I.J. Information Technology and Computer Science, **2016, 2, 32-40**
Published Online February 2016 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijitcs.2016.02.04

A Morphological Analyzer for Reduplicated Manipuri Adjectives and Adverbs: Applying Compile-Replace

Ksh. Krishna B Singha
Department of Computer Science, Assam University, Silchar, 788011, India
E-mail: ksworks2010@gmail.com

Kh. Dhiren Singha
Department of Linguistics, Assam University, Silchar, 788011, India
E-mail:dhirensingha@rediffmail.com

Bipul Syam Purkaystha
Department of Computer Science, Assam University, Silchar, 788011, India
E-mail:bipul_sh@hotmail.com

Abstract—Finite-state implementations naturally denote concatenations of morphemes and are limited to modeling concatenative morphotactics. The non-concatenative structure, such as reduplication, in the computational morphology of many world languages cannot be handled completely by finite-state technology. This paper describes the non-concatenative phenomena of reduplication, occurs in the adjective and adverb word classes of Manipuri language using the formalism of finite-state morphology tools and techniques. The discussion covers the non-concatenative nature and the challenges in capturing the various reduplication phenomena exhibited by the two classes; then present a morphological analyzer of the reduplicated adjectives and adverbs. It has been implemented using XFST and LEXC with the application of compile-replace algorithm to the morphotactics description of the language, which includes finite-state operations other than concatenation, to capture reduplication phenomena.

Index Terms—Restricted and Complete Reduplication, Compile-Replace Algorithm, Regular Expression, Morphological Analysis, Finite-State Transducers, Manipuri language.

I. Introduction

Most of the computational morphologists consider reduplication as a residual problem. Reduplication, unlike concatenative morphology creates a new word form from some other words by copying a part or whole of it. So reduplication is a non-concatenative phenomenon, and it involves copying of a part of the word or the whole, of the same. The reduplication may occur with a prefix, root, stem, suffix in the word or a combination of any of these morphemes. Reduplicated words may appear in a dictionary as distinct words from the original word which underwent reduplication [1]. The outcome of reduplication process is to either strengthen/ emphasize the original word for grammatical and/or semantic purposes.

The process of reduplication may yield more than one word and yet is considered as a single word from the morphological point of view. As a matter of fact, reduplication cannot be defined as a concatenative morphology and so cannot apply concatenation operation in a finite – state network. Therefore, reduplicated words are always an issue while performing morphological analysis of a language.

Manipuri language exhibits a productive process of verb stem reduplication, occurs in cases of reduplicated forms of adverbs and adjectives, etc. These forms cannot be described simply by adding reduplicated stems into the dictionary, since it is not the verb root but the full, or possibly extended stem of the verb that is reduplicated. In Manipuri, wh- words, adjectives, adverbs, nouns, pronouns, etc. can be reduplicated. It can also be mentioned here that the language has both complete and restricted reduplication. Of the two types, the former copies the entire word while the later one copies only a part of the word form. In both the cases, roots or affixes, or both can be repeated. In the process of reduplication, some affixes can be identified to take part, but not all the affixes of the language take part.

In this paper we present a finite-state solution to capture the reduplication phenomena in Manipuri adjectives and adverbs. Like many other Tibeto-Burman languages, these two word classes in the language are derived from the verb roots with the help of affixes, particularly the prefixes and the suffixes. In the language, the order of reduplicated adjectives and adverbs are: (i) prefix plus root together, and (ii) root plus suffix. Interestingly both the cases are found in case of adjective and adverb reduplication processes. By looking at such varieties presented by such a productive morphological

phenomena in these two word classes, it would be a very challenging task but at the same time, very interesting as well, to capture both restricted and complete reduplication for morphological analysis, which will show a way for solving the issue for other word classes as well.

It is important to note here that Multi Word Expressions (MWE) in the language can also occur by reduplication [2]. However, our study did not focus on various issues of Multi Word Expressions in the language.

The next section is a brief profile of the language, available reduplicated word forms and an elaboration on reduplicated adjectives and adverbs in the language. The third section discusses the occurrence of reduplication with their corresponding lexical form. Here the initial lexical network and application of the compile-replace algorithm to the lower side language of the transducer network is illustrated in length from the implementation perspectives. Analysis of the implementation outcome follows the third section and concludes with a hint of future research directions in this field for the language.

II. Reduplication in Manipuri

A. Brief profile of Manipuri Language

Manipuri belongs to the Kuki-Chin group of Tibeto-Burman language family [Grierson, 67]. It is mainly spoken in Manipur and its neighboring states in Northeast India, and in neighboring countries like Myanmar and Bangladesh. Typologically, the language is a tonal, agglutinating, and verb final language.

Being an agglutinative language, Manipuri is characterized by a very rich and complex morphological word structure. A single word form may have a good number of morphemes concatenated one after another like beads on a string. There are concatenations of up to twelve morphemes in a single Manipuri word. Normal Manipuri words may be, most of the time, equivalent to average to long English sentences.

Like many other South Asian languages, reduplication is one of the typical features of Manipuri. It shows extensive use of reduplication for grammatical and semantic functions. Manipuri exhibits both complete and restricted reduplication.

Complete reduplication in Manipuri language is seen in nouns, numerals, wh-words and reduplicated adverbs. Examples that illustrates the complete reduplication in Manipuri are–

khun -> khun khun /every village(noun)
yum-> yum yum /every house (noun)
ahum->ahum ahum /three each (numeral)
kəna -> kəna kəna /who who (wh-word)
kəri -> kəri kəri /what what (wh-word)
cenna -> cenna cenna /by running (adverb)
səkna -> səkna səkna /by singing (adverb)
təpna -> təpna təpna /(more) slowly (adverb)

Manipuri adjectives, adverbs and some selective wh-words exhibit restricted reduplication as shown below-

əcət-> əcət əcətpə /those who go
əp^{h}ə -> əp^{h}ə əp^{h}əbə /the good ones
p^{h}əjə -> p^{h}əjə p^{h}əjəbə /the nice ones
p^{h}əjə -> p^{h}əjə p^{h}əjənə /nicely
kərəm -> kərəm kərəmbə /which ones
kədai -> kədai kədaidə /where where

B. Reduplication in Adjectives and Adverbs

Like many other Tibeto-Burman languages, Manipuri distinguishes two open classes- nouns and verbs. Word classes, such as adjectives and adverbs are derived from the verbs. So the formation of adjectives and adverbs in Manipuri is important to demonstrate for a meaningful discussion on the present topic.

i) Adjective Formation from Verb Roots

Adjectives are formed by prefixing formative prefix (FP) *ə* and suffixing an adjectival (NZR) suffix *bə* to a monosyllabic verb root (VR) [4].

(a) ə + p^{h}ə + bə /good
(b) ə +waŋ + bə /tall

that is, it is of the lexical form

$$FP + VR + NZR$$

or, by suffixing nominalizer *bə* to a polysyllabic verb word forms which are either bi-morphemic or compound ones.

(c) hərau + bə /joyful
(d) p^{h}əjə + bə /beautiful
(e) phu-rəm-bə /the beaten ones

So, the general syntax in lexicalized form is

$$VR +(Suffix)+ NZR$$

ii) Reduplicated Adjectives in Manipuri

Reduplicated adjectives in Manipuri are formed in two ways-

A) Duplication of (FP+VR) and
B) Duplication of (VR+suffix(es))

In case of A) the reduplicated form of the word takes the form

$$(FP + VR)\ (FP + VR) + NZR$$

Examples of such reduplication are-

(f) əsaŋ əsaŋbə / the long ones
(g) ətum ətumbə / the pointed ones
(h) ələk ələkpə / the coming ones

It is to be noted here that the verb root (VR) is a monosyllabic root, which requires a formative prefix (FP) *ə-* and a nominalizing suffix (NZR) *bə*/*pə* to form adjectives.

Consider the following adjective, to see how it goes through the process of reduplication.

(i) əcoubə (the big ones)

When (i) is broken into its constituent morphemes and their corresponding lexical forms, it yields the following:

Lexical form:	FP	VR	NZR
Surface form	ə	cou	bə

The simple and reduplicated forms of the adjective are as in Table 2.1-

Table 2.1. Simple and Reduplicated forms of Adjectives of (i)

	Simple form	Reduplicated form	
Lexical form:	FP+VR+NZR	FP+VR	FP+VR+NZR
Surface form:	ə+cou+bə	ə+cou	ə+cou+bə

The reduplication process in case of B) is of the form

(VR + suffixes) (VR + suffixes) + NZR

Some examples are-

(j) caniŋ caniŋbə /the ones who wished to eat
(k) p^həjə p^həjəbə, /the nice ones
(l) tarəm tarəmbə /the fallen ones

The reduplication process copies the polysyllabic forms of the verb *caniŋ, p^həjə* and *tarəm*. The verb forms are comprised of the verb root (VR) and other allowed verb suffixes (viz. *niŋ*, *rəm*, see Krishna B Singha, et al, 2013 for adjective suffixes) required for adjective formation in the language, although *p^həjə* is an opaque polysyllabic verb root.

So for the reduplicated form of the adjective *tarəmbə* can be represented in its surface to lexical form correspondence as in Table 2.2

Table 2.2 Constituent Morphemes of the Reduplicated Adjective in (l)

Surface Form	***Lexical Form***
ta	VR
rəm	Deictic
ta	VR
rəm	Deictic
bə	NZR

The bracketed portion is shown to have been copied in the reduplicated form of the adjective i.e. *tarəm tarəmbə*, where deictic suffix *rəm* is copied along with the verb root *ta.*

In both A) and B) it is a case of restricted reduplication. Only the difference being the monosyllabic roots take the FP while the polysyllabic roots do not.

iii) Adverb formation from Verb roots

As mentioned earlier that Manipuri does not have a distinct category of adverbs. They are derived from the verb roots by suffixing the adverbial suffix (ADVZ) *-nə* as illustrated below-

(m) səknə / by singing
(n) cətnə /by going

Here, the suffix *-nə* is attached to *sək* and *cət*, to form adverbs. The lexical form of it is in the language is schematized as below:

VR + ADVZ

In addition to the above, adverbs in Manipuri are also derived from the verb root by attaching the prefix *tə-*, i, etc. to the verb root followed by the adverbial suffix *-nə*. Some examples of adverbs in the language are:

(o) tə-nik-nə /by twinkling
(p) tə-ram-nə /politely

An exceptional rule to the use of prefix i to form adverb is to reduplicate the verb root. Its use in an adverb does not qualify the word as a valid wordform without reduplication. Here the only the verb root is reduplicated and this reduplicated form has the formative prefix i and the adverbial suffix nə, i.e. it does not exist as an adverb without the reduplicated form. Some examples are-

(q) i-seŋ-seŋ-na /clearly
(r) i-təp-təp-na /slowly

So the lexical form of same in the language can be schematized as under-

FP + VR + VR+ ADVZ

iv) Reduplicated Adverbs in Manipuri

Reduplicated adverbs in Manipuri are formed in four ways-

A) duplication of (VR+ADVZ)
B) duplication of (FP+VR)
C) duplication of (VR + suffix (es)
D) duplication of (VR)

Case A) is an exhibition of the complete reduplication as in case of adjectives; adverbs in the language undergo the process of reduplication by copying the whole part; as in-

(s) səknə səknə /by singing
(t) cətnə cətnə /by going
(u) təpnə təpnə /(more) slowly

To see the reduplication phenomena in this case, let us consider the example (t) to analyze the constituent morphemes in its simple and reduplicated form as follows-

Table 2.3. Simple and Reduplicated forms of (t)

	Simple form	*Reduplicated form*	
Lexical form:	*cət+nə*	*cət+nə*	*cət+nə*
Surface form:	*VR+ADVZ*	*VR+ADVZ*	*VR+ADVZ*

Following table shows the reduplicated process in one to one correspondence form of (t) in its surface to lexical representation –

Table 2.4. Constituent morphemes of the Reduplicated Adverb in (t)

Surface form	*Lexical form*
cət	VR
nə	ADVZ
cət	VR
nə	ADVZ

So, it is a case of complete reduplication which emphasizes the adverbial meaning.

The second case, i.e. B), is a case of partial reduplication, as it copies the part of the word, sans the adverbial nə. The reduplicated part has a prefix to the VR, which reduplicates along with the VR.

(v) tə-seŋ tə-seŋ-nə /(crystal) clearly
(w) tə-yaŋ tə-yaŋ-nə /lightly
(x) tə-ru tə-ru-nə /cleanly

The simple and reduplicated form for (v) can be shown by the following table-

Table 2.5. Simple and Reduplicated Forms of (v)

	Simple form	Reduplicated form	
Lexical form:	FP+VR+ADVZ	FP+VR	FP+VR+ADVZ
Surface form:	tə-seŋ-nə	tə-seŋ	tə-seŋ-nə

When this process is captured at the morpheme level in its surface to lexical form of the above wordform (q), it can be shown in the following way-

Table 2.6. Constituent Morphemes of the Reduplicated Adverb in (v)

Surface form	Lexical form
tə	FP
seŋ	VR
tə	FP
seŋ	VR
nə	ADVZ

Like adjectives, adverbs are derived from polysyllabic stems (case C) and undergo reduplication without the adverbial suffix, such as

(y) phəjə phəjə-nə /nicely
(z) kəp-niŋ kəp-niŋ-nə /cryingly
(aa) təpləp təpləpnə /(little bit) slowly

So in the simple and reduplicated form, in this case, it undergoes the same process of derivation as that of adjectives, the following shows it for (z)-

Table 2.7. Simple and Reduplicated Forms of (z)

	Simple form	Reduplicated form	
Lexical form:	VR+Suff+ADVZ	VR+Suff	VR+Suff+ADVZ
Surface form	kəp+niŋ+nə	kəp+niŋ	kəp+niŋ+nə

The general form of this reduplication can be illustrated as-

(VR + suffixes) (VR + suffixes) + ADVZ

It is worth mentioning here that only the polysyllabic part of the verb stem is doubled. The adverbial suffix is attached after the process of reduplication forming the reduplicated adverb.

The reduplication in the last case is a case of restricted reduplication. Only the verb root doubles here. The formative prefix i as well as the adverbial suffix does not take part in reduplication. Examples are-

(ab) icum cumnə /very rightly
(ac) itəp təpnə /very slowly

This method is a variation among the other types of reduplication. It can be as shown below for (aa) –

Table 2.8. Constituent Morphemes of the Reduplicated Adverb in (aa)

Surface form	Lexical form
i	FP
cum	VR
cum	VR
nə	ADVZ

The verb root can also be replaced by a polysyllabic root, as in-

(ad) ip^h p^{h}əɟənə /nicely
(ae) iŋək ŋəktunə /surprisingly

In this case the following illustrates the process of reduplication for (ad) in its surface to lexical mapping form.

Table 2.9. Constituent morphemes of the reduplicated adverb in (ad)

Surface form	Lexical form
i	FP
ŋək	VR
ŋək	VR
tu	Determiner
nə	ADVZ

For all the four cases (A, B, C, and D) of reduplication, the following table shows the constituent reduplicated morphemes and their corresponding lexical forms-

Table 2.10. Constituent Reduplicated Morphemes of the four (A, B, C, & D) Cases of Reduplicatio

Form	Surface form	Lexical form
A) VR+ADVZ	cət	VR
	nə	ADVZ
	cət	VR
	nə	ADVZ
B) FP+VR	tə	FP
	seŋ	VR
	tə	FP
	seŋ	seŋ
	nə	ADVZ
C) VR+Suffix	kəp	VR
	niŋ	SUFF
	kəp	VR
	niŋ	SUFF
	nə	ADVZ
D) VR	i	FP
	seŋ	VR
	seŋ	VR
	nə	nə

It can be observed that the reduplicated forms of adverbs and adjectives exhibit a similar pattern in case of the polysyllabic roots. Both of these classes double the polysyllabic root and take the respective adjectival or adverbial suffixes.

The following section will describe ways to implement the above reduplication processes by the basic method of describing lexicon in regular expression terms and the application of the compile-replace algorithm of the Xerox tool package.

III. Capturing Reduplication

The concatenation operation of finite-state automata theory and its ability to represent concatenative morphotactics through regular expressions is well known in the field. Most languages build words primarily by concatenating morphemes together. The mechanism of continuation classes in Two-Level Morphology [5] translates into CONCATENATION in regular expressions which are the basic mechanism for describing morphotactics in LEXC [6].

However, the mechanism failed for being inadequate in describing the morphotactics phenomena of nonconcatenative morphology like reduplication, separated dependencies, stem interdigitation, infixation, etc.

Using basic finite state methods, SALAMA, the Swahili Language Manager [7], reduplication was implemented by using the basic finite state concatenation operation. K. R. Beesley and Karttunnen, L., 2003 [6], showed the handling of fixed-length form reduplication in Tagalog using classic Two-Level Morphology [7] which involves copying the first CV syllable of a verb root. They have also shown the Malay full stem reduplication using the compile-replace algorithm of the finite-state morphology[6]. We will adhere to the method based on finite state calculus for implementing reduplication –by using the compile-replace algorithm and show how this algorithm is best suited for describing the grammar of reduplication for both adjectives and adverbs of Manipuri language.

A. Compile-Replace Algorithm

The compile-replace algorithm of Xerox tool package makes it possible to include finite state operations other than concatenation into the morphotactics description [6]. And the basic idea behind this is the fact that regular expressions can easily notate to concatenations of a string, any number of times. Meaning, in finite-state terms any string s can be reduplicated as {*s*}^2 which denote concatenations of s twice.

The basic requirement for the application of compile-replace algorithm is that the regular-expression substrings in the network to be modified by **compile-replace** must be delimited by the **^[** and **^]** multicharacter symbols. These delimiters must be declared as Multichar Symbols in the LEXC file description. It is also required to be ensured that every substring of characters appearing on a subpath between **^[** and **^]** must be compilable as a regular expression before **compile-replace** can work successfully [6].

Using compile-replace is a two step process of compiling finite state networks for capturing the reduplication phenomena. Both the steps produce finite-state transducer networks. The initial transducer (using **LEXC** or regular expressions) relates two regular languages of which the lower language contains delimited substrings those themselves are valid XFST regular expressions.

The description of the LEXC file to illustrate grammars for reduplication of adverbs and adjectives are defined as follows- upper-side strings are built by the usual fashion of concatenation of morphemes and their respective continuation classes; while the lower-side strings are built by straightforward concatenation of a prefix ^[, a root enclosed in braces, and an abstract overt-adverbial/adjectival suffix and ^2 followed by the closing delimiter ^], in the respective order of their formation, as shown below in I-VI.

i. *AdjM.txt (LEXC of reduplicated adjectives for the duplication of (FP+VR))*

```
Multichar_Symbols
+Redp ^[  ^] FP+ +VR +ADJ +NZR
LEXICON Root
        FP+: ^[{ə          VerbRoots;
LEXICON VerbRoots
        kən      Verb;
        ca       Verb;
LEXICON Verb
        +VR:0    Redupli;
LEXICON Redupli
        +Redp:}^2^]      AdjSuffix;
LEXICON AdjSuffix
        +NZR:bə          Adjective;
LEXICON Adjective
        +ADJ:0  #;
END
```

The concatenation of suffix bə to the duplicated word is represented in the lexicon in the usual fashion of LEXC file structure, using a continuation class. By simply putting the portion of the stem to be doubled, inside the ^[{ and }^2^], the suffix is concatenated using AdjSuffix continuation class.

The result of the compilation produces a network of which the lower forms are yet other regular expressions. The upper and lower symbols have been lined up for easy reading.

```
UPPER:  FP+kən+VR+Redp+NZR+ADJ
LOWER:^[{ə kən    }  ^ 2 ^]  bə

UPPER:    FP+ca+VR+Redp+NZR+ADJ
LOWER:^[{ ə ca     } ^ 2 ^]  bə
```

The ultimate network after applying the compile-replace algorithm to the lower language is shown below:

```
LEXICAL:FP+kən+VR+Redp+NZR+ADJ
SURFACE: əkən əkənbə

LEXICAL:FP+ca+VR+Redp+NZR+ADJ
SURFACE: əca əcabə
```

In case of polysyllabic roots and stems, reduplicated forms of adverbs and adjectives simply double the root/stem and requires respective suffix. The LEXC file description for reduplicated adjective, derived from the polysyllabic root p^həʃə is given below in II below-

ii. <u>AdjP.txt</u> *(LEXC for reduplicated adjective duplication of (VR+ suffix(es))*

```
Multichar_Symbols
        +VR  ^[  ^]  +NZR +Redp  +ADJ
LEXICON ROOT
        0:^[{     VerbRoots;
LEXICON VerbRoots
        pʰəʃə     Verb;
LEXICON Verb
        +VR:0   Redupl;
LEXICON Redupl
        +Redp:0}^2^]     AdjSuffix;
LEXICON AdjSuffix
        +NZR:bə  Adjective;
LEXICON Adjective
        +ADJ:0  #;
END
```

The XFST compilation of the above LEXC file yields the following upper and lower languages respectively.

```
UPPER:    phəʃə+VR+Redp+NZR+ADJ
LOWER:^[{phəʃə    } ^ 2 ^] bə
```

The lower language produced are regular expressions, to which the compile-replace algorithm would be applied to produce the valid reduplicated surface forms of the polysyllabic root. The result of applying the compile-replace algorithm hence gives the following valid reduplicated surface forms of the adjective for the polysyllabic root p^həʃə.

```
LEXICAL:phəʃə+VR+Redp+NZR+ADJ
SURFACE: phəʃə phəʃəbə
```

The LEXC file description for adverb formation mechanism of verb root tou and ca by duplicating the verb root and adverbial suffix nə is as shown below in III.

iii. <u>AdvM.txt</u> *(LEXC for reduplicated adverbs for the duplication of (VR+ADVZ))*

```
Multichar_Symbols
        +VR  ^[  ^]  +ADV +Redp +ADVZ
LEXICON ROOT
        0:^[{     VerbRoots;
LEXICON VerbRoots
        tou      Verb;
        ca       Verb;
LEXICON Verb
        +VR:0   ADVSuff;
LEXICON ADVSuff
        +ADVZ:nə  Redupl;
LEXICON Redupl
        +Redp:0}^2^]     Adverb;
LEXICON Adverb
        +ADV:0#;
END
```

When the above file is compiled through XFST, it produces a transducer network, of which the lower side language itself is a regular expression. The following is the result from the first compilation of the network. The symbols have been lined up for easy comparison with the grammar.

```
UPPER:    tou+VR+ADVZ+Redp+ADV
LOWER:^[{tou       nə    } ^ 2 ^]
UPPER:    ca +VR+ADVZ+Redp+ADV
LOWER:^[{ca        nə    } ^ 2 ^]
```


As it can be seen that the lower strings are not a valid surface string, but are yet other regular expressions of the form {s}^2, which is the abstract meta-description of full-stem reduplications; where s is a string of any characters.

When the compile-replace algorithm is applied to the lower language of this network, it recognizes each individual delimited regular-expression substring such as ^[{touna}^2^], compiles and replaces it with the result of the compilation, to give a finite-state transducer with the following upper and lower languages.

LEXICAL: tou+VR+ADVZ+Redp +ADV
SURFACE: tounə tounə

LEXICAL: ca +VR+ADVZ+Redp +ADV
SURFACE: canə canə

Now, the lower languages aka surface forms are the valid reduplicated adverbs in the language. However, it is a matter of choice whether to include a hyphen, a space between the reduplicated stems, or leave it in its concatenated form. In any case, it should be treated as a single word form and tagged as an adverb by a POS tagger (or mapped as a reduplicated adverb by a morphological analyzer).

iv. *AdvFPFP.txt (LEXC for reduplication of formative prefix tə with monosyllabic verb roots seŋ, and yaŋ (FP+VR))*

```
Multichar_Symbols
+VR ^[ ^] +ADVZ +Redp FP+ +AdV
LEXICON ROOT
        FP+:^[{tə       VerbRoots;
LEXICON VerbRoots
        yaŋ     Verb;
        səŋ     Verb;
LEXICON Verb
        +VR:0   Redupl;
LEXICON Redupl
        +Redp:0}^2^]    AdvSuffix;
LEXICON AdvSuffix
        +ADVZ:nə Adverb;
LEXICON Adverb
        +AdV:0 #;
END
```

The compilation of the above lexicon results a transducer network with the following upper and lower language for the verb roots yaŋ and seŋ respectively,

UPPER:FP+yaŋ+VR+Redp+ADVZ +AdV
LOWER:^[{tə yaŋ} ^ 2 ^] nə

UPPER:FP+seŋ+VR+Redp+ADVZ +AdV
LOWER:^[{təseŋ} ^ 2 ^] nə

Compile-replace algorithm application to the lower language results in the following valid reduplicated adverbs of the two verb roots, viz. yaŋ and seŋ.

LEXICAL:FP+yaŋ+VR+Redp+ADVZ +AdV
SURFACE:təyaŋ təyaŋnə

LEXICAL:FP+səŋ+VR+Redp+ADVZ+AdV
SURFACE: təsəŋ təsəŋnə

v. *AdvP.txt (LEXC for reduplicated adverbs of the form (VR + suffix (es))*

```
Multichar_Symbols
+VR ^[ ^] +ADV +Redp +ADVZ
LEXICON ROOT
        0:^[{   VerbRoots;
LEXICON VerbRoots
        phəʃə   Verb;
        caniŋ Verb;
LEXICON Verb
        +VR:0   Redupl;
LEXICON Redupl
        +Redp:0}^2^]    ADVSuff;
LEXICON ADVSuff
        +ADVZ:nə Adverb;
LEXICON Adverb
        +ADV:0#;
END
```

The XFST compilation of the above lexicon V yields the following upper and lower language.

UPPER:phəʃə+VR+Redp+ADVZ+ADV
LOWER:^[{phəʃə } ^ 2 ^] nə

UPPER: caniŋ+VR+Redp+ADVZ+ADV
LOWER:^[{caniŋ } ^ 2 ^] nə

Here also, the lower language produced are regular expressions, to which the compile-replace algorithm would be applied to produce the valid reduplicated surface forms of the polysyllabic root.

An application of the compile-replace algorithm to the above lower languages produces the following valid reduplicated surface forms of both adjective and adverb respectively, for the polysyllabic root $p^həʃə$ and caniŋ respectively.

LEXICAL:phəʃə+VR+Redp+ADVZ+ADV
SURFACE: phəʃə phəʃənə

LEXICAL:caniŋ+VR+Redp+ADVZ+ADV
SURFACE:caniŋ caniŋnə

The lexicon for reduplicated adverbs where reduplication is only the verb root is shown below in VI. As illustrated in the section 2, this adverb form does not exist in its simple form and is different from other forms of reduplication. Without the formative prefix (FP) i, the reduplication takes the form of lexicon I. i.e.AdvM.txt.

vi. *VI. AdvFP.txt (LEXC for reduplicated adverbs of the form (VR))*

```
Multichar_Symbols

+VR  ^[  ^]  +ADVZ +Redp  FP+ +AdV
LEXICON ROOT
        FP+:i     VerbRoots;
LEXICON VerbRoots
        0:^[{     Roots;
LEXICON Roots
        cum       Verb;
        tən       Verb;
LEXICON Verb
        +VR:0   Redupl;
LEXICON Redupl
        +Redp:0}^2^]     AdvSuffix;
LEXICON AdvSuffix
        +ADVZ:nə  Adverb;
LEXICON Adverb
        +AdV:0 #;
END
```

The following is the upper and lower language resulted from the application of the XFST compilation of the above lexicon VI.

The XFST compilation of VI:

UPPER:FP+cum+VR+Redp+ADVZ+AdV
LOWER:i^[{cum} ^ 2 ^] nə

UPPER: FP+tən+VR+Redp+ADVZ+AdV
LOWER:i^[{tən }^ 2 ^] nə

And the result of applying compile replace algorithm to the lower language of the network gives the reduplicated adverbs of the two verb roots.

LEXICAL:FP+cum+VR+Redp+ADVZ+AdV
SURFACE: icum cumnə

LEXICAL:FP+tən+VR+Redp+ADVZ+AdV
SURFACE: itən tənnə

We perform the compilations shown above by using a source file. The five lexicons I, II, III, IV, V, and VI are combined together with the union operation to form a single network transducer. The compile-replace is then applied to the lower language.

The lexical form of the reduplicated word $p^həʃə$ $p^həʃəbə$ is analyzed by the resulting network as follows:

XFST[1]: up $p^həɸəphəɸbə$
$p^həɸ$+VR+Redp+NZR+ADJ
XFST[1]:

The network can perform from the other direction also, i.e. generating a reduplicated form from a given verb root.

XFST[1]: down $p^həɸ$+VR+Redp+NZR+ADJ
$p^həɸp^həɸbə$
XFST[1]:

IV. Analysis

The finite-state network has been tested with 400 (100 monosyllabic and remaining polysyllabic) verb roots on each of the six lexicon. The implementation of the reduplication process of both the monosyllabic and polysyllabic roots in Manipuri shows that the usual LEXC network and the compile-replace algorithm in the XFST tool package can successfully capture and analyze the reduplicated adjectives and adverbs of the language. Also as a matter of fact, the morphophonemic alternation rules can be applied easily to the lower language after the application of the compile-replace algorithm.

The present writing system of Manipuri uses Bengali script. The script cannot represent the language perfectly and completely and so the issue remains the same for spelling rules in the language as there is no standardized document on grammar and spelling rule of the language while using Bengali script. An instance of such an occurrence is while applying morphophonemic rule for the adjectival suffix bə/pə following a t (unvoiced) or an t̬ (voiced).

The rule in the XFST regular expression notation is

bə->pə || t _;

where bə is converted to a pə in an environment where it follows an unvoiced t.

There is no proper standard mechanism to represent such situations occurring inside a word structure with the available grammar and spelling rules for computational implementation of these linguistic variants.

V. Conclusion

We have presented two non-concatenative processes, complete and restricted reduplication, that take place in adjectives and adverbs of Manipuri language at the word level. The paper has also discussed the challenges involved in the morphological analysis of the two classes. We have presented and tested methods for capturing and analyzing the reduplication using XFST and LEXC. On the basis of the result, we conclude that the compile-replace algorithm in the environment offered by the Xerox tool package offers elegant solutions to the problems discussed. As a matter of fact, it is worthy to mention that this tactics of solving the problem of morphological analysis of reduplicated words in the language will definitely contribute to the mechanism of finite-state based morphological analysis of Manipuri language

The presented technique in the finite-state environment can be extended to other classes of the language, such as nominal category and wh-words, for similar problems of reduplication.

REFERENCES

[1] Muhirwe, Jackson, and Trond Trosterud: Finite State Solutions For Reduplication In Kinyarwanda Language, IJCNLP. 2008.

[2] Nongmeikapam, Kishorjit, and Sivaji Bandyopadhyay: Identification of MWEs Using CRF in Manipuri and Improvement Using Reduplicated MWEs, Proceedings of the International Conference on Natural Language Processing, India. 2010.

[3] Grierson:G.A. Linguistic Survey of India, Vol. III, Pt. II Delhi-Varanasi: Motilal Banarasidas (ed.). 1903-28, (Reprinted 1967-68).

[4] Singha, Ksh Krishna B., Kh Raju Singha, and Bipul Syam Purkayastha.: Morphotactics of Manipuri Adjectives: A Finite-State Approach, International Journal of Information Technology and Computer Science (IJITCS) 5.9 (2013): 94.

[5] Koskenniemi, K..: *Two-level morphology: a general computational model for word-form recognition and production*. Publication No. 11. University of Helsinki: Department of General Linguistics. 1983.

[6] Beesley, K. AND Karttunen L.: Finite State Morphology: CSLI Studies in Computational Linguistics. Stanford University, CA: CSLI Publications. 2003.

[7] Hurskainen, Arvi.: Solutions for Handling Non-concatenative Processes in Bantu Languages, Inquiries into Words, Constraints and Contexts (2005): 45.

[8] Bhat, D. N. S.: The Adjectival Category. Amsterdam, John Benjamins Publishing Company, 1994.

[9] Chelliah, L. Subhana,: A Grammar of Meithei. Berlin, Mouton de Gruyter, 1997.

[10] Hurskainen, Arvi.: "Swahili language manager: A storehouse for developing multiple computational applications." Nordic Journal of African Studies 13.3 (2004): 363-397.

[11] Singh, Yashawanta Ch.: Manipuri Grammar. New Delhi, Rajesh Publications, 2000.

[12] Kotze, Petronella M., and Winston N. Anderson. "A computational morphological analyser for Northern Sotho deverbative nouns: applying Xerox finite-state software to traditional grammar." South African journal of African languages 25.1 (2005): 59-70.

[13] Beesley, Kenneth R. "Finite-state morphological analysis and generation of Arabic at Xerox Research: Status and plans in 2001." ACL Workshop on Arabic Language Processing: Status and Perspective. Vol. 1. 2001.

Authors' Profiles

Ksh. Krishna B Singha is a Research scholar, in the Department of Computer Science, Assam University, Silchar, India since 2010. She did her Master of Computer Applications from Manipur University, Canchipur, India. Recently she has completed her degree for the Doctor of Philoosophy in Computer Science. Her research interests include Morphological Analysis, Machine Learning, POS Tagging, Fuzzy Set Theory, Fuzzy Logic, etc.

Dr. Kh. Dhiren Singha is an Associate Professor in the Department of Linguistics, Assam University, Silchar, India. In addition to being a Principal Investigator for a project on 'MEYOR LANGUAGE' of Arunachal Pradesh, India, under the scheme for protection and preservation of endangered languages of the central institute of indian languages (ciil), mysore, he is also the editor of indian language review. He is the author of three books: An Introduction to Dimasa Phonology, Ahni Grao: My Language and Dimasa Word Book: A Classified Vocabulary. His research interests include Phonology, Morpho-syntax, Language Typology and Tibeto-Burman Linguistics.

Dr. Bipul Syam Purkaystha is a Professor and Head of the Department of Computer Science, Assam University, Silchar, India. Currently he is also the Dean of the School of Physical Sciences of the same University. His research interests include Computational Linear Algebra, Natural Language Processing.

How to cite this paper: Ksh. Krishna B Singha, Kh. Dhiren Singha, Bipul Syam Purkaystha,"A Morphological Analyzer for Reduplicated Manipuri Adjectives and Adverbs: Applying Compile-Replace", International Journal of Information Technology and Computer Science(IJITCS), Vol.8, No.2, pp.32-40, 2016. DOI: 10.5815/ijitcs.2016.02.04