevant suffix list. Therefore, that would make much difference.

අප <cp>ඉදිරියේ<\cp> ඇති අභියෝග ජය <cp>ගනිමින්<\cp> ජාතික <sp>සංවර්ධනයන්<\sp>, ජාතික <sp>සමඟියන්<\sp> රැක

Figure 5.7: Prefix & Suffix Lists increase False Positives

As in Figure 3.7, a stem word can be affixed with many suffixes, to make a suffix group for the stem and Figure 3.8 illustrates that, a suffix group can be affixed with many stem words, to make a stem group for a suffix group. The total list of prefixes and suffixes are available in Appendix D & Appendix E.

Suffixes match with the same stem word, are grouped together and same group number is given for each suffix in the group. Different numbers are given for the different groups. Hence the relevant stem list could be referenced by the group number. When there are no groups available for a suffixes or when the particular group is not available, the common dictionary was referred as the default stem list. The stem word part was extracted for suffix groups and. 12 groups were identified as shown in Figure 5.8 a, and the total list is available in Appendix G.

In the same way prefix groups are built as shown in the Figure 5.8 b. It was observed that, out of the said 20 ‘උපසර්ග ’there are only few prefixes which contain much stem words. Hence we have only 4 prefix groups. Similarly the prefix lists are available in Appendix D.

කරුවන් 13
කරුවන්ගේ 13
කර 2
කරගත්ත 2
කළ 0
කළේ 0
Figure 17 a: Suffixes and their Groups

අ 1
ඉ 0
අව 2
අනු 0
Figure 17 b: Prefixes and their Groups

Figure 5.8: Prefixes and Suffixes with their Groups

Prefixes are used either to enhance or to diminish or to deny or to change the meaning of a word. All the 20 prefixes in Sinhala language accompany with noun stems. නො is the only prefix, which combines with verb stems as well as with noun stems. There are hundreds of suffixes and those are able to accompany with either noun stems or verb stems. There are many words which have both prefix and suffix.

In the case of prefixes, this behavior is different to suffixes. There are separate stem groups for each prefix. There are few stem words found in a list except one.

This prefix-root-suffix method reduces the size of the word list referencing, for the stem word, and avoids making forced matches

The words with no HIT at the stage 1 are checked for suffixes first, as it's rather logical to have a suffix than to a prefix because prefixes have short stem lists. The suffix may be bit lengthier than to a prefix. The longer the string matched with, the higher the confidence to be the real word.

There may be more than one suffix matches with the OCRed string as shown in Figure. 5.9. In that case, the entire stem – suffix combinations are considered to generate candidate words and highest score of the likelihood is considered in selecting one at the end.

අයිතිකරුවන්ගෙන්
අයිති-කරුවන්ගෙන්
අයිතිකරු-වන්ගෙන්
අයිතිකරුවන්-ගෙන්

Figure 5.9: Many suffixes matching with OCRed string

In this stage, word is sought to have suffix or a prefix. If one or both found, the relevant stem list is referred for the remaining word part. We could identify different Stem lists coupled to different categories of suffixes and prefixes. Depending on the suffix or prefix or both, the stem list to be considered is different. But in our research to avoid more complicated lists we limit searching to suffix groups only when a prefix –suffix both exists. As the prefixes we considered belong to a special featured group of the Sinhala language all the members in a stem list belongs to a prefix are members of the relevant stem list for the suffix.

In the stage 2 all the possible words are considered in searching a hit word in dictionary, and if it made a hit, the word is added to the word hypothesis net as well as in stage 1.The final selection would have the maximum score for the likelihood. Frequency of the stem word and statistics of the probability of the suffix/prefix contributes to the score.

There are different combinations of prefix and suffix, and stem words in the input text as in Figure 3.8. There is more possibility being a word part inaccurate as depicted in the Figure 5.10. The statistics for training data proved this too. Hence, we introduced considering confusion characters in this stage too. When the system separates the components into prefix stem and suffix and matches are sought, the system is capable of correcting single errors in each component, allowing correcting up to 3 errors in the string as well. Confusion level measure and word frequency of the stem word is considered, similar to the procedure used in stage 1, for a successful candidate in calculating its likelihood.

සාහිත්‍යකරුවන්
යටත්ජීතකරණයේ
සමමානයෙන්

Figure 5.10: Errors present in word parts

A word is a combination of glyphs, which are vertically separated from their neighbourhood. We observed the frequency of the each glyph in the dictionary by isolation of character boxes that are vertically separated from their neighbourhood. This is called mono-gram. All the probable n-grams and their frequencies were observed in n-grams up to 5-grams as shown in Figure 5.11. Appendix H contains the list. We ranked those in descending order of frequencies. Then we analyzed for probable word parts for prefixes or suffixes. This logical word parts can be incorporated with grammar rules to generate words from a root word list. In addition, these n-grams can be used with statistical methods in post processing. But, in our research we do not go to that depth, so we search only the logical and meaningful word parts for suffixes and prefixes.

√ච්ය	4087
√ගැන	4050
√කාර	4041
√ීන	3955
√ළඟ	3955
√කිරී	3926
√න්ගෙන්	3913
×ැද	3835
√ච්	3817

Figure 5.11: Word Parts by N-grams probable for suffix/prefix with Frequency

Stage 3: Confusion Groups

Provided having unsuccessful HIT in above mentioned 2 methods, to correct multiple errors in the remaining words, either Edit Distance Method or Confusion Groups List

is used. Confusion Group is a group of confusion characters with similar shape.

.8 හ ග ශ ඝ ඟ ඁ
.8 ඩ ධ ඬ ඨ
.7 උ ල ළ
.6 ඊ ඒ
.7 ද ද ඳ දු
.8 ළු ළූ ළු ඵ ඨ ඪ

Figure 5.12: Confusion Groups

Characters with slightly differences grouped and ranked at high values, whereas characters with much difference are grouped and ranked at low values as in Figure 5.12. The total list is available in Appendix F. But for the suspicious misrecognized character in the group is considered first and the full value, which is 1, is given for its similarity, whereas the remaining members in the group, are given the group similarity measure.

Character at any position in the input word can be replaced by its confusion group member and this makes a large list of candidates as in Figure 5.13. When this replacement is done for each and every position in the input word, a large word hypothesis net is created. In addition there may be more than one confusion group for a character.

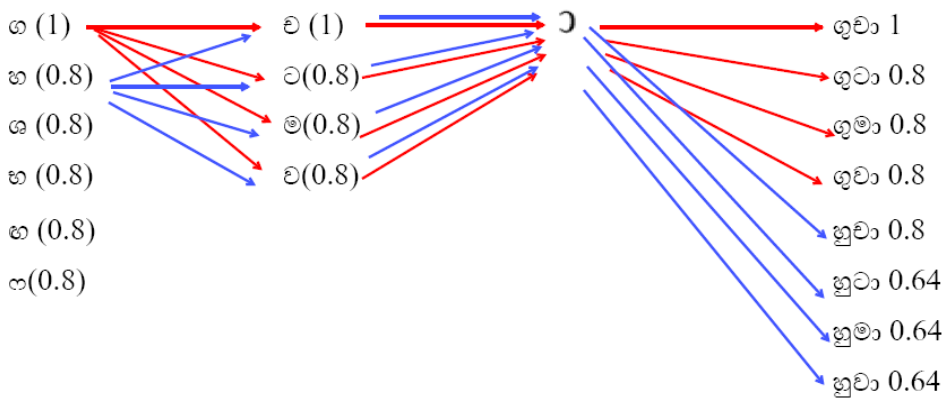


Figure 5.13: Generating of Word Hypothesis (simplified for clarity)

Therefore, a long list of words is generated to be matched with the dictionary. Hence, it's a time and resource consuming exhaustive search. Therefore, penalties can be introduced to limit the word net.

Each replacing character group is bundled with a measure for the degree of similarity as in the confusion vector pairs. With each replacement, the confusion level increases so that a penalty is set for 1/100 allowing 6 possible replacements. If the confusion level goes beyond the limit, consideration for generating probable words will stop for the node penalty level exceeds and continue generating the words with the other nodes. Confusion level is calculated by multiplying of similarity measures for all the possible node characters whether they are recognized or found in confusion groups.

Finally likelihood of the candidate words are manipulated by multiplication of the penalty measure and the word frequency and the best scored word is selected as in the other cases.

This method affects unnecessary burden on the system. Hence we selected this at last, provided no option is successful. But this method is capable of correcting the multiple errors occurred at any position in the word together. Too long words are also omitted in considering in this method. According to the Appendix I, most frequent words are shorter words and there are few inter word space missing in recognition. Hence, this will not hope to give adverse result.

The word hypothesis generated in the stage 3 of the system is too large to end up the resources. Grammar/ Syntactical rules are also incorporated into the stage 3 in order to reduce the set of word hypothesis. When a word is going to generate, it is checked for some rules, for which combinations can not happen together.

We tried few syntactical rules for filtering real characters. For confusion character replacement some of the rules are considered in order to limit the exhaustive search.

They are

1. no vowel is followed by another vowel
2. no vowel can present in a middle of a word
3. no vowel can have a dependent vowel (if present consider for replacement of both)
4. consonants can have only a definite set of vowel forms
5. the first letter in a word can not start with a pure consonant

The dictionary and all the stem word lists are implemented using a "Trie" data structure for its high search efficiency as the system forces many words to get searched for a given input word. The system was coded in c++ using Trie library routines in [46]

Algorithms

Algorithm for the System

```
// corrects single error in a word
Repeat for all the OCRed in a file
     Extract a word
     If Sinhala word search it in the dictionary
          If a match found, write into the output
          else
               Generate words with confusion pair list[1]
               if word with confusion character found
                    write the best match into the output
               else
                    Generate words with prefix-root-
               suffix combinations[2]
                    If a match found write the best match
into output file
                    else
                         Generate words with confusion
                    groups[3]
```

```
                    If a match found write the best
match into output
     Else write it into output file // assume digit/
English alpha/ punctuation
```

Algorithm for confusion pairs

```
Repeat for each component in a string from left to right
     For each confusion pair in the list{
          If match found
               Generate word replacing component with
          confusion
               Test the word against the Dictionary
               If a hit add the word to candidate list
and manipulate the likelihood
     }
Select the highest scored candidate
```

Algorithm for suffix/ prefix

```
//corrects single error in a word part

Repeat for each suffix in the list{
     Check the right substring with the suffix
     For each exact match and match with confusions{
          Extract the word stem
          Match with a root word in the stem group for
          the suffix
          If match found
               Generate the candidate word by combining
          stem & suffix
```

```
            Else
                  Generate the confusion words for stem
                  Check for hits in the suffix group
                  If match found
                        Generate the candidate word
                  Else
                        Test for prefix in the substring
                        For each exact or confusion match {
                              Extract the stem word
                              If match found
                                    Generate the candidate word
                                    with prefix/stem/suffix
                              Else
                                    Generate confusion word
                              stems
                                    Check for hits in the group
                                    If found
                                          Generate the candidate
            word
                                    Else // no matching with
            prefix/suffix

                        }


      }

}
Repeat for each prefix in the list {// assumed no suffix
      Check for prefix in the left substring
      For each exact match or match with confusion {
            Extract the stem word
```

```
          If match found
               Generate the candidate word with prefix
          stem & suffix
          Else
               Generate the word stems with confusions
               Check for hits in the group
               If found
                    Generate the candidate word
               Else // no matching found
     }
}
// no prefix no suffix is omitted considering here as it
is covered by step1
Select the highest scored candidate
```

Algorithm for confusion groups

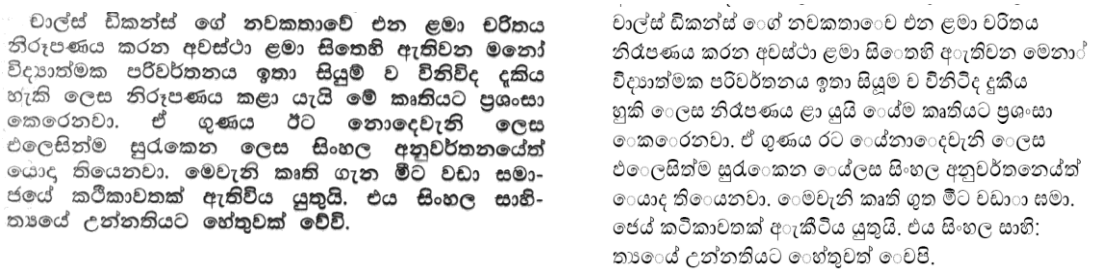
```
Read a component in a string from left to right
Do {
     While not end of the confusion pair list {
          Check a confusion group with the component
          If match
               Add the add the group members for the
position
          Else
               Consider only the component for the
position
          If the right part of the string from the
component not empty
               Iterate the routine for right substring
          Else generate a word
```

```
                By assembling the components for the
          positions
          Test the assembled word against the Dictionary
          If a hit add to candidate list and manipulate
the likelihood
     }
} While not end of string
Select the highest scored candidate
```

6. EVALUATION OF PROTOTYPE IMPLEMENTATION

The system was trained by 10 numbers of samples and the set of data for testing was 115 articles published on a daily paper. The training sample space contains 2689 words whereas the testing sample consists of 30,240 numbers of words.

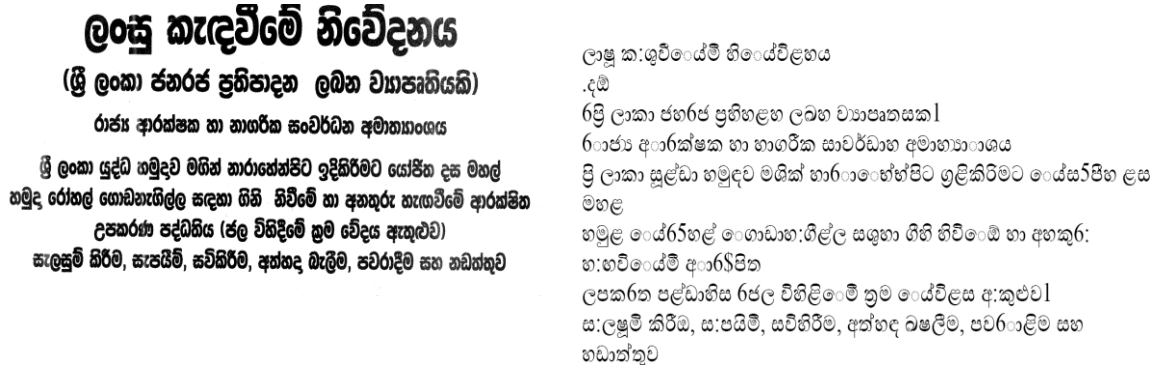
Image file and its OCR output for a body text and for a styled text are shown in Figure 6.1 and Figure 6.2 respectively. It shows that if the image files contain only the body text, the OCR process recognizes it more accurately than the image files with bold or any other formatted text or styled text. Hence documents only with body text are selected as the sample data for our purpose.



චාල්ස් ඩිකන්ස් ගේ නවකතාවේ එන ළමා චරිතය නිරූපණය කරන අවස්ථා ළමා සිතෙහි ඇතිවන මනෝ විද්‍යාත්මක පරිවර්තනය ඉතා සියුම් ව විනිවිද දැකිය හැකි ලෙස නිරූපණය කළා යැයි මේ කෘතියට ප්‍රශංසා කෙරෙනවා. ඒ ගුණය ඊට නොදෙවැනි ලෙස එලෙසින්ම සුරැකෙන ලෙස සිංහල අනුවර්තනයේත් යොදා තියෙනවා. මෙවැනි කෘති ගැන මීට වඩා සමා-ජයේ කථිකාවතක් ඇතිවිය යුතුයි. එය සිංහල සාහි-ත්‍යයේ උන්නතියට හේතුවක් වේවි.

චාල්ස් ඩිකන්ස් ෙග් නවකතාෙව් එන ළමා චරිතය
නිරූපණය කරන අවස්ථා ළමා සිෙතහි අැතිවන මෙනෝ
විද්‍යාත්මක පරිවර්තනය ඉතා සියුම් ව විනිවිද දුකිය
හැකි ෙලස නිරූපණය ළා යුයි ෙයේම කෘතියට ප්‍රශංසා
ෙකෙරනවා. ඒ ගුණය රට ෙයේනාෙදවැනි ෙලස
එෙලසින්ම සුරැෙකන ෙයේලස සිංහල අනුවර්තනෙයේත්
ෙයාද තිෙයනවා. ෙමවැනි කෘති ගැත මීට වඩාා සමා.
ජෙය් කථිකාවතක් අැතිවිය යුතුයි. එය සිංහල සාහි:
ත්‍යෙය් උන්නතියට ෙහේතුවත් ෙවපි.

Figure 6.1: Sample tiff and its OCRed text for a body text



ලංකා කැඳවීමේ නිවේදනය

(ශ්‍රී ලංකා ජනරජ ප්‍රතිපාදන ලබන ව්‍යාපෘතියකි)

රාජ්‍ය ආරක්ෂක හා නාගරික සංවර්ධන අමාත්‍යාංශය

ශ්‍රී ලංකා යුද්ධ හමුදාව මගින් නාරාහේන්පිට ඉදිකිරීමට යෝජිත දැස මහල් හමුදා රෝහල් ගොඩනැගිල්ල සඳහා ගිනි නිවීමේ හා අනතුරු හැඟවීමේ ආරක්ෂිත උපකරණ පද්ධතිය (ජල විහිදීමේ ක්‍රම වේදය ඇතුළුව) සැලසුම් කිරීම, සැපයීම්, සවිකිරීම, අත්හදා බැලීම, පවරාදීම සහ නඩත්තුව

ලාෂූ ක:ශුවීෙේම් හිෙේවිළහය
.දඕ
6ශ්‍රී ලාකා ජහරජ ප්‍රහිහළහ ලබහ ව්‍යාපෘතසක1
6ොජ්‍ය අොරක්ෂක හා හාගරීක සාවර්ධාහ අමාහ්‍යාොශය
ප්‍රී ලාකා සූල්ඩා හමුදව මශික් හාරොෙන්පිට ග්‍රළිකිරිමට ෙේස5ීහ ළස මහළ
හමුළ ෙේ65හල් ෙගොඩාහ:ගිල්ල සශුහා ගිහි හිවීෙඕ හා අහකූ6:
හ:හවිෙේම් අා6ීපිත
ලපක6ත පල්ඩාහිස 6ජල විහිළිෙම් ක්‍රම ෙේවිළස අ:කුළුව1
ස:ලෂූම් කිරීඔ, ස:පයිම්, සවිහිරීම, අත්හද බෂලීම, පව6ොළිම සහ හඩාත්තුව

Figure 6.2: Sample tiff and its OCRed text for a styled text

As we need statistics about the language, the Dictionary was used as the resource for that purpose too as stated in Chapter 3 Section 2.

Errors found in the OCRed text were observed as in Table 6.1 and listed in Appendix A. It was observed that, OCRed text is in glyph sequence. To get those run on our system, they had to be rewritten in correct Unicode sequence. In addition, we observed many unwanted ZWJ marks, which were hidden on the OCRed text due to insertions in the OCR process due to some reason, Unless they are really need at those places, they were removed. There were short breaks (00A0) for both line ends and paragraph ends, and those were replaced by space characters. In addition, @ proceeded by 0DB8 character was removed and hyphens appeared as . or - or were dropped for word continuation. Further we observed, that 6 was recognized instead of ◌ි 0DBB at few places, 2 was recognized for / and 5 was recognized for the rightmost part of long O character. As total percentage of them was 0.2 (0.2%), we focused on the areas where majority of errors occur.

Majority of the above said errors, were observed to be, misrecognized characters. Only a few errors were found with inserted or deleted type errors. Hence, we focused on misrecognized errors with the training samples.

Total no. of words = 2689
Total no. of components =13675
Misrecognized characters = 264
Misrecognition errors = 1248
No. of @◌ found = 55
No. of 6 found for ◌ි = 8
Hyphenations found = 6
Short breaks found at end of each line
ZWJ in actual use was 1% out of the ZWJs found in the OCRed text.

Table 6.1: Summary of Errors found on OCRed Training Sample

Recognized Characters - Real Character	Number of errors
ත්-න්	161
ච-ව	153
යේ-ෙ	38
දී-දි	38
ත-න	38
රෘ-රු	32
ටි-වි	25
කී-නි	22
තූ-තු	20
කි-නි	19
දූ-දු	18
චූ-වු	16
කූ-තු	16
සූ-සු	14
ත්-ක්	14

When we checked, each recognized word in the dictionary, some of them were found and some were not found. Some of the hit words were also different to its original word. Few were found, but, with no meaning, which were not valid words. This happens because of the unclean data in the lexicon. Both these error types are belonged to False Positives. In the other hand, some of the words were not found, even though it should be because of the reason that the lexicon does not contain that word - they are called False Negatives. The two types are illustrated in Figure 6.3a. In order to reduce false positives we cleaned words in the dictionary as far as possible, and to reduce false negatives we added the valid words, which were not marked as correct in the training data. There were false positives, even for the text on training samples. The reason for that is either the acceptance of invalid words which

will makes hits in the lexicon, or the acceptance of valid words, which too makes a hit, but the context is different to the original word.

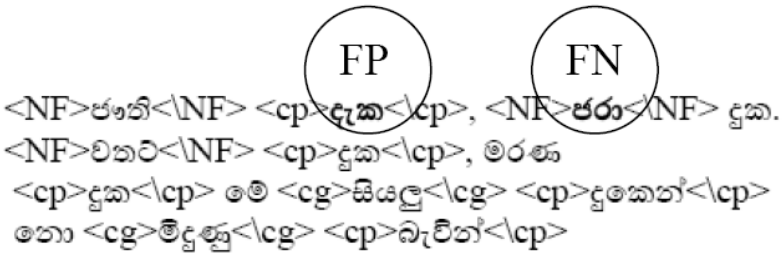


Figure 6.3: False Positives & False Negatives

Output of the System
<cp>සිද්ධිය<\cp> වූ <cp>දින<\cp> <sp>යාලුවොන්<\sp> එක්ක එකතු වී

<cg>අවුරුදු<\cg> පාට
දමා <cp>විනෝද<\cp> වී තිබෙනවා.

Figure 6.3b: False Positives: more probable in single component words

False positives are valid words but not the original, whereas false negatives are original words but invalid according to lexicon used. The original lexicon was not purely cleaned, and comprehensive. Hence there were false positives and false negatives respectively. In order to reduce false positives we cleaned words in the dictionary as far as possible, and to reduce false negatives we added the valid words, which were not marked as correct in the training data.

We observed that there is more probability for a recognized character, which is of a single component, validated being False Positive as in Figure 6.3b. There are many foreign words blended with Sinhala language, and they have no meaning in the sense of Sinhala, but, they may have hits with the lexicon. Valid Sinhala words with single component will contribute to the same as well.

Statistics for the Samples

Table 6.2: Output of OCR text without error correcting

OCR output	Training Data	Testing Data
Total number of input words in the samples	2689	30240
The number of words detected as valid	1331	18092
The number of words detected as not valid	1358	12148
False positives (validated but not the original)	19	435*
False negatives (not validated but original)	30	1164
Accuracy	50.9%	59.8%

* Extrapolated value by taking 10% of the sample

Next, the research focused on error correction. Stage 1 of the system, consists of a simple but efficient error correction mechanism based on Confusion Vector Pairs. For the training samples of 2689 number of words, 684 words were corrected with confusion pairs leaving 697 words detected as erroneous.

Table 6.3: Output of OCR text after stage 1

After Stage 1	Training Data	Testing Data
Total number of input words in the samples	1358	12148
The number of words detected as valid	684	7588
The number of words detected as not valid	674	4560
False positives	21	200*
False negatives	30	1164
The % of the errors corrected at this stage	51.44%	62.46%
Accuracy after Stage1	74.93%	84.92%

* Extrapolated value by taking 10% of the sample

Then we used one of the linguistic features, prefixes and suffixes of the Sinhala script to increase the accuracy of recognition of the OCR process. We tried the listed n-grams in Appendix H to obtain prefixes and suffixes, but it was in vain. Hence n-grams were combined with linguistic features to produce a list of sensible prefixes

and suffixes. That was also of not much use. Then we stuck to the structure of the words.

Some words, which were not validated at stage 1 like අංශයේදී were validated in this stage by breaking the word into two parts and matching with prefixes, suffixes and stem lists. At the same time there were false positives among the validated words such as සමුදාට (ස-මුදා-ට), because the stem remaining after breaking up of prefixes and suffixes was forced to match with the default list. Hence, the group lists were identified and for suffixes there were 12 in number whereas it is 4 in number for prefixes. The dictionary is considered as the default stem list, if and only if no group list is available.

There are different combinations of prefixes and suffixes, and stems in the OCR output text. Sometimes, the word parts may or may not be accurate as depicted in the Table 6.5. Hence, we extended the system to detect and correct single errors in the word part. When the system separates the components into prefix stem and suffix and matches are sought, it is able to correct a single error in each component as well. The output of samples is listed in Table 6.4.

Table 6.4: Output of OCR text after stage 2

After Stage 2	**Training Data**	**Testing Data**
Total number of input words in the samples	623	4560
The number of words detected as valid	100	692
The number of words detected as not valid	523	3868
False positives	25	306*
False negatives	30	1164
The % of the errors corrected at this stage	16.05%	15.17%
Accuracy after Stage2	80.55%	87.2%

* Extrapolated value by taking 10% of the sample

Table 6.5: Stage 2 output of the words with different errors

Input Word	Output Word	Explanation
ලස්සන	ලස්සන	No error
ලස්යන	ලස්සන	Stem error corrected
ලස්සනවන	ලස්සනවන	Hit in suffix list
අවලස්සන	අවලස්සන	Hit in prefix list
අවලස්සනවන	අවලස්සනවන	All 3 parts without errors
ලස්යනවන	ලස්සනවන	Correction in stem
අවලස්සක	අවලස්සන	Correction in stem
ලස්සනවක	ලස්සනවන	Correction in suffix
අමලස්සන	අවලස්සන	Correction in prefix
ලස්යනවක	ලස්සනවන	Correction in both stem & suffix
අමලස්යන	අවලස්සන	Correction in both prefix & stem
අවලස්යනවන	අවලස්සනවන	Correction in stem
අවලස්සනමන	අවලස්සනමක	Alternative correction for suffix found before වන
අමලස්සනවන	අවලස්සනවන	Correction in prefix
අවලස්යනමන	අවලස්සනමක	Correction in stem & suffix
අමලස්සනමන	අවලස්සනමක	Correction in stem & suffix
අමලස්යනවන	අවලස්සනවන	Correction in prefix & stem
අමලස්යනමන	අවලස්සනමක	Correction in all 3 prefix, stem & suffix
අවලස්යතවන	අවලස්යතවන	No hit

In Table 6.5, we observed that there are false positives introduced even in this consideration, because මන in the word අවලස්සනමන found a suffix corrected for confusion as මක and the group for the suffix is not available so that the system read the default list that is in the dictionary and made a hit.

The system matches the word parts in the relevant list either suffix or prefix. Sometimes a direct match cannot be found. Then, the word part is altered using the confusion pairs in stage1, and checked for a match. We set the probability for the both cases equally. But the winning word is the word with the highest score. Hence, the measure given to the word part should be different, and a direct match should be given a higher value than to the altered match. Further each word part has a different frequency of occurrence as well. In addition, we considered the first found word part for the altered one, instead of all the possible words for economy. The above reasons contribute to false positives as shown in the lines 15 and 16 in Table 6.4

Stage 3 of the research is based on Confusion Groups. The multiple error correction mechanism is really involved with this technique on OCRed text. Since this is an exhaustive search the process consumes both considerable time and reasonable resources. For lengthier text it will be worse. We observe that sometimes the inter word space character is missing and a long string is there to feed to the system. Hence a limit has to be set to avoid the unnecessary burden on the system. In our research, it is 5 composite character lengths. Confusion characters for the components in the OCRed string haves individual similarity measure. When generating a word, component by component as shown in Figure 6.5, the similarity measure multiplies and the confidence level drops down. Since there is no use of having words with too low confidence levels, a penalty level is set. In our case it is 0.01 which allows words up to 6 confusion levels as shown in Figure 6.4.

Input
කරුණාවත්ත කරුණාවත්ත කරුණාවත්ත කරුණාවත්ත
කරුණාමත්ත කරුණාමත්ත තරුණාමත්ත තරුණෑමත්ත

Output
කරුණාවත්ත <cp>කරුණාවත්ත<\cp>
<cg>කරුණාවත්ත<\cg> <cg>කරුණාවත්ත<\cg>
<cg>කරුණාවත්ත<\cg> <cg>කරුණාවත්ත<\cg>
<cg>කරුණාවත්ත<\cg> <NF>තරුණෑමත්ත<\NF>

Figure 6.4: Sample for Confusion group Corrections

0 ගුවා 1	0 හුමා 0.64	0 භුවා 0.8	0 භුමා 0.64
0 ගුටා 0.8	15 හුවා 0.64	0 භුටා 0.64	0 භුවා 0.64
0 ගුමා 0.8	0 ශුවා 0.8	0 භුමා 0.64	0 ෆුවා 0.8
0 ගුවා 0.8	0 ශුටා 0.64	0 භුවා 0.64	0 ෆුටා 0.64
0 හුවා 0.8	0 ශුමා 0.64	0 භුවා 0.8	0 ෆුමා 0.64
0 හුටා 0.64	0 ශුවා 0.64	0 භුටා 0.64	0 ෆුවා 0.64

Figure 6.5: Exhaustive search of a word with likelihood score &Confusion Level

Word hypothesis net is constructed as follows. The group ට ම ව ව has similarity of 0.8 and භ ග ශ භ භ ෆ has 0.8 for that too. Each confusion member in the group is taken one at a time to generate words. At any position, the leading part is considered as fixed and the trailing part can be one of the all possible combinations generated by

replacing each component by its group member at the relevant location as depicted in Figure 6.5. The matching component in the group is considered first with similarity measure as 1, and the other members are of the group similarity. Confusion level is attached to each generated word and it is a multiplication of all the similarities of the combinations linked together. The likelihood is a multiplication of the generated word frequency and confusion level and it is shown before the word as in Figure 6.5.

By using this technique at stage 3 of our system, there are 108 number of words out of 523 words, which is a 20.65% validated in the training sample, whereas the number for testing sample is 1631 from the total of 3868 making the validation by 42.17%.

Table 6.6: Output of OCR text after stage 3

After Stage 3	**Training Data**	**Testing Data**
Total number of input words in the samples	523	3868
The number of words detected as valid	108	1631
The number of words detected as not valid	415	2237
False positives	12	35*
False negatives	30	1164
The % of the errors corrected at this stage	20.65%	42.17%
Accuracy after Stage3	84.57%	92.6%

* Extrapolated value by taking 10% of the sample

Therefore the total accuracy of the system increases from 52.2% to 84.57% for training data and it's an increase from 59.8% to 92.6% for testing data, after integrating all three techniques. A sample of final output is shown in Table 6.6.

The result after applying the proposed system is shown in Figure 6.6.

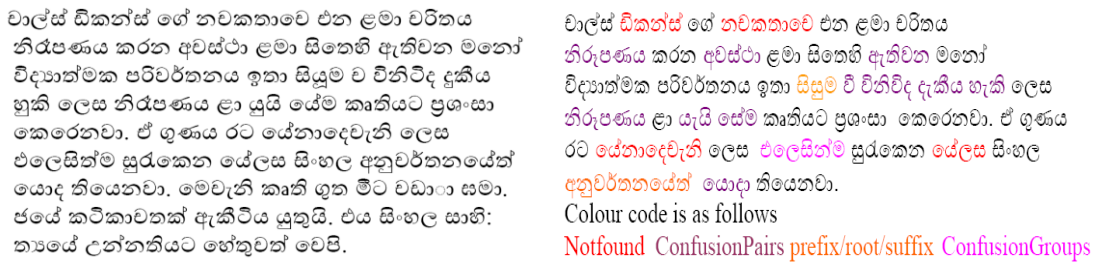


Figure 6.6: sample after 1st pass and final output

Summary of the system is shown in Table 6.7 and Table 6.8.

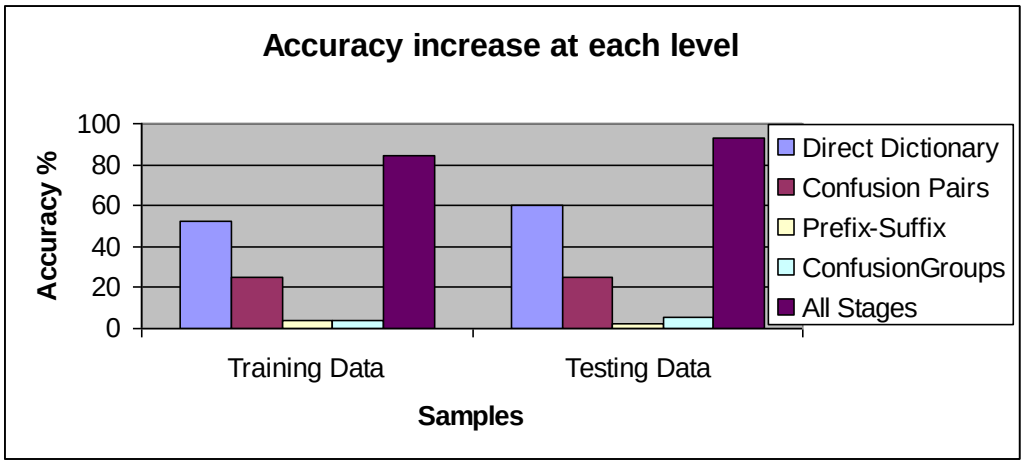
Table 6.7: Error Detection & Correction for Training Data

For Training Data	Sinhala OCRed Text	With Confusion Pairs	With Prefix suffix	With Confusion Groups	All three stages together
# of words input	2689	1283	623	523	2689
# of hits (stage)	1406	660	100	108	868
# of hits (total)	1406	2066	2166	2274	2274
# of none-hits invalid	1283	623	523	415	415
Error Detection % to total errors	47.71	47.71	23.17	19.4	47.71
Error Correction % on the detected	-	51.44	16.05	20.65	84.56
# of False Positives	36	21	25	12	94
# of False Negatives	30	x	x	x	30
Real error correction	1370	639	75	96	2180
Time taken		-	-	-	00:16:10
Accuracy at the end%	52.2	76.83	80.55	84.57	84.57
Real accuracy %	50.9	74.7	77.5	81.07	81.07

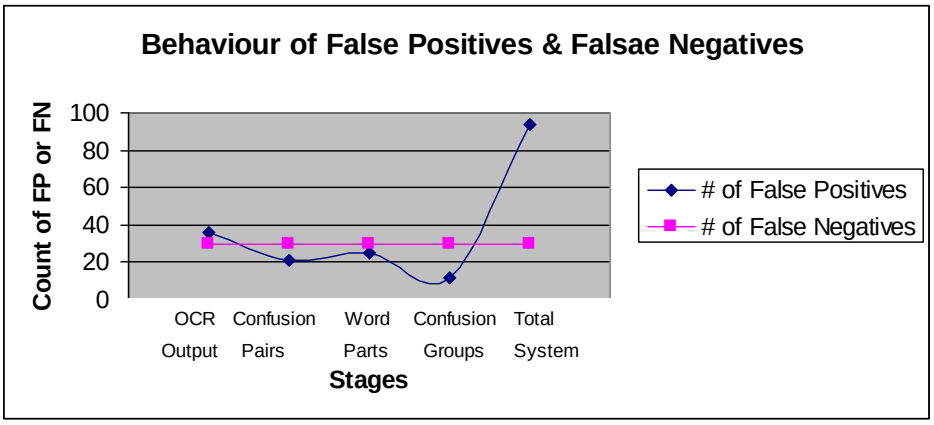
Table 6.8: Error Detection and Correction for Testing Data

For Testing Data	Sinhala OCRed Text	With Confusion Pairs	With Prefix suffix	With Confusion Groups	All three stages together
# of words	30240	12148	4560	3868	30240
# of hits	18092	7588	692	1631	27311
# of hits (total)	18092	25680	26372	28003	28003
# of none-hits	12148	4560	3868	2237	2237
Error Detection % at each level	40.172	40.17196	15.08	12.79	40.172
Error Correction % on the detected %	-	62.46	15.17	42.16	81.59
# of False Positives	435*	200*	306*	35*	976*
# of False Negatives	1164	x	x	x	1164
Real error correction	17719	7411	386	1596	27112
Time taken		-	-	-	03:43:39*
Accuracy %	59.8	84.3	87.2	92.6	92.6
Real accuracy %	58.59	84.25	86.19	89.65	89.65

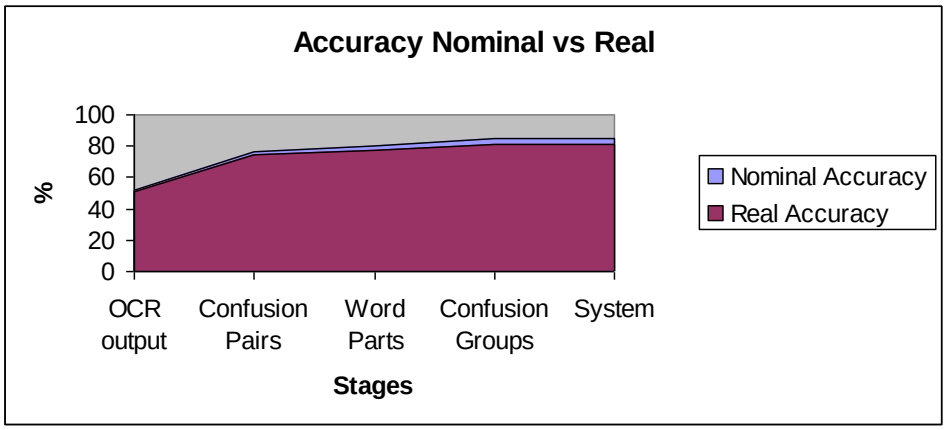
* Extrapolated value by taking 10% of the sample



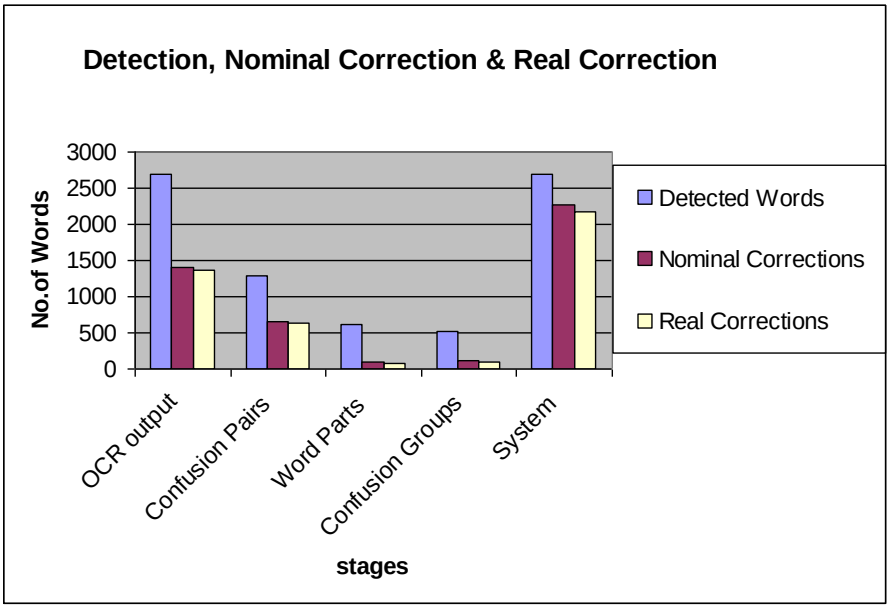
Graph 6.1: Accuracy increase at each Stage



Graph 6.2: False positives and false negatives for training data at each stage



Graph 6.3: Nominal accuracy vs. real accuracy for training data



Graph 6.4: Detected Errors and Corrections Training Samples

It can be seen that the exhaustive search space is reduced to greater extent after implementing few grammar rules and it is depicted in Figure 6.7

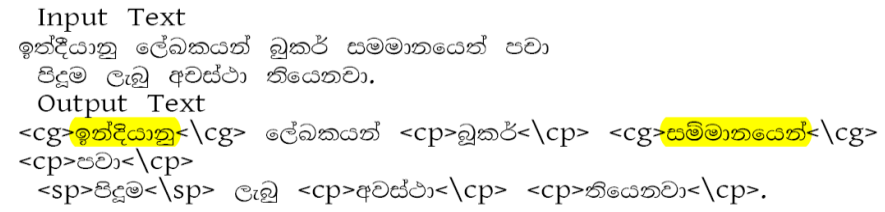


Figure 6.7 a: Text Containing Multiple Errors

Figure 6.7 b: Part of Exhaustive Search without Grammar Rules

Role in Grammar and Syntactical rules in an exhaustive search is superb. It reduces the set of word net by implementing few syntactical rules from 20790 to 14400 as in Appendix K and it reduces run time in proportional to the word count. Hence, it is need not say, that the importance of rules on exhaustive search.

We could observe that the correct words generated in words hypothesis net, but they were not validated since those words are not in the lexicon as in Figure 6.8. This is true in each stage, but it would not appear on the final output.

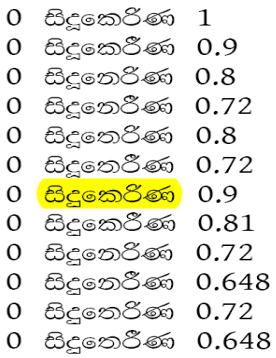


Figure 6.8: Unrecognized False Positives in Exhaustive Search

7. CONCLUSION

Dictionary use in this research was three-fold; firstly in error detection, to mark the words recognized as correct, secondly in error correction, to propose a probable word, and thirdly as an indirect method to analyse linguistic features such as prefixes, stems and suffixes.

The total errors found on the OCR output for training data was 1283 making the error rate 47.8% and for testing data it was 12148 making the error 40.2%.

The testing data provides us a better result than the training. Two reasons may cause that result; one is the improvement in OCR Engine with time so that character accuracy had been improved and secondly, the testing image files were based on direct outputs of word processing soft copies which may have reduced much noise.

With the result of the system we observed false negatives, which are caused by two reasons; recognition errors in the word and real word is missing in the dictionary. Definitely , errors have to be corrected, hence, the missing words, even though smaller in number are to be added into the dictionary dynamically, especially the words which are uncommon in general text or specific names. To reduce the errors in the recognized words, techniques are to be sought, in this research.

The difficult task was reducing false positives. Although cleansing the dictionary data lessens the problems, only that will not be adequate because false positives occur not only because of the lack of the word in the dictionary, but also the word that makes the hit is not the original word. Removing obsolete words which make false positives may reduce that and dynamically updated statistics for the words adds a value in reducing the same. But, this will not end the problem.

There were 36 numbers of false positives found without any error correction mechanism, and 94 numbers of false positives found with error correction mechanisms. It indicated that even though any error is not detected for a word, it may

not be accurate. So even in this situation, alternative candidates have to be considered for the best match, especially for single component words, which tends to make false positives at most times. In addition error correction techniques too contribute to increase false positives by accepting non-real words, especially, the single error correction technique at stage 1. False positives introduce at stage 2, prefix –suffix combinations, can be reduced by using grouping concept. Stage 3 introduced very few false positives because of the exhaustive search. The ideal situation is having the words passed though a context sensitive filtering process to get the best match. Word bi-grams will also be helpful in this regard.

The confusion Vector Pair List method corrects only single errors. The method alone increased the word accuracy to 76.83% for training data and to 84.9% for testing data by taking the system accuracy of recognition to 74.93 % and 84.92 % respectively. When compared to the accuracy at the OCR output, this is a tremendous step. Adding credit to that, it consumes limited resources. Since, the majority of errors belong to the set of single errors; this technique would suit as the primary solution.

Saturating the dictionary solves false negatives. However it is not practical. Therefore, other techniques have to be used to reduce false positives in number. For that, a method is proposed by this research in stage 2 by considering word parts separately as prefix, stem and suffix word parts. There were 623 words in the training data marked as invalid, in stage 2 and 16.05% of words from that could be corrected whereas that for testing data was 15.17%. Hence, the accuracy of training and testing samples increased up to 80.55% and 87.2%.

This method was used in two fold, gaining computational economy by reducing the words left for exhaustive search at stage 3 and some words falls into false negative category can be validated by breaking them into parts. But, at the same time, it increases the number of false negatives because of the forced matches, especially for long words. By introducing the stem word groups, forced matches too can be avoided. By having the groups of stem words for suffix or prefix, the number of false positives had been dropped by about 10% in the training data.

In addition to that, the research applies confusion character correction for those words left for stage 2. This will result in reducing false negatives further making it a viable solution for correcting multiple errors with limited resources.

The final step, Confusion Groups was an exhaustive solution consuming much resource making it uneconomical. However, this method is capable of correcting multiple errors with 6 confusion characters for our set penalty level. But it alone increases the word accuracy by 3.57% for training data and 5.4% for testing data, whereas the accuracy was 20.65% and 42.16% for words left for stage 3, which is a considerable improvement. Hence, the final output increase to 84.57% and 92.6% for the samples of training data and testing data respectively.

With all three stages together, in this proposed system the accuracy increases from 52.2% to 84.57% for training data and from 59.8% to 92.6% for testing data. But the real accuracy, that is without false positives, for training data increased from 50.9% to 81.07%.

A few syntactical rules were embedded into the code for filtering out ZWJ and in identifying the real vowel characters with inverted glyphs like ◌a for ◌ා. Even the vowel ඓ can be recognized as ඓ.

Syntactical rules can be applied for that too, but in our samples none was found. A few syntactical rules implemented on exhaustive search drops the word hypothesis net generated in stage 3, by half and it is a massive saving in computational resources. The more the rules implemented, the more the benefits returns.
Even at other stages grammar rules and syntactical rules will contribute to reduce the number of searching. However, the numbers of matching words found are few at those stages, there will not be a significance difference.

None of the aforesaid approaches solved inserted or deleted errors. To work with those errors edit distant methods or statistical n-gram methods have to be employed.

Eventually, it can be said that, the goals of this research have been achieved to a satisfactory level, even though it needs bit more work to use the system for a commercial level. The fact disclosed from this research was the words validated by lexicon at glance, may not be the original words and there may a difference, between the nominal accuracy and real accuracy.

7.1 Further Work

This strategy can be combined with the OCR engine itself by doing slight modifications at code level. The advantage of this is the ranking errors of characters at the recognition level can be considered for confusion characters and that may reduce false positives as well as limit expensive exhaustive search.

The system uses confusion groups at the final attempt. The longer the word left for this technique, the larger the word hypothesis. This makes the system uneconomical in terms of computing resources and time. If this system is coupled to the OCR engine, ranking character errors can be considered instead of the confusion groups to save computational time and resources.

To couple this system to the OCR engine, all the lists of words referencing in the code such as dictionary, stem lists, prefix and suffix lists, confusion vector list and confusion group list, should be written in the order of the glyphs appearing in the words. Filtering out of the unwanted characters such as ZWJ can also be implemented to the same. The filtering process can be extended to identify the misrecognized characters, such as 6 for ◌ో, 5 for ◌ొand @ for ◌ొ and make the necessary substitutions.

As we did not see a significant difference on confusion characters within the group, we limited to use the confusion groups. Therefore, Confusion Matrix method was not implemented. In a matrix, different similarity measures can be given to each confusion character in the group. With extensive research, the confusion groups can be extended to form confusion matrixes consisting 2 dimensions, in which each individual character replacement has an individual confusion measure. By extending the matrix 3 dimensions, each confusion character with different vowel forms can be considered separately with individual confusion measures. Statistics of the errors and probability of character misrecognition will be more meaningful in valuating the similarity measures.

In prefix, stem, and suffix word formation methods, we do not consider suffix and prefix groups together to avoid the complexity of the code. But this also can be implemented by taking the union of the two groups.

In group lists of suffixes, a single component entry is not allowed. Eg: - ද-වත්න. But in the situations, where the default list is searched, it should not be allowed to accept single component stems, as it may cause false positives. However, there are many single character words. Hence in dictionary search for the whole word, single character search should be allowed.

In OCR recognizing process, at least a part of the word should be correct and that can be used for filtering out unwanted candidate words. In this case n-grams of the recognized text words may be useful. Dictionary words can be reference by tagging them by n-grams on the word. For any n-grams identified on OCRed text can be tested on dictionary words, to find out the words with those n-grams occur at the same position in same sized words [19]. Starting from the longest n-gram and considering the starting and ending spaces to the word would result a reliable output. By adding size tolerance and position tolerance for the position of n-gram appearing on the dictionary word, inserted and deleted type errors can also be treated in this.

To find the best matching candidate word, the context of the same document can also be used as another source of information. This will be helpful in reducing false positives to a great extent as the possibility of appearing the same word in the same document is much higher than to the general frequency of the word. This can be done by maintaining a dynamic list of words with the word count and this area is left untouched in our research.

We implemented few syntactical rules to reduce the set of word hypothesis net in exhaustive search in stage 3 in our research. Those rules are general rules for word formation and can be applied at any place, where a word is formed. It played a great role in reduction in word hypothesis net. Not only it saves computational time but

also the resources consuming. The research can be extended to implement those rules in other 2 stages as well. There are many syntactical rules as stated in chapter2. They can also be implemented on the system to improve this research further. In addition there are many grammar rules combined with words for their formation. These rules can be used with root or stem word list to ensure the recognized words are correct [27].

In our case, similarity measure is assigned for each pair in proportional to errors found in the samples and the similarity between the two, matching character and the proposed character. Each member in a confusion vector pair or a confusion group has a different probability of occurrence. Hence, it should be given a different measure for each pair/ group according to the statistics obtained for the component frequency and probable error count to get the best result.

Even in the stage 2 of this system, probable values can be assigned to each suffix or prefix in the lists to increase the confidence level individually and not to fall into the category of false positives. In our system, we assigned equal probabilities for every suffix and prefix, so the system can be improved to that extent by assigning individual probabilities to individual suffix or prefix to improve the real accuracy.

A method for the automatic correction of OCR errors would be clearly beneficial [29]. Whatever robust the OCR is, unless it reduces real word errors (false positives) as well as non-word errors automation is helpless. In addition, false positives will become a key problem detected and emphasized at the end. Statistical methods of word bigrams have to be applied to reduce those. One such method would be conditional probability of word bi-grams [20]. Otherwise Sinhala text store on web can also be used as statistical information.

One of the practical methods dealing with inserted and deleted character errors is edit distance method. That will work with transitions errors in typing as well as character misrecognitions. That is one of the areas left to touch in improving this research.

The requirement for an OCR system for Sinhala script needs more attention in reproducing the documents of the National Archives, Sri Lanka, and old books. Scanning images from that material would lead to destroy the remaining leaves. In addition it will be difficult to use with available columnar and pictorial data. OCR systems may lead to poor and insignificant results if their input source is physically out of condition, of old age, having low printing quality, bad physical condition, poor printing quality and containing imperfections and distortions such as rips, stains, blots, and discolorations sources such as old books, poor-quality [41]. In addition repeated photocopies and faxes can still be difficult to process and may cause many OCR errors [29]. Hence the accuracy must be improved to utmost level employing all the improvements mentioned above on the system we implemented incorporating to a robust OCR engine to save the contents of the valuable and invaluable documents as well as its life.

REFERENCES

[1] Thien M Ha , H Bunke, “Image Processing Methods For Document Image Analysis,” in *Handbook Of Character Recognition And Document Image Analysis*, World Scientific Publishing Company, Singapore, May 1997,ch 1, pp 1-47.

[2] Mohamed Cheriet, Nawwaf Kharma, Cheng-Lin Liu, Hing Y. Suen, *Character Recognition Systems - A Guide For Students And Practitioners*, A John Wiley and Sons, New Jersey, 2007.

[3] A. Dengel, R. Hoch, F. Hones, T. Jager, M. Malburg, and A. Weigel. “Techniques for Improving OCR Results,” in *Handbook of Character Recognition and Document Image Analysis*, World Scientific Publishing Company, Singapore , May 1997, ch 4, pp 227–258.

[4] T.K. Ho, J.J. Hull, and S.N. Srihari, "Word Recognition with Multi-Level Contextual Knowledge,” in *Document Analysis and Recognition Int. Conf.*, Saint Malo, France, 1991.

[5] Tin Kam Ho, Jonathan J. Hull and Sargur N. Srihari, “A Word Shape Analysis Approach to Lexicon Based Word Recognition,” *A Pattern Recognition Letters,* vol. 13, pp 821-826, Nov. 1992.

[6] Unicode Consortium, (May 2010) Sinhala Unicode Character Code Set, Available: http://unicode.org/charts/PDF/U0D80.pdf

[7] A. Lawrence Spitz, "Shape-Based Word Recognition," *Document Analysis and Recognition*, Vol. 1, pp178-190, May. 1999.

[8] C. Weliwitage, A. Harvey, A. Jennings, (January 2010), "Whole of Word Recognition Methods for Cursive Script," Available: http://www.aprs.org.au/wdic2003/CDROM/111.pdf.

[9] Christian M. Strohmaier, Christoph Ringlstetter, Stoyan Mihov, “Lexical Postcorrection of OCR-Results: The Web as a Dynamic Secondary Dictionary?” *Document Analysis and Recognition Annu. Int. Conf.,* 1993

[10] Thomas A. Lasko, Susan E. Hauser, “Approximate String Matching Algorithms for Limited Vocabulary OCR Output Correction,” in *Document Recognition and Retrieval VIII,* San Jose, CA , 2001

[11] Jason J. S. Chang, Shun-Der Chen, “The Post processing Of Optical Character Recognition Based On Statistical Noisy Channel And Language Model” , in *Language, Information and Computation Pacific Asia Conf. ,* 1995

[12] F. Lebourgeois, J.L. Henry, H. Emptoz, “An OCR System for Printed Documents,” *IAPR Workshop on Machine Vision Applications*, Tokyo, Japan, Dec. 1992

[13] Tao Hong and Jonathan Hull, “Visual Inter-Word Relations and Their Use in OCR Post Processing,” in *Document Analysis and Recognition Int. Conf.,* 1995.

[14]H.L. Premaratne, J. Bigun, “A Segmentation-Free Approach To Recognise Printed Sinhala Script Using Linear Symmetry,” *Pattern Recognition*, Vol. 37, pp. 2081-2089, Oct. 2004.

[15] Lalith Premaratne Yaregal Assabie Josef Bigun, “Recognition of Modification-Based Scripts Using Direction Tensors,” in *4th Computer Vision, Graphics and Image Processing*, India, 2004.

[16]Ruvan Weerasinghe, Asanka Wasala, Kumudu Gamage, “A Rule Based Syllabification Algorithm for Sinhala,” in *2nd Int. Conf. Natural Language Processing*, Jeju Island, Korea, 2005.

[17] Sukalpa Chanda, Srikanta Pal and Umapada Pal, “Word-Wise Sinhala Tamil and English Script Identification Using Gaussian Kernel SVM,” in *Int. Conf. Pattern Recognition*, Kolkata, India, 2008.

[18] Veena Bansal And R. M. K. Sinha, "Integrating Knowledge Sources In Devanagari Text Recognition System," PhD thesis, Indian Institute of Technology, Kanpur, India, Mar. 1999.

[19] V. Bansal, R. M. K. Sinha, "Partitioning and Searching Dictionary for Correction of Optically Read Devanagari Character Strings," *Document Analysis and Recognition*, vol. 4, pp 269-280, Jul. 2002.

[20] Lalith Premaratne, E Jarpe, Josef Bigun, "Lexicon and Hidden Markov Model based Optimisation of the recognizes Sinhala Script," *Pattern Recognition* Letters, vol. 27, pp 696-705, Apr. 2006

[21] Xiaofan Lin, "DRR Research beyond COTS OCR Software: A Survey," in SPIE Conf. *Document Recognition and Retrieval XII*, San Joes, CA, Jan**.** 2005

[22] S. Hewavitharana, H. C. Fernando, N.D. Kodikara, "Off-line Sinhala Handwriting Recognition using Hidden Markov Models," in *Proc. Indian Conf. Computer Vision , Graphics & Image Processing*, Ahmedabad, India, 2002.

[23] Ajay S Bhaskarabhatla, Sriganesh Madhvanath, "Experiences in Collection of Handwriting Data for Online Handwriting Recognition in Indic Scripts," HP Laboratories, India, 2005

[24] Dulip Herath, Nishantha Medagoda, "Research Report on the Preprocessing Engine of the Optical Character Recognition System for Sinhala Scripts," Language Technology Research Laboratory,Univ. Colombo, Sri Lanka.

[25] Karthika Mohan, C. V. Jawahar, "A Post-Processing Scheme for Malayalam using Statistical Sub-character Language Models," in *Proc. 9th IAPR Int. Workshop Document Analysis Systems*, New York, Pages 493-500, 2010.

[26] Kai Niklas, "Unsupervised Post-Correction of OCR Errors," Diploma thesis, Univ. Hannover, Jun. 2010.

[27] G S Lehal, Chandan Singh, “A post-processor for Gurmukhi OCR,” *Sadhana* , vol. 27, Part 1, February 2002, pp. 99–111.

[28] Takahashi, H.,; Itoh, N., Amano, T., Yamashita, A., “A spelling correction method and its application to an OCR system,” *Pattern Recognition*, vol. 23, p. 363-377, 1990

[29] Xiang Tong and David A. Evans, “Statistical Approach to Automatic OCR Error Correction in Context.”

[30] B. B. Chaudhuri**,** U. Pal**,** “OCR Error Detection and Correction of an Inflectional Indian Language Script**,”** in *Proc. Int. Conf. Pattern Recognition*, vol. III, p 245, 1996,

[31] *Sinhala Character Code for Information Interchange Sinhala Unicode Standard*, SLS 1134:2004, 2004.

[32] ජාතික අධ්‍යාපන ආයතනය, *සිංහල ලේඛන රීතිය*, තෙවන මුද්‍රණය, ජාතික අධ්‍යාපන ආයතනය, කොළඹ, ශ්‍රී ලංකා, 2001.

[33] *“*Optical_character_recognition*“* (2012, Oct 10), Available : http://en.wikipedia.org/wiki/Optical_character_recognition

[34] ජේ. බී. පීරිස්, *සිංහල අක්ෂර විචාරය*, ප්‍රථම මුද්‍රණය, සුමිත ප්‍රකාශන, ශ්‍රී ලංකා, 2006.

[35] ජේ. බී. පීරිස්, *නූතන සිංහල ලේඛන වියාකරණය*, ප්‍රථම මුද්‍රණය, සීමා සහිත ලේක් හවුස් ඉන්වෙස්ට්මන්ට්ස් සමාගම, කොළඹ, ශ්‍රී ලංකා, 1990

[36] University of Colombo School of Computing Sri Lanka, PAN Localization Project of Language Technology Research Laboratory, Optical Character Recognition System for Sinhala

[37] University of Colombo,(2012 Jan.), “UCSC/LTRL Sinhala Corpus Beta Version,” available: http://www.ucsc.cmb.ac.lk/ltrl/downloader.php?resource=crpsb

[38]H. Bunke, “A Fast algorithm for finding the nearest neighbor of a word in a dictionary,” in *Proc. 2nd Int. Conf. .Document Analysis and Recognition*, Nov. 1993

[39]Riseman, E.M., Hanson, A.R., “A Contextual Postprocessing System for Error Correction Using Binary n-Grams,” *Computers, IEEE Transactions*, vol. C-23, p 480 - 493 , May 1974

[40]B.B Chaudhuria, U Pala, “Complete printed Bangla OCR system,” *Pattern Recognition*, vol. 31, Pages 531–549, Mar. 1998.

[41] Youssef Bassil, Mohammad Alwani, “OCR Post-Processing Error Correction Algorithm Using Google's Online Spelling Suggestion,” *Emerging Trends in Computing and Information Sciences*, ISSN 2079-8407, Vol. 3, Jan. 2012.

[42] Dharam Veer Sharma, Gurpreet Singh Lehal, Sarita Mehta, “Shape Encoded Post Processing of Gurmukhi OCR,” in *10th Int. Conf. Document Analysis and Recognition*, 2009

[43] Tao Hong and Jonathan J Hull, “Improving OCR Performance with Word Image Equivalence,” in *4th Symp. Document Analysis and Information Retrieval*, Las Vegas, NV, pp. 177-190, Apr. 1995.

[44] Kenneth Ward Church, Patrick Hanks, “Word Association Norms, Mutual Information, and Lexicography,” *Computational Linguistics*, vol. 16, Mar. 1990

[45] Wagner, Fisher, “The String to String Correlation,” 1974

[46] (2012 Sep.) C library routines for Trie Data Structure implementation available: http://tommyds.sourceforge.net/doc/tommytrie_8h.html

[47] Harsha Wijewardhane, Personal Communication, Nov. 2012

[48] Ray Smith, An Overview of the Tesseract OCR Engine, Proc. in 9*th Int. Conf. Document Analysis and Recognition*, IEEE Computer Society (2007), pp. 629-633, 2007

[49] Tesseract OCR, Available: http//code/google.com.p/tesseract-ocr accessed on December, 2012-12-29

[50] University of Colombo, “Sinhala corpus of 10 million words,” (2012 Apr.) Available: http://www.ucsc.cmb.ac.lk/ltrl

Appendix B: Summary of Errors found on Training Samples

Recognized & Actual Characters	Total Error Count
ත න	202
ච ව	165
ී ි	79
ූ ු	60
ක ත	43
යේ ෙ	38
රැ රෑ	32
ි ී	30
ටි ව	25
කී ති	22
ක න	17
ත ක	17
චූ වූ	16
ඩ ව	14
න ත	14
තු තෑ	13
හ හ	13
ද දා	12
ප ෂ	12
මි මී	11
ෑු ඈ	10
ඩ ධ	10
ද ද්‍ර	10
හු හෑ	10
ද දු	8
මී මි	8
ැ ෑ	8
ශ්‍රී ශ්‍රී	7
ම මී	7
ය ඝ	7
ඹ ඹ	6
කූ තූ	6
ට ව	6
සූ සු	6
හ භ	6
එ ඵ	5
එ ළු	5
න ා	5
පු පෑ	5
ය ඝ	5
ර ඊ	5
ගු ගෑ	4
ග ෙ	4
ඝ ඝ	4
චූ ඩූ	4
ඩ ච	4
ද දෑ	4
දූ දෑ	4
නූ නෑ	4
භ ඟ	4
ම ඹ	4
ව් වි	4
ු ූ	4
2 /	3
ඔ ඕ	3
ඛ ඛ	3
ග භ	3
ච ව්	3
ට ව්	3
ත්‍රී ශ්‍රී	3
ද දෑ	3
පූ ප්‍ර	3
ජ් ඒ	3
ඛ ඛ	3
භ ග	3
ඹ ම	3
ය ඝ	3
ලූ ලෑ	3
ශ ග	3
ෂ ෙ	3
සූ සෑ	3
පa සa	2
ඔ ම	2
ඔ ඕ	2
ග ශ	2
චූ වෑ	2
ජී ඒ	2
ට ච	2
ඩ ග	2
ඩි වී	2
කූ කූ	2
තෑ නෑ	2
දෘ දෑ	2
ද්‍රා ලා	2
ෂි වි	2
පූ ෂූ	2
ජ් ෂ	2
භ භ	2	ම	2
ම මි	2
ඹ ඹ	2
ය ෂ	4
යූ යෑ	2
යූ සූ	2
රැ රෑ	2
ව ච	2
වි චි	2
වි වී	2
ශ ඝ	2
ඝ ප	2
ඝ ය	2
ා ෙ	2
0 6	1
0 ං	1
1 ාේ	1
5 ාේ	1
ංද්‍රඃ ඝ	1
ට ළ	1
එ ඒ	1
එ ඪ	1
එ ඵ	1
ක භ	1

කෙූ	නෑ	1
ක්‍රි	ත්‍රි	1
ක්‍රි	ශ්‍රී	1
ක්‍රී	ත	1
ග	ඟ	1
ග	ව	1
ග	ස	1
ග	ෆ	1
ගෑ	ග	1
ගෑ	ෙ	1
ගි	හී	1
ගු	ඉ	1
ග්‍ර	ඉ	1
ඟි	හී	1
ච	ඩ	1
ච	වී	1
චි	ච්	1
ච්	වි	1
ජ	ඡ	1
ජ	ජ්	1
ඤ	ඤා	1
ට	ම	1
ටි	ට	1
ටි	ඊ	1
ටී	ව්	1
ඨී	ධි	1
ණු	ණෑ	1
ණ්	ණි	1
ඬ	ඩ	1
ති	තී	1
ති	නි	1
තී	නි	1

තී	නී	1
තු	නෑ	1
ත්‍ර	නු	1
ත්‍රි	ශ්‍රී	1
ත්‍රී	ශ්‍රී	1
ද”	ද්	1
ද	ඳ	1
දෘ	දී	1
දී	)	1
දෘා	දා	1
දූ	දා	1
දූ	දු	1
ඳ	)	1
දූ	දෑ	1
ප	වි	1
ප	ස	1
ප්‍ර	ටු	1
ප්‍ර	ත්‍ර	1
ප්‍ර	ද	1
ප්‍ර	ෂ්‍ර	1
ප්‍රී	ශ්‍රී	1
ළු	ළු	1
බ්	බ්	1
බ්	බ්	1
බ්	බ	1
බ්	බී	1
භූ	හෑ	1
ම	ඹ	1
මී	ම්	1
ය	ස	1
යි	ට	1
ර	ඝී	1

රැ	රෑ	1
ර්	ර	1
ලූ	කි	1
ලෙූ	දෙ	1
ව	ච	1
ව	ට	1
වී	ච්	1
වී	ව්	1
වූ	ධූ	1
වූ	ව	1
වූ	වෑ	1
ව්	වී	1
ශි	වි	1
ෂූ	ත්‍රූ	1
ෂ්	ඒ	1
ස”	ස්	1
ස	න	1
ස	හ	1
හ	ඟ	1
හ	න	1
හ	ස	1
හ්	න්	1
ළ	නූ	1
ළූ	දූ	1
ළූ	මූ	1
ා	ග	1
ැැ	ෑ	1
ෙ	ම	1
@ම	ම	
6	ර	

Summary of Errors

Appendix C: Recognized-Confusion pair with Similarity Measure

ත	න	.7
ව	ව	.8
ී	ී	.9
ූ	ු	.9
ක	ත	.7
ේ	ෙ	.5
රූ	රු	.8
ී	ී	.9
ට	ව	.5
කී	ති	.5
ක	න	.7
ත	ක	.7
වූ	වු	.6
ඩ	ව	.7
න	ත	.7
තූ	තෑ	.5
හ	හ	.7
ද	දා	.7
ප	ෂ	.7
ම්	ම	.8
අූ	ඇ	.5
ඩ	ධ	.6
ද	ද්ර	.6
හූ	හෑ	.5
ද	දු	.5
ම්	ම්	.8
ැ	ෑ	.9
ප්රී	ග්රී	.5
ම	ම්	.5
ය	ස	.7
ඔ	ඔ	.8
කූ	තූ	.6
ට	ව	.7
සූ	සු	.8
හ	හ	.8
ළු	ළූ	.7
ළු	ළු	.8
න	ා	.5
පු	පෑ	.5
ය	ස	.7
ර	ඊ	.7
ගූ	ගෑ	.5
ග	ෙ	.5
ඝ	ඝ	.7
වූ	ඩු	.5
ඩ	ව	.7
ද	දෑ	.5
දූ	දෑ	.5
නූ	නෑ	.5
හ	ඟ	.7
ම	ඹ	.7
ව්	වි	.7
ූ	ු	.9
2	/	.5
ඔ	ඕ	.6
ඛ	බ	.8
ග	හ	.7
ච	ව්	.5
ට	ට්	.5
ත්රී	ග්රී	.6
ද්ර	දෑ	.5
පූ	ප්ර	.6
ඡ	ඒ	.6
බ	ඛ	.8
හ	ග	.7
ඹ	ම	.7
ය	ඝ	.7
ලූ	ලෑ	.5
ශ	ග	.7
ෂ	ෙ	.5
සූ	සෑ	.5
සa	සa	.7
ඹ	ම	.7
ඔ	ම	.7
ග	ශ	.7
චූ	වෑ	.5
ඡී	ඒ	.6
ට	ච	.7
ඩ	ග	.5
ඩි	වී	.5
ත්වූ	ත්වූ	.8
තෑ	නෑ	.7
ද්යා	දෑ	.5
ද්රා	ලා	.5
පි	වි	.5
පූ	ෂූ	.5
ජ්	ෂ	.5
හ	හ	.7
ම	ම්	.6
ම්	ඕ	.6
ය	ෂ	.5
යූ	යෑ	.5
සූ	සු	.5
රූ	රු	.7
ව	ව	.7
වි	වි	.5
වී	ව්	.6
ශ	ස	.5
ස	ප	.7
ස	ය	.7
ා	ෙ	.5
0	6	.5
0	ං	.7
ළ	ල	.5
ළු	ළූ	.6
ළු	ඩ	.7
ළු	ළු	.8
ක	හ	.6
ක්රි	ත්රි	.7
ක්රි	ශ්රී	.5
ක්රී	ත	.3
ග	ඟ	.6
ග	ස	.5
ග	ෆ	.7
ගෑ	ග	.5
ගී	ඟී	.5
ගූ	ඉ	.5
ග්ර	ඉ	.4
ඟි	ඟී	.7
ච	ඩ	.7
ච	වී	.6
චි	ච්	.7
ච්	වි	.5
ජ	ඡ	.7
ජ	ජ්	.7
ඥ	ඥා	.7
ට	ම	.7
ටි	ට	.8
ටි	ඊ	.5
ටී	ව්	.5
යී	ධි	.6
ණූ	ණෑ	.5
ණ්	ණි	.5
ඬ	ඩ	.7
ති	නි	.7
තී	නි	.6
තී	නී	.6
තූ	නෑ	.5
ත්ර	නූ	.5
ත්රි	ශ්රී	.5
ත්රී	ශ්රී	.5

ද”	ද්	.6
ද	ඳ	.7
දි	දී	.8
දී	)	.5
ද්යා	දා	.5
ද්ර	දා	.7
ද්ර	දු	.7
දී	)	.5
දූ	දෑ	.5
ප	ව්	.5
ප	ස	.6
ජ්රි	ටු	.5
ජ්රි	ට්රි	.5
ජ්රි	ද	.4
ජ්රි	ලි	.4
ජ්ර	ශ්ර	.5
ළු	ළු	.7
බ්	බ්	.7
බී	බ්	.7
බ්	බ	.6
බ්	බී	.7
භූ	භෑ	.5
ම	ඹ	.5
මී	ම	.6
ය	ස	.7
ය්	ට	.5
ර	ජී	.5
රෘ	රෘ	.8
ර්	ර	.6
ලූ	කී	.4
ව	ච	.7
ව	ට	.6
වී	ච්	.5
වී	ච්	.6
වූ	ධූ	.5
වූ	ව	.5
වූ	වෑ	.5
ව්	වී	.6
ශි	ව්	.5
ෂූ	ත්රි	.5
ෂ්	ඒ	.5
ස”	ස්	.7
ස	න	.6
ස	හ	.6
හ	ඟ	.7
හ	න	.7
හ	ස	.6
හ්	න්	.6
ළ	නු	.4
ළි	දි	.4
ළි	මු	.4
◌ා	ග	.3
◌ැ◌	◌ැ	.5
◌	ම	.5
@ම	ම	.8
6	ර	.8

Confusion.txt

Appendix D: Prefix List

අ	1
ඉ	0
අව	3
අති	2
බහු	0
ප්‍රති	0
නි	0
ස	0
සු	0
නො	0
වි	0
උප	4
අනු	0
පර	0
අප	0
අධි	0
දු	0
පිළි	0
නී	0
උ	0
ප	0
ස්ව	0

Appendix E: Suffix.List

කරුවන්ගේ	13	වයක්	0	වූ	1		
කරුවන්	13	වය	0	වූත්	1		
කරුවා	13	ව	0	වූයේ	1		
කරු	13	වලදී	3	වුණාය	1		
කරගන්න	2	වලද	3	වරයකු	4		
කර	2	වලට	3	වරයාගේ	4		
කරනු	2	වලදැයි	3	වරයාට	4		
කරන	2	වලින්	3	වරයකු	4		
කරමින්	2	වල	3	වරයාහට	4		
කිරීමෙන්	2	වට	11	වරයෙකුට	4		
කිරීමේ	2	වාට	0	වරයෙකු	4		
කිරීමට	2	වත්ව	0	වරයෙක්	4		
කිරීම	2	වතට	0	වරයාටත්	4		
කරයි	2	වත්	0	වරයාය	4		
කොට	2	වක්	11	වරයා	4		
කැර	2	වකට	11	වරියයි	4		
කාරී	2	වකදී	11	වරිය	4		
කළ	0	වක	11	වරුන්ට	4		
කළේ	0	වකි	11	වරුන්	4		
කට	0	වක්ද	11	වරු	4		
කම්වලින්	0	වක්	11	වසක	0		
කමක්	0	වකු	0	වෙකුට	0		
කමට	0	විය	1	වෙකු	0		
කමය	0	වී	1	වෙක්	0		
කම්	0	වේ	1	වෙනවා	1		
කම	0	වෙන්	0	වෙනව	1		
කයි	0	වෝ	0	වෙන	1		
කී	0	වැනි	0	වෙනුයි	0		
කින්	0	වන	1	වෙනි	0		
කදීම	0	වම	0	වෙනු	0		
කදී	0	වීමට	1	වෙන්	0		
ක්ද	12	වීමටය	1	වෙන්ම	0		
ක්	12	වීමකින්	1	වෙමින්	1		
ක	12	වීමකි	1	වෙමු	0		
කි	12	වීමද	1	වෙන්න	1		
කොට	0	වීමදී	1	වෙයි	1		
වනට	1	වීමේදී	1	වෙලා	1		
වන	1	වීමයි	1	වේ	1		
වන්ටත්	1	වීමත්	1	වේගන	1		
වනතුරු	1	වීම	1	වේද	1		
වන්ට	1	වීමෙන්	1	වේදී	0		
වන්නට	1	වුණ	1	වෙච්ච	0		
වන්න	1	වුණා	1	ගේ	0		
වන්නේ	1	වුණත්	1	ගන	10		
වන්ගෙන්	1	වුණි	1	ගන්න	10		
වන්	1	වුණේ	1	ගැසීම	0		
වනු	1	වුන්	0	ගැසීමේ	0		

ගෙන	10		යි	0		දෙමින්	0		
ගෙන්	0		යා	0		දෙන	0		
ගැන	0		යාට	0		දුන්න	0		
ගත්	10		යුතු	0		ධුර	0		
ගැනීම	10		යත්	0		ධුරය	0		
ගැනීමට		10	යෙකින්	6		ධුරයට	0		
ගැනීමේ		10	යෙකි	6		ත්	0		
ගැනීමදී	10		යෙකුට	0		තාව	0		
ගාන්න	0		යෙකු	0		තාවය	0		
ගත්ත	10		යෙක්	0		තුමා	0		
ගන්නා	10		යද	6		තුමාගේ	0		
ය	6		යදැයි	6		තුමාට	0		
යන්ගෙන්		6	යේයි	6		තුමාගෙන්		0	
යන්ගේ	6		යේය	6		තුරු	0		
යන්ට	6		යේ	6		පත්	0		
යන්ද	6		යේන්	0		ලන	0		
යන්හි	6		යේට	0		ලත්	9		
යන්	6		යේද	6		ලෙද	0		
යනුත්	6		යේම	6		ලෙස	0		
යන	0		යේයි	6		ලගෙ	9		
යට	6		යේදී	6		ලාට	9		
යෙනි	6		යේදීය	6		ලා	9		
යෙන්ම	6		ට	0		න්	0		
යෙන්	6		ටත්	0		නුත්	0		
යෙහිම	6		ටම	0		නායක	0		
යෙහි	6		ම	0		නගින	0		
යේ	6		මග	0		නැගෙන		0	
යෙනුත්	6		මලක	0		නකු	0		
යේත්	6		මගකද	0		නම්	0		
යට	6		මගකදී	0		හැක	0		
යටත්	6		මගදී	0		පති	0		
යටම	6		මක	0		පත්	0		
යම	6		මට	0		පත්වීම	0		
යාම	0		මින්	0		බව	0		
යව	0		ද	0		බවට	0		
යකි	6		දැයි	0		බැවින්	0		
යකින්	6		දාය	0		බැව්	0		
යක	6		දහසක්	7		භව	0		
යකට	6		දී	8		භාවය	0		
යකද	6		දීම	8		භාවයට	0		
යකය	6		දීමක	8		භාවයෙන්		0	
යනම	6		දීමක්	8		සිය	7		
යක්	6		දීමක්ද	8		සියකට	7		
යක්ද	6		දීමටත්	8		සියවසක		7	
යක්ම	6		දීමට	8		සීයක්	7		
යකු	0		දීය	0		හට	0		
යකුට	0		දෙන	0		හැකි	0		
යකැයි	0		දෙනෙකු		7	හි	0		
යයි	0		දෙනෙක්		7	ලු	0		

නම 7 නමක් 7

Suffix.txt

Appendix F: Confusion Groups

.9 ෆ ෆ
.9 ු ූ
.4 ය ස ප ඝ ෂ
.4 ට ම ව ච
.4 න ක ත
.7 රූ රැ රෑ රෑ
.5 ඒ ජ්
.6 ඔ ඕ ම ම් ඔ ඕ
.7 ම මි මී
.6 ග්‍ර ශ්‍ර
.4 භ ග ශ භ ඟ ෆ
.4 ඩ ධ ඬ ඨ
.7 උ ල ළ
.6 ර් ඊ
.7 ද දා ද්‍ර දු
.7 එ ළු ඵ ථ ඨ ඝ
.7 බ ඛ
.7 ඛ ඛි ඛී ඛ බි බී
.7 ඇ ඈ අු අූ
.7 ඡ ජ ඣ

Appendix G: More Frequent N-Grams

න	216356	යි	13268	යු	7470	රණ	4759	සඳ	3680
ෙ	194531	ද්	12462	මන	7400	දන	4756	යුතු	3633
ය	174410	යන්	12394	වු	7136	නම	4663	නක	3575
ක	165684	ව්	12261	ෙබ	7087	කිර	4563	රීම	3559
ර	162682	හි	12250	ටි	7010	කරන	4537	රට	3532
ත	154694	පා	12101	ඔ	6933	නැ	4490	ඩි	3520
ු	115426	දි	12009	උ	6932	නී	4486	ලන	3487
ද	97185	ප්ර	11841	පැ	6768	ාට	4475	ද්ශ	3476
ැ	93952	්ය	11761	ෙර	6518	ත්ත	4414	ෙද්ශ	3473
න්	87858	ඉ	11747	රය	6442	කළ	4401	පස	3463
හ	73448	පි	11711	අත	6171	ජන	4396	පස	3463
ට	72976	ලි	11671	ඒ	6143	ෙගන	4377	කැ	3463
අ	66557	මී	11645	වය	6097	වට	4367	කැ	3463
ග	62583	තා	11591	මහ	6065	පත්	4318	ැකි	3457
ත්	47679	ඳ	11291	රම	6053	රිය	4275	ධි	3397
බ	43477	දු	10877	පු	6005	ඩා	4251	මම	3308
ක්	43237	දී	10581	ෙත	5931	ට	4218	අතර	3289
ණ	30133	ාල	10458	ෙයන්	5862	ැර	4198	ංග	3270
ම්	29115	වැ	10311	ගැ	5775	යුත	4198	සඳහ	3256
නි	28794	ග්	10253	සිට	5666	ලය	4183	ස්ව	3186
වි	28464	රා	10051	කට	5650	මය	4166	ාහ	3180
ස්	25213	අැත	9982	වස	5624	මැ	4149	බන	3176
සි	22981	ෙක	9975	වත්	5596	ාන්	4143	පන	3174
ෙම	22110	මු	9859	ෙනා	5540	රත	4137	දර	3161
ඩ	20157	ෙම්	9591	පර	5453	ලක	4128	නට	3157
ශ	19583	සා	9513	ෙකා	5406	මක	4127	හැක	3145
නි	19359	අා	9398	න්ද	5372	අැති	4124	සැ	3125
එ	18886	සු	9260	ක්ෂ	5366	විය	4087	කිරීම	3120
කා	18763	යක්	9215	වූ	5343	ගැන	4050	අෙ	3109
වන	18728	ෙස	9122	ත්ව	5326	කාර	4041	කම	3072
ර්	18463	වර	9098	ෙස්	5300	මින	3955	ගන්	3063
කි	18298	නය	9003	අව	5272	අප	3955	පිළ	3050
යා	17666	වී	8915	න්ත	5255	කිරී	3926	පිළි	3033
ෙග	17095	ෙල	8855	ෙන්	5214	න්ෙන්	3913	ෙත්	3021
ල්	16911	ෙව්	8753	හු	5213	ණු	3892	ාෙව්	3014
නා	16134	කු	8617	නව	5205	ෙපා	3886	යම	2980
කර	15613	යට	8572	ගා	5197	ැද	3835	රුව	2964
අැ	15356	වල	8228	ළි	5171	වීම	3817	ෙලස	2953
ං	15136	ැති	8069	රක	5157	ලස	3807	ංක	2949
ජ්	14802	ෙග්	8041	මන්	5111	ල්ල	3798	ත්ර	2945
ය්	14686	මට	7903	ගි	5110	කිය	3769	මින්	2882
ෙය්	14636	න්ෙ	7903	බා	4995	ැල	3758	වක්	2877
ෙන	14528	වන්	7823	රෙ	4921	පාල	3757	විස	2873
වා	14516	හැ	7642	දැ	4881	සිය	3726	විෙ	2865
හා	14164	න්න	7605	ාර්	4840	බි	3712	ළු	2863
මා	14112	ගන	7519	ෙද්	4785	කය	3701	සිටි	2863
ෙද	13289	ලා	7472	එක	4771	භ	3694	හැකි	2856

ලබ	2854	සම්	2391	තක	2092	ලිය	1850	ටත්	1616
මෙ	2854	කාට	2388	කින	2090	වින	1844	අම	1616
තුව	2842	ණ්ඩ	2385	්රී	2081	වද	1838	හන්	1612
ෙමන	2820	තිබ	2382	ගැනීම	2073	මී	1837	වස්	1611
නම්	2793	කී	2377	ම්බ	2069	පුර	1836	ලී	1610
ටු	2790	ෙදන	2373	ත්වය	2066	යව	1834	පති	1606
ලැ	2782	අය	2327	එම	2065	ායක	1828	අල	1606
ලැ	2782	මාන	2319	තිව	2064	පිර	1828	බන්	1605
තාව	2768	පෙ	2313	නෙය	2063	මග	1826	ග්ර	1605
මණ	2739	ෙවන්	2306	ඉන්	2058	කාල	1822	නෙය්	1604
තව	2733	රජ	2304	පරි	2049	නැත	1815	පන්	1603
ෙමම	2721	ාශ	2284	අද	2049	ාෙහ	1796	නාය	1601
ලු	2721	තය	2277	නිය	2046	රති	1794	ලැබ	1597
ලු	2721	ධාන	2270	ද්ධ	2018	දැන	1776	න්නට	1591
බැ	2715	දු	2266	ලක්	2012	ෙරෝ	1775	පිය	1590
බු	2708	ලට	2262	පමණ	2005	පය	1763	රද	1589
ෙගා	2674	හර	2261	රග	2003	තුර	1760	මාර	1584
ස්ථා	2671	න්ව	2259	යාව	2002	ජී	1759	තිෙබ්	1583
යත	2665	කර	2257	ෙර්	1996	දක්	1746	ප්රති	1581
රර	2657	ෙකාට	2246	පහ	1992	ශා	1743	ප්රති	1581
සඳහා	2653	නිසා	2241	පහ	1992	විය	1742	ෙලෝ	1578
ඳහා	2653	නිසා	2241	රක්	1989	එක්	1742	ලෝ	1578
තුළ	2650	ක්රි	2240	දහ	1976	ලංක	1734	වලට	1577
ගත්	2640	ෙල්	2239	විට	1973	පාර	1733	භය	1576
ෙහෝ	2634	නවා	2229	ාඩ	1963	විසි	1729	වාද	1569
හෝ	2634	විද	2228	එය	1959	චා	1726	රාජ	1557
තුර	2634	ඩු	2222	වුන්	1953	පිළිබ	1723	ච්	1556
ෙමන්	2628	වුන	2221	රෝ	1953	සමා	1710	ත්ෙත්	1554
පසු	2609	තිෙබ	2219	දින	1952	ර්ම	1708	ෝග	1552
රන්	2600	මහත	2214	ාළ	1947	ඔවුන	1708	වයි	1551
සර	2599	ගය	2211	ලබා	1945	ලංකා	1701	පිරි	1550
තිය	2583	අනු	2211	ලං	1943	වන්ෙ	1698	කව	1542
නෙ	2573	සී	2194	විත	1934	කම්	1689	කාෙ	1538
ගල	2571	න්ෙග	2193	ඊ	1933	යකි	1681	තත	1536
ැට	2566	වැන	2188	ාලි	1930	න්ෙග්	1681	හතා	1534
සිදු	2535	මක්	2185	සන්	1930	න්ධ	1676	විධ	1532
ගැනී	2514	සත	2170	ශෙ	1927	ගිය	1674	මහතා	1530
සක	2510	අාර	2166	ඔවු	1925	දය	1661	සය	1529
ෙයෝ	2509	රණය	2163	ධි	1916	අවස	1660	බී	1522
ඔහ	2504	ෙහා	2162	හාර	1913	ක්රිය	1659	රීමට	1510
හස	2489	ැනීම	2148	සින්	1913	ළිබඳ	1653	ාවි	1502
ෙබ්	2486	ැයි	2139	ලින්	1913	පිළිබඳ	1653	ධාර	1500
වර්	2476	මැති	2137	ෙට	1899	දුර	1639	තෙ	1500
වැඩ	2457	මත්	2123	ජාත	1886	වම	1637	ජීව	1500
ත්ෙ	2436	වශ	2120	ෙජ්	1877	ාති	1632	ාවක්	1498
නාව	2427	ැනි	2119	වලි	1866	පළ	1628	දෙ	1498
න්ට	2426	ගු	2116	කින්	1866	ප්රත	1625	මව	1494
පැව	2420	රියා	2101	වැනි	1860	ෙපාල	1623	ාවට	1492
දිය	2402	සල	2093	හත්	1859	රම්	1622	වනු	1479

අර	1474	ක්රියා	1334	රින	1231	ොපාලි	1133	ජී	1079
හිට	1473	අස	1333	ටිය	1230	තිොය	1132	එන	1077
හිට	1473	ොබා	1328	වාර	1229	නීමට	1130	දන්	1076
ාවන	1464	වීමට	1328	එහි	1226	ශයන්	1128	ොක්	1075
යකු	1460	ස්ථාන	1327	ඉන්ද	1223	යද	1127	ාලිස	1075
ැවැ	1456	වැඩි	1325	ංශ	1223	කෘ	1127	වශෙ	1071
ශය	1455	හ්	1324	නිෙ	1222	මයි	1124	පී	1071
ඉදි	1452	මස	1324	වලින්	1221	අාද	1124	ාකාර	1070
විසින	1449	ැඳ	1318	ාතික	1220	ණෙ	1121	ශ්ය	1069
යාප	1441	ාධි	1316	වාස	1219	ශ්ෂ	1119	ශ්ය	1069
ගර	1436	කතු	1310	රෙය්	1219	කෙය	1119	විෙශ	1067
සට	1434	පැවැ	1307	සාම	1217	ෙපන	1117	විෙශ්	1065
තත්	1430	වති	1305	ෙපර	1207	ෙශ්ෂ	1116	විෙශ්ෂ	
පිට	1420	අැත්	1303	යාෙ	1206	ෂි	1112		1064
ධන	1418	හළ	1302	භාව	1206	ෙසා	1110	නිව	1064
ර්ය	1413	කාශ	1300	නුෙ	1204	ැත්ෙත		ගී	1061
ෙකෙ	1405	ායි	1299	විද්ය	1200		1110	නයක්	1060
ගන්න	1405	ළ්	1291	ළා	1198	න්දු	1109	බවට	1059
හමු	1403	ෙළ්	1289	ම්ම	1197	යාන	1108	බන්ධ	1057
කිරීමට	1399	ටයුතු	1289	ම්ය	1195	ෙබෝ	1107	කතුව	1056
කියා	1397	කටයුතු	1289	ම්ය	1195	ෙදෙ	1107	ඉඩ	1055
ගෙ	1396	්යාප	1288	සුව	1193	ැක්	1107	තවත	1053
ැත්ෙ	1395	කැර	1288	කෙළ	1193	බෝ	1107	තුන	1052
අඩ	1395	එහ	1287	දැක	1192	තිවර	1105	නකු	1050
ැනිව	1383	පද	1284	බාෙ	1189	ාත්ම	1102	රෙද	1049
දුව	1383	රැද	1280	ෙයා	1188	ගාඩ	1102	ජල	1049
නක්	1380	දිරි	1279	ැකිය	1188	ප්රකා	1101	රත්	1047
ලන්	1379	ැවන්	1273	ගින	1187	යාම	1098	අත්	1043
හල	1378	ස්ෙ	1271	ක්ෂණ	1187	ඉතා	1097	මහි	1038
යින්	1376	සක්	1271	යැ	1186	ෙගාඩ	1096	බර	1037
ටයු	1372	රැවන්	1271	ජාතික	1171	නස	1096	ණයක	1032
කටයු	1372	හිත	1267	ෙමර	1170	ගණ	1092	ගාව	1031
ාස්	1368	මහ	1265	ාල්	1170	ළම	1091	න්ය	1029
කරණ	1367	වලින	1264	ෙමෙ	1167	පාෙ	1091	එකතුව	1028
කන	1366	ාදී	1262	රැ	1166	නිල	1091	ැල්	1026
ැප	1363	ක්රී	1262	පක්ෂ	1165	වරය	1090	කෙළ්	1026
ජාති	1363	පාලි	1260	ක්රීඩ	1163	වස්ථ	1089	කෙළ්	1026
පාලන	1359	ස්ක	1255	කයන්	1162	දල	1089	ෙලෝක	
න්ම	1359	ෙහ්	1254	වරැ	1159	ෙවත	1086		1024
සව	1355	ඩ්	1253	ෙබන	1157	සිත	1085	ලෝක	1024
ණක	1355	යත්	1252	දීම	1151	යන්ෙ	1085	ැවි	1023
වන්ෙන්		එකතු	1248	සංව	1148	අග	1085	ැතැ	1023
	1350	දහස	1246	ඡ	1148	අනුව	1084	ක්ත	1023
තැන	1347	නායක	1241	එෙහ	1147	ෙපෙ	1083	ව්යා	1018
කිස	1346	නාෙ	1240	ෙගන්	1146	සිර	1083	වශ්ය	1018
ථාන	1343	කාව	1240	ක්ක	1143	වස්ථා	1083	ගමන	1018
ෙස්ව	1338	ත්ය	1239	තරග	1142	හිම	1081	නයට	1017
මාණ	1338	ශි	1236	යම්	1139	යකට	1079	ජ්ය	1017
කරන්	1336	න්නා	1234	සිං	1138	ටී	1079	කුණ	1016

පැම	1013	ධික	1011	ෘති	1007	ඝ	1005	තවත්	1001
මරිකා	1012	තරම	1010	වරණ	1006	රෙද්ශ	1003	තවත්	1001
ෙට	1011	පැමි	1008	ඉදිරි	1006	ප්රෙද්	1002		

Appendix J: Exhaustive Search for a Word

0 ඇනුලු	0 ඇකුලු	47 ඇතුලු
0 ඇනුලූ	0 ඇකුලූ	0 ඇතුලූ
0 ඇනුලැ	0 ඇකුලැ	0 ඇතුලැ
0 ඇනුලු	0 ඇකුලු	47 ඇතුලු
0 ඇනුළු	0 ඇකුළු	370 ඇතුළු
0 ඇනුළූ	0 ඇකුළූ	0 ඇතුළූ
0 ඇනුළැ	0 ඇකුළැ	0 ඇතුළැ
0 ඇනුළු	0 ඇකුළු	370 ඇතුළු
0 ඇනූලු	0 ඇකූලු	0 ඇතූලු
0 ඇනූලූ	0 ඇකූලූ	0 ඇතූලූ
0 ඇනූලැ	0 ඇකූලැ	0 ඇතූලැ
0 ඇනූලු	0 ඇකූලු	0 ඇතූලු
0 ඇනූළු	0 ඇකූළු	0 ඇතූළු
0 ඇනූළූ	0 ඇකූළූ	0 ඇතූළූ
0 ඇනූළැ	0 ඇකූළැ	0 ඇතූළැ
0 ඇනූළු	0 ඇකූළු	0 ඇතූළු
0 ඇනෑලු	0 ඇකෑලු	0 ඇතෑලු
0 ඇනෑලූ	0 ඇකෑලූ	0 ඇතෑලූ
0 ඇනෑලැ	0 ඇකෑලැ	0 ඇතෑලැ
0 ඇනෑලු	0 ඇකෑලු	0 ඇතෑලු
0 ඇනෑළු	0 ඇකෑළු	0 ඇතෑළු
0 ඇනෑළූ	0 ඇකෑළූ	0 ඇතෑළූ
0 ඇනෑළැ	0 ඇකෑළැ	0 ඇතෑළැ
0 ඇනෑළු	0 ඇකෑළු	0 ඇතෑළු
0 ඇනූලු	0 ඇකූලු	0 ඇතූලු
0 ඇනූලූ	0 ඇකූලූ	0 ඇතූලූ
0 ඇනූලැ	0 ඇකූලැ	0 ඇතූලැ
0 ඇනූලු	0 ඇකූලු	0 ඇතූලු
0 ඇනූළු	0 ඇකූළු	0 ඇතූළු
0 ඇනූළූ	0 ඇකූළූ	0 ඇතූළූ
0 ඇනූළැ	0 ඇකූළැ	0 ඇතූළැ
0 ඇනූළු	0 ඇකූළු	0 ඇතූළු