A Sinhala Word Joiner

Rajith Priyanga **Surangika Ranatunga** **Gihan Dias**
Department of Computer Science and Engineering
University of Moratuwa
Sri Lanka
rpriyanga@yahoo.com, surangika@cse.mrt.ac.lk, gihan@uom.lk

Abstract

Sinhala is an agglutinative language where many words are formed by joining several morphemes. Word joining is a basic operation in Sinhala morphology, and is based on sandhi rules.

The Sinhala word joiner is a software component which implements sandhi rules to synthesise a word from two or more morphemes. We studied Sinhala word join rules based on grammar and usage and implemented a library and a standalone application to synthesise Sinhala words. In addition to the joined word, it also outputs the rule used for joining. The tool uses a combination of a corpus and hand-coded rules to improve accuracy.

Keywords: Sinhala word joiner, Morphophonemic tools, corpus based scoring algorithm, sandhi rules

1 Introduction

Sinhala belongs to the Indo Aryan sub branch of the Indo-European language family. It is a descendent of the Sanskrit language, but was heavily influenced by the Pāli language from the second century B.C. as a result of the introduction of Buddhism to Sri Lanka. Other than from Pāli, Sinhala was influenced mainly by Tamil, Arabic, Portuguese, Dutch and English languages. Sinhala is written in its own script which is a descendent of the Brahmi script.

Even though the Sinhala language and its script have many similarities with their ancestors from India, they have evolved uniquely over two millennia.

There have been some attempts to implement morphological synthesizers and analysers for Sinhala verbs and nouns (Hettige and Karunananda, 2011). The basic operation of Sinhala word formation is joining a word with affixes or other words. There is currently no software tool to implement this operation, or the disjoin operation for morphological analysis.

The objective of this work was to implement a word joining tool for the Sinhala language tools stack.

The functionality of the target tool is summarised by a function *f* that has inputs and outputs as follows:

(combined word, rule) = f(left, right)

Where

1. left and right are valid Sinhala words or morphemes.
2. *combined word* is the valid joined form of left and right. null is also a valid value.
3. rule is the name of the join rule used for the joining when *combined* is not null.

The rest of this paper is organized as follows. We provide a brief introduction to the Sinhala morphology and join rules in section 2. In section 3, we briefly explain the related work done on the areas related to Sinhala morphology and word joining. In sections 4 and 5 we present the challenges faced and our methodology of solving this problem. Finally, in sections 6 and 7, we present our results and conclusions.

2 Sinhala Morphology

Like Sanskrit, Sinhala is rich in inflectional and derivational morphology. In inflection, grammatical forms of a word are formed by applying morphological operations on the lemma (base word). (Karunarathilaka, 1995)

e.g. : *minis* + *u* → *minissu* (මිනිස් + උ → මිනිස්සු)

Proc. of ICON-2017, Kolkata, India. December 2017 ©2017 NLPAI, pages 220–226

minis is the lemma of the noun man. *u* is the suffix to generate the plural subject form of the noun.

In derivational morphology, words of different word classes or with different meanings are formed by applying morphological operations on the lemma.

e.g.: *duk* + *pat* → *duppat* (දුක් + පත් → දුප්පත්)

duk means suffering. *pat* means become. The combined word *duppat* is an adjective that means poor.

Based on the two-level morphology concept, these morphological operations can be represented in two stages. (Kimmo, 1984)

1. Lexical Representation
2. Surface Representation

e.g.:
1. Lexical representation of the plural subject form of the noun lemma *minis* is (*minis* + *u)* (මිනිස් + උ)

2. Surface representation of (*minis* + *u*) is *minissu* (මිනිස්සු)

For morphological synthesis, both lexical and surface representation rules should be applied. Surface representation rules are generally based on phonology and may transform both the left- and right-hand morphemes. This transformation is called morphophonemics.

In Sinhala, the most common types of word joining are:

prefix + word → word
word (or lemma) + suffix → word
word + word → word

Where + is the join operator.

Most inflectional morphology operations are of the "lemma + suffix → word" form.

e.g.: *minis* + *u* → *minissu* (මිනිස් + උ → මිනිස්සු)

Many derivational morphology operations are of the "prefix + word → word" and "word + word → word" form.

e.g.: *duk* + *pat* → *duppat* (දුක් + පත් → දුප්පත්)

Meaning is as explained above.

pol + *attạ* → *pollattạ* (පොල් + අත්ත → පොල්ලත්ත)

pol means coconut. *atta* means branch. *pollatta* means the branch of a coconut tree.

In Sinhala, this set of morphophonemic rules are called *sandhi* (join rules). Similar to the *Ashṭādhyāyī* of *Pāṇini* in Sanskrit, The Sinhala grammar book *Sidat Saňgarā* written in 13th century A.D. by *Vēdēha Swāmi* describes some of the grammatical aspects of the Sinhala language.

There are nine join rules in Sinhala language according to *Sidat Saňgarā.*

Following is an example of how the *Sidat Saňgarā* has explained join rules. This join rule is named *Pūrwạ Swạrạ Lōpạ*.

"*pera sarạ lopā parạ sarạ gatạtạ pæminạ*"
(පෙර සර ලොපා පර සර ගතට පැමිණ)

Meaning: Vowel part of the last letter of the left word is replaced by the first vowel in the second word.
According to the above definition, there are two conditions for this rule to be valid.

1. The last letter of the left word must be a combined letter that has a consonant and a vowel part
2. The first letter of the right word must be a vowel.

The other join rules are similarly defined.

2.1 Sinhala Word Join Rules

We represent the join rules described in Sidat Saňgarā in an easily understandable format as follows.
Where
- Ci = consonant (e.g. *k* - ක්)
- Vi = vowels (e.g. *a* - අ)
- Individual letters at the word boundary are written in square brackets. (e.g. [C1])
- Combined letters that have a consonant and a vowel in it is written in [Ci|Vi] form.
  e.g.: *ka* = [*k* | *a*] (ක = ක් + අ)
- L and R are the remaining parts of the joining morphemes.

1. **Pūrwạ Swạrạ Lōpạ**
L[C1|V1] + [V2]R → L[C1|V2]R

2. **Parạ Swạrạ Lōpạ**
L[C1|V1] + [V2]R → L[C1|V1]R

3. **Swạrạ**
L[C1] + [V1]R → L[C1|V1]R

4. **Swạrādeshạ**
L[C1|*a*] + [*i*]R → L[C1|*e*]R
L[C1|*a*] + [*u*]R → L[C1|*o*]R
L[C1|*a*] + [*u*]R → L[C1|*ō*]R

5. **Gatrādeshạ**
L[C1|V1] + [C2|V2]R → L[C1|V1][C3|V2]R

Where C3 is a member of {*y, v, h, k, t, p, n, m*}

6. **Pūrwạ Rūpạ**
L[C1] + [C2|V2]R → L[C1][C1|V2]R

7. **Gatrākshạrạ Lōpạ**
L[*n*] + [C2|V2]R → L[*ňg*|V2]R
L[*n*] + [C2|V2]R → L[*ňb*|V2]R

8. **Āgạmạ**
L[C1] + R → L[C1|u]R
L[C1] + R → L[C1|i]R

L[C1|V1] + [V2]R → L[C1|V1][C3|V2]R
Where C3 = {*y, v, r*}

9. **Dvitvạ Rūpạ**
L[C1|V1] + [V2]R → L[C1][C1|V2]R

In addition to the above 9 join rules, we identified a few more join rules in current Sinhala. Some of them are directly taken from Sanskrit and are used in loanwords. The following join rule is an example of a rule that is not in *Sidat Saňgarā*, but currently in use. (Karunathilaka, 1995)

11. **Parạ Rūpạ**

L[C1] + [C2|V2]R → L[C2][C2|V2]R

3 Previous Work

In implementing an English to Sinhala machine translator, Hettige and Karunananda (2011) have implemented a morphological synthesizer. They generate all the forms of all noun classes considering the changes to the letters at the word boundaries in the transformation. They have not used generic joining rules for joining Sinhala words and morphemes but have defined a large number of specific finite state automata to handle multiple letter combination at the word boundaries. However, they do not cover all combinations.

To obtain the indistinct singular subject form of the noun lemma *miti* (short person) and *balu* (dog) they implement 2 different automata, which result in *mittā* and *ballā* respectively.

miti + *ā* → *mittā* (මිටි + ආ → මිටටා)
Remove *ti* (ටි) and append *ttā* (ටටා)

balu + *ā* → *ballā* (බලු + ආ → බල්ලා)
Remove *lu* (ලු) and append *llā* (ල්ලා)

They have implemented 85 FSA for Sinhala noun formations. However, both of the above transformations use the common join rule called *Dvitvạ Rūpạ*, and may be defined as a single FSA.

Also in their method, the FSA must be input to the noun form synthesizer. For the same letter combinations at word boundaries, different finite state automata must be used for different word morpheme combinations. It is not possible to locate the correct FSA without a comprehensive knowledge of the Sinhala language.

There are no other significant work done in the area of Sinhala morphological synthesis or word joining.

Word joiners have been implemented for other Indic languages such as Hindi and Sanskrit. (Jha et al., 2009; Hyman, 2009; Gupta and Goyal, 2017; Kumar et al., 2010) Some of them use the join rules mentioned in the *Ashṭādhyāyī* of *Pāṇini*. (Jha et al., 2009; Hyman, 2009; Gupta and Goyal, 2017)

Most of them have used finite state transducers to do the morphophonemic operations to obtain the surface form. Most of the morphological tool implementations for European languages also use finite state transducers to obtain the surface representation for the lexical representation (Lauri et al., 1992). Finite state transducers have been widely used in solving this morphology problem in different language families.

4 Challenges

In Sinhala, for a given pair of morphemes, there may be multiple matching join rules based on the boundary conditions. Also, even when a single join rule is applied, there can be multiple possible outputs, all of which are not necessarily valid for a given pair.

Accordingly, we have identified the following scenarios for a pair of or morphemes.

1. There is only one matching join rule for the pair. The combined form/forms generated by the rule are
    a. valid
    b. partially valid
    c. invalid
2. There are multiple matching join rules for the pair and they yield the same combined form. The combined form is
    a. valid
    b. invalid
3. There are multiple matching join rules and they yield the different combined forms. The combined forms are
    a. valid
    b. partially valid
    c. invalid

In order to detect the correct join rule and the correct join forms accurately, two options were considered.

1. Using another set of rules, which can be applied on top of the standard join rules to eliminate false positives. e.g.: When joining the *naLu* (නළු) and *a* (ආ), the *Dvitvạ Rūpạ* join rule is also selected. It generates the form *nalla*. (නළ්ළා). But there is an elimination rule that says the letter *L* (ළ) is not duplicated. Therefore, the form *nalla* is eliminated.

2. Check against the set of all valid Sinhala words, so that you can eliminate invalid Sinhala words.

An attempt was made to collect the language rules that can be used to detect the correct join operation for a given pair. But it was found that the documented secondary rule set is not complete, so that in some scenarios, access to the Sinhala vocabulary is needed to check the validity of the combined forms.

Also, an attempt was made to learn these extra rules from a sample data set with tuples of left, right and combined forms. Sinhala being a low resource language, it is difficult to collect an accurately enriched data set large enough to perform the learning to learn the complete rule set.

Having access to a database of all valid Sinhala words is also not practical. Also there are some valid words generated as combined forms by the join rules, that are not valid combined forms of the given 2 words or morphemes.

Hence a combined solution is proposed. It involves finite state transducers for each join rule and non-tagged corpus of Sinhala words.

5 Methodology

In this research, we implemented a generic Sinhala word joining tool based on the 9 base rules and 4 additional join rules that are currently in use.

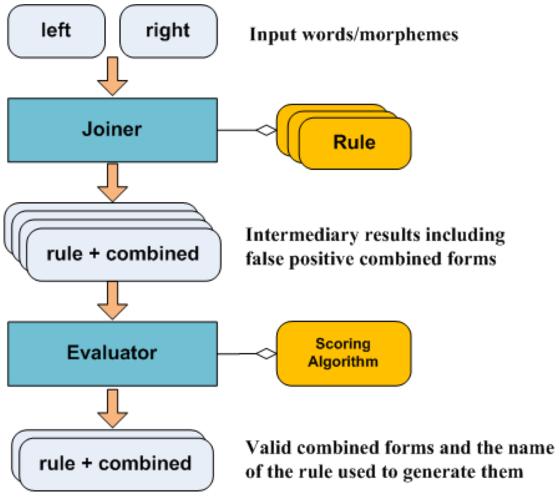
left
right
Input words/morphemes
Joiner
Rule
rule + combined
Intermediary results including false positive combined forms
Evaluator
Scoring Algorithm
rule + combined
Valid combined forms and the name of the rule used to generate them

Figure 1: Word joining process

Figure 1 shows the bird's eye view of the word joining process.

Due to the nature of the Sinhala join rules and exceptions in the Sinhala language, finite state transducers in isolation cannot solve the problem accurately.

For a given left right word/morpheme pair, the joiner applies all the applicable join rules. Some finite state transducers arrive at end states and yield combined forms. All the pairs of combined forms and the join rules used to generate them are returned as intermediate results. There can be both false positives and true positives among them.

A scoring algorithm is introduced to evaluate all the combined forms generated by the join rules. The purpose of the scoring algorithm is to assign a score to each combined form generated by finite state transducers.

The evaluator then selects the results with a score larger than a threshold. The best value for the threshold with respect to a given scoring algorithm is obtained by regressing the joining operation with different threshold values for a sample data set with a manually verified results set.

5.1 Scoring Algorithm

The following parameters are passed to the scoring algorithm.

1. Left most word or morpheme
2. Right most word or morpheme
3. Combined form of the left and right.
4. Name of the join rule used to generate the combined form

The algorithm returns an integer value as the score for the given quadruple.

Our software application uses a corpus and some hand coded elimination rules to derive the score.

It first checks whether the combined form is an invalid joined form of the left and right words or morphemes according to the elimination rules. If it is invalid, the score is set as -1.

If the combined form is not invalid, the word is looked up in the corpus. The occurrence frequency of the word is set as the score. It is a non-negative number.

For the current corpus, the threshold is set as 2. This value has will depend on the size and quality of the corpus. If this set to 0, the number of false positives increases due to the impurities in the corpus. If this is set to a larger value, the number of false negatives increases since the evaluator tends to reject valid combined forms that have a low frequency of occurrence in the corpus.

New scoring algorithms may be plugged-in to the application to obtain better results.

6 Results

The precision and recall were measured for the joining results of the following data sets.

6.1 Dataset 1

8 different grammatical forms of 50 Sinhala nouns were generated by joining their lemma and relevant suffixes. 412 noun forms are expected for the 400 word-morpheme pairs.

6.2 Dataset 2

50 pairs of complete words are joined to generate combined words.

6.3 Precision and Recall

True positives are the correct combined forms for a given pair generated by the application as the end results.

False positives are the incorrect combined forms for a given pair generated by the application as the end results.

False negatives are the expected combined forms for a given pair, but not given by the application as end results. Some of them were eliminated by the scoring algorithm.

	True positives	False positives	False negatives
Dataset 1	401	22	9
Dataset 2	46	0	4
Total	447	22	13

Precision = True positives / (True positives + False positives)
= 447/(447+22)
= 0.9531

Recall = True positives / (True positives + False negatives)

= 447/(447+13)
= 0.9717

7 Conclusion

7.1 False Positives

The analysis showed that the main reason for the false positives is the lack of elimination rules.

E.g.
The input : *ali* + *ā* (අලි + ආ)
ali is the lemma of the noun elephant
ā is the suffix to form the singular subject form

Expected output : (*aliyā* - අලියා, *āgamạ*)

Actual outputs : (*aliyā* - අලියා, *āgamạ*), (*allā* අල්ලා, *dwithwạ rūpạ*)

aliyā means elephant. *allā* means god Allah.

The word *allā*, though it occurs frequently in the corpus, is not a valid combined form of the lemma *ali* and the suffix *ā*.

Some false positives are due to the impurities in the corpus.

We may add an exceptions database and introduce more elimination rules to the scoring algorithm to reduce the false positives.

7.2 False Negatives

The analysis showed that the main reason for the false negatives is the incompleteness of the corpus. The occurrence frequencies of the valid combined forms that are not available in the corpus are set as 0. Therefore, they are eliminated by the evaluator.

It is not possible to create a corpus with all valid Sinhala words. Therefore, we may use statistical or machine learning methods to learn further scoring rules.

7.3 Performance

Since our corpus contains 1.2 million entries (including impurities) the database lookup takes a considerable time on test machines. Therefore, an average join operation for a given word pair takes around 20 milliseconds on a 2.5 GHz processor.

7.4 Future Enhancements

A possible future enhancement would be to generate a large sample dataset of tuples of left and right words or morphemes, combined forms and rule name using the current word joiner tool version, get them verified using human input and use that dataset to mine the elimination rules using statistical methods.

The elimination rules mined by this exercise may also be used to implement a scoring mechanism for words that are not available in the corpus.

References

[Department of Census 2012] Department of Census and Statistics Sri Lanka. (2012). Census of Population and Housing. Retrieved from http://www.statistics.gov.lk/PopHouSat/CPH2011/Pages/Activities/Reports/FinalReport/Population/FinalPopulation.pdf

[Geiger 1938] Wilhelm Geiger (1938). A Grammar of the Sinhalese Language. Ceylon Branch of the Royal Asiatic Society (Colombo). Colombo.

[Gupta and Goyal 2017] Priyanka Gupta, and Vishal Goyal (2017), Implementation of Rule Based Algorithm for Sandhi-Vicheda Of Compound Hindi Words. International Journal of Computer Science Issues, vol. 14, no. 2, pp. 45–49

[Hettige and Karunananda, 2011] B. Hettige and A. S. Karunananda. (2011), Computational model of grammar for English to Sinhala Machine Translation. 2011 International Conference on Advances in ICT for Emerging Regions (ICTer)

[Hyman, 2009] Malcolm D. Hyman. (2009). From Pāṇinian Sandhi to Finite State Calculus. Lecture Notes in Computer Science - Sanskrit Computational Linguistics: 253–265

[Ido et al, 1997] Ido Dagan, Lillian Lee, and Fernando Pereira. (1997). Similarity-based methods for word sense disambiguation. Proceedings of the 35th annual meeting on Association for Computational Linguistics

[Jha et al, 2009] Jha G.N. et al. (2009). Inflectional Morphology Analyzer for Sanskrit. In: Huet G., Kulkarni A., Scharf P. (eds) Sanskrit Computational Linguistics. Lecture Notes in Computer Science, vol 5402. Springer, Berlin, Heidelberg

[Karunarathilaka, 1995] W.S.Karunathilaka. (1995). Sinhala Bhasha Vyakaranaya, M.D. Gunasena, Colombo, Sri Lanka

[Kimmo, 1984] Kimmo Koskenniemi. (1984) A general computational model for word-form recognition and production. Proceedings of the 22nd annual meeting on Association for Computational Linguistics

[Kumarathunga, 2000] Munidasa Kumarathunga (2000). Vyakarana Vivaranaya. S. Godage. Colombo, Sri Lanka

[Kumar et al, 2010] Anil Kumar, Vipul Mittal, and Amba Kulkarni (2010). Sanskrit Compound Processor. Lecture Notes in Computer Science Sanskrit Computational Linguistics: 57–69

[Lauri et al, 1992] Lauri Karttunen, Ronald M. Kaplan, and Annie Zaenen. (1992). Two-level morphology with composition. Proceedings of the 14th conference on Computational linguistics

[Murali et al, 2014] N. Murali, R.j. Ramasreee, and K. V. R. K. Acharyulu. (2014). Kridanta Analysis for Sanskrit. International Journal on Natural Language Computing, 3(3):33–49

[Porter, 1980] M.F. Porter (1980). An algorithm for suffix stripping, Program, Vol. 14 Issue: 3, pp.130-137

[Sharma et al, 2002] Utpal Sharma, Jugal Kalita, and Rajib Das (2002). Unsupervised learning of morphology for building lexicon for a highly inflectional language. Proceedings of the ACL-02 workshop on Morphological and phonological learning