Syllabification of Lao Script for Line Breaking

Phonpasit Phissamay,Valaxay Dalolay,Chitaphone Chanhsililath, Oulaiphone Silimasak Sarmad Hussain, Nadir Durrani
Science Technology and Environment Agency
CRULP
phonpasit@stea.gov.la[1],valaxay@stea.gov.la,vnoy2002@yahoo.com[1],
sarmad.hussain@nu.edu.pk[2],nadirdurrani@yahoo.com[2]

Abstract

Lao like other South East Asian languages is written in a continuum. Space is never or at least rarely used character in Lao language. Tokenization is a foremost obligatory task in almost all NLP tasks. Word identification becomes easier if we can extract syllables first and then combine syllables to form words based on collocation. This paper discusses syllable identification in Lao script. The technique gives more than 98% results.

1. Definition

Lines are broken as result of one of two conditions. The first condition is the presence of an explicit line breaking character. The second condition results from a formatting algorithm having selected among available line break opportunities; ideally the chosen line break results in the optimal layout of the text.

Different formatting algorithms may use different methods to determine an optimal line break. For example, simple implementations consider a single line at a time, trying to find a *locally optimal* line break. A basic, yet widely used approach is to allow no compression or expansion of the inter-character and inter-word spaces and consider the longest line that fits. When compression or expansion is allowed, a locally optimal line break seeks to balance the relative merits of the resulting amounts of compression and expansion for different line break candidates.

When expanding or compressing inter-word space according to common typographical practice, only the spaces marked by U+0020 SPACE, U+00A0 NO-BREAK SPACE, and U+3000 IDEOGRAPHIC SPACE are subject to compression, and only spaces marked by U+0020 SPACE, U+00A0 NO-BREAK SPACE, and occasionally spaces marked by U+2009 THIN SPACE are subject to expansion. All other space characters normally have fixed width. When expanding or compressing inter-character space the presence of U+200B ZERO WIDTH SPACE or U+2060 WORD JOINER is always ignored.[1]

There are three important style of context analysis used to determine line break opportunities.

Western — spaces and hyphens are used to defined breaks: It is commonly used for scripts employing the space character. Hyphenation is often used with space-based line breaking to provide additional line break opportunities. However, it requires knowledge of the language and in addition; it may need user interaction or overrides.[2]

East Asian — lines can break anywhere, unless prohibited: In these scripts, lines can break anywhere, except before or after certain characters. The precise set of prohibited line breaks may depend on user preference or local custom and is commonly tailorable.[3]

South East Asian — line breaks require morphological analysis: The third style is used for scripts such as Lao, which do not use spaces, but which restrict word-breaks to syllable boundaries, the determination of which requires knowledge of the language comparable to that required by a hyphenation algorithm. Such an algorithm is beyond the scope of the Unicode Standard.[4]

[1] http://ftp.unicode.org
[2] http://sqs.cmr.sfc.keio.ac.jp/tdiary/20060902.html
[3] http://ftp.unicode.org
[4] http://unicode.org/unicode/reports/tr14/

2. Introduction

Syllable is a unit of spoken language, which may have a common meaning or not. A unit of spoken language may have one single syllable or many syllables. Additional complexity is introduced by lack of a space character. Though humans can process multiword string while reading and extract words from it, this process is very difficult for computers. However, unless words can be separated, it is impossible to perform even the simple task of determining how to break at the end of a typed line when characters exceed the line length, without the possibility of breaking between a word and worse between syllables.

In Lao language, the breaking of the text flow after each word (like in English) is not common. One possible way line breaking can be achieved in Lao is to use a Lao lexicon which is currently incompletion. Lao collation is complex because it does not sort on key-press order but on basis of its intricate syllable structure. Thus, before any processing is done, a Lao character string has to be syllabified. This paper develops a Lao string syllabification algorithm. This algorithm is essential for processing basic input character string to enable more advanced language processing including searching, sorting, line breaking and lexical development.

3. Lao Character Set

Lao syllable structure contains characters and marks for consonants, vowels and tones. Table 1 below lists these possible characters and marks. The marks are combining characters and are shown adjacent to 'x', latter representing a consonant. Character names and their Unicode is also given (ref. to Unicode)

a. Vowels ('x' is a placeholder for a consonant character)

Table 1: Vowels

◌ะ	Vowel A	◌ู	Vowel UU
◌ั	Vowel Maikan + Consonants	◌ิ	Vowel E +Main Consonants
◌า	Vowel AA	◌ี	Vowel E + Vowel Maikan + Consonantal
◌ำ	Vowel AM	เX	Vowel E + Main Consonants
◌ิ	Vowel I	แX	Vowel EI + Main Consonants + Vowel A
◌ี	Vowel II	ไX	Vowel EI + Vowel Maikan + Consonantal
◌ึ	Vowel Y	ใX	Vowel EI + Main Consonants
◌ื	Vowel YY	◌ๅ	Vowel O + Main Consonants + Vowel A
◌ุ	Vowel U		

b. Consonants ('x' marks a placeholder for a consonant character)

Table 2: Consonants

ก	Letter KO	ม	Letter MO
ข	Letter KHO SUNG	ย	Letter YO
ฃ	Letter KHO TAM	ร	Letter LO LING
ง	Letter NGO	◌ฺ	SEMI Vowel Sing LO
จ	Letter CO	ล	Letter LO LOOT
ส	Letter SO SUNG	ว	Letter WO
ฉ	Letter SO TAM	ห	Letter HO SUNG
ย	Letter NYO	หง	Letter HO SUNG+ Letter NGO
ด	Letter DO	หย	Letter HO SUNG + Letter NYO
ต	Letter TO	หม	Letter HO SUNG + Letter NO
ถ	Letter THO SUNG	ฆ	Letter HO NO
ท	Letter THO TAM	หม	Letter HO SUNG + Letter MO
น	Letter NO	ฌ	LETTER HO MO
บ	Letter BO	หล	Letter HO SUNG + Letter LO LOOT

ປ	Letter PO	ຫຼ	Letter HO SUNG + SEMIVOWEL SIGN LO
ຜ	Letter PHO SUNG	ຫວ	Letter HO SUNG + Letter WO
ຝ	Letter FO TAM	ອ	Letter O
ພ	Letter PHO TAM	ຮ	Letter HO TAM
ຟ	Letter FO SUNG		

c. Tones ('x' is a placeholder for a consonant character)

Table 3: Tone Marks

◌	Lao Tone MAI EK
◌	Lao Tone MAI THO
◌	Lao Tone MAITI
◌	Lao Tone MAI CATAWA

Sign ('x' is a placeholder for a alternate consonant character)

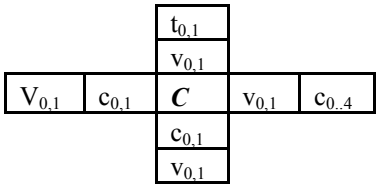
Table 4: Special Sign

ໆ	Lao KO LA
ฯ	Lao ELLIPSIS

Table 1 shows there are 18 vowel marks and characters, 30 character marks and characters, 4 tonal marks and 3 special symbols. Vowels can occur before, above, below a consonantal character or on the baseline. Characters occur on the baseline. Tonal marks always occur on top of consonantal characters.

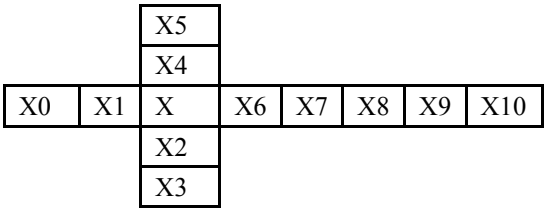
4. Basic Syllable Structure

Lao language writing system is based on a central or nuclear consonantal character. This consonant may have optionally vowel character or marks around it (before, after, above or below). In addition, this nuclear consonantal character may also have optional a tonal mark above it and optionally more consonantal characters following it. This is illustrated in Figure 1 below. Capital C indicates the nuclear consonantal character. The subscripts indicate that all are optional except the nucleus ***C***.



4.1. Constraints on Syllable Structure

Not all characters and marks are allowed at all the placeholders shown in Figure 1. Figure 2 and Table 2 below collectively further classify these consonants, vowels and tones to indicate these limitations. The number *n* in X*n* represents the key-press order or typing sequence of these characters and marks relative to each other, except X (which is typed between X1 and X2). Thus tone is typed after the initial vowel X0, consonant X1, nucleus X, and vowel mark X3 and X4 (if X0, X1 and X2 are not null; X can never be null).



The following table is shown the characters exist in each structure position

X0	X1	X	X2	X3	X4	X5	X6	X7	X8	X9	X10
ເx1	ຫ	ກ01 ຂ02 ຄ03	◌ຼ1	◌ຸ1	◌ິ1	◌ ◌ 1	◌ 1	◌ 1	◌ 1	◌ 1	◌ 1
◌ x2		ງ04 ຈ05 ສ06	◌ ◌ 2	◌ ◌ 2	◌ ◌ 2	◌ ◌ 2	◌ 2	◌ 2	◌ 2	◌ 2	◌ 2
◌ x3		ຊ07 ຍ08 ດ09	◌ 3		◌ ◌ 3	◌ ◌ 3	◌ 3	◌ 3	◌ 3	◌ 3	◌ 3
◌ x4		ຕ10 ຖ11 ທ12	◌ 4		◌ ◌ 4	◌ ◌ 4			◌ 4	◌ 4	
◌ x5		ນ13 ບ14 ປ15			◌ ◌ 5				◌ 5	◌ 5	
		ຜ16 ຝ17 ພ18			◌ ◌ 6				◌ 6	◌ 6	
		ຟ19 ມ20 ຢ21			◌ ◌ 7				◌ 7		
		ຣ22 ລ23 ວ24							◌ 8		
		ຫ25 ອ26 ຮ27									
		ໝ28 ໝ29									

X0 represents a vowel which occurs before the nuclear consonant. It is can always defined the beginning of syllable.

X1 is a combination consonant which comes before the nuclear consonant, only if nuclear consonant is one of {ງ, ຍ, ລ, ວ, ຼ, ມ, ນ, ຣ}.

X is represents the nuclear consonants.

X2 is a combination consonant which comes after the nuclear consonant, which placing under or next to the nuclear consonant.

X3 is represents a vowel which occurs under the nuclear consonant.

X4 is represents a vowel which occurs upper the nuclear consonant.

X5 is represents a tone marks which occurs upper the nuclear consonant or upper vowels.

X6 is represents consonant vowel, which occurs after nuclear consonant. It functions when the syllable doesn't have any vowels. And it always exists with X8.

X7 is represents an after vowels. However $X7_1$ is always represents the end of syllable and it is never exist with tone mark.

X8 is represents alternate consonants.

X9 is represents alternate consonant to pronounce foreign language, it always exist with $X10_3$.

X10 represents a sign mark. X10 is always occurs at the end of syllable, but mostly people keep it separate from syllable.

The nuclear consonant always exists in syllable with some vowel or alternate consonants. The position at which vowels appear can guide to define the beginning and end of each syllable. Using following rules we can find potential syllable boundary.

1. **For $x0_1$ = ◌x**
   - 1.1. ◌x = $x0_1$ (x1)X(x2)(x5)(x8)(x9:x10)
   - 1.2. ◌ิx, ◌ีx = $x0_1$ (x1)X(x2)$x4_{1-2}$(x5)(x8) (x9:x10)
   - 1.3. ◌ึx, ◌ืx = $x0_1$ (x1)X(x2)$x4_{3-4}$(x5) $x6_2$ (x8) (x9:x10)
   - 1.4. ◌xะ, ◌xาะ = $x0_1$ (x1)X(x2)($x7_2$)$x7_1$
   - 1.5. ◌ัx◌า = $x0_1$ (x1)X(x2) $x4_6$ (x5) $x7_2$
   - 1.6. ◌ัx(x8) = $x0_1$ (x1)X(x2) $x4_7$ (x5) x8 (x9:x10)
   - 1.7. ◌ัx(x8) = $x0_1$ (x1)X(x2) $x4_7$ (x5) x8 (x9:x10)
   - 1.8. ◌xๅ, ◌ัxๅ = $x0_1$ (x1)X(x2) ($x4_7$)(x5)$x6_3$
2. **For $x0_2$ = ◌ก x**
   - 2.1. ◌กx = $x0_2$ (x1)X(x2)(x5)(x6)(x8) (x9:x10)
   - 2.2. ◌กxะ = $x0_2$ (x1)X(x2)$x7_1$
   - 2.3. ◌กัx(x8) = $x0_2$ (x1)X(x2) $x4_7$ (x5) x8 (x9:x10)
3. **For $x0_3$ = เ◌x**
   - 3.1. เ◌x, เ◌xอ = $x0_3$ (x1)X(x2)(x5)(x8) (x9:x10)
   - 3.2. เ◌xะ = $x0_3$ (x1)X(x2)$x7_1$
   - 3.3. เ◌ัxอ, เ◌ัxย = $x0_3$ (x1)X(x2)$x4_7$(x5) $x8_{3:8}$
4. **For $x0_4$ = ไx** = $x0_4$ (x1)X(x2)(x5)$(x6_1)$ (x9:x10)
5. **For $x0_5$ = ใx** = $x0_5$ (x1)X(x2)(x5) $(x6_1)$
6. **For x3 = ุx & ูx** = (x1)X(x2)x3(x5)(x8) (x9:x10)
7. **For $x4_{1-4}$ = ◌ิx & ◌ีx & ◌ึx & ◌ืx** = (x1)X(x2)$x4_{1-4}$(x5)(x8) (x9:x10)
8. **For $x4_5$ = ◌่x** = (x1)X(x2)$x4_5$(x5)($x7_2$)(x9:x10)
9. **For $x4_6$ = ◌้x**
   - 9.1. ◌้x(x8) = (x1)X(x2)$x4_6$(x5)x8(x9:x10)
   - 9.2. ◌้xอะ = (x1)X(x2)$x4_6$(x5) $x6_1$ $x7_1$
10. **For $x4_7$ = ◌ั x** = ◌ัx, ◌ัxอ = (x1)X(x2)$x4_7$(x5)($x6_1$)x8(x9:x10)
11. **For x6 = xอ & xอ & xๅ** = (x1)X(x2)(x5)$x6_{1-3}$x8(x9:x10)
12. **For $x7_1$ = xะ** = (x1)X(x2)(x5)$x7_1$
13. **For $x7_2$ = xา** = (x1)X(x2)(x5)$x7_2$ (x8)(x9:x10)
14. **For $x7_3$ = x◌ำ** = (x1)X(x2)(x5)$x7_3$ (x9:x10)

5. Lao Syllabification Algorithm

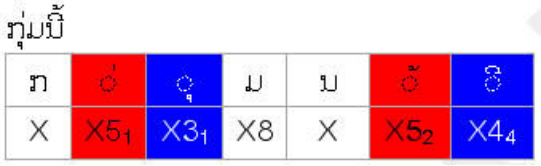
Lao Syllabification algorithm can be defined as follows:

Traverse through the input array and mark syllable boundary as soon as you encounter a punctuation mark, space or a character that does not belong to Lao character set. Example:

Syllable (boundaries due to non-Lao character)

Filter out Lao characters out of input array leaving behind punctuation marks, spaces, and non-Lao characters. Example:

Reorder character in case of typing variations (Sometime people maybe typed X5 before X2, X3, X4 and X4 maybe typed before X2). Example:



ກ	◌่	◌ຸ	ມ	ນ	◌ໍ	◌ິ
X	$X5_1$	$X3_1$	X8	X	$X5_2$	$X4_4$

Syllable In this case: $X3_1$ should typed before $X5_1$ and $X4_4$ should typed before $X5_2$

ກ	◌ຸ	◌่	ມ	ນ	◌ິ	◌ໍ
X	$X3_1$	$X5_1$	X8	X	$X4_4$	$X5_2$

Mark each character in run with all possible X_n values it can take. Example:

ຄົນລາວ	ຄ	◌ົ	ນ	ລ	າ	ວ
	X	$X4_6$	X	X	$X7_2$	X
			X8	X2		X2
				X9		X6
						X8

Use rules discussed in previous section to find out syllable boundaries.

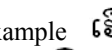
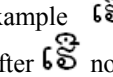
In case if more then one condition suggests a syllable boundary chose the one with longest run.
In case none of the conditions suggest syllable boundary try including last character from previous syllable to current syllable.

Test all the conditions for previous syllable because removing last character might make it invalid. If it is still a valid syllable then try conditions starting from newly added character, other wise restore the previous syllable and skip first character from current syllable and try testing from next character. Keep skipping characters till find a valid syllable boundary.

If program finds boundary for the new syllable it should continue naturally other wise restore previously disturbed syllable by putting back the removed character. Skip the current characters and re-test the conditions, keep skipping unless you find valid syllable.

For example: - In case of  program should identify syllable boundary like

 then it would be unable to detect any condition working for  so it should shift syllable boundary one step back and try to include '  ' with  . But before testing this new syllable it should test the previous syllable and find if it is still a valid syllable which in the current scenario is a valid syllable. Now try testing for  this is also a valid syllable so program would continue naturally and go for testing next syllable.

Now consider this example  program suggests syllable boundary after now as we try test  we would find none of the conditions working so we try to remove from previous syllable and test if is still a valid. In case it is we test we still find none of the conditions working so we restore previous syllable skip current character and carry on testing conditions from . Keep skipping characters unless you find valid syllable. In the current example you only have to skip it once.
Traverse till the end of array.

Put the Lao characters back into original array that have punctuation marks and other non-Lao characters.

Example: ໂຄງການL10Nພັ ດທະນາພາສາລາວ

Step 1 & 2: Traverse through the input array mark syllable boundary where counter non-Lao character. Extract only Lao text and put it in new array.

ໂຄງການ | ພັດທະນາພາສາລາວ

Step 3: Look for possible re-ordering of text. For example user might type 'È' X5 before '×' X47 so we must do the re-ordering.
Step 4: Mark each character with possible $X_{N.}$

Step 5: Use rules and conditions discussed above to define syllable boundaries.

Step 6: Put the Lao characters back into original array.