Survey on Publicly Available Sinhala Natural Language Processing Tools and Research

Nisansa de Silva

Abstract—

Sinhala is the native language of the Sinhalese people who make up the largest ethnic group of Sri Lanka. The language belongs to the globe-spanning language tree, Indo-European. However, due to poverty in both linguistic and economic capital, Sinhala, in the perspective of Natural Language Processing tools and research, remains a resource-poor language which has neither the economic drive its cousin English has nor the sheer push of the law of numbers a language such as Chinese has. A number of research groups from Sri Lanka have noticed this dearth and the resultant dire need for proper tools and research for Sinhala natural language processing. However, due to various reasons, these attempts seem to lack coordination and awareness of each other. The objective of this paper is to fill that gap of a comprehensive literature survey of the publicly available Sinhala natural language tools and research so that the researchers working in this field can better utilize contributions of their peers. As such, we shall be uploading this paper to arXiv and perpetually update it periodically to reflect the advances made in the field.

Index Terms—Sinhala, Natural Language Processing, Resource Poor Language

1 INTRODUCTION

Sinhala\(^1\) language, being the native language of the Sinhalese people [2–4], who make up the largest ethnic group of the island country of Sri Lanka, enjoys being reported as the mother tongue (L1) of approximately 16 million people [5, 6]. When both L1 and L2 speakers are counted, 79.7% of the total Sri Lankan population are literate in Sinhala [7].

To give a brief linguistic background for the purpose of aligning the Sinhala language with the baseline of English, primarily it should be noted that Sinhala language belongs to the same the Indo-European language tree [8, 9]. However, unlike English, which is part of the Germanic branch, Sinhala belongs to the Indo-Aryan branch. Further, Sinhala, unlike English, which borrowed the Latin alphabet, has its own writing system, which is a descendant of the Indian Brahmi script [10–16]. By extension, this makes Sinhala Script a member of the Aramaic family of scripts [17, 18].

Thus by inheritance, Sinhala writing system is abugida (alphasyllabary), which to say, that consonant-vowel sequences are written as single units [19]. We show the evolutionary eras of the Sinhala script in Appendix A. It should be noted that the modern Sinhala language have loanwords from languages such as Tamil, English, Portuguese, and Dutch due to various historical reasons [20]. Regardless of the rich historical array of literature spanning several millennia (starting between 3\(^{rd}\) to 2\(^{nd}\) century BCE [21, 22]), modern natural language processing tools for the Sinhala language are scarce [23].

\(^1\)Englebretson and Genetti [1] observe that in some contexts the Sinhala language is also referred as Sinhalese, Singhala, and Singhalese.

Natural Language Processing (NLP) is a broad area covering all computational processing and analysis of human languages. To achieve this end, NLP systems operate at different levels [24–26]. A graphical representation of NLP layers and application domains are shown in Figure 1. On one hand, according to Liddy [25], these systems can be categorized into the following layers: phonological, morphological, lexical, syntactic, semantic, discourse, and pragmatic. The phonological layer deals with the interpretation of language sounds. As such, it consists of mainly speech-to-text and text-to-speech systems. In cases where one is working with the written text of the language rather than speech, it is possible to replace this layer with tools which handle Optical Character Recognition (OCR) and language rendering standards (such as Unicode [27]). The morphological layer analyses words at their smallest units of meaning. As such, analysis on word lemmas and prefix-suffix-based inflection are handled in this layer. Lexical layer handles individual words. Therefore tasks such as Part of Speech (PoS) tagging happens here. The next layer, syntactic, takes place at the phrase and sentence level where grammatical structures are utilized to obtain meaning. Semantic layer attempts to derive the meanings from the word level to the sentence level. Starting with Named Entity Recognition (NER) at the word level and working its way up by identifying the contexts they are set in until arriving at overall meaning. The discourse layer handles meaning in textual units larger than a sentence. In this, the function of a particular sentence maybe contextualized within the document it is set in. Finally, the pragmatic layer handles contexts read into contents without having to be explicitly mentioned [24, 25]. Some forms of anaphora (coreference) resolution [28–32] fall into this application.

On the other hand, Wimalasuriya and Dou [26] categorize NLP tools and research by utility. They introduce three categories with increasing complexity; Information Retrieval...
Information Retrieval covers applications, which search and retrieve information which are relevant to a given query. For pure IR, tools and methods up-to and including the syntactic layer in the above analysis are used. Information Extraction, on the other hand, extracts structured information. The difference between IR and IE is the fact that IR does not change the structure of the documents in question. Be them structured, semi-structured, or unstructured, all IR does is fetching them as they are. In comparison, IE, takes semi-structured or unstructured text and puts them in a machine readable structure. For this, IE utilizes all the layers used by IR and the semantic layer. Natural Language Understanding is purely the idea of cognition. Most NLU tasks fall under AI-hard category and remain unsolved [24]. However, with varying accuracy, some NLU tasks such as machine translation are being attempted. The pragmatic layer of the above analysis belongs to the NLU tasks while the discourse layer straddles information extraction and natural language understanding [24].

The objective of this paper is to serve as a comprehensive survey on the state of natural language processing resources for the Sinhala language. The initial structure and content of this survey are heavily influenced by the preliminary surveys carried out by de Silva [23] and Wijeratne et al. [24]. However, our hope is to host this survey at arXiv as a perpetually evolving, living research article which continuously gets updated as new research and tools for Sinhala language are created and made publicly available. We also discuss how the non-compliance of policies to put data and code online [36], after the research is concluded and the paper is published, have negatively impacted the growth and sustainability of Sinhala NLP. Hence, it is our hope that this work will help future researchers who are engaged in Sinhala NLP research to conduct their literature surveys efficiently and comprehensively. For the success of this survey, we shall also consider the Sri Lankan NLP tools repository, lknlp. This manuscript is at version 5.2.0. The latest version of the manuscript can be obtained from arXiv or ResearchGate.

Figure 6 in Appendix B shows the most prolific researchers in the domain of Sinhala NLP. The nodes contain the name of the researcher along with the total number of Sinhala NLP papers that researcher has authored. The edges between two researchers are labeled with the number of Sinhala NLP papers the relevant pair of researchers have co-authored. When selecting authors, we have applied a threshold of 3 Sinhala NLP publications. Given that the objective of the visualization is to portray the co-operation between researchers, we have also added the strongest edge that connects each researcher to the rest of the researchers in the graph. The few isolated nodes are researchers that have authored at least 3 Sinhala NLP publications but do not have any collaborators.

---

2. This is, however, not without the criticism of being nothing more than a Chinese room [33] rather than true NLU.
not have any coauthored papers with anyone else on the graph. We have also added labels to clusters in cases where all or the majority of researchers in those clusters have the same affiliation. It is observable that the cluster from the Department of Computer Science & Engineering, University of Moratuwa is the most prolific in Sinhala NLP research.

Figure 7 in Appendix B shows the probability of studies from the institutions to which the most prolific authors from Figure 6 are affiliated citing institutions of the same set. We observe both interesting and disturbing trends which we discuss in the figure caption.

Figure 8 in Appendix B shows the mapping between authors with at least 10 papers in the Sinhala NLP domain and their research interests denoted by the subsection titles of the Section 3 of this paper. If two or more authors on the diagram have co-authored a paper, it gets counted for each of the authors separately without any bias on the author order listed on the publication. For example, the paper Building a wordnet for Sinhala [37] is counted for both Nisamsa de Silva and Gihan Dias. If a single paper contributes to more than one research area, that paper is counted for all of the research areas to which it contributes. For example, the paper Sinhala Text Classification: Observations from the Perspective of a Resource Poor Language [23] which introduces a new Sinhala text classification data set is counted for both Data Sets (Section 3.2) and Text Classification (Section 3.10).

The remainder of this survey is organized as follows; Section 2 introduces some important properties and conventions of the Sinhala language which are important for the development and understanding of Sinhala NLP. Section 3 discusses the various tools and research available for Sinhala NLP. In this section we would discuss both pure Sinhala NLP tools and research as well as hybrid Sinhala-English work. We will also discuss research and tools which contributes to Sinhala NLP either along with or by the help of Tamil, the other official language of Sri Lanka. Section 4 gives a brief introduction to the primary language sources used by the studies discussed in this work. Finally, Section 5, concludes the survey.

2 Properties of the Sinhala Language

Before moving on to discussing Sinhala NLP resources, we shall give a brief introduction to some of the important properties of Sinhala language, which impact the development of Sinhala NLP resources. Sinhala grammar has two forms: written (literary) and spoken. These forms differ from each-other in their core grammatical structures [1, 38, 39]. The written form strictly adheres to the SOV (Subject, Object, and Verb) configuration [40, 41]. Further, in the written form, subject-verb agreement is enforced [42] such that, in order to be grammatically correct, the subject and the verb must agree in terms of: gender (male/female), number (singular/plural) and person (1st/2nd/3rd). However, in spoken Sinhala, the SOV order can be neglected [43] and male singular 3rd person verb can be used for all nouns [42]. Similar to many Indo Aryan languages, animacy plays a major role in Sinhala grammar in syntactic and semantic roles [46–48]. Comparative studies done by Noguchi [49] and by Miyagishi [39, 50] have found that animacy extends its influence from phrase level to sentence level in Sinhala (e.g., Usage of post-positions [40, 51]). On this matter, Table 1 explains grammatical cases and inflections of animate common nouns while Table 2 explains grammatical cases and inflections of inanimate common nouns. We provide a comparative analysis of parsing the very simple English sentence “I eat a red apple” and its Sinhala, Hindi, and French translations in Fig 2. English and French parsing was done using the Stanford Parser 6. Hindi parsing was done using the IIT-Hyderabad Parser 7 and the study by Singh et al. [52].

Herath et al. [53, 54] argue that pure Sinhala words did not have suffixes and that adding suffixes was incorporated to Sinhala after 12th century BC with the influx of Sanskrit words. With this, they declare Sinhala to have to following types of words:

1) Suffixes
2) Nouns
3) Cases
4) Verbs
5) Conjunctions and articles
6) Adjectives
7) Demonstratives, Interrogatives, and negatives
8) Particles and prefixes

They further divide nouns into five groups: material, agentic, common, abstract, and proper. In addition to these, they also introduce compound nouns. We show the noun categorization proposed by Herath et al. [53] in Table 3.

Herath et al. [54] categorize Sinhala suffixes along the attributes of: gender, number, definiteness, case, and conjunctive. They further claim that there are 3 types of suffixes: Suf1 adds gender, number, and definiteness; Suf2 adds case; and Suf3 adds conjunctive. Conjunctive is claimed to be equivalent to too and and in English. We show an extension of the suffix structure proposed by Herath et al. [54] in Table 4. In their analysis on register variation (vocabulary) of 60 languages, Li et al. [55] observes that Sinhala exhibits homogeneity between 0.5 and 1.0 in the three corpora considered: CC (macro-web register - Common Crawl), TW (social media register - Twitter), and WK (Wikipedia register - March 2020).

3 Sinhala NLP resources

In this section we generally follow the structure shown in Figure 1 for sectioning. However, in addition to that, we also discuss topics such as available corpora, other data sets, dictionaries, and WordNets. We focus on NLP tools and research rather than the mechanics of language script handling [56–61]. One of the earliest attempts on Sinhala NLP was done by Herath et al. [62]. However, progress on that project has been minimal due to the limitations of their time. The later work by Nandasara [63] has not caught much of the advances done up to the time of its publication. Given that it was a decade old by the time the first edition
TABLE 1: Examples for grammatical cases and inflection of animate common nouns

| Form   | Case     | Singular | Plural |
|--------|----------|----------|--------|
|        |          | Masculine | Feminine | Masculine | Feminine |
|        |          | Definite | Indefinite | Definite | Indefinite | Definite | Indefinite |
| 1      | Nominative | ෙපʣත | ෙපʣතú | ෙපʣƮ |
| 2      | Accusative | ෙපʣත | ෙපʣතú | ෙපʣƮ |
| 3      | Dative    | ෙපʣත | ෙපʣත | ෙපʣƮ |
| 4      | Genitive  | ෙපʣත | ෙපʣŋ | ෙපʣŋ |
| 5      | Locative  | ෙපʣŋ | ෙපʣŋ | ෙපʣŋ |
| 6      | Instrumental | ෙපʣŋ | ෙපʣŋ | ෙපʣŋ |
| 7      | Vocative  | ෙපʣŋ | ෙපʣŋ | ෙපʣŋ |

TABLE 2: Examples for grammatical cases and inflection of inanimate common nouns

Note that the grammatical cases of Auxiliary and Vocative do not exist for inanimate nouns in Sinhala.

| Form   | Case     | Singular | Plural |
|--------|----------|----------|--------|
|        |          | Masculine | Feminine | Masculine | Feminine |
|        |          | Definite | Indefinite | Definite | Indefinite | Definite | Indefinite |
| 1      | Nominative | ෙපʣත | ෙපʣතú | ෙpname_1 |
| 2      | Accusative | ෙpname_2 | ෙpname_3 | ෙpname_4 |
| 3      | Dative    | ෙpname_5 | ෙpname_6 | ෙpname_7 |
| 4      | Genitive  | ෙpname_8 | ෙpname_9 | ෙpname_10 |
| 5      | Locative  | ෙpname_11 | ෙpname_12 | ෙpname_13 |
| 6      | Instrumental | ෙpname_14 | ෙpname_15 | ෙpname_16 |
| 7      | Vocative  | ෙpname_17 | ෙpname_18 | ෙpname_19 |

TABLE 3: Noun categorization by Herath et al. [53]

| Type    | Examples                                      |
|---------|-----------------------------------------------|
| Material| ෙකඳු, කොඳි                              |
| Agentive| අංගලිංග, ආංගලිංග                        |
| Common  | උබුන්, කෑඹාඹා                        |
| Abstract| උබුන්, ආංංංං                            |
| Proper  | ෙකඳු, ෙකඳු                                |
| Compound| ෙකඳු (බුද්ධ+බුද්ධ), ෙකඳු (බුද්ධ+බුද්ධ) |

of this survey was compiled, we observe the existence of many new discoveries in Sinhala NLP which have not been taken into account by it. A review on some challenges and opportunities of using Sinhala in computer science was done by Nandasara and Mikami [64]. At this point, it is worth noting that the largest number of studies in Sinhala NLP has been on optical character recognition (OCR) rather than on higher levels of the hierarchy shown in Figure 1. On the other hand, the most prolific single project of Sinhala NLP we have observed so far is an attempt to create an end-to-end Sinhala-to-English translator [19, 65–82]. Tamil, the other official language of Sri Lanka is also a resource-poor language. However, due to the existence of larger populations of Tamil speakers worldwide, including but not limited to economic powerhouses such as India, there are more research and tools available for Tamil NLP tasks [24]. Therefore, it is rational to notice that Sinhala and Tamil NLP endeavours can help each other. Especially, given the above fact, that these are official languages of Sri Lanka, results in the generation of parallel data sets in the form of official government documents and local news items. A number of researchers make use of this opportunity. We shall be discussing those applications in this paper as well. Further, there have been some fringe implementations, which bridge Sinhala with other languages such as Japanese [9, 22, 83–85].

3.1 Corpora

For any language, the key for NLP applications and implementations is the existence of adequate corpora. On this matter, a relatively substantial Sinhala text corpus was created by Upeksha et al. [86, 87] by web crawling. It was later extended by adding Jathaka Stories and more web-crawled news articles. Later a smaller Sinhala news corpus was created by de Silva [23]. Both of the above corpora are publicly available. However, none of these come close to the massive capacity and range of the existing English corpora. A word corpus of approximately 35,000 entries was developed by Weerasinghe et al. [88]. But it does not seem to be online anymore. Guzmán et al. [89] provided two monolingual corpora for Sinhala. Those were a 155k+ sentences of filtered Sinhala Wikipedia and 5178k+ sentences of Sinhala common crawl. Wijeratne and de Silva [90] have publicly released a massive corpus of text and stop words taken from a decade of Sinhala Facebook posts. While the stop word extraction of Wijeratne and de Silva [90] algorithmic, Lakmal et al. [91] has introduced a manually curated stop word list. A parallel corpus of Sinhala and English was collected by Bañón et al. [92] containing 217,407 sentences of filtered Sinhala Wikipedia containing 217,407 sentences of Sinhala Wikipedia.
Fig. 2: Parse trees for the sentence “I eat a red apple” in four languages.

sentences and available to download from their website\textsuperscript{16}. However, the later audit by Caswell et al. [93] raised issues on the quality of that data set. A parallel corpus\textsuperscript{17} of aligned Sinhala-English documents and sentences obtained from crawling the web was released by Sachintha et al. [94]. The study by Warusawithana et al. [95] created a refined version\textsuperscript{18} of the OpenSLR-52 speech corpus for Sinhala\textsuperscript{19} by Kjartansson et al. [96]\textsuperscript{20}. Dhananjaya et al. [98] created sin-cc-15M corpus\textsuperscript{21}, which they claim, at the time of their publication to be the largest monolingual Sinhala corpus. A text corpus collected from the Sihala blog Kalaya\textsuperscript{22} along with the relevant code is available on Github\textsuperscript{23}.

As for Sinhala-Tamil corpora, Hameed et al. [99] claim to have built a sentence aligned Sinhala-Tamil parallel corpus and Mohamed et al. [100] claim to have built a word aligned Sinhala-Tamil parallel corpus. However, at the time of writing this paper, neither of them was publicly available. A very small Sinhala-Tamil aligned parallel corpus created

\begin{itemize}
\item[16.] https://www.paracrawl.eu/
\item[17.] https://github.com/kdissa/comparable-corpus
\item[18.] https://bit.ly/3IF7t19
\item[19.] https://openslr.org/52
\item[20.] A later further overview on the corpus is also available [97].
\item[21.] 3https://tinyurl.com/42un7a9y
\item[22.] http://www1.kalaya.org/
\item[23.] https://bit.ly/KalayaCorpus
\end{itemize}
### Table 4: Extension of the suffix structure proposed by Herath et al. [54]

| Suf1 | Suf2 | Suf3 | Noun     | Attributes |
|------|------|------|----------|------------|
|      |      |      | -tree-   |            |
|      |      |      |          | Number | Definite | Case | Conjunct |
|      |      |      |          |        |          |      |          |
| -    | -    | -    | ශයූ (the-) | P      | U        | N, A, V | N        |
| මුන් | -    | -    | මුන් (a-) | S      | D        | N, A, V | N        |
| මුන් | -    | -    | මුන් (of a-) | S      | I        | G      | N        |
| මුන් මුන් | -    | -    | මුන් (to the-) | S      | D        | Da     | N        |
|      | මුන් | මුන් | මුන් (in the-) | S      | D        | Au     | N        |
|      | මුන් | මුන් | මුන් (on -s) | P      | U        | L      | N        |
|      | මුන් | මුන් | මුන් (to -s) | P      | U        | Da     | N        |
|      | මුන් | මුන් | මුන් (from a-) | S      | I        | Au     | N        |
| මුන් මුන් | මුන් | මුන් | මුන් (from a- too) | S      | I        | N, A, Y | Y        |
|      | මුන් | මුන් | මුන් (-and) | S      | D        | N, A, Y | Y        |
| මුන් මුන් | මුන් | මුන් | මුන් (of the- too) | S      | I        | N, A, Y | Y        |
| මුන් මුන් | මුන් | මුන් | මුන් (of a - too) | S      | U        | N, A, Y | Y        |
| මුන් මුunnel | මුunnel | මුunnel | මුunnel (to the- too) | S      | D        | Da     | Y        |
| මුunnel මුunnel | මුunnel | මුunnel | මුunnel (in -s too) | S      | D        | Au     | Y        |
| මුunnel මුunnel | මුunnel | මුunnel | මුunnel (on -s too) | P      | U        | L      | Y        |
| මුunnel මුunnel | මුunnel | මුunnel | මුunnel (to -s too) | P      | U        | Da     | Y        |
| මුunnel මුunnel | මුunnel | මුunnel | මුunnel (of the- too) | S      | D        | G      | Y        |
| මුunnel මුunnel | මු_tunnel | මු_tunnel | මු_tunnel (from a- too) | S      | I        | Au     | Y        |

The suffix structure proposed by Herath et al. [54] includes the following categories:
- **Number**: Singular (S) / Plural (P)
- **Definite**: Definite (D) / Indefinite (I) / Undecided (U)
- **Case**: Nominative (N) / Accusative (A) / Dative (Da) / Genitive (G) / Instrumentive (In) / Auxiliary (Au) / Locative (L) / Ablative (Ab)
- **Conjunct**: with suffix th (Y) / without suffix th (N)

Table 4 shows the extension of this suffix structure with additional attributes such as the type of noun and the possible combinations of suffixes.

---

**3.2 Data Sets**

Specific data sets for Sinhala, as expected, is scarce. However, a Sinhala PoS tagged data set [104–106] is available to download[^27]. Further, a Sinhala NER data set created by Manamini et al. [107] is also available to download from github[^26]. Liyanage et al. [108] analyzed Sinhala fastText and Word2Vec in the context of cross-lingual embedding spaces.

Facebook has released FastText [109–111] models for the Sinhala language trained using the Wikipedia corpus. They are available as both text models[^29] and binary files[^30]. Using the above models by Facebook, Lakmal et al. [91] have created an extended FastText model trained on Wikipedia, News, and official government documents. The binary file[^31] of the trained model is available to be downloaded.

Herath et al. [112], Herath and Medagoda [113] have compiled a report on the Sinhala lexicon for the purpose of establishing a basis for NLP applications. A comparative analysis of Sinhala word embedding has been conducted by Lakmal et al. [91]. A similar study was then conducted by Silva [114] analysing the progress that had been made in NLP in the intervening years. A dataset[^32] consisting of 3576 Sinhala documents drawn from Sri Lankan news websites and tagged (CREDIBLE, FALSE, PARTIAL or UNCERTAIN) was published by Jayawickrama et al. [115]. A benchmark data set for Sinhala spell correction was created by Sonnadara et al. [116] which they put online on Github[^33].

[^24]: http://bit.ly/2HTMEme
[^25]: https://github.com/Chaarangan/tamizhinet-corpus
[^26]: https://bit.ly/3VIK2Lj
[^27]: http://bit.ly/2Krhrbv
[^28]: http://bit.ly/2XrwCoK
[^29]: http://bit.ly/2JXAyL8
[^30]: http://bit.ly/2JY5J9c
[^31]: http://bit.ly/2WowH0h
[^32]: https://github.com/yudhanjaya/MisinformationCorpusSinhala/
[^33]: https://github.com/chason94/SinNeuSpellCorrector
De Saa and Ranathunga [117] has released a data set for Sinhala hate speech detection which consists of comments from youtube. Another data set consisting of Sinhala hate speech comments pulled from Facebook is available on Kaggle but it does not have an accompanying paper. Similarly, Perera et al. [118] have released a hate annotated dataset of 1600+ annotated Sinhala tweets. A text-to-speech data set with 3300 Sinhala sentences with 7.5 hours of recordings was released by the Path Nirvana Foundation. The open parallel corpus (OPUS) [39] by Tiedemann [119] curates the largest collection of parallel data between Sinhala and other languages. Zhang et al. [120] sampled data from OPUS for 100 languages (including Sinhala) and created the OPUS100 data set as a benchmark for neural machine translation (NMT). Thier code is also available on GitHub. Further, Tiedemann [121] created an NMT benchmark named Tatoeba MT Challenge using OPUS data based on Tatoeba. The English-Sinhala data set can also be directly downloaded from a link on Github.

A benchmark Sinhala-English translation data set named FLORES [41] was created by [89]. This includes a 600k+ Sinhala-English subtitle pairs initially collected by [122], 45k+ Sinhala-English sentence pairs from GNOME [46], KDE [47], and Ubuntu [48]. They further provided two monolingual corpora for Sinhala. Those were a 155k+ sentences of filtered Sinhala Wikipedia and 5178k+ sentences of Sinhala common crawl. In addition to the data set, they also have made their code publicly available. This work was then extended by NLLB Team et al. [123] of which the data and code is available on github under the No Language Left Behind (NLLB) project. Very importantly, they also provide a toxicity data set for Sinhala which can be used for hate speech detection in or by translation or otherwise.

Jenarthanan et al. [124] introduced the ACTSEA dataset which contains Sinhala and Tamil tweets annotated with emotions. They have 318,308 Sinhala tweets annotated. Dhananjaya et al. [98] have made publicly available three data sets developed for their study: 1) They have taken the corpus created by Sachintha et al. [94] and derived a Sinhala Neus source classification data set. They have taken the corpus created by de Silva [23] and derived a Sinhala Neus source classification data set, and 3) They have taken the corpus created by Upeksha et al. [86] and derived a Sinhala writing style classification data set. A data set of Sinhala swear and/or obscene words is publicly available in both Unicode and Singlish formats. Fernando et al. [125] has created aligned corpora for Sinhala–English, Tamil–English, and Sinhala–Tamil language pairs. The corpora are available on github as well as the code for their document and sentence alignment task. Buddhika et al. [126] introduce a crowd sourcing tool that they named Voicer to collect speech data. They claim that the tool is open source and that they had created a Sinhala Speech corpus of 10 hours with 39 different sentences in the banking domain. Neither the code nor the data is publicly linked in the research paper. However, a subsequent work by Hellarawa and Thayasivam [127] uses this data set. Thus, it can be assumed that this data set may be available if the authors are contacted.

Ranasinghe et al. [128] created a benchmark dataset they named SOLD: Sinhala Offensive Language Dataset which contains 10,000 posts from Twitter annotated both at sentence-level and token-level with the two classes offensive and not offensive. In the same paper, they also introduce the dataset SemiSOLD which contains 145,000 Sinhala tweets annotated with the same classes but with a semi-supervised approach. The relevant code is also available on Github. Sinhala is included in the 2800+ language metadata set composed by van Esch et al. [129]. This can be used for comparative analysis of Sinhala against other languages in the data set. Ruder et al. [130] created the multi-language dataset named XTREME-UP which contains Sinhala data sets for OCR and Transliteration tasks. Pratap et al. [131] created the The Massively Multilingual Speech (MMS) data set which contains Sinhala for the Spoken Language Identification (LID) task. A data set and a model for the Language Identification (LID) task including Sinhala were created by Burchell et al. [132]. A large Sinhala-English dictionary with 1,368,416 unfiltered and 195,255 filtered En-Si pairs has been made publicly available on github by Wickramasinghe and de Silva [133].

3.3 Dictionaries
A necessary component for the purpose of bridging Sinhala and English resources are Sinhala-English dictionaries. The earliest and most extensive Sinhala-English dictionary available for consumption was by Malalasekera [134]. However,
this dictionary is locked behind copyright laws and is not available for public research and development. This copyright issue is shared with other printed dictionaries [135–140] as well. The dictionary by Kulatunga [141] is publicly available for usage through an online web interface but does not provide API access or means to directly access the data set. The largest publicly available English-Sinhala dictionary data set is from a discontinued Firefox plug-in EnSiTip [142] which bears a more than passing resemblance to the above dictionary by Kulatunga [141]. Hettige and Karunanananda [68] claim to have created a lexicon to help in their attempt to create a system capable of English-to-Sinhala machine translation. A review on the requirements for English-Sinhala smart bilingual dictionary was conducted by Samarawickrama and Hettige [143]. The study by Wickramasinghe and de Silva [133] introduced a large Sinhala-English dictionary which has 1,368,416 unfiltered and 195,255 filtered En-Si pairs. Both their code and the two versions of the dictionary are publicly available on github:[69]

There exists the government sponsored trilingual dictionary [144], which matches Sinhala, English, and Tamil. However, other than a crude web interface on the ministry website, there is no efficient API or any other way for a researcher to access the data on this dictionary. Weerasinghe and Dias [145] have created a multilingual place name database for Sri Lanka which may function both as a dictionary and a resource for certain NER tasks.

3.4 WordNets

WordNets [146] are extremely powerful and act as a versatile component of many NLP applications. They encompass a number of linguistic properties which exist between the words in the lexicon of the language including but not limited to: hypernymy, hyponymy, synonymy, and meronymy. Their uses range from simple gazetteer listing applications [26] to information extraction based on semantic similarity [147, 148] or semantic oppositeness [149]. An attempt has been made to build a Sinhala Wordnet by Wijesiri et al. [37]. For a time it was hosted on [150] but it too is now defunct and all the data and applications are lost other than what Arukgoda et al. [151] have cloned to use in their application uploaded on github:[60]. However, even at its peak, due to the lack of volunteers for the crowd-sourced methodology of populating the WordNet, it was at best an incomplete product. Another effort to build a Sinhala Wordnet was initiated by Welgama et al. [152] independently from above; but it too have stopped progression even before achieving the completion level of Wijesiri et al. [37].

3.5 Morphological Analyzers

As shown in Fig 1, morphological analysis is a ground level necessary component for natural language processing. Given that Sinhala is a highly inflected language [23, 42, 43], a proper morphological analysis process is vital. The earliest attempt on Sinhala morphological analysis we have observed are the studies by Herath et al. [53, 54]. They are more of an analysis of Sinhala morphology rather than a working tool. As such we discussed the observations and conclusions of these works at Section 2. It is also worth to note that these works predate the introduction of Sinhala unicode and thus use a transliteration of Sinhala in the Latin alphabet.

The next attempt by Herath et al. [153] creates a modular unit structure for morphological analysis of Sinhala. Much later, as a step on their efforts to create a system with the ability to do English-to-Sinhala machine translation, Hettige and Karunanananda [65] claim to have created a morphological analyzer (void of any public data or code), which links to their studies of a Sinhala parser [66] and computational grammar [19]. Hettige et al. [77] further propose a multi-agent System for morphological analysis. Welgama et al. [154] attempted to evaluate machine learning approaches for Sinhala morphological analysis. Yet another independent attempt to create a morphological parser for Sinhala verbs was carried out by Fernando and Weerasinghe [155]. Later, another study, which was restricted to morphological analysis of Sinhala verbs was conducted by Dilshani and Dias [156]. There was no indication on whether this work was continued to cover other types of words. Further, other than this singular publication, no data or tools were made publicly accessible. Nandathilaka et al. [157] proposed a rule based approach for Sinhala lemmatizing. The work by Welgama et al. [158] claim to have set a set of gold standard definitions for the morphology of Sinhala Words; but given that their results are not publicly available, further usage or confirmation of these claims cannot not be done. The table 5 provides a comparative summary of the discussion above. The combined study introduced a rule based stemmer [159] and a tokenizer [160] for Sinhala. A later work by Kumarasinghe et al. [161] named SinMorphy used a comprehensive vocabulary of Sinhala words to conduct rule-based morphological analysis on Sinhala.

3.6 Part of Speech Taggers

The next step after morphological analysis is Part of Speech (PoS) tagging. The PoS tags differ in number and functionality from language to language. Therefore, the first step in creating an effective PoS tagger is to identifying the PoS tag set for the language. This work has been accomplished by Fernando et al. [104] and Dilshani et al. [105]. Expanding on that, Fernando et al. [104] has introduced an SVM Based PoS Tagger for Sinhala and then Fernando and Ranathunga [106] give an evaluation of different classifiers for the task of Sinhala PoS tagging. While here it is obvious that there has been some follow up work after the initial foundation, it seems, all of that has been internal to one research group at one institution as neither the data nor the tools of any of these findings have been made available for the use of external researchers. Several attempts to create a stochastic PoS tagger for Sinhala has been done with the studies by Herath and Weerasinghe [162], Jayaweera and Dias [163], and Jayasuriya and Weerasinghe [164] being the most notable.

Within a single group which did one of the above stochastic studies [163], yet another set of studies was carried out to create a Sinhala PoS tagger starting with the foundation of Jayaweera and Dias [165] which then extended

69. https://github.com/jseanm1/aruthSWSD
70. https://github.com/kasunw22/sinhala-para-dict/tree/main
to a Hidden Markov Model (HMM) based approach [166] and an analysis of unknown words [167, 168]. Further, this group presented a comparison of few Sinhala PoS taggers that are available to them [169]. A RESTful PoS tagging web service created by Jayaweera and Dias [170] using the above research can still be accessed via POST and GET. A hybrid PoS tagger for Sinhala language was proposed by Gunasekara et al. [171]. The study by Kothalawala et al. [172] discussed the data availability problem in NLP with a Sinhala POS tagging experiment among others. Withanage and Silva [173] proposed a stochastic PoS tagger based on a small 10,000 word corpus drawn from Facebook and Twitter. Wijerathna [174] used Support Vector Machines (SVM) to tag Sinhala text with 30 the PoS tags that were proposed by Fernando et al. [104]. The study by Sathsarani et al. [175] compared rule-based and stochastic models against deep learning models in the task of Sinhala PoS tagging.

### 3.7 Parsers

The PoS tagged data then needs to be handed over to a parser. This is an area which is not completely solved even in English due to various inherent ambiguities in natural languages. However, in the case of English, there are systems which provide adequate results [176] even if not perfect yet. The Sinhala state of affairs, is that, the first parser for the Sinhala language was proposed by Hettige and Karunananda [66] with a model for grammar [19]. The study by Liyanage et al. [43] is concentrated on the same given that they have worked on formalizing a computational grammar for Sinhala. While they do report reasonable results, yet again, do not provide any means for the public to access the data or the tools that they have developed. Kanduboda [42] have worked on Sinhala differential object markers relevant for parsing.

The first attempt at a Sinhala parser, as mentioned above, was by Hettige and Karunananda [66] where they created prototype Sinhala morphological analyzer and a parser as part of their larger project to build an end-to-end translator system. The function of the parser is based on three dictionaries: Base Dictionary, Rule Dictionary, and Concept Dictionary. They are built as follows:

- **The Base Dictionary**: prakarthi (base words), nipatha (prepositions), upasarga (prefixes), and vibakthi (Irregular Verbs).
- **The Rule Dictionary**: inflection rules used to generate various forms of verbs and nouns from the base words.
- **The Concept Dictionary**: synonyms and antonyms for the words found in the base dictionary.

**Parsers are, in essence, a computational representation of the grammar of a natural language.** As such, in building Sinhala parsers, it is crucial to create a computational model for Sinhala grammar. The first such attempt was taken by Hettige and Karunananda [19] with special consideration given to Morphology and the Syntax of the Sinhala language as an extension to their earlier work [66]. Here, it is worthy to note that, unlike in their earlier attempt [66], where they explicitly mentioned that they are building a parser, in this study [19], they use the much conservative claim of building a computational grammar. Under Morphology, they again handled Sinhala inflection. Their system is based on a Finite State Transducer (FST) and Context-Free Grammar (CFG) where they they modeled 85 rules for nouns and 18 rules for verbs. The specific implementation is more partial to a rule-based composer rather than parser. It is also worthy to note that this system could only handle simple sentences which only contained the following 8 constituents: Attributive Adjunct of Subject, Subject, Attributive Adjunct of Object, Object, Attributive Adjunct of Predicate, Attributive Adjunct of the Complement of Predicate, Complement of Predicate, and Predicate. With these, they propose the following grammar rules for Sinhala:

\[
S = \text{Subject} \text{ Akkyanaya} \\
\text{Subject} = \text{SimpleSubject} \mid \text{ComplexSubject} \\
\text{ComplexSubject} = \text{SimpleSubject} \text{ ConSub} \\
\text{SimpleSubject} = \text{Noun} \mid \text{Adjective Noun} \\
\text{ConSub} = \text{Conjunction SimpleSubject} \\
\text{Akkyanaya} = \text{VerbP} \mid \text{Object VerbP} \\
\text{Object} = \text{SimpleObject} \mid \text{ComplexObject} \\
\text{ComplexObject} = \text{Conjunction SimpleObject} \\
\text{SimpleObject} = \text{Noun} \mid \text{Adjective Noun} \\
\text{VerbP} = \text{Verb} \mid \text{Adverb Verb}\
\]

The later work by Liyanage et al. [43] also involves formalizing a computational grammar for Sinhala. They claim that Sinhala can have any order of words in practice. However, they do not note that this is happening because practices of the spoken language, which does not share the strong SOV conventions of the written language, are slowly seeping into written text. However, they do make note of how Sinhala grammar is modeled as a head-final

### Abbreviations

- **Nouns (Nu)**, **Verbs (Ve)**, **Adjectives (Aj)**, **Adverbs (Av)**, **Function Words (Fn)**, **Root (R)**, **Person (P)**, **Number (Nb)**, **Gender (G)**, **Article (A)**, **Case (C)**

| Modus Operandi                                      | Handle | Outputs | Dictionary                  | Base           | Handles       | Output   |
|------------------------------------------------------|--------|---------|-----------------------------|----------------|---------------|----------|
| Finite State Transducer                               | RB     | Y Y N N | Nu, Ve, Aj, Av, Fn, R, P, Nb, G | Hettige and Karunananda [65] | Y Y Y Y Y Y Y | Y N A O O O O O |
| Agent-based                                           | RB     | N/A     | Nu, Ve, Aj, Av, Fn, R, P, Nb, G | Hettige et al. [77] | Y Y Y Y Y Y Y | Y N A O O O O O |
| Rule-based                                             | RB     | N/A     | Nu, Ve, Aj, Av, Fn, R, P, Nb, G | Nanadithalaka et al. [157] | Y N N N N N N | Y N A O O O O O |
| Method                                                               |          |         | Nu, Ve, Aj, Av, Fn, R, P, Nb, G | Welgama et al. [154] | Y Y Y Y Y Y Y | Y N A O O O O O |
| Finite State Transducer                               | RB     | N/A     | Nu, Ve, Aj, Av, Fn, R, P, Nb, G | Fernando and Weerasinghe [155] | Y Y Y Y Y Y Y | Y N A O O O O O |
| Method                                                   |          |         | Nu, Ve, Aj, Av, Fn, R, P, Nb, G | Dilshani and Dias [156] | Y Y Y Y Y Y Y | Y N A O O O O O |

71. [http://bit.ly/2F0jKID](http://bit.ly/2F0jKID)

---

**TABLE 5: Morphological Analyzers comparison**

| Base Dictionary                                      | Modus Operandi                  | Handles       | Output   |
|-------------------------------------------------------|---------------------------------|---------------|----------|
| RB Finite State Automata                               | Y Y N N                         | Nu, Ve, Aj, Av, Fn, R, P, Nb, G | Y Y Y Y Y Y Y |
| Agent-based                                           | N/A                             | Nu, Ve, Aj, Av, Fn, R, P, Nb, G | Y Y Y Y Y Y Y |
| Rule-based                                             | N/A                             | Nu, Ve, Aj, Av, Fn, R, P, Nb, G | Y Y Y Y Y Y Y |

---
language [44]. They propose the Sinhala Noun Phrase (NP) to be defined as shown in equation 1 where NN is a noun which can be of types: common noun (N), pronoun (PrN) or proper noun (PropN). The adjectival phrase (ADJP) is then defined as shown in equation 2 where: Det is a Determiner, Adj is the adjective, and Deg is an optional operator Degrees which can be used to intensify the meaning of the adjective in cases where the adjective is qualitative. While they note that according to Gunasekara [177], there has to three classes of adjectives (qualitative, quantitative, and demonstrative), they do not implement this distinction in their system. Similarly, they propose Sinhala Verb Phrase (VP) to be defined as shown in equation 3 where V is a single verb. They here note that they are ignoring compound verbs and auxiliary verbs in their grammar. The adverbial phrases (ADV P) are then recursively defined as shown in equation 4.

\[ NP = [ADJP][NN] \]  
\[ ADJP = \left[ \text{Det} \left[ \text{Deg} \left[ \text{Adj} \right] \right] \right] \]  
\[ VP = [ADV P][V] \]  
\[ ADV P = \left[ \text{NP} \left[ \text{ADV P} \left[ \text{Deg} \left[ \text{ADV} \right] \right] \right] \right] \]

Similar to Hettige and Karunananda [19], the work by Liyanage et al. [43] also builds a CFG for Sinhala covering 10 out of the 25 types of simple sentence structures in Sinhala reported by Abhayasinghe [45]. This parser is unable to parse sentences where inanimate subjects do not consider the number. Further, sentences which contain, compound verbs, auxiliary verbs, present participles, or past participles cannot be handled by this parser. If the verbs have imperative mood or negation those too cannot be handled by this. Non-verbal sentences which end with adjectives, oblique nominals, locative predicates, adverbials, or any other language entity which is not a verb cannot be handled by this parser.

The study by Kanduboda [42] covers not the whole of Sinhala parsing but analyzes a very specific property of Sinhala observed by Aissen [178] which states that it is possible to notice Differential Object Marking (DOM) in Sinhala active sentences. Kanduboda [42] define this as the choice of /wa/ and /ha/ object markers. They further observe three unique aspects of DOM in Sinhala: (a) it is only observed in active sentences which contain transitive verbs, (b) it can occur with accusative marked nouns but not with any other cases, (c) it exists only if the sentence has placed an animate noun in the accusative position. They do a statistical analysis and provide a number of short gazetteer lists as appendices. However, they observe that further work has to be done for this particular language rule in Sinhala given that they found some examples which proved to be exceptions to the general model which they proposed.

### 3.8 Named Entity Recognition Tools

As shown in Fig 1, once the text is properly parsed, it has to be processed using a Named Entity Recognition (NER) system. The first attempt of Sinhala NER was done by Dahanayaka and Weerasinghe [179]. Given that they were conducting the first study for Sinhala NER, they based their approach on NER research done for other languages. In this, they gave prominent notice to that of Indic languages. On that matter, they were the first to make the interesting observation that NER for Indic languages (including, but not limited to Sinhala) is more difficult than that of English by the virtue of the absence of a capitalization mechanic. Following prior work done on other languages, they used Conditional Random Fields (CRF) as their main model and compared it against a baseline of a Maximum Entropy (ME) model. However, they only use the candidate word, Context Words around the candidate word, and a simple analysis of Sinhala suffixes as their features.

The follow up work by Senevirathne et al. [180] kept the CRF model with all the previous features but did not report comparative analysis with an ME model. The innovation introduced by this work is a richer set of features. In addition to the features used by Dahanayaka and Weerasinghe [179], they introduced, Length of the Word as a threshold feature. They also introduced First Word feature after observing certain rigid grammatical rules of Sinhala. A feature of clue Words in the form of a subset of Context Words feature was first proposed by this work. Finally, they introduced a feature for Previous Map which is essentially the NE value of the preceding word. Some of these feature extractions are done with the help of a rule-based post-processor which utilizes context-based word lists.

The third attempt at Sinhala NER was by Manamini et al. [107] who dubbed their system Ananya. They inherit the CRF model and ME baseline from the work of Dahanayaka and Weerasinghe [179]. In addition to that, they take the enhanced feature list of Senevirathne et al. [180] and enrich it further. They introduce a Frequency of the Word feature based on the assumption that most commonly occurring words are not NEs. Thus, they model this as a Boolean value with a threshold applied on the word frequency. They extend the First Word feature proposed by Senevirathne et al. [180] to a First Word/ Last Word of a Sentence feature noting that Sinhala grammar is of SOV configuration. They introduce a (PoS) Tag feature and a gazetteer lists based feature keeping in line with research done on NER in other languages. They formally introduce clue Words, which was initially proposed as a sub-feature by Dahanayaka and Weerasinghe [179], as an independent feature. Utilizing the fact that they have the ME model unlike Dahanayaka and Weerasinghe [179], they introduce a complementary feature to Previous Map named Outcome Prior, which uses the underlying distribution of the outcomes of the ME model. Finally, they introduce a Cutoff Value feature to handle the over-fitting problem.

The table 6 provides a comparative summary of the discussion above. It should be noted that all three of these models only tag NEs of types: person names, location names and organization names. The Ananya system by Manamini
et al. [107] is available to download at GitHub. The data and code for the approaches by Dahanayaka and Weerasinghe [179] and by Senevirathne et al. [180] are not accessible to the public. Azeez and Ranathunga [181] proposed a fine-grained NER model for Sinhala building on their earlier work on NER [107] and PoS tagging [106]. Anuruddha [182] proposed a method based on reinforcement learning for Sinhala NER. A Sinhala NER system restricted to the sports domain was introduced by Wijesinghe and Tissera [183], where they attempted to utilise classical machine learning models. The work by Mallikarachchi et al. [184] used support vector machines to detect Sinhala named entities of types person, location and, organization.

3.9 Semantic Similarity

A Sinhala semantic similarity measure has been developed for short sentences by Kadupitiya et al. [185]. This work has been then extended by Kadupitiya et al. [186] for the application use case of short answer grading. Data and tools for these projects are not publicly available. A sentence similarity measurement which uses Siamese neural networks was developed by Nilaxan and Ranathunga [187], where they demonstrate their results for Sinhala and Tamil. A cross-lingual document similarity measurement using the use-case of Sinhala and English was developed by Isuranga et al. [188].

3.10 Text Classification

Text classification is a popular application on the semantic layer of the NLP stack. A very basic Sinhala text classification using Naïve Bayes Classifier, Zipf’s Law Behavior, and SVMs was attempted by Gallege [189]. A smaller implementation of Sinhala news classification has been attempted by de Silva [23]. As mentioned in Section 3.2, their news corpus is publicly available. Another attempt at Sinhala text classification using six popular rule-based algorithms was done by Lakmali and Haddela [190]. Even though they talk about building a corpus named SinNG5, they do not indicate of means for others to obtain the said corpus. Another study by Kumari and Haddela [191] utilizes the SinNG5 corpus as the data set for their attempt to use LIME [192] for human interpretability of Sinhala document classification. However, they too do not provide access corpus. Nanayakkara and Ranathunga [193] have implemented a system which uses corpus-based similarity measures for Sinhala text classification. Gunasekara and Haddela [194, 195] claim to have created a context-aware stop word extraction method for Sinhala text classification based on simple TF-IDF. An LSTM based textual entailment system for Sinhala was proposed by Jayasinghe and Sirts [196]. Demotte and Ranathunga [197] proposed a dual-state capsule network architecture for text classification where they demonstrated their methodology on the Sinhala data set established by Senevirathne et al. [198]. The attempt by Sameemdeen and Selvanthan [199] considers three classical machine learning algorithms (Naïve Bayes, SVM, and KNN) and then goes on to briefly discusses the pros and cons of previous attempts by: Nanayakkara and Ranathunga [193], Gunasekara and Haddela [194], Lakmali and Haddela [190], and Buddhika et al. [200]. Then they propose active learning [201, 202] as an alternative. However, no experimental results of how active learning would improve Sinhala text classification are given. The study by Bandara et al. [203] proposed an ontology-based approach for Sinhala fake news detection. However, their literature survey did not cover seminal papers in OBIE such as the work by Wimalasuriya and Dou [26]. This has impacted the overall methodology that was presented. A novel, domain-dependent, and domain-adaptive text classification framework named AdaptText for Sinhala text classification was proposed by Kodithuwakkul and Hettiarachchi [204]. A simple TF-IDF based Sinhala text classification system was proposed by Koralage [205]. The study by [206] used Genetic Algorithm on Lucene search queries to obtain interpretable classification models for Sinhala text documents. The approach proposed by Rathnayake et al. [207] uses adapter-based [208–213] fine-tuning on XLM-R [214], for code-mixed and code-switched text. Kirindage and Godewithana [215] used LDA to develop Sinhala news topic hierarchies and categorize Sinhala news documents using the said topic hierarchies. Hettigoda [216] collected English-Sinhala code-mixed comments from Facebook pages belonging to Clothing industrial online businesses in Sri Lanka and classified them in to the classes: Inquiries, Maybe Inquiries, and Not Inquiries. A comparative analysis on BERT based models for Sinhala text classification was conducted by Dhananjaya et al. [98]. Both their code and data (sin-cc-15M corpus), Sinhala News source classification data set, Sinhala News category classification data set, and Sinhala writing style classification data set are publicly available. The study by Wijayarathna and Jayalal [217] used class-
cal machine learning techniques including Random Forest to classify fake news in Sinhala on Twitter. Chathuranga and Ranathunga [218] used capsule-based methods recommended by Seneviratne et al. [198] to classify Sinhala-English code-mixed data. The work by Weerasiri et al. [219] compared Word2Vec, FastText and Doc2vec against TF-IDF for Sinhala news document classification. The study by Caldera et al. [220] used stacked LSTM to classify Sinhala and Singlish text discussing COVID-19 on Youtube and Twitter.

3.11 Sentiment Analysis
A simple MLP-based method to classify sentiments in Sinhala text was initially proposed by Medagoda [221] based on their prior work [222]. A word2vec based tool [79] for sentiment analysis of Sinhala news comments is available. A methodology for constructing a sentiment lexicon for Sinhala Language in a semi-automated manner based on a given corpus was proposed by Chathuranga et al. [223]. Demotte et al. [224] proposed a sentiment analysis system based on sentence-state LSTM Networks for Sinhala news comments. In the subsequent work [225], they used word similarity to generate a Sinhala semantic lexicon. They followed this up with a further study [198] which discussed a number of other deep learning models such as RNN and Bi-LSTM in the domain of Sinhala sentiment analysis. Jayasuriya et al. [226] proposed a method to classify Sinhala posts in the domain of sports into positive and negative class sentiments. Ranathunga and Liyanage [227] claimed that using word embedding models as semantic features can compensate for the lack of well developed language-specific linguistic or language resources in the case of analysing sentiment of Sinhala news comments. Jayasuriya et al. [228] conducted a comparative study between word N-grams and character N-grams in the task of semantic classification of Sinhala content in social media. Which they soon followed up with an ensemble approach [229]. The work by Karunarathne [230] used word embedding for to analyse the sentiment of manually annotated Sinhala Tweets. Abeyratne and Jayaratne [231] conducted a multi model analysis on classifying Sinhala songs by emotion. The work by Jayawardhama et al. [232, 233] used the data set released by Wijeratne and de Silva [90] to predict the reactions induced by Sinhala Facebook posts. They then extended the work [234] and compared the results obtained with their data set against that of Seneviratne et al. [198]. Aththanayaka and Naleer [235] used Random forest, Support vector machines, and Multinomial Naïve Bayes models to analyse sentiment in Sinhala-English code-mixed text from social media. The ACTSEA dataset [80] for Sinhala sentiment analysis was introduced by Jenarathanan et al. [124]. The dataset contains 318,308 Sinhala tweets annotated with emotions.

3.12 Hate Speech Detection
As mentioned in the Section 3.2, a large annotated data set for Sinhala hate speech detection was created by NLLB Team et al. [123]. A machine learning approach to detect hate speech in Sinhala was proposed by De Silva [236]. A feature model and a data set for Sinhala hate speech detection for youtube was proposed by De Saa and Ranathunga [117]. Sandaruwan et al. [237] have attempted to identify abusive Sinhala comments in social media using text mining and machine learning techniques. A cyberbullying comment classification study for Sinhala was conducted by [238] where they used classical machine learning algorithms. The study by Hettiarachchi et al. [239] used classical machine learning methods to detect hate speech in Romanized Sinhala social media posts. While the basic idea is the same, they have avoided mentioning transliteration in their paper. The study by Samarasinghe et al. [240] proposed using CNNs for detecting hate speech in Sinhala text. Kariyawasam [241] proposed a machine learning approach for identifying toxic Sinhala language on social media. Guruge et al. [242] used an ensemble of Naïve Bayes, Support Vector Machine, XGBoost, MLP, and AdaBoost to detect hate speech in 49019 Tweets they collected from February of 2021 to April of 2021. Sandaruwan et al. [243] collected 3000 Sinhala comments corpus and conducted Multinomial Naïve Bayes to detect hate speech. Sheran [244] claims to have used machine learning to detect hate speech written in Sinhala or Singlish on social media. A deep learning approach for detecting hate speech in Sinhala tweets was explored by Munasinghe and Thayasivam [245]. The study by Shalinda and Munasinghe [246] utilized classical machine learning techniques such as Linear Support vector classifier, Random Forest Classification, SGD classifier, Logistic Regression, XGBoost classifier and multinomial Naïve Bayes classifier on both Sinhala and Singlish (Romanized Sinhala) to identify hate words. They report that the SGD classifier over TF-IDF with uni-grams and bi-grams gives them the highest accuracy. Gamage et al. [247] conducted a comparative analysis on a number of embedding systems as well as classical frequency-based methods for Sinhala hate speech detection. The study by Fernando et al. [248] also claims to use machine learning and deep learning to detect hate speech in Sinhala. The study by Perera et al. [249] predicted Sinhala hate speech using user behaviour on twitter by applying ensemble Machine Learning to classify them by the five big personality traits. The follow-up study by Perera et al. [118] analyses Sinhala hate speech propagation on Twitter. They have also published a dataset [82] of 1600+ annotated Sinhala tweets.

3.13 Word Sense Disambiguation
There have been multiple attempts to do word sense disambiguation (WSD) [250–254] for Sinhala. For this, Arukgoda et al. [151] have proposed a system named Aruth based on the Lesk Algorithm [255]. An online tool [83], an API [84] of the algorithm, and code along with data on github [85] are available. For the same task, Marasinghe et al. [256] have proposed a system based on probabilistic modeling. A dialogue...

79. http://bit.ly/2QK99Np
80. https://bit.ly/3oZoziI
81. 81. https://bit.ly/3FTNaMZ
82. 82. https://github.com/Isurie/Text-Classification-Module/tree/master/Dataset
83. 83. http://aruth.herokuapp.com/
84. 84. https://bit.ly/3sIEyb5
85. 85. https://github.com/jseanm1/aruthSWSD
act recognition system which utilizes simple classification algorithms has been proposed by Palihakkara et al. [257]. A word sense disambiguation tool named *Sinsense* was introduced by Subasingha [258]. They used cross-lingual sense disambiguation where English sense disambiguation was used to obtain Sinhala sense disambiguation. However, neither their tool nor their full research is publicly available.

### 3.14 Text Summarizing

A deterministic process flow for automatic Sinhala text summarizing was proposed by Welgama [259]. The study by Wimalasuriya [260], which has the same name as the above work by Welgama [259], uses the graph-based TextRank algorithm for automatic Sinhala text summarizing. The use case of Sinhala Text summarization for government gazettes was explored by Jayawardene [261]. The study by Rathnayake et al. [262] compared the results of extractive and abstractive summarization on Sinhala text books. The study by Jahan and Wijesekara [263] compared the abstractive summarizing methods of TF-IDF and Text-Rank for Sinhala using ROUGE as the evaluation score. They concluded TF-IDF to be the superior choice.

### 3.15 Other Semantic Tools

Applications of the semantic layer are more advanced than the ones below it in Figure 1. But even with the obvious lack of resources and tools, a number of attempts have been made on semantic level applications for the Sinhala Language. The earliest attempt on semantic analysis was done by Herath et al. [264] using their earlier work which dealt with Sinhala morphological analysis [53].

### 3.16 Phonological Tools

On the case of phonological layer, a report on Sinhala phonetics and phonology was published by Wasala and Gamage [265]. Wickramasinghe et al. [266] discussed the practical issues in developing Sinhala Text-to-Speech and Speech Recognition systems. *The Massively Multilingual Speech (MMS)* [86] data set created by Pratap et al. [131] has Sinhala data for the Spoken Language Identification (LID) task.

#### 3.16.1 Text-to-Speech

Based on the earlier work by Weerasinghe et al. [267], Wasala et al. [268] have developed methods for Sinhala grapheme-to-phoneme conversion along with a set of rules for schwa epenthesis. This work was then extended by Nadungodage et al. [269]. Weerasinghe et al. [270] developed a Sinhala text-to-speech system. However, it is not publicly accessible. They internally extended it to create a system capable of helping a mute person achieve synthesized real-time interactive voice communication in Sinhala [271]. A rule based approach for automatic segmentation of a small set of Sinhala text into syllables was proposed by Kumara et al. [272]. An *ew prosodic phrasing* method to help with Sinhala Text-to-Speech process was proposed by Bandara et al. [273, 274, 275]. Sodimana et al. [276] proposed a text normalization methodology for Sinhala text-to-speech systems. Further, Sodimana et al. [277] formalized a step-by-step process for building text-to-speech voices for Sinhala. Both Jayamanna [278] and Mishangi [279] have created Sinhala document readers for visually impaired persons to be used on Android devices. An OCR and Text-to-Speech system for Sinhala named Bhashitha was proposed by De Zoysa et al. [280]. The works by Lakmal et al. [281] and Senarathna et al. [282] adapted MaryTTS [283] to synthesize Sinhala speech. The study by Jayawardhana et al. [284] used *Deep Voice* [285] for Sinhala and English TTS. Gamage et al. [286] included a Sinhala text-to-speech module as one of the three modules present in their currency recognition system. Anuradha and Thelijjagoda [287] proposed a machine translation system to convert Sinhala and English Braille documents into voice. A separate group has done work on Sinhala text-to-speech systems independent to above [288].

#### 3.16.2 Speech-to-Text

On the converse, Nadungodage et al. [289] have done a series of work on Sinhala speech recognition with special notice given to Sinhala being a resource poor language. This project divides its focus on: continuity [290], active learning [291], and speaker adaptation [292]. A Sinhala speech recognition for voice dialing which is speaker independent was proposed by Amarasingha and Gamin [293] and on the other end, a Sinhala speech recognition methodology for interactive voice response systems, which are accessed through mobile phones was proposed by Manamperi et al. [294]. A Sinhala speech to Unicode text converter for the disaster relief domain was proposed by Prasangini and Nagahamulla [295]. Priyadarshani [296] proposes a method for speaker dependant speech recognition based on their previous work on: dynamic time warping for recognizing isolated Sinhala words [297], genetic algorithms [298], and syllable segmentation method utilizing acoustic envelopes [299]. The method proposed by Gunasekara and Meegama [300] utilizes an HMM model for Sinhala speech-to-text. A Sinhala speech recognizer supporting bi-directional conversion between Unicode Sinhala and phonetic English was proposed by Punchimudiyanse and Meegama [301]. The work by Karunayake et al. [302] transfer learns CNNs for transcribing free-form Sinhala and Tamil speech data sets for the purpose of classification. Dilshan [303] conducted a study for the specific use case of transcribing number sequences in continuous Sinhala speech. Gamage et al. [304] explored the use of combinational acoustic models such as Deep Neural Network - Hidden Markov Model (DNN-HMM) [305] and Subspace Gaussian Mixture Model (SGMM) [305] in Sinhala speech recognition. In the follow-up work, Gamage et al. [306] extended that work with end-to-end Lattice-Free Maximum Mutual Information (e2e LF-MMI) model [307] which is claimed to be a viable solution for low resource language speech recognition by Carmantini et al. [308]. However, it was shown that the new model slightly underperforms compared to the state-of-the-art result. Karunathilaka et al. [309] explore Sinhala speech recognition using deep learning models such as: pre-trained DNN, DNN, TDNN, TDNN+LSTM.
The first half of the study by Arafath [310] dealt with recognizing Sinhala speech using LSTMs. Gamage et al. [286] included a Sinhala speech recognition module as one of the three modules present in their currency recognition system. Time-delay neural network architectures (including multistream CNN architecture) was used for for acoustic modeling of Sinhala Automatic Speech Recognition (ASR) by Warusawithana et al. [311]. They have used the Kaldi speech recognition toolkit [312] for training the ASR models. As part of their child [sic] cognitive ability assessment model, Kahawanugoda et al. [313], proposed a Sinhala speech recognition system. The study by Azir et al. [314] attempts to identify number sequences spoken in Sinhala. TacoSi introduced by Arachchige and Weerasinghe [315] is based on Tacotron [316] and has been evaluated with 10 human evaluators to determine its text-to-speech quality. Nanayakkara [317] used DeepSpeech87 by Mozilla for Sinhala speech recognition.

3.16.3 Speech-to-Speech

Layansan et al. [318] created a speech-to-speech translation system for Sinhala on the Android platform. The system developed by Rajapakshe et al. [319] is also speech-to-speech in the sense that, it is a chat bot for scheduling medical appointments and giving medical advice where the front end contained speech recognition and voice synthesizer components that interfaced with a chat bot component in the back end.

3.16.4 Speech-to-Intent

The work by Karunanayake et al. [320] used English phoneme-Based Automatic Speech Recognition (ASR) for intent identification in Sinhala and Tamil. Ignatius and Thayasivam [321] proposed a speaker-invariant speech-to-intent classification model with i-vector based speaker normalization, which was then evaluated on Sinhala, and Tamil speech intent data sets. The later work by Yadav et al. [322] used pre-trained embeddings for Sinhala speech intent classification. Hellarawa and Thayasivam [127] proposes a BiLSTM-based ASR system for intent classification which they have tested on the banking domain Sinhala speech dataset created by Buddhika et al. [126]. For this, they report an accuracy of 98.53%.

3.16.5 Speech classification

The Sinhala speech classification system proposed by Buddhika et al. [200] does so without converting the speech-to-text. However, they report that this approach only works for specific domains with well-defined limited vocabularies. The work by Dinushika et al. [323] uses automatic speech recognition of Sinhala for speech command classification. Extending that, Kavmini et al. [324] presented a Sinhala speech command classification system which can be used for downstream applications. The work by Wolarathna et al. [325] used CNNs to classify emotions (sad, disgust, surprise, neutral, happy, calm, fear, and angry) of Sinhala speech by Autistic children. The voice assistant system created by Senarathne et al. [326] is capable of handling Sinhala voice commands.

3.16.6 Lip Synchronization

The study by Weerathunga et al. [327] worked on lip synchronization for Sinhala speech where videos of people speaking Sinhala were mapped to a visemes alphabet created by them. Further of this line of study, Wakkumbura et al. [328] came up with Phoneme-ViSeme mapping for Sinhala speech that they intended to be used for future applications of robotics.

3.17 Optical Character Recognition Applications

While it is not necessarily a component of the NLP stack shown in Fig 1, which follows the definition by Liddy [25], it is possible to swap out the bottom most phonological layer of the stack in favour of an Optical Character Recognition (OCR) and text rendering layer.

3.17.1 Printed Text

The XTREME-UP88 data set created by Ruder et al. [130] contains a Sinhala data set for the OCR task. The data was obtained from book transcriptions. An attempt for Sinhala OCR system has been taken by Rajapakse et al. [329] before any other work has been done on the topic. Much later, a linear symmetry-based approach was proposed by Premaratne and Bigun [330, 331]. They then used hidden Markov model-based optimization on the recognized Sinhala script [332]. Similarly, Hewavitharana et al. [333] used hidden Markov models for off-line Sinhala character recognition. Herath et al. [112], Herath and Medagoda [113] developed a prepossessing engine based on a template matching for printed Sinhala OCR. Statistical approaches with histogram projections for Sinhala character recognition is proposed by Hewavitharan and Kodikara [334], by Ajward et al. [335], and by Madushanka et al. [336]. Karunanayaka et al. [337] also did off-line Sinhala character recognition with a use case for postal city name recognition. A separate group had attempted Sinhala OCR [338] mainly involving the nearest-neighbor method [339, 340]. A study by Ediriweera [341] uses dictionaries to correct errors in Sinhala OCR. An early attempt for Sinhala OCR by Dias et al. [342] has been extended to be online and made available to use via desktops [343] and hand-held devices [344] with the ability to recognize handwriting. A simple neural network based approach for Sinhala OCR was utilized by Rimas et al. [345]. A fuzzy-based model for identifying printed Sinhala characters was proposed by Gunarathna et al. [346]. Premachandra et al. [347] proposes a simple back-propagation artificial neural network with hand crafted features for Sinhala character recognition. Another neural network with specialized feature extraction for Sinhala character recognition was proposed by Jayamaha and Naleer [348]. On the matter of neural networks and feature extraction, a feature selection process for Sinhala OCR was proposed by Kumara and Ragel [349]. Jayawickrama et al. [350] worked on Sinhala printed characters with special focus on handling diacritic vowels. However, they opted to refer to diacritic vowels as modifiers in their work. Gunawardhana and Ranathunga [351] proposed a limited approach to recognize Sinhala letters on Facebook images. A CNN-based methodology
to improve printed Sinhala character OCR was proposed by Liyanage [352]. Printed Character Recognition (PCR) was used by Vasanthuraj and Thayasivam [102] to create a large-scale Tamil-Sinhala-English parallel corpus. A meta-study on the effects of text genre, image resolution, and algorithmic complexity needed for Sinhala OCR from books and newspapers was conducted by Anuradha et al. [353]. Anuradha et al. [354] used Tesseract 3\textsuperscript{rd} [355] for Sinhala OCR. The later study by [356] improved the accuracy of Tesseract OCR engine on Sinhala from 53.22\% to 86.16\% for the data set they tested on. Maduranga and Jayalal [357] used Artificial Neural Network (ANN) based on Universe of Discourse and Self Organization Map methods to recognize multi-style printed Sinhala characters. An OCR and Text-to-Speech system for Sinhala named Bhashitha was proposed by De Zoysa et al. [280]. A study on Sinhala text extraction from social media images (memes) was conducted by Sama-rajeeva and Ranathunga [358]. They specifically handled the character-touching issue. The study by Walawage and Ranathunga [359] attempted to devise a feature set to separately identify Sinhala and English text on social media images (memes). The study by de Silva and Liyanage [360] used Convolutional Spiking Neural Networks to extract Sinhala text from YouTube thumbnails. Chanda et al. [361] proposed a Gaussian kernel SVM based method for word-wise Sinhala, Tamil, and English script identification. The work by Vasanthuraj et al. [103] adapted the Tesseract engine to handle non-Unicode (legacy fonts) in pdf documents to create a Tamil-Sinhala-English parallel corpus.

3.17.2 Handwritten Text

Fernando et al. [362] claim to have created a database for handwriting recognition research in Sinhala language and further claims that the data set is available at National Science Foundation (NSF) of Sri Lanka. However, the paper provides no URLs and we were not able to find the data set on the NSF website either. The work by Karunanayaka et al. [363] is focused on noise reduction and skew correction of Sinhala handwritten words. A genetic algorithm-based approach for non-cursive Sinhala handwritten script recognition was proposed by Jayasekara and Udawatta [364]. Nilaweera et al. [365] compare projection and wavelet-based techniques for recognizing handwritten Sinhala script. Silva and Kariyawasam [366] worked on segmenting Sinhala handwritten characters with special focus on handling dia-critic vowels. A comparative study of few available Sinhala handwriting recognition methods was done by Silva et al. [367]. Silva et al. [368] uses contour tracing for isolated characters in handwritten Sinhala text. A Sinhala handwriting OCR system which utilizes zone-based feature extraction has been proposed by Dharmapala et al. [369]. The study by Walawage and Ranathunga [370] and its follow up study by Walawage [371] specifically focus on segmentation of overlapping and touching Sinhala handwritten characters. Silva and Jayasundere [372] focused on recognizing character modifiers in Sinhala handwriting. The similarly named studies by Mariyathas et al. [373] and Wasaltilake and Kartheeswaran [374], both utilize CNN to recognize Sinhala handwriting; as does the study by Weerasinghe [375]. Ifhaam and Jayalal [376] used genetic algorithms to recognize Sinhala handwritten postal addresses for postal sorting. A segmentation-based approach that utilizes an n-gram model to recognize and validate Sinhala words written on touch screens was proposed by Mahesh and Priyankara [377]. They used a CNN classifier and was able to classify 19 different Sinhala characters. The study by Rowel et al. [378] frames their work as an E-Learning platform for hearing impaired children. However, their research does not contain any work done towards Sinhala sign language to be included in Section 3.22. What they do have is an OCR system that they claim to recognize letters and digits. Even there, we are only given an example of a recognised digit. Whether or not their system can recognize Sinhala letters is not explicitly discussed. As part of their child [sic] cognitive ability assessment model, Kahawanugoda et al. [313], proposed a Sinhala handwriting letter recognition system. The study by Withana and Rupasinghe [379] used Sinhala handwritten text classification to detect Dyslexia and Dysgraphia.

3.17.3 Ancient Text

Summarizing on optically recognized old Sinhala text for the purpose of archival search and preservation was explored by Rathnasena et al. [380]. The work of Peiris [381] also focused on OCR for ancient Sinhala inscriptions. Building upon the architecture proposed by Ruwanmini et al. [382], a neural network-based method for recognizing ancient Sinhala inscriptions was proposed by Karunarathne et al. [383]. The study by Wickramarathna and Ranathunga [384] created a system to recognize Brahmi characters, correct errors, and generate Sinhala meanings.

3.18 Translators

A meta-study on the viability of machine translators replacing English to Sinhala human translators was conducted by Dilshani and Seneviratna [385]. However, this study only involves 100 combined and complex English sentences translated to Sinhala by human translators as well as MT software. Given that reason and the fact that they seem to only used Google translate and Akura Sinhala dictionary app for comparison, the conclusions of this study may not be generalized. Another meta-study on the impact of pre-trained multilingual sequence-to-sequence models on low-resource language translation has been conducted by Lee et al. [386]; while they consider Sinhala as one of the examples, they do not go much into the specific impact due to the general nature of the paper. The meta-analysis by Ramadasa et al. [387] attempts to evaluate the goodness of the Google Sinhala-to-English translation by using the Google cloud API to translate Sinhala to English and then analysing the accuracy of the Sentiment Analysis task and the Named Entity Recognition task on the translated text. The meta-analysis by Das et al. [388] compared the results of translating English to 15 Indic languages including Sinhala using statistical translation methods. They used datasets from OPUS [119] for model building and utilized Flores-200 for fine-tuning. The NMT for Indic languages study conducted by Sheshadri et al. [389] discusses the Sinhala language translation in the abstract and conclusion but

89. https://tesseract-ocr.github.io/
the paper itself focuses more on the languages spoken in India. Nevertheless, it puts Sinhala into a regional linguistic perspective. The meta-study by Bapna et al. [390] discusses the task of building clean, web-mined datasets for a number of languages including Sinhala for the task of machine translation. This discussion was continued by Jones et al. [391] who discussed the bilingual lexica (BILEx) in the context of a number of languages including Sinhala. As mentioned in Section 3.1, the study by [93] raised questions on the quality of the existing Sinhala-English corpora. Sen et al. [392] on the other hand attempted to improve the quality of the Sinhala-English parallel corpora using fuzzy string matching where they tried to match the English translation of the given Sinhala sentence to the English sentence in the dataset pair. A study on the viability of using Google Translate for the legal domain English-Sinhala and Sinhala-English translation was conducted by [393]. They used human experts to extensively analyse the end result of the translation with many concrete examples of legal phrases that needed to be translated.

3.18.1 Sinhala-English Non-NMT

A series of work has been done by a group towards English to Sinhala translation as mentioned in some of the above subsections. This work includes; building a morphological analyzer [65], lexicon databases [68], a transliteration system [69], an evaluation model [74], a computational model of grammar [19], and a multi-agent solution [80]. After working on human-assisted machine translation [70], Het-tige and Karunananda [73, 75] have attempted to establish a theoretical basics for English to Sinhala machine translation. A very simplistic web based translator was proposed [71, 72]. The same group have worked on a Sinhala ontology generator for the purpose of machine translation [79] and a phrase level translator [81] based on the previous work on a multi-agent system for translation [78]. Further, an application of the English to Sinhala translator on the use case of selected text for reading was implemented [76]. They later continued their work on multi-agent English to Sinhala translation with the AGR organizational model [82].

Another group independently attempted English-to-Sinhala machine translation [394] with a statistical approach [395]. Wijerathna et al. [396] and De Silva et al. [397] have proposed simple rule based translators. An example-based method applied on the English-Sinhala sentence aligned government domain corpus was proposed by Silva and Weerasinghe [398]. A translator based on a look-up system was proposed by Vidanaralage et al. [399]. In a preprint, Joseph et al. [400] proposes an evolutionary algorithm for Sinhala to English translation with a basis of Point-wise Mutual Information (PMI) and claims that the code will be shared once the paper is accepted. However, they do not report any quantitative results to be compared and the reported qualitative results are also superficial. Pushpananda et al. [401] utilized statistical machine translation to translate between Sinhala and Tamil. Fernando et al. [402] tries to solve the Out of vocabulary (OOV) problem for Sinhala in the context of Sinhala-English-Tamil statistical machine translation. The approach proposed by Rajitha et al. [403] uses statistical machine translation and transliteration to align Sinhala and English documents.

3.18.2 Sinhala-English NMT

Fonseka et al. [404] used Byte Pair Encoding (BPE) for English to Sinhala neural machine translation. As another solution to the OOV problem, an analysis of subword techniques to improve English to Sinhala Neural Machine Translation (NMT) was conducted by Naranpanawwa et al. [405]. A data augmentation method to expand bilingual lexicon terms based on case-markers for the purpose of solving the OOV problem in the domain of NMT was proposed by Fernando et al. [406] which they later extended further [407]. Epaliyana et al. [408] proposed iterative filtering and data selection be used to improve Sinhala-English NMT. Perera et al. [409] used English Part-of-Speech (PoS) tags to improve English to Sinhala NMT. Lin et al. [410] used a model based on fairseq [411] to improve machine translation between English and Sinhala. The second half of the study by Arafath [310] dealt with translating Sinhala speech to other languages. Kugathasan and Sumathipala [412] proposed an NMT system for Sinhala-English Code-Mixed text using the standardized Sinhala Code-Mixed text they proposed earlier [413].

3.18.3 Singlish to Sinhala (Transliteration)

The XTREME-UP90 data set created by Ruder et al. [130] contains a Sinhala data set for the transliteration task. The early work by Goonetilleke et al. [414] attempted Sinhala transliteration through the Latin alphabet. However, their work does not use the word transliteration and instead focuses on the predictive aspect. Priyadarshani et al. [415] used statistical machine learning for transliteration of names between Sinhala, Tamil, and English. A rule-based system on trigrams was proposed by Liwera and Ranathunga [416] for Singlish to Sinhala transliteration of social media text. A Singlish to Sinhala converter which uses an LSTM was proposed by De Silva [417]. A rule-based approach for the same was proposed by de Silva and Ahangama [418]. The study by Nanayakkara et al. [419] introduced an English-to-Sinhala transliteration system. Sumanathilaka et al. [420, 421] proposed a Trie [422] data structure-based algorithm for word suggestion in Sinhala transliterations. An extended analysis of the same work was presented in a later work [423]. An LSTM-based sequence-to-sequence model was used by Sandaruwan et al. [424] for Singlish to English NMT task. Amarasekara et al. [425] proposed a rule-based method supported by N-gram analysis and a corpus dataset to transliterate Singlish tweets to Sinhala.

3.18.4 Between Sinhala and Non-English Languages

Most of the cross-Sinhala and Tamil work has been done in the domain of machine translation. A neural machine translation for Sinhala and Tamil languages was initiated by Tennage et al. [426, 427]. Then they further enhanced it with transliteration and byte pair encoding [428] and used synthetic training data to handle the rare word problem [429]. This project produced Si-Ti [430] a machine translation system of Sinhala and Tamil official documents. In the statistical machine translation front, Farhath et al. [431] worked on integrating bilingual lists. The attempts
by Weerasinghe [432] and Sripirakas et al. [433] were also focused on statistical machine translation while Jayakaran and Weerasinghe [434] attempted a kernel regression method. A yet another attempt was made by Pushpananda et al. [435] which they later extended with some quality improvements [436]. An attempt on real-time direct translation between Sinhala and Tamil was done by Rajpirathap et al. [437]. Dilshani et al. [438] have done a study on the linguistic divergence of Sinhala and Tamil languages in respect to machine translation. Mokanarang [439] claims to have built a named entity translator between Sinhala and Tamil for official government documents. But this work is locked behind an institutional repository wall and thus is not accessible by other researchers. Arukgoda et al. [440] studied the possibility of using deep learning techniques to improve Sinhala-Tamil translation which they further improved later [441]. Pramodya et al. [442] compared Transformers, Recurrent Neural Networks, and Statistical Machine Translation (SMT) in the context of Tamil to Sinhala machine translation. The work by Nissanka et al. [443] used monolingual word embedding to improve NMT between Sinhala and Tamil. Thilainathan et al. [444] uses pre-trained mBART models for six directional translations between Sinhala, Tamil, and English. Yashothara and Uthayasanker [445] discussed the use of the Hierarchical Phrase-Based Model for Tamil to Sinhala and Sinhala to Tamil translations. 

There have been attempts to link Sinhala NLP with Japanese by Herath et al. [22, 83, 84], Thelijjagoda et al. [85], Thelijjagoda et al. [446], and Kanduboda [9]. There has also been an attempt to use dictionary-based machine translation [447] between Sinhala and the liturgical language of Buddhism, Pali [448–450]. The study by Anuradha and Thelijjagoda [287] uses machine translation on the unique application of converting Sinhala and English Braille documents, which they have run OCR on, into voice.

### 3.19 Spelling and Grammar

The open-source data driven approach proposed by Wasala et al. [451, 452] claims to be able to check and correct spelling errors in Sinhala. The approach by Jayalatharachchi et al. [453] attempts to obtain synergy between two algorithms for the same purpose. These efforts [451, 453] were then extended by Subhagya et al. [454]. A rule-based Sinhala spell checker named SinSpell based on Hunspell [91] was introduced by Liyanapathirana et al. [455]. They have also made the tool available [92] online for use. The study by Sithamparanathan and Uthayasanker [456] extended the Generic Environment for context-aware spell correction to handle Sinhala and Tamil. Sonnadara et al. [116] created a benchmark data set for Sinhala spell correction along with a neural model. A multi agent-based spell checker, named LaSi Spell for Sinhala spell checking was introduced by Samarawickrama et al. [457]. The study by Udagedara et al. [458] specifically solved the problem of spell-checking Sri Lankan names and addresses. A system named Eroff was proposed by Sudesh et al. [459] to correct real-word errors in Sinhala text.

A model for detecting grammatical mistakes in Sinhala was developed by Pabasara and Jayalal [460]. They followed this up with a grammatical error detection and correction model [461]. Gunasekara et al. [462] used annotation projection for semantic role labelling for Sinhala. A Sinhala grammar checker based on Hidden Markov models was developed by Fernando and Arudchelvam [463]. Widjaratna [464] used a sequence-to-sequence model with attention, which is generally used for translation tasks, to translate sentences with common grammatical errors to their corrected counterparts. A rule-based system to convert Sinhala sentences from active voice to passive voice while correcting grammatical errors was proposed by Ilukkumbura and Rupasinghe [465]. The study by Goonawardena et al. [466] a rule based system to spell-check Sinhala text as well as detect and correct grammatical errors.

### 3.20 Chat Bots

A simple Sinhala chat bot which utilizes a small knowledge base has been proposed by Hettige and Karunamanda [67]. A study on the effect of word embeddings on a Sinhala chat bot was conducted by Gamage et al. [467] where they used, the fasttext model trained by Facebook [109–111], on a RASA [93] chat bot. A Sinhala chat bot for train information was proposed by Harshani [469]. Similarly, the tool proposed by Chandrasena et al. [470] serves as a chat bot-based recommendation system for Sri Lankan traditional dancers. The chat bot discussed by Kumanayake [471] has the very specific purpose of answering user inquiries about the degree programs at University of Ruhuna. Aishika et al. [472] used off the shelf RASA NLU Engine [468] and Microsoft Bot Network [473] to set up a generic Sinhala chat bot architecture. They demonstrated the effectiveness of their architecture by creating a food ordering chat bot. A web-based code-less chat bot development platform for Sinhala was proposed by Dissanayake et al. [474]. Further, they claimed that their system can handle OOV tokens as well as Sinhala-English code-switching. The work by Dasanayaka and Warajith [475] used a deep learning Intent Mapping (IM) model to map consumer responses in their Sinhala chat bot framework. Rajapakshe et al. [319] proposed a Sinhala conversational interface for the purpose of scheduling medical appointments and giving medical advice. The chat bot component was in the back end while the front end contained speech recognition and voice synthesizer components.

### 3.21 News and/or Social Media Recommendation

A trending topic detection model for Sinhala tweets using simple clustering and ranking algorithms was proposed by Jayasekara and Ahangama [476]. Sandamini et al. [477] proposed a post recommendation system, which supports Singlish, for social media. Tennakoon and Gamalath [478] proposes a hybrid system which uses skip-gram and collaborative Filtering on Multi-Layer Perceptron for recommending categorized Sinhala news articles. Tennakoon et al. [479] then extends the the system to also be able identify grey sheep users while preforming the aforementioned hybrid recommendation using LDA and SVM. Following the above

91. http://hunspell.github.io/
92. http://nlp-tools.uom.lk/sinspell/
93. https://rasa.com/
work, a news aggregator with news categorization, comment filtering, and two types of recommendation systems was proposed by Malsha et al. [480]. Madhushika et al. [481] analysed Twitter trending topics to understand how Sinhala Twitter data affects news dissemination on mass media. They proposed calculating a news value to a tweet which can be utilized to sort tweets by their news-worthiness in order to give better recommendations.

### 3.22 Sinhala Sign Language

In the domain of Sinhala sign language, Liyanaarachchi et al. [482] claims to have created a signing dataset for the Sinhala sign language however only the abstract of their work can be publicly accessible. DISSANAYAKE et al. [483] also claims to have created a data base of gestures that are included in Sinhala sign language along with a Sinhala sign language recognition system. Wijegoonarathna [484] has created a neural network-based approach for real-time Sinhala sign language gesture recognition. The approach by Hettiarachchi and Meegama [485], the approach by Dilakshan and Priyadarshana [486], as well as the approach by Peiris [487] have used CNN to recognize the fingerspelling alphabet of Sinhala sign language. Perera and Jayalal [488] also has used CNN to translate Sinhala sign Language to Sinhala text. However, they have also explicitly used Scale Invariant Feature Transform (SIFT) [489] features. A study for the limited use case of translating 15 Sinhala signs to text was conducted by Fernando and Wimalaratne [490]. Strides have been made in the domains of computer interpreting for written Sinhala [491] and animation of finger-spelled words and number signs [492].

### 3.23 Sinhala Braille

The work by de Silva and Liyanage [360] uses KNN, SVM, and a simple ANN system to recognize Sinhala Braille text. Vithanage [493] also claims to have created a conversion engine to easily convert the Braille text into the corresponding Sinhala text. The study by Madubashana [494], the study by Ariyaratna et al. [495] as well as the study by Weerasekara [496] also focused on creating an automated Braille to Sinhala recognition system. Anuradha and Thelijjagoda [287] uses OCR on Sinhala and English Braille documents on which they then run a machine translation system in order to convert them into voice.

### 3.24 Plagiarism Detection

An extremely simple plagiarism detection tool which only uses n-grams of simply tokenized text was proposed by Basnayake et al. [497]. Another simple plagiarism detection tool that uses synonymy and Hyponymy-Hypernymy (which they call Generalization in the paper) was attempted by Rajamanthri and Thelijjagoda [498]. They later extended this work [499] to propose a more advanced plagiarism detection tool which uses Internet resources. Kasthuri Arachchi and Charles [500] proposed using Word2Vec vector cosign similarity to detect plagiarism. A multi-document Sinhala similarity detection tool to detect plagiarism was proposed by Piyaratna [501]. Punchiweva et al. [502] developed a character-level model which can identify the author for Sinhala text in student answers.

### 3.25 Sinhala-English Code-Mixing

The problem of recognizing Sinhala and English code-mixed data where the Sinhala text is written in Singlish was explored by Smith and Thayasivam [503] and later by Smith [504] using an XGB classifier and a CRF model building on their previous work [505], which analysed such data. Shannmugalangam and Sumathipala [506] also attempted to identify the language in Sinhala-English code-mixed text using Support Vector Machines (SVM), Naive Bayes, Logistic Regression, Random Forest, and Decision Trees. A dictionary based approach to standardize Sinhala Code-Mixed text was proposed by Kugathasan and Sumathipala [413]. They later used it for NMT in Sinhala-English Code-Mixed text [412]. As discussed in Section 3.10, Rathnayake et al. [207] used adapter-based [208–213] fine-tuning on XLM-R [214], for classifying code-mixed and code-switched Sinhala text. Hettigoda [216] classified English-Sinhala code-mixed comments from Facebook. As discussed in Section 3.11, Aththanayaka and Naleer [235] utilized traditional machine learning methods for sentiment analysis on Sinhala-English code-mixed text from social media. Chathuranga and Ranathunga [218] used capsule based methods to classify Sinhala-English code-mixed data. Dissanayake et al. [474] claim that the web-based code-less chat bot development platform for Sinhala proposed by them is capable of handling Sinhala-English code-switching. The feature set derived by Walawage and Ranathunga [359] for text on social media images (memes) attempts to separately identify Sinhala and English text.

### 3.26 Miscellaneous Applications

In this section, we discuss NLP tools and research which are either hard to categorize under above sections or reasonably involve multiples of them.

Dissanayake and Hettige [507] implemented a question and answer generator for Sinhala with the limited PoS: pronouns, adjectives, verbs, and adverbs. Jayakody et al. [508] uses simple KNN and SVM methods on a PoS tagged Sinhala corpus to create a question-answering system which they name Mahoshadha. On the other hand, Amarasinghe and Ranathunga [509] used an ontology-based approach to generate Sinhala essay questions. The work by Liyanage and Ranathunga [510, 511] attempt to use LSTMs for mathematical word problem generation in Sinhala and other languages. The follow-up work by Niyarepora et al. [512] used pre-trained mBART and mT5 models for the same task and showed better results. The study by Kao and Ilmini [513] used LSTM to generate Sinhala lyrics. Bandara et al. [514] introduced Sibil AI, a GTP-2 based Children’s story generator for Sinhala. However, they did not train a GTP-2 model in Sinhala. Instead, they trained an English GTP-2 model and generated stories in English which were then translated to Sinhala using Google translate API.

Fernando [515] proposed a method for inexact matching of Sinhala proper names. A study on determining canonical word order of colloquial Sinhala sentences using priority information was conducted by Kanduboda and Tamaoka [516] which they later extended [517–519].

A dataset consisting of Sinhala documents drawn from Sri Lankan news websites was published by Jayawickrama
et al. [115] along with the benchmark misinformation classification models. A hybrid approach to detect Sinhala fake news on Social media was proposed by Wijayarathna and Jayalal [520] where the text content of the post is checked against credible sources and the authenticity of the user account posting the relevant post is evaluated by a rule-based points allocation schema. Wijayarathna and Jayalal [521] collected a set of 120 fake news tweets and 250 non-fake news tweets which they then converted to vectors by taking the fasttext vector for words and averaging. The vector representations of the tweets were then compared to predict whether the news containing it is real or fake. The study by Udurawana et al. [522] proposes to use an accuracy score (obtained by analysing the news content) and a credibility score (obtained by using a scoring mechanism) to detect fake news in Sinhala text. They also incorporate a module that classifies on the basis of passive aggressiveness. A cross-language information retrieval system where Sinhala search queries are converted to English search queries by mapping Sinhala word embeddings to English word embeddings was proposed by Hisan et al. [523].

Sandathara et al. [524] proposed a system which they named Arunalu that they claimed to use Voice recognition, Natural Language Processing, Machine Learning, and Deep Learning concepts to help individuals with dyslexia overcome problems of reading Sinhala. The learning bot MiMi proposed by Vithana et al. [525] assists children to learn to speak without stuttering. Rajitha et al. [403] has proposed a statistical machine translation and transliteration method to align Sinhala and English documents. Rajitha et al. [526] used the data set that they introduced in their previous work [94] to prove that task specific supervised distance learning metrics outperform their unsupervised counterparts, for document alignment. Fernando et al. [125] have used pre-trained multilingual language models (PMLMs) to improve document and sentence alignment between Sinhala–English, Tamil–English, and Sinhala–Tamil language pairs.

Kumari and Hettiarachchi [527] proposed an algorithm for Sinhala topic modelling based on LDA [528] and RAKE [529]. Batawalarachchi [530] proposed two methods using statistical features to select words to be included in Sinhala document titles. Arambewela et al. [531] proposed a Sinhala writing assistant tool utilising CNNS. Jayaweera et al. [532] used classical machine learning methods to propose dynamic stop word removal. They claim to have released a corpus of 100,000+ Sinhala documents in their paper. But they provide no information on where to obtain this corpus. A two-tiered model to embed Sinhala sentences was proposed by Weeraprameshwara et al. [533]. According to their results, Seq2Seq GRU with attention run on fastText word embedding obtains the best results for Sinhala sentence embedding. Liyanage et al. [534] have created a small Sinhala treebank of 100 sentences. Burchell et al. [132], show that they obtain better results for the LID task for most of the languages compared to NLLB [123]. However, for Sinhala the results by Burchell et al. [132] are the same as that of NLLB. Minixhofer et al. [535] have used Sinhala from OPLIS100 [120] dataset to pre-train their multilingual punctuation-agnostic sentence segmentation model. Given that Sinhala is not one of their main target languages, they do not report sentence segmentation results for Sinhala in much detail. However, they do report that their method improves Sinhala sentence splitting F1 score to 86.0 above their SpaCy baseline of 75.8. The study by Petrov et al. [536] compared the performance 17 off-the-shelf language tokenizers on a large number of languages including Sinhala. They have found that some languages such as English have shorter tokenizing lengths (close to 1.0) across the board while some can raise to undesirable heights (they observe 15.0 as the highest). They report undesirable numbers for Sinhala, mostly in the 8.83 to 12.86 range. They report reasonable results on MBart50 [537] and mT5 [538, 539] with 1.35 and 1.66 respectively. The best result (1.00), is given by CANINE [540]. A meta-study on how language data from social media are used in research was conducted by Hewapathirana [541]. A Sinhala programming language named Helaa based on Java was proposed by Yasasri and Karunarathna [542].

4 PRIMARY SOURCES

Even though the main objective of this survey is to cover NLP tools and research, we noticed that much of these NLP tools and research depend on primary sources of Sinhala language such as printed books in the role of knowledge sources and ground truth. Therefore, for the benefit of other researchers who venture into Sinhala NLP, we decided to add a short introduction to the available primary sources of Sinhala language used by their peers. We note that the body of work by a single scholar, Disanayake [2, 40, 543–553], is quite prominent in the case of being used for NLP applications. For formal introduction of the language, the books by Disanayake [2] and Perera [3] are commonly used. In cases which deal with the Sinhala alphabet, the introduction by Indrasena [554] and by Disanayake [543, 544] have been used. An analysis of modern Sinhala linguistics has been done by Jayathilake [555] and by Pallatthara and Weihene [41]. The early study by Henadeerage [556] covers a number of topics on the Sinhala language such as grammatical relations, argument structure, phrase structure and focus constructions.

As we discussed in Section 3.3, a number of printed Sinhala dictionaries exist, Malalasekera [134] being the most prominent English-Sinhala dictionary among them. In addition to that seminal work, previous researchers of Sinhala NLP have utilized a number of other dictionaries of various configurations such as: English-Sinhala [136, 139, 140], Sinhala-Sinhala [135, 138], and English-Sinhala-Tamil [137]. A number of NLP applications have utilized first sources intended to teach children [557–561] or foreigners [177, 562, 563]44. This makes sense given that an introduction written for children would start with basic principles and thus be ideal for crafting rule based NLP systems and an introduction written for foreigners would have Sinhala language described in terms of English, making easy the process of rule based translation of English NLP tools to Sinhala.

For applications where a rule based approach for Sinhala spelling correction is utilized, the books by Disanayake [550, 551], by Koparahewa [564], and by Gair and Karunatilleke [565] are used to provide a basis. A number of NLP
applications which handle spoken Sinhala in the capacity of phonological layer (Section 3.16) applications or otherwise, make note of the fact that spoken Sinhala is considerably distinguishable from written Sinhala, as such, they refer primary sources which explicitly deal with spoken Sinhala [40, 549, 553, 559, 562, 566, 567].

Primary sources used in NLP application for Sinhala grammar are varied. A number of them provide overviews of the entirety of Sinhala grammar [20, 41, 44, 552, 568–577]. There are specific primary sources focusing on verbs [548, 560, 578], nouns [547, 561], prepositions [559], compounds [546], derivation [545], case system [579], and sentence structure [45] of the Sinhala language. The book by Rajapaksha [580] is commonly used in NLP applications as a guide for word tagging and punctuation mark handling. NLP studies that tackle the hard problem of handling questions expressed in Sinhala often refer to the book by Kariyakarawana [581]. Kekulawala [582] has aptly discussed the much controversial topic of the situation of future tense of Sinhala.

5 Conclusion

At this point, a reader might think, there seems to be a significant number of implementations of NLP for Sinhala. Therefore, how can one justify listing Sinhala as a resource poor language? The important point which is missing in that assumption is that in the cases of almost all of the above listed implementations and findings, the only thing that is publicly available for a researcher is a set of research papers. The corpora, tools, algorithm, and anything else that were discovered through these research are either locked away as properties of individual research groups or worse lost to the time with crashed ancient servers, lost hard drives, and expired web hosts. This reason and probably academic/research rivalry have caused these separate research groups not to cite or build upon the works of each other. In many cases where similar work is done, it is a re-hashing on the same ideas adopted from resource rich languages because of, the unavailability of (or the reluctance to), referring and building on work done by another group (Refer Fig. 7 in Appendix B). This has resulted in multiple groups building multiple foundations behind closed doors but no one ending up with a completed end-to-end NLP work-flow. In their analysis of low resourced languages, Ranathunga and de Silva [583] observed that only 11.43% of Sinhala NLP papers have released the relevant data sets. Further, according to them, code being released sits at 9.71% while tools being released sits at 5.71%. While the 11.43% figure of data release may induce a feeling of availability, Ranathunga and de Silva [583] further observes that it is only 1.14% that has been released in public repositories. Research publications promising access to data and code only to be found lacking later is a common academic shortfall according to Gabelica et al. [36] however, given that Sinhala NLP is already having minimal work done as a whole compared to some other languages, we simply cannot afford to lose any of the generated data or code. In conclusion, what can be said is that, even though there are islands of implementations done for Sinhala NLP, they are of very small scale and/or are usually not readily accessible for further use and research by other researchers. Thus, so far, sadly, Sinhala stays a resource poor language.

Acknowledgments

The authors would like to thank Romain Egele for checking the examples we have provided in French for their accuracy. Similarly, the authors would also like to thank Shrvan Kale for checking the examples we have provided in Hindi for their accuracy.

References

[1] R. Englebretson and C. Genetti, “Santa barbara papers in linguistics: Proceeding from the workshop on sinhala linguistics,” Santa Barbara, CA: Department of Linguistics at the University of California, Santa Barbara, 2005.
[2] J. B. Disanayake, National Languages of Sri Lanka: Sinhala. Department of cultural affairs, 1976.
[3] T. G. Perera, Sinhala Bhashawa, 1983.
[4] L. Bauer, Linguistics Student’s Handbook. Edinburgh University Press, 2007.
[5] Department of Census and Statistics Sri Lanka. Percentage of population aged 10 years and over in major ethnic groups by district and ability to speak sinhala, tamil and english languages. [Online]. Available: https://goo.gl/nmVZsd
[6] J. W. Gair and W. Karunatilaka, Literary Sinhala. ERIC, Cornell University, New York, 1974.
[7] Department of Census and Statistics, Sri Lanka. (2012) Census of Population and Housing of Sri Lanka. [Online]. Available: https://bit.ly/3bAgXe
[8] H. Young. A language family tree - in pictures — education — the guardian. [Online]. Available: https://www.theguardian.com/education/gallery/2015/jan/23/a-language-family-tree-in-pictures
[9] A. B. Kanduboda, “The role of animacy in determining noun phrase cases in the sinhalese and japanese languages,” Science of words, vol. 24, pp. 5–20, 2011.
[10] P. Fernando, “Palaeographical development of the brahmi script in ceylon from 3rd century bc to 7th century ad,” 1949.
[11] D. Bandara, N. Warnajith, A. Minato, and S. Ozawa, “Creation of precise alphabet fonts of early brahmi script from photographic data of ancient sri lankan inscriptions,” Canadian Journal on Artificial Intelligence, Machine Learning and Pattern Recognition, vol. 3, no. 3, pp. 33–39, 2012.
[12] P. T. Daniels and W. Bright, The world’s writing systems. Oxford University Press on Demand, 1996.
[13] M. H. Sirisoma, “Brahmi inscriptions of sri lanka from 3rd century bc to 65 ad,” pp. 3–54, 1990.
[14] M. Dias, “Lakdiwa sellipawalin heliwaana sinhala bhashaw prathyartha namayange vikashanaya,” Department of Archaeology, Colombo Sri Lanka, p. 1, 1996.
[15] A. S. Hettiarachchi, “Investigation of 2nd, 3rd and 4th century inscriptions,” Inscriptions: Volume Two, Archaeological Department Centenary (1890–1990), Commemorative Series. Colombo: Department of Archaeology, pp. 57–104, 1990.
[16] S. Paranavitana and S. L. P. Departamemtuva, Inscriptions of Ceylon. Dept. of Archaeology, 1970.
[17] R. Salmonon, Indian epigraphy: a guide to the study of inscriptions in Sanskrit, Prakrit, and the other Indo-Aryan languages. Oxford University Press, 1998.
[18] H. Falk, Schrift im alten Indien: ein Forschungsbericht mit Anmerkungen.. Gunter Narr Verlag, 1993, vol. 56.
[19] B. Hettinger and A. S. Karunananda, “Computational model of grammar for english to sinhala machine translation,” in Advances in ICT for Emerging Regions (ICTer), 2011 International Conference on. IEEE, 2011, pp. 26–31.
[20] A. M. Gunasekara, A Comprehensive Grammar of the Sinhalese Language. Asian Educational Services, New Delhi, Madras, India, 1986.
[21] H. P. Ray, The archaeology of seafaring in ancient South Asia. Cambridge University Press, 2003.
[22] A. Herath, Y. Hyodo, Y. Kawada, T. Ikeda, and S. Herath, “A practical machine translation system from japanese to modern sinhalese,” Giftu University, pp. 153–162, 1994.
D. Sachintha, L. Piyarathna, C. Rajitha, and S. Ranathunga, “Sinhala language corpora and D. Warusawithana, N. Kulaweera, L. Weerasinghe, and D. Lakmal, S. Ranathunga, S. Peramuna, and I. Herath, “Word

I. Caswell, J. Kreutzer, L. Wang, A. Wahab, D. van Esch, N. Ulzii-Orshikh, A. Tapo, N. Subramani, A. Sokolov, C. Sikasote, M. Setyawan, S. Sarin, S. Samb, B. Sagot, C. Rivera, A. Rios, I. Papadimitriou, S. Osei, P. J. O. Suarez, I. Orife, K. Ogueji, R. A. Niyongabo, T. Q. Nguyen, M. Müller, A. Müller, S. H. Mahmod, N. Muhammad, A. Mnyakeri, J. Mirzakhailov, T. Mataniz Rojas, L. Pla Sempere, G. Ramírez-Sánchez, E. Sarriàs, M. Strebra, B. Thompson, W. Waite, D. Wiggins, and J. Zaragoza, “Paracrawl: Web-scale acquisition of parallel corpora,” in Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 4555–4567.

I. Caswell, J. Kreutzer, L. Wang, A. Wahab, D. van Esch, N. Ulzii-Orshikh, A. Tapo, N. Subramani, A. Sokolov, C. Sikasote, M. Setyawan, S. Sarin, S. Samb, B. Sagot, C. Rivera, A. Rios, I. Papadimitriou, S. Osei, P. J. O. Suarez, I. Orife, K. Ogueji, R. A. Niyongabo, T. Q. Nguyen, M. Müller, A. Müller, S. H. Mahmod, N. Muhammad, A. Mnyakeri, J. Mirzakhailov, T. Mataniz Rojas, L. Pla Sempere, G. Ramírez-Sánchez, E. Sarriàs, M. Strebra, B. Thompson, W. Waite, D. Wiggins, and J. Zaragoza, “Paracrawl: Web-scale acquisition of parallel corpora,” in Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 4555–4567.

D. Warusawithana, N. Kulaweera, L. Weerasinghe, and B. Karunarathne, “A systematic approach to derive a refined speech corpus for sinhala,” 2022.

O. Sartinjua, L. Piyarathna, C. Rajitha, and S. Ranathunga, “Exploiting parallel corpora to improve multilingual embedding based document and sentence alignment,” arXiv preprint arXiv:2106.06776, 2021.

D. Warusawithana, N. Kulaweera, L. Weerasinghe, and B. Karunarathne, “A systematic approach to derive a refined speech corpus for sinhala,” 2022.

O. Sartinjua, L. Piyarathna, C. Rajitha, and S. Ranathunga, “Exploiting parallel corpora to improve multilingual embedding based document and sentence alignment,” arXiv preprint arXiv:2106.06776, 2021.

D. Warusawithana, N. Kulaweera, L. Weerasinghe, and B. Karunarathne, “A systematic approach to derive a refined speech corpus for sinhala,” 2022.

O. Sartinjua, L. Piyarathna, C. Rajitha, and S. Ranathunga, “Exploiting parallel corpora to improve multilingual embedding based document and sentence alignment,” arXiv preprint arXiv:2106.06776, 2021.

D. Warusawithana, N. Kulaweera, L. Weerasinghe, and B. Karunarathne, “A systematic approach to derive a refined speech corpus for sinhala,” 2022.

O. Sartinjua, L. Piyarathna, C. Rajitha, and S. Ranathunga, “Exploiting parallel corpora to improve multilingual embedding based document and sentence alignment,” arXiv preprint arXiv:2106.06776, 2021.

D. Warusawithana, N. Kulaweera, L. Weerasinghe, and B. Karunarathne, “A systematic approach to derive a refined speech corpus for sinhala,” 2022.

O. Sartinjua, L. Piyarathna, C. Rajitha, and S. Ranathunga, “Exploiting parallel corpora to improve multilingual embedding based document and sentence alignment,” arXiv preprint arXiv:2106.06776, 2021.
International Conference on Advances in ICT for Emerging Regions (ICICTer2012). IEEE, 2012, pp. 3–6.

[294] W. Manamperi, D. Karunathilaka, T. Madhushani, N. Galagedara, and D. Dias, “Sinhala speech recognition for interactive voice response systems accessed through mobile phones,” in 2018 Moratuwa Engineering Research Conference (MERCon). IEEE, 2018, pp. 241–246.

[295] N. Prasangini and H. Nagahamulla, “Sinhala speech to sinhala unicode text conversion for disaster relief facilitation in sri lanka,” in 2018 IEEE International Conference on Information and Automation for Sustainability (ICIAS), 2018, pp. 1–6.

[296] P. G. N. Priyadarshani, “Speaker dependent speech recognition on a selected set of sinhala words,” 2012.

[297] P. G. N. Priyadarshani, N. G. J. Dias, and A. Punchihewa, “Dynamic time warping based speech recognition for isolated sinhala words,” in 2012 IEEE 55th International Midwest Symposium on Circuits and Systems (MWSCAS). IEEE, 2012, pp. 892–895.

—. “Genetic algorithm approach for sinhala speech recognition,” in 2012 IEEE 55th International Midwest Symposium on Circuits and Systems (MWSCAS). IEEE, 2012, pp. 896–899.

[299] P. G. N. Priyadarshani and N. G. J. Dias, “Automatic segmentation of separately pronounced sinhala words into syllables,” 2011.

[300] M. R. K. Gunasekara and R. G. N. Meegama, “Real-time translation of discrete sinhala speech to unicode text,” in 2015 Fifteenth International Conference on Advances in ICT for Emerging Regions (ICTer). IEEE, 2015, pp. 140–145.

[301] M. Punchimudiyanse and R. G. N. Meegama, “Unicode sinhala and phonetic english bi-directional conversion for sinhala speech recognizer,” in 2015 IEEE 10th International Conference on Industrial and Information Systems (ICIIS). IEEE, 2015, pp. 293–298.

[302] Y. Karunanayake, U. Thayasivam, and S. Ranathunga, “Transfer learning based free-form speech command classification for low-resource languages,” in Proceedings of the 57th Conference of the Association for Computational Linguistics: Student Research Workshop, 2019, pp. 288–294.

[303] K. A. D. C. Dilshan, “Transcribing number sequences in continuous sinhala speech,” 2018.

[304] B. Gamage, R. Pushpananda, R. Weerasinghe, and T. Nadungodage, “Usage of combinational acoustic models (dnn-hmm and sgmm) and identifying the impact of language models in sinhala speech recognition,” in 2020 20th International Conference on Advances in ICT for Emerging Regions (ICTer). IEEE, 2020, pp. 17–22.

[305] D. Giuliani and B. BabaAli, “Large vocabulary children’s speech recognition with dnn-hmm and sgmm acoustic modeling,” in Sixteenth Annual Conference of the International Speech Communication Association, 2015.

[306] R. Pushpananda, T. Nadungodage, and R. Weerasinghe, “Improving sinhala speech recognition through e2e Lf-mmi model,” in Proceedings of the 18th International Conference on Natural Language Processing (ICON), 2021, pp. 213–219.

[307] V. Manohar, H. Hadian, D. Povey, and S. Khudanpur, “Semi-supervised training of acoustic models using lattice-free mmi,” in 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2019, pp. 892–895.

[308] A. Ignatius and U. Thayasivam, “Speaker-invariant speech-to-text classification for low-resource languages,” in International Conference on Speech and Computer. Springer, 2021, pp. 279–290.

[309] H. Yadav, A. Gupta, S. K. Kallabandi, A. W. Black, and R. R. Shah, “Intent classification using pre-trained embeddings for low-resource languages,” arXiv preprint arXiv:2110.09264, 2021.

[310] T. Dinushika, L. Kavmini, P. Abeyawardhana, U. Thayasivam, and S. Jayasena, “Speech command classification system for sinhala language based on automatic speech recognition,” in 2019 International Conference on Asian Language Processing (IALP). IEEE, 2019, pp. 205–210.

[311] L. Kavmini, T. Dinushika, U. Thayasivam, and S. Jayasena, “Improved speech command classification system for sinhala language based on automatic speech recognition,” International Journal of Asian Language Processing, p. 2050009, 2020.

[312] K. T. Welarathna, V. Kulasekara, K. Pulasinghe, and V. Piyawardana, “Automated sinhala speech emotions analysis tool for autism children,” in 2021 10th International Conference on Information and Automation for Sustainability (ICIAS). IEEE, 2021, pp. 500–505.

[313] K. H. I. R. Senarathe, J. M. I. Nirash, H. M. C. P. Herath, V. D. Bandara, D. Wijendra, and J. Krishara, “Automated sinhala voice assistant to manage tasks using natural language processing - voice,” in 2022 3rd International Informatics and Software Engineering Conference (IISEC). IEEE, 2022, pp. 1–5.

[314] C. Weerathunga, R. Weerasinghe, and D. Sandaruwan, “Lip synchronization modeling for sinhala speech,” in 2020 20th International Conference on Advances in ICT for Emerging Regions (ICTer). IEEE, 2020, pp. 208–213.

[315] W. G. V. K. Wakkumbura, R. A. H. Madhubhashana, P. M. K. A. W. S. Koon, W. G. C. W. D. Koon, and M. N. A. F. Koon, “Phoneme viseme mapping for sinhala speaking robot for sri lanka health care applications,” in 2022 IEEE 4th Eurasia Conference on Biomedical Engineering, Healthcare and Sustainability (ECBIOS). IEEE, 2022, pp. 258–262.

[316] R. K. Rajapakse, A. R. Weerasinghe, and E. K. Seneviratne, “A neural network based character recognition system for sinhala script,” Department of Statistics and Computer Science, University of Colombo, 1995.

[317] H. L. Premaratne and J. Bigun, “Recognition of printed sinhala characters using linear symmetry,” in The 5th Asian Conference on Computer Vision, 2002, pp. 23–25.

[318] A. Ignatius, “A segmentation-free approach to recognise printed sinhala script using linear symmetry,” Pattern recognition, vol. 37, no. 10, pp. 2081–2089, 2004.

[319] H. L. Premaratne, E. Järpe, and J. Bigun, “Lexicon and hid-
den markov model-based optimisation of the recognised sinhala script,” Pattern recognition letters, vol. 27, no. 6, pp. 696–705, 2006.

[333] S. Hewavitharana, H. C. Fernando, and N. D. Kodikara, “Off-line sinhala handwriting recognition using hidden markov models.” in Proc. of the International Information Technology Conference (IITC), Colombo, Sri Lanka, 2002.

[334] S. Hewavitharana and N. D. Kodikara, “A statistical approach to sinhala handwriting recognition,” in Proc. of the International Information Technology Conference (IITC), Colombo, Sri Lanka, 2002.

[334] S. Ajward, N. Jayasundara, S. Madushika, and R. Ragel, “Converting printed sinhala documents to formatted editable text,” in 2010 Fifth International Conference on Information and Automation for Sustainability (IEEE, 2010), pp. 138–143.

[335] P. T. C. Madushanka, R. Bandara, and L. Ranathunga, “Sinhala handwritten character recognition by using enhanced thinning and curvature histogram based method,” in 2017 IEEE 2nd International Conference on Signal and Image Processing (ICSIP). IEEE, 2017, pp. 46–50.

[336] M. L. M. Karunanyaka, N. D. Kodikara, and G. D. S. P. Wimalaratne, “Off line sinhala handwriting recognition with an application for postal city name recognition,” IJIC 2004, 2004.

[337] R. Weerasinghe, A. Wasala, D. Herath, and V. Welgama, “Nlp applications of sinhala: Tts & ocr,” in Proceedings of the Third International Joint Conference on Natural Language Processing: Volume II, 2004. IEEE.

[338] A. R. Weerasinghe, D. L. Herath, and N. P. K. Medagoda, “A nearest-neighbor based algorithm for printed sinhala character recognition,” Innovations for a Knowledge Economy, p. 11, 2006.

[339] A. Weerasinghe, D. L. Herath, and N. Medagoda, “A knn based algorithm for printed sinhala character recognition,” in Proceedings of 8th International Information Technology Conference, 2006.

[340] D. L. Ediriweera, “Improving the accuracy of the output of sinhala ocr by using a dictionary,” Ph.D. dissertation, University of Moratuwa Sri Lanka, 2012.

[341] G. Dias, T. N. P. Patikirikorala, C. I. Arambewela, R. P. M. Darshana, and N. D. Alahendri, “Sinhala optical character recognition for desktops,” 2013.

[342] G. Dias, T. N. P. Patikirikorala, C. I. Arambewela, R. P. M. Darshana, and N. D. Alahendri, “Online sinhala handwritten character recognition for desktops,” 2013.

[343] M. H. P. Ranmuthugala, G. D. N. C. Pathiragoda, S. H. C. Jayasundara, G. Dias, and A. S. Karunanyanda, “Online sinhala handwritten character recognition on handheld devices,” Innovations for a Knowledge Economy, p. 1, 2006.

[344] M. Rimas, R. P. Thilakumara, and P. Koswatta, “Optical character recognition for sinhala language,” in 2013 IEEE Global Humanitarian Technology Conference: South Asia Satellite (GHTC-SAS). IEEE, 2013, pp. 149–153.

[345] G. I. Gunawardana, M. A. P. Chamikara, and R. G. Ragel, “A fuzzy based model to identify printed sinhala characters,” in 7th International Conference of Information for Automation and Information for Sustainability, IEEE, 2014, pp. 1–6.

[346] H. W. H. Premachandra, C. Premachandra, T. Kimura, and H. Kawanaka, “Artificial neural network based sinhala character recognition,” in International Conference on Computer Vision and Graphics, Springer, 2016, pp. 594–603.

[347] J. M. H. M. Jayamaha and H. M. M. Naleer, “Feature extraction technique based character recognition using artificial neural network for sinhala characters,” 2016.

[348] T. N. Kumara and R. Ragel, “A systematic feature selection process for a sinhala character recognition system,” in 2016 IEEE International Conference on Information and Automation for Sustainability (I4SAS). IEEE, 2016, pp. 1–6.

[349] B. R. Jayawickrama, L. Ranathunga, K. L. Mahalijanaaraachchi, L. G. B. Subhagy, and W. H. A. Nimasha, “Letter segmentation and modifier detection in printed sinhala signature,” in 2018 18th International Conference on Advances in ICT for Emerging Regions (ICTer). IEEE, 2018, pp. 203–208.

[350] S. Gunawardhana and L. Ranathunga, “Segmentation and identification of presence of sinhala characters in facebook images,” in 2018 IEEE 13th International Conference on Industrial and Information Systems (ICIIS). IEEE, 2018, pp. 78–82.

[351] K. L. N. D. Liyanage, “Improving Sinhala OCR using Deep Learning,” 2018.

[352] I. Anuradha, C. Liyanage, and R. Weerasinghe, “Estimating the effort of off-line recognition when resolution and algorithmic complexity are increased,” International Journal on Advances in ICT for Emerging Regions (ICTer), vol. 14, no. 3, 2021.

[353] I. Anuradha, C. Liyanage, H. Wijayawardhana, and R. Weerasinghe, “Deep learning based sinhala optical character recognition (ocr),” in 2020 20th International Conference on Advances in ICT for Emerging Regions (ICTer). IEEE, 2020, pp. 289–299.

[354] R. Smith, “Overview of the tesseract ocr engine,” in Ninth international conference on document analysis and recognition (ICDAR 2007), vol. 2. IEEE, 2007, pp. 629–633.

[355] B. P. K. Balasooriya, “Improving and Measuring OCR Accuracy for Sinhala with Tesseract OCR Engine,” Ph.D. dissertation, 2021.

[356] Y. Maduranga and S. Jayalal, “Multi-style printed sinhala character recognition and digitalization using artificial neural network,” in 2022 2nd International Conference on Advanced Research in Computing (ICARC). IEEE, 2022, pp. 120–124.

[357] S. Samarajeewa and L. Ranathunga, “An approach for resolving double character segmentation in sinhala social media text images,” in 2020 From Innovation to Impact (FITI), vol. 1. IEEE, 2020, pp. 1–6.

[358] N. M. T. de Silva and S. R. Liyanage, “Sinhala braille character recognizer,” in Proceedings of the Fifth International Conference on Pattern Recognition, 2008.

[359] S. Chanda, S. Pal, and U. Pal, “Word-wise sinhala tamil and english script identification using gaussian kernel svm,” in 2008 19th International Conference on Pattern Recognition. IEEE, 2008, pp. 1–4.

[360] H. C. Fernando, N. D. Kodikara, and S. Hewawitharana, “A database for handwriting recognition research in sinhala language,” in Proceedings of the International Conference on Information and Automation, 2005.

[361] M. L. M. Karunanyaka, C. A. Marasinghe, and N. D. Kodikara, “Thresholding, noise reduction and skew correction of sinhala handwritten words,” in MVA, 2005, pp. 355–358.

[362] B. Jayasekara and L. Udawatta, “Non-cursive sinhala handwriting script recognition: A genetic algorithm based alphabet training approach,” in Proceedings of the International Conference on Information and Automation, 2005.

[363] N. P. T. I. Nilaweera, H. L. Premaratne, and D. U. J. Sonnadara, “Comparison of projection and wavelet based techniques in recognition of sinhala handwritten scripts,” in Proceedings of the 25th National IT Conference, 2007.

[364] C. Silva and K. Kariyawasam, “Segmenting sinhala handwritten characters,” International Journal of Conceptions on Computing and Information Technology, vol. 2, no. 4, pp. 22–26, 2014.

[365] C. M. Silva, N. D. Jayasundere, and C. Kariyawasam, “State of handwriting recognition of modern sinhala script,” 2014.

[366] —, “Contour tracing for isolated sinhala handwritten character recognition,” in 2015 Fifteen International Conferences on Advances in ICT for Emerging Regions (ICTer). IEEE, 2015, pp. 25–31.

[367] K. A. K. N. D. Dharmapala, W. P. M. V. Wijesooriya, C. P. Chandrasekara, U. K. A. U. Rathnapriya, and L. Ranathunga, “Sinhala handwriting recognition mechanism using zone based feature extraction,” 2017.

[368] K. S. A. Walawage and L. Ranathunga, “Segmentation of overlapping and touching sinhala handwritten characters,” in 2018 3rd International Conference on Information Technology Research (ICTIR). IEEE, 2018, pp. 1–6.

[369] K. S. A. Walawage, “Segmentation of overlapping sinhala handwritten characters,” Ph.D. dissertation, 2019.

[370] C. M. Silva and N. D. Jayasundere, “Character modifier combination in sinhala character recognition,” in 2016 IEEE International Conference on Information and Automation for Sustainability (I4SAS). IEEE, 2016, pp. 1–6.

[371] J. Mariyathas, V. Shanmuganathan, and B. Kuhaneswaran, “Sinhala handwriting character recognition,” in Proceedings of the Third International Conference on Advances in ICT for Emerging Regions (ICTer). IEEE, 2013, pp. 355–358.

[372] B. Jayaweera, I. Anuradha, and S. Liyanage, “Improving Industrial and Information Systems (ICIIS). IEEE, 2018, pp. 78–82.

[373] K. L. N. D. Liyanage, “Improving Sinhala OCR using Deep Learning,” 2018.

[374] I. Anuradha, C. Liyanage, and R. Weerasinghe, “Estimating the effort of off-line recognition when increased resolution and algorithmic complexity are increased,” International Journal on Advances in ICT for Emerging Regions (ICTer), vol. 14, no. 3, 2021.
Priyadarshana, “Sinhala word suggestion algorithm for ad hoc romanized sinhala transliterations using a trie.” 2023.

T. G. D. K. Sumanathilaka, R. Weerasinghe, and Y. H. P. P. Priyadarshana, “Swa-bhasha: Romanized sinhala to sinhala reverse transcription using a hybrid approach,” in 2023 3rd International Conference on Advanced Research in Computing (ICARC). IEEE, 2023, pp. 136–141.

F. Bodon and L. Rónyai, “Trie: an alternative data structure for data mining algorithms,” Mathematical and Computer Modelling, vol. 38, no. 7-9, pp. 739–751, 2003.

T. G. D. K. Sumanathilaka, “Romanized sinhala to sinhala reverse transcription using a hybrid approach,” Ph.D. dissertation, 2023.

D. Sandaruwan, S. Fernando, and S. Sumathipala, “Neural machine translation approach for singlish to english translation,” The International Journal on Advances in ICT for Emerging Regions, vol. 14, no. 03, pp. 36–42, 2021.

M. D. C. Amarasekara, R. A. D. P. Rajapaksha, H. M. D. T. Jayarathna, H. M. G. K. Karunarathna, I. T. S. Pyatilek, and C. P. Wijesiriwardana, “Developing a system to transliterate singlish twitter posts to sinhala,” 2023.

P. Tennyse, P. Sandaruwan, M. Thilakarathne, A. Herath, S. Ranathunga, S. Jayasena, and G. Dias, “Neural machine translation for sinhala and tamil languages,” in Asian Language Processing (ALP), 2017 International Conference on. IEEE, 2017, pp. 189–192.

P. N. Tennyse, M. W. D. P. Sandaruwan, J. K. M. M. Thilakarathne, A. N. Herath, S. Ranathunga, S. Jayasena, and G. Dias, “Neural machine translation for sinhala-tamil,” 2017.

P. Tennyse, A. Herath, M. Thilakarathne, P. Sandaruwan, and S. Ranathunga, “Transliteration and byte pair encoding to improve sinhala to tamil neural machine translation,” in 2018 Moratuwa Engineering Research Conference (MERCOn). IEEE, 2018, pp. 390–395.

P. Tennyse, P. Sandaruwan, M. Thilakarathne, A. Herath, and S. Ranathunga, “Handling rare word problem using synthetic training data for sinhala and tamil neural machine translation,” in Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018), 2018.

S. Ranathunga, F. Farhath, U. Thayasivam, S. Jayasena, and G. Dias, “Si-ta: Machine translation of sinhala and tamil official documents,” in 2018 National Information Technology Conference (NITC). IEEE, 2018, pp. 1–6.

F. Farhath, S. Ranathunga, S. Jayasena, and G. Dias, “Integration of bilingual lists for domain-specific statistical machine translation for sinhala-tamil,” in 2018 Moratuwa Engineering Research Conference (MERCOn). IEEE, 2018, pp. 538–543.

R. Weerasinghe, “A statistical machine translation approach to sinhala-tamil language translation,” Towards an ICT enabled Society, pp. 36, 2016.

S. Sripirakas, A. R. Weerasinghe, and D. L. Herath, “Statistical machine translation of systems for sinhala-tamil,” in Advances in ICT for Emerging Regions (ICTer), 2010 International Conference on. IEEE, 2010, pp. 62–68.

M. Jeyakaran and R. Weerasinghe, “A novel kernel regression based machine translation system for sinhala-tamil translation,” in Proceedings of 4th Annual UICSC Research Symposium, 2013.

R. Pushpananda, R. Weerasinghe, and M. Niranjana, “Towards sinhala tamil machine translation,” in Advances in ICT for Emerging Regions (ICTer), 2013 International Conference on. IEEE, 2013, pp. 288–288.

T. Mokanaranagan, “Sinhala–tamil machine translation: Towards better transliteration quality,” in Proceedings of the Australian Language Technology Association Workshop 2014, 2014, pp. 129–133.

S. Rajpirathap, S. Sheeyam, K. Umasuthan, and A. Chelvarathna, “Real-time direct translation system for sinhala and tamil languages,” in 2015 Federated Conference on Computer Science and Information Systems (FedCSIS). IEEE, 2015, pp. 1437–1443.

W. S. N. Dilshani, S. Yashothara, R. T. Uthayasanker, and S. Jayasena, “Linguistic divergence of sinhala and tamil languages in machine translation,” in 2018 International Conference on Asian Language Processing (ALAP). IEEE, 2018, pp. 13–18.

T. Mokanaranagan, “Translation of named entities between sinhala and tamil for official government documents,” 2019.

A. Arukgoda, A. R. Weerasinghe, and R. Pushpananda, “Improving sinhala–tamil translation through deep learning techniques,” 2019.

A. S. Arukgoda, “Improving sinhala–tamil translation through deep learning techniques,” Ph.D. dissertation, 2021.

A. Pramodya, R. Pushpananda, and R. Weerasinghe, “A comparison of transformer, recurrent neural networks and SMT in Tamil to Sinhala MT,” in 2020 20th International Conference on Advances in ICT for Emerging Regions (ICTer). IEEE, 2020, pp. 155–160.

L. Nissanka, B. Pushpananda, and A. Weerasinghe, “Exploring neural machine translation for sinhala-tamil languages pair,” in 2020 20th International Conference on Advances in ICT for Emerging Regions (ICTer). IEEE, 2020, pp. 202–207.

S. Thillainathan, S. Ranathunga, and S. Jayasena, “Fine-Tuning Self-Supervised Multilingual Sequence-to-Sequence Models for Extremely Low-Resource NMT,” in 2021 Moratuwa Engineering Research Conference (MERCOn). IEEE, 2021, pp. 432–437.

S. Yashothara and R. T. Uthayasanker, “The utility of hierarchical phrase-based model machine translation for low resource languages,” in Computational Linguistics and Intelligent Text Processing: 19th International Conference, CICLing 2018, Hanoi, Vietnam, March 18–24, 2018, Revised Selected Papers, Part I. Springer, 2023, pp. 279–288.

S. Thelijjagoda, Y. Imai, and T. Ikeda, “Japanese-sinhalese machine translation system jaw/sinhalase,” Journal of the National Science Foundation of Sri Lanka, vol. 35, no. 2, 2007.

R. M. M. Shalini and B. Hettige, “Dictionary based machine translation system for pali to sinhala,” in SLAAI-International Conference on Artificial Intelligence, 2017, p. 23.

R. C. Childers, A dictionary of the Pali language. Trübner, 1875.

S. Salavie, An introduction to the study of eastern liturgies. Sands & Company, 1938.

A. J. Liddicoat, “Choosing a liturgical language,” Australian Review of Applied Linguistics, vol. 16, no. 2, pp. 123–141, 1993.

A. Walaia, R. Weerasinghe, R. Pushpananda, C. Liyanage, and E. Jayalatharachchi, “A data-driven approach to checking and correcting spelling errors in sinhala,” Int. J. Adv. ICT Emerg. Reg, vol. 3, no. 01, 2010.

R. A. Walaia, R. Weerasinghe, R. Pushpananda, C. Liyanage, and E. Jayalatharachchi, “An open-source data driven spell checker for sinhala,” ICTer, vol. 3, no. 1, 2011.

E. Jayalatharachchi, A. Walaia, and R. Weerasinghe, “Data-driven spell checking: the synergy of two algorithms for spelling error detection and correction,” in International Conference on Advances in ICT for Emerging Regions (ICTer2012). IEEE, 2012, pp. 7–13.

L. G. B. Subhayaga, L. Ranathunga, W. H. A. Nimasha, B. R. Jayawickrama, and K. L. Mahalilianaarchchi, “Data driven approach to sinhala spellchecker and correction,” in 2018 18th International Conference on Advances in ICT for Emerging Regions (ICTER). IEEE, 2018, pp. 01–06.

U. Liyanapathirana, K. Gunasinghe, and G. Dias, “Sinspell: A comprehensive spelling checker for sinhala,” arXiv preprint arXiv:2007.02983, 2020.

L. Sithamparanathan and T. Uthayasanker, “A sinhala and tamil extension to generic environment for context-aware correction,” in 2019 National Information Technology Conference (NITC). IEEE, 2019, pp. 102–106.

L. Samarawickrama, H. L. Premarathne, S. C. M. De Silva, and S. B. Hettige, “LaSi Spell: Language Agents for Sinhala Spellings,” 4th International Conference on Advances in Computing and Technology (ICACT . . . , 2019.

Y. Udagedara, B. Elikerewa, and U. Thayasivam, “Language model-based spell-checker for sri lankan names and addresses,” in 2022 Moratuwa Engineering Research Conference (MERCOn). IEEE, 2022, pp. 202–207.

A. D. N. Herath, S. Ranathunga, S. Jayasena, and G. Dias, “An error correction system for pali to sinhala,” in Proceedings of 4th Annual UICSC Research Symposium, 2013.

A. Arukgoda, A. R. Weerasinghe, and R. Pushpananda, “Improving sinhala–tamil translation through deep learning techniques,” 2019.
K. Munidasa, *Viyakarana Vivaranaya*. MD Gunasena Publishers, Colombo, 1938.

National Institute of Education, *Sinhala Lekhana Roethiya*, 1989.

K. Jayathilake, *Nuthana Sinhala Vyakaranaye Mul Potha*. Pradeepa Publications, 1991.

U. S. Sannasgala and A. Perera, *Viyakarana Vimansawa*, 1995.

V. G. Balagalle, *Basha Adauananayasa Sinhala Vivharaya*, 1995.

W. S. Karunatilaka, *Sinhala Basha Viharanaya*. M D Gunasena Publishers, Colombo, 2004.

T. Jayawardana, *The surface case system in Sinhala*. University of Kelaniya, 1989.

D. Rajapaksha, *Sinhala bhashave pada bedima saha virana lakshana bhavithaya*, 2008.

S. M. Kariyakarawana, *The syntax of focus and wh-questions in Sinhala*. Karunaratne & Sons Limited, 1998.

S. L. Kekulawala, *The future tense in Sinhalese – an ‘unorthodox’ point of view*. Vidyalankara University of Ceylon, 1972.

S. Ranathunga and N. de Silva, “Some languages are more equal than others: Probing deeper into the linguistic disparity in the nlp world,” arXiv preprint arXiv:2210.08523, 2022.

J. Lankage, “Sinhala warna malawe vikashanaya,” Ph.D. dissertation, 1988.

M. Mudiyanse, *Sinhala Warna Malawe Vikashanaya*. Godage International Publishers, 2018.

W. Thennakoon, *Parani Lankawa ha Shilalipi*. M D Gunasena, 1957.

W. S. Karunarathne, *Sinhala Shila Lekhana*. S Godage & Brothers, 1956.

S. U. Deraniyagala, *The Prehistory of Sri Lanka*. Department of Archaeological Survey Sri Lanka, 1992.

Wiktionary, “anusvāra,” https://en.wiktionary.org/wiki/anusvara, (Accessed on 02/05/2023).

同様に, “visarga,” https://en.wiktionary.org/wiki/visarga, (Accessed on 02/05/2023).

S. T. Nandasara, “Development and standardization of sinhala script code for digital inclusion of native computer users,” Ph.D. dissertation, 2019.

A. Staatsdruckerei, *Alfabete des gesammten Erdkreises*. Druck und Verlag der Kaiserlich-Königlichen Hof- und Staatsdruckerei, 1855. [Online]. Available: https://books.google.lk/books?id=dswbybAEACAA

Microsoft, “Font list windows 10 - typography - microsoft learn,” https://learn.microsoft.com/en-us/typography/fonts/windows_10_font_list, 1998, (Accessed on 02/05/2023).
APPENDIX A
EVOLUTIONARY ERAS OF THE SINHALA SCRIPT

The evolutionary eras of the Sinhala script have raised much scholarly debate. We have shown some prominent categorizations in Fig 3. Other than Mudiyanse [586], scholars generally agree with the fact that the oldest records start at around 300 BCE. Other than Deraniyagala [589], all others agree that an era border should exist at 1 AD. However, this is a peculiar observation. Lankage [584, 585], Mudiyanse [586], Karunarathne [588], and Nandasara and Mikami [64] place an era border at 300 CE while Lankage [584, 585], Mudiyanse [586], Thennakoon [587], and Nandasara and Mikami [64] place an era border at 500 CE. The next most common border placement is at 800 CE by Lankage [584, 585], Mudiyanse [586], Thennakoon [587], and Karunarathne [588]. The last relatively common border placement is agreed between Lankage [584, 585], Mudiyanse [586], Thennakoon [587], and Karunarathne [588]. Given that the widest coverage is provided by Nandasara and Mikami [64], we have used their era definitions and examples in Fig 4 and Fig 5. The sources from which Nandasara and Mikami [64] have extracted the ancient script are given by the relevant superscripts as follows:

| Era | Source |
|-----|--------|
| 300 BCE - 1 CE | 1. Periyankulama (207-197 BCE) | 2. Mihintale (207-197 BCE) | 3. Situlpawwa (161-137 BCE) | 4. Korawakgala (77-63 BCE) | 5. Ritigala Weeweltanne (22-7 BCE) | 6. Yatahalena Vihara (22-7 BCE) | 7. Gallena Vihara (22-7 BCE) | 8. Nuwarragala (22-7 BCE) | 9. Ritigala Andiyakanne (22-7 BCE) | 10. Boowattegala (22-7 BCE) | 11. Rajagala (44-22 BCE) |
| 1 CE - 300 CE | 12. Anuradhapura (1-7 CE) | 13. Situlpawwa (1-7 CE) | 14. Maharattamale (7-18 CE) | 15. Wallipuram (67-111 CE) | 16. Viharagala (60-67 CE) | 17. Pahala Kainattama (60-67 CE) | 18. Tonigala (301-328 CE) | 19. Ruwanweliseya (337-365 CE) | 20. Thissamaharama (406-428 CE) | 21. Anuradhapura (437-452 CE) |
| 300 CE - 500 CE | 22. Kandanadu (517-518 CE) | 23. Dhakshinathupa (639-650 CE) | 24. Baron Paviliyan (639-650 CE) | 25. Kuchchaweli (639-650 CE) | 26. Murutawa (639-650 CE) |
| 500 CE - 700 CE | 27. Thiriyaya (733-771 CE) | 28. Viyuvalpopa (853-887 CE) | 29. Dorabewila (915-925 CE) | 30. Baddulla (946-954 CE) | 31. Polonnaruwa (982-1029 CE) | 32. Indikatuseya (982-1029 CE) |
Fig. 4: The evolution of the Sinhala script under the era classification proposed by Nandasara and Mikami [64] - Part 1.

**Note:** The IPA for (*) anusvāra [590] and (**) visarga [591] are approximations as there is no consensus on their representation. 1200 AD and 1500 AD from inscriptions and pillars [64]. 1737 from the first printed Sinhala book collected by Nandasara [592]. 1876 from “Alfabete des gesammten Erdkreises” (Alphabets of the entire world) [593] reported as CINGALESISCH [592]. 1891 from the alphabet shown by Gunasekara [177]. 1996 from the character set Sarasavi developed by S T Nandasara [592]. Nandasara and Mikami [64] report that the gray cells are not included due to being possible to be produced using consonant modifiers. 1998 from the Iskoola Pota UNICODE font by Microsoft [594].
Fig. 5: The evolution of the Sinhala script under the era classification proposed by Nandasara and Mikami [64] - Part 2

Note: 1200 AD and 1500 AD from inscriptions and pillars [64], 1737 from the first printed Sinhala book collected by Nandasara [592], 1876 from “Alfabete des gesammten Erdkreises” (Alphabets of the entire world) [593] reported as CINGALESISCH [592]. 1891 from the alphabet shown by Gunasekara [177]. 1996 from the character set Sarasavi developed by S T Nandasara [592]. Nandasara and Mikami [64] report that the gray cells are not included due to being possible to be produced using consonant modifiers. 1998 from the Iskoola Pota UNICODE font by Microsoft [594].
APPENDIX B
AUTHOR META ANALYSIS

Fig. 6: Co-author graph of the most prolific researchers in the Sinhala NLP domain (Selected at the threshold of at least 3 publications)
Fig. 7: The Probability of research from an institution citing that of other institutions. Note that the calculations are limited by 4 factors: (1) The availability of free to download pdf of the paper, (2) The aforementioned pdf containing references list (Some extended abstracts do not come with the references list), (3) The text extraction capabilities of pdf2text⁹⁵, (4) The accuracy of the research paper title look up. With all those limitations in mind, we still can make a few interesting observations. The institution with the highest number of publications, UCSC - University of Colombo seems to be getting the most citations from most sources. However, they themselves seem to almost exclusively cite their own papers (0.7543). The only exception is the smaller number of works they cite (0.2114) from the Department of CSE - University of Moratuwa, the institute with the second highest number of publications. This leaves them with a 0.0343 probability of citing anyone else. Comparatively, Department of CSE - University of Moratuwa seems to be more egalitarian in citing. They have a lower self-citation probability (0.5251) and a higher probability of citing UCSC - University of Colombo (0.3513). This results in a probability of citing others at 0.1236. Both the Johns Hopkins University and University of Edinburgh prefer to cite Google instead of work from Sri Lanka. Faculty of IT - University of Moratuwa prefers to cite UCSC - University of Colombo (0.2339) rather than Department of CSE - University of Moratuwa (0.0968) which is from the same parent institution.
Fig. 8: Authors with at least 10 papers in the Sinhala NLP domain, mapped to their research interests denoted by the subsection titles of the Section 3 of this paper.