Research Report on Bangla Verb and Noun Morphological Analysis

Md. Zahurul Islam
Center for Research on Bangla Language Processing, BRAC University
zaharul@bracu.ac.bd

Abstract

This report describes the inflection Bangla verb and noun morphology and rules, lexicons and grammar for Bangla morphological analysis.

1. Introduction

This report describes Bangla verb and noun morphology and also the two level rules, lexicon and unification based grammar for Bangla verbs and nouns. These rules, lexicon and grammar are based on PC-KIMMO (a two level morphological Analyzer) and JKimmo (A multilingual computation morphology frame work for PC-KIMMO). JKimmo is a multilingual wrapper around PC-KIMMO that enables to use Bangla language for input and output.

2. Bangla morphological analysis

2.1. Verbs

Verbs divide into two classes: finite and non-finite. Non-finite verbs have no inflection for tense or person, while finite verbs are fully inflected for person (first, second, third), tense (present, past, future), aspect (simple, perfect, progressive), and honor (intimate, familiar, and formal), but not for number. Conditional, imperative, and other special inflections for mood can replace the tense and aspect suffixes. The number of inflections on many verb roots can total more than 200. A few example of Bangla verb inflexion are given below in Table 1.

Table 1: Verb inflections

	Present				**Past**				**Future**	
Verb Root	**Simple**	**Continuous**	**Perfect**	**Subjunctive**	**Simple**	**Habitual**	**Continuous**	**Perfect**	**Simple**	**Subjunctive**
কর (1st person)	করি	করছি	করেছি		করলাম	করতাম	করছিলাম	করেছিলাম	করব	
বল (2nd Person)	বল	বলছ	বলেছ	বলো	বললে	বলতে	বলছিলে	বলেছিলে	বলবে	বলো
খাওয়া (3rd person)	খাওয়ায়	খাওয়াচ্ছে	খাইয়েছে	খাওয়াক	খাওয়াল	খাওয়াত	খাওয়াচ্ছিল	খাইয়েছিল	খাওয়াবে	খাওয়াক

2.2. Noun

Nouns are inflected for case, including nominative, objective, genitive (possessive), and locative. The case marking pattern for each noun being inflected depends on the noun's degree of animacy. When a definite article such as -টা (singular) or -গুলো (plural) is added, as in the Tables (2 and 3) below, nouns are also inflected for number.

Table 2: Singular noun inflections

	Animate	Inanimate
Nominative	ছাত্রটা	জুতাটা
Objective	ছাত্রটাকে	জুতাটা
Genitive	ছাত্রটার	জুতাটার

	Animate	Inanimate
Locative		জুতাটাতে

Table 3: Plural noun inflections

	Animate	Inanimate
Nominative	ছাত্ররা	জুতাগুলো
Objective	ছাত্রদের(কে)	জুতাগুলো
Genitive	ছাত্রদের	জুতাগুলোর
Locative		জুতাগুলোতে

3. Components

3.1. Transliteration file

The original PC-KIMMO software is written in C programming language and uses only Latin alphanumeric characters for input and output purposes. For inputs using scripts other than Latin, the user has to come up with his/her own transliteration scheme that uses Latin characters corresponding to characters of the non-Latin script. Viewing and understanding the input and output strings in such a way can be cumbersome and non-intuitive for the user.

JKimmo solves this problem in a modular, abstract fashion. It requires that the whole transliteration scheme be written down in a separate file. The user can then load that transliteration file. Once the transliteration file is loaded, the user can input strings and view output strings in his preferred language in an intuitive way. Transliteration scheme for Bengali language is given in Table 4.

3.2. Rule file

Two level orthographic rules are required for JKimmo and PC-KIMMO. The general structure of the rules file is a list of declarations composed of a keyword followed by data. The set of valid keywords in a rules file includes COMMENT, ALPHABET, NULL, ANY, BOUNDARY, SUBSET, RULE, and END. The COMMENT, SUBSET and RULE declarations are optional and also can be used more than once in a rules file. The END declaration is also optional, but can only be used once. PC-KIMMO only recognizes Latin characters in rule file. To implement rule for language that uses other than Latin script we must follow the transliteration scheme. There is a free rule compiler for PC-KIMMO called kgen is available. It takes rule specification and it generate rule for PC-KIMMO. There are more free tools available that can be used for rule generation. A sample rules file has shown in Table 5.

Table 4: Bengali transliteration scheme

Bangla	Latin	Bangla	Latin	Bangla	Latin	Bangla	Latin	Bangla	Latin
◌্	^	◌া	a	গ	G	ণ	N	র	r
অ	A	◌ি	I	ঘ	G	ত	t	ল	l
আ	F	◌ী	I	ঙ	?	থ	T	শ	S
ই	H	◌ু	u	চ	C	দ	d	ষ	$
ঈ	L	◌ূ	U	ছ	C	ধ	D	স	s
উ	M	◌ৃ	R	জ	J	ন	n	হ	h
ঊ	Q	◌ে	e	ঝ	J	প	p	ড়	'
ঋ	V	◌ৈ	E	ঞ	Q	ফ	P	ঢ়	"
এ	W	◌ো	o	ট	V	ব	b	য়	Y
ঐ	X	◌ৌ	O	ঠ	W	ভ	B	◌ং	%
ও	Z	ক	k	ড	X	ম	m	◌ঃ	&
ঔ	F	খ	K	ঢ	Z	য	y	◌ঁ	~

Table 5: Sample rule file

```
;MAIN RULE FILE BANGLAMORPHOLOGY.RUL

ALPHABET
   k K g G ? c C J q v w x z N t T d D n p P b B m y r l S $ s h ' " Y & ~ ^ a i I u U R e E o O A F H L M Q V W X Z f + j
NULL 0
ANY @
```

```
BOUNDARY #
SUBSET Cons   k K g G ? c C j J q v w x z N t T d D n p P b B m y r l S $ s h ' " Y & ~ ^
SUBSET KhaGon   K d p D

RULE "defaults" 1 31
   k K g G ? c C J q v w x z N t T d D n p P b B m y r l S $ s @
   k K g G ? c C J q v w x z N t T d D n p P b B m y r l S $ s @
 1: 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

RULE "defaults" 1 30
   h ' " Y & ~ ^ a i I u U R e E o O A F H L M Q V W X Z f + @
   h ' " Y & ~ ^ a i I u U R e E o O A F H L M Q V W X Z f 0 @
 1: 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

RULE " a:e <= KhaGon:@ _ +:0 Y e C | KhaGon:@ _ +:0 l | KhaGon:@ _ +:0 t" 6 10

    a  a  KhaGon  +  Y  e  C  l  t   @
    e  @       @  0  Y  e  C  l  t   @
 1:  1  1      2  1  1  1  1  1  1   1
 2:  1  3      2  1  1  1  1  1  1   1
 3:  1  1      2  4  1  1  1  1  1   1
 4:  1  1      2  1  5  1  1  0  0   1
 5:  1  1      2  1  1  6  1  1  1   1
 6:  1  1      2  1  1  1  0  1  1   1
END
```

3.3. Lexicon files

PC-KIMMO lists lexical items (indivisible words and morphemes) in their underlying forms, and encodes morphotactic constraints. Its main job is to decompose a word into a sequence of morphemes using a simple positional analysis. The positional analysis need only go far enough to ensure that all correct parses are produced but not too many incorrect parses. Co-occurrence restrictions between morpheme positions are best handled in the word grammar, not the lexicon. A lexicon consists of one main lexicon file plus one or more files of lexical entries. The general structure of the main lexicon file is a list of keyword declarations. The set of valid keywords is ALTERNATION, FEATURES, FIELDCODE, INCLUDE, and END. To write lexicons that will be used in JKimmo for language that uses other than Latin script then we has to follow the transliteration scheme. Main lexicon file is given in Table 6.

Table 6: Main lexicon file

```
;MAIN                         LEXICON                         FILE
BANGLAMORPHOLOGY.LEX
ALTERNATION Vroot      VROOT
ALTERNATION Aspect      SADHARON GHOTOMAN
PURAGHOTITO NITTOBRITTO ANUGGA
ALTERNATION  Time                  BORTOMAN  ATIT
BHOBISHSHOT
ALTERNATION Person       DITIO TRITIO PROTHOM
DITIOTU DITIOGOU TRITIOGOU
```

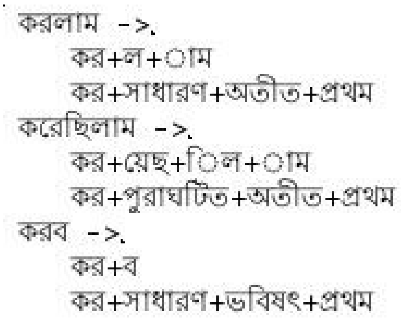
3.4. Grammar file

Grammar is optional for PC-KIMMO. When the morphemes are given by the lexicon section they are combined using a word parser. Here correlations between different morphemes are considered using feature unification. The grammar file contains three sections. The first section of the grammar file contains feature abbreviations. Feature abbreviations can be used either in lexical entries or in grammar rules and are expanded by "Let" statements. For example, the feature abbreviation *pl* is expanded into the feature structure [number: PL]. The second section of the grammar file contains category templates. These are feature specifications that are attached to lexical categories such as Noun and Adjective. This greatly reduces the amount of information that must be stored in the lexicon. For example, the statement Let N be <number> = SG means that all nouns are assigned singular number. The third section of the grammar file contains the word grammar rules. Associated with each rule are feature constraints. A feature constraint consists of two feature structures that must unify with each other. Feature constraints have two functions: they constrain the operation of a rule and they pass features from one node to another up the parse tree. A sample grammar file is given in Table 7.

4. Results

4.1. Recognition

Sample recognition output is given in Figure 1.



```
করলাম ->.
    কর+ল+◌াম
    কর+সাধারণ+অতীত+প্রথম
করেছিলাম ->.
    কর+য়েছ্+◌িল+◌াম
    কর+পুরাঘটিত+অতীত+প্রথম
করব ->.
    কর+ব
    কর+সাধারণ+ভবিষৎ+প্রথম
```

Figure 1: Sample recognition output

Table 7: Sample grammar file

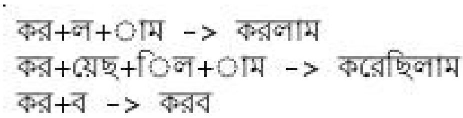
```
;GRAMMER FILE BANGLAMORPHOLOGY.GRM
Let vr be <pos> = VB
Let sd be <aspect> = SD
Let pg be <aspect> = PG
Let gh be <aspect> = GH
Let an be <aspect> = AN
Let nt be <aspect> = NT
Let b be <time> = B
Let a be <time> = A
Let v be <time> = V
Let p be <person> = P
Let d be <person> = D
Let t be <person> = T
Let dtu be <person> = DTU
Let dgo be <person> = DGO
Let tgo be <person> = TGO

Let SADHARON be <cat>=ASPECT
Let GHOTOMAN be <cat>=ASPECT
Let PURAGHOTITO be <cat>=ASPECT
Let NITTOBRITTO be <cat>=ASPECT
Let ANUGGA be <cat>=ASPECT

RULE
Word -> Verbroot VSuffix1
            <Word aspect> = <VSuffix1 aspect>
            <Word time> = <VSuffix1 time>
            <Word person> = <VSuffix1 person>
            <Word pos> = <Verbroot pos>
END
```

4.2. Generation

Sample generation output is given in Figure 2.



```
কর+ল+◌াম -> করলাম
কর+য়েছ্+◌িল+◌াম -> করেছিলাম
কর+ব -> করব
```

Figure 2: Sample generation output

5. Conclusion

This report presents Bangla verb and noun morphology and rules, lexicons and grammar for inflectional verb and noun morphology.

6. References

[1] M.Z. Islam and M. Khan, “JKimmo: A Multilingual Computational Morphology Framework for PC-KIMMO”, *Proc. of 9th International Conference on Computer and Information Technology, ICCIT 2006*, Dhaka, Bangladesh, 2006.

[2] S. Dasgupta and M. Khan, “Morphological Parsing of Bangla Words Using PC-KIMMO”, *Proc. 7th International Conference on Computer an Information Technology, ICCIT 2004*, Dhaka, Bangladesh, 2004.

[3] S. Dasgupta and M. Khan, “Feature Unification for Morphological Parsing in Bangla”, *Proc. 7th International Conference on Computer an Information Technology, ICCIT 2004*, Dhaka, Bangladesh, 2004.

[4] A.K.M. Morshed, *Adunik Vasatatto* - 2nd version, Noa Uddog, Kokata, 1997.