Development of a morphological analyser for Bengali

Abu Zaher Md. Faridee
Dept. of Comp. Science and Eng.,
Bangladesh Univ. of Eng. and Tech.,
Dhaka 1000
zaher14@gmail.com

Francis M. Tyers
Dept. Llenguatges i Sist. Informàtics,
Universitat d'Alacant,
Alacant. E-03070
ftyers@dlsi.ua.es

Abstract

This article describes the development of an open-source morphological analyser for Bengali Language using finite-state technology. First we discuss the challenges of creating a morphological analyser for a highly inflectional language like Bengali and then propose a solution to that using lttoolbox, an open-source finite-state toolkit. We then evaluate the performance of our developed system and propose ways of improving it further.

1 Introduction

Bengali language (also referred as *Bangla* by its native speakers) is an Indo-Aryan language which is mainly spoken in the region of eastern South Asia (also known as Bengal) comprising Bangladesh, the Indian State of West Bengal, southern Assam and part of Tripura. A literally rich language, its evolution can be traced back to Magadhi Prakrit and Sanskrit languages. There are an estimated 230 million speakers of the language world wide, making it sixth among the most spoken languages of the world.

Building a morphological analyser for Bengali using finite-state technology requires taking into account the highly inflectional properties of the language and the diverse vocabulary. This diversity can be attributed to the influence of different cultures and languages ranging from European languages like English, French, Dutch, Portuguese to Middle Eastern languages like Arabic, Persian as well as its own kin Hindi, Urdu, and Sanskrit language, resulting in a rich set of inflections. Negatives and adverbs sometimes also take the form of enclitic, being another reason for higher the inflection.

Bengali exhibits *diglossia*,[1] one of the two major forms is called *Shadhubhasha* and the other is called *Chalitabhasha* or SCB (Standard Colloquial Bengali). The main difference between these two forms are seen in verbal inflections and higher usage of *tatshama* words in *Shadhubhasha*. Though a lot of Bengali literature and formal or legal documents have been historically written in *Shadhubhasha*, in modern times, SCB has almost replaced it in day-to-day usage; therefore, we chose to create our morphological analyser based on SCB.

Bengali is a *classifier language* (Samit Bhattacharya and Basu, 2005), meaning the verb does not change its form based on the gender and number of the subject or object, rather on the tense, aspect, modality and person. Our work found Bengali having three tenses, *present*, *past* and *future*; two moods, *indicative* and *imperative*; four aspects, *simple*, *continuous*, *perfect* and *habitual*, which differs a little from the grammar books. We identified 59 separate inflections for each of the verbs. A sample suffix list for the Bengali verb কর [kar][2] 'do' is shown in table 1. It should be noted that the suffix added to the verb root is highly dependent on its syllabic structure leading to six ba-

[1] http://lrc.cornell.edu/asian/courses/bengali

[2] *NLK transliteration scheme, details at* http://en.wikipedia.org/wiki/Bengali_script#Romanization_Reference.

J.A. Pérez-Ortiz, F. Sánchez-Martínez, F.M. Tyers (eds.)
Proceedings of the First International Workshop on Free/Open-Source Rule-Based Machine Translation, p. 43–50
Alacant, Spain, November 2009

sic inflection rule for verbs (the actual number of inflectional paradigms for the verbs are however much more than that).

For creating this morphological analyser/generator we took a slightly different approach from more traditional Bengali grammars. For example, there are seven grammatical cases in Bengali traditional grammars (Chowdhury et al., 2000), but morphologically there are only four cases.[4] Animacy also plays a major role in the inflectional pattern for nouns. We identified that there are four levels of animacy encountered in Bengali text, *inanimate*, *animate*, *human* and *elite*. The animacy level of the nouns govern which case and plurality suffix are added to them. For example, in nominative case, to construct plural form the suffix রা [rā] and গণ [gan] are added after *human*, *elite* nouns respectively, but গুলো [gulō] is added after *inanimate* and *animate* nouns. The details are given in table 2. The inflection rules for gender of nouns vary radically, they are actually more derivational than inflectional. Since right now we only deal with inflectional morphology, we decided to treat genders as separate words.

Inflectionally similar to nouns, the pronouns do not have have genders. Proper nouns are harder to deal with in Bengali, there is no special way to readily identify proper nouns, unlike in English. We have several sub-categories like *anthroponym, toponym, cognomen, organization* etc. for them. A small number of Adjectives inflect on degree of comparison. They can also inflect on gender, but these can be considered *sankskritism* rather than general phenomena (Islam et al., 2007) and do not have much use in SCB. Comparative and superlative forms are normally generated by applying তর [tara] and তম [tama] suffixes. Adverbs do not inflect on degree.

Like most other languages, there are small number of exceptions to all the general set of rules, and these words are treated separately.

2 Design

The morphological analyser/generator was developed as a part of a new language pair for Apertium, hence the analyser data conforms to Apertium's `dix` format (Ortiz-Rojas et al., 2005). The data for the analyser/generator is contained in an XML file, the Bengali monolingual dictionary (*monodix*). This dictionary has two principal parts, the first one is the *pardef* section, which contains the inflection rules for a particular type of word, the other is the entry section, which contains the list of words stating which *pardef* they belong to. A part of pardef definition for a verb is given in figure 1.

When the transducer encounters the surface form 'ফিরছিল' [phirchila], it outputs lemma 'ফের' [phēr] along with tag `vblex, past, cnt, p3, infml`. Thus the morphological analyser realises *3rd person, informal, past continuous* form of verb 'ফের' from 'ফিরছিল'.

The benefit of using Apertium (particularly lt-toolbox, the toolset responsible for managing the monodix) is its robust architecture. The primary advantage is that the analyser can also double as a morphological generator. On the other hand, given an XML based monodix it can create a compiled dictionary which is generally faster than a normal text or database based dictionary, optimal for running small memory footprint devices.

Bengali is distinct in nature as certain clitics are used to convey several adverbial meaning. For example, আপনি [āpni] - *you*, but আপনিও [āpnio] - *you too*. Here the clitic ও is used to convey the meaning of adverb 'also'. So we create a new *pardef* for this. A sample *pardef* for enclitic can be seen in figure 2.

As can be seen in table 1, the lemma form that we chose for the verb is a bit different from traditional dictionary formats where the verbs are generally represented in their gerund form. We chose to use the *present indefinite, second person, informal* inflected form as the lemma. The justification for this decision lies in the fact that this inflection is the shortest (in length) for every verb. Also, according to Sanskrit grammar, this form is analogous to the original stem (i.e. *dhatu*) after which affixes are added to create the real verb. Anubadok,[5] an open-source English to Bengali machine translation system from which a lot of linguistic data was derived, also uses this format for verb lemma. A potential disadvantage

[3] Although no suffix is added, the pronunciation here is different [karā]

[4] `http://en.wikipedia.org/wiki/Bengali_grammar#Case`

[5] `http://anubadok.sf.net`

কর [kar]		First	Second			Third and Impersonal	
			Polite	Familiar	Informal	Polite	Familiar
Ger.	া [ā]	n/a	n/a	n/a	n/a	n/a	n/a
Inf.	তে [tē]	n/a	n/a	n/a	n/a	n/a	n/a
Gen.	ার [ār]	n/a	n/a	n/a	n/a	n/a	n/a
Pres. Spl.	n/a	ি [i]	েন [ēn]	ø[3]	িস [is]	েন [ēn]	ে [ē]
Pres. Cont.	n/a	ছি [chi]	ছেন [chēn]	ছ [cha]	ছিস [chis]	ছেন [chēn]	ছে [chē]
Fut. Spl.	n/a	ব [ba]	বেন [bēn]	বে [bē]	বি [bi]	বেন [bēn]	বে [bē]
Past Spl.	n/a	লাম [lām]	লেন [lēn]	লে [lē]	লি [li]	লেন [lēn]	ল [la]
Past Hbl.	n/a	তাম [tām]	তেন [tēn]	তে [tē]	তি [ti]	তেন [tēn]	ত [ta]
Past Cont.	n/a	ছিলাম [chilām]	ছিলেন [chilēn]	ছিলে [chilē]	ছিলি [chili]	ছিলেন [chilēn]	ছিল [chila]
Perf.	n/a	েছি [ēchi]	েছেন [ēchēn]	েছ [ēcha]	েছিস [ēchis]	েছেন [ēchēn]	েছে [ēchē]
Plu Perf.	n/a	েছিলাম [ēchilām]	েছিলেন [ēchilēn]	েছিলে [ēchilē]	েছিলি [ēchili]	েছিলেন [ēchilēn]	েছিল [ēchila]
Part. Past.	ে [ē]	n/a	n/a	n/a	n/a	n/a	n/a
Part. Cond.	লে [lē]	n/a	n/a	n/a	n/a	n/a	n/a
Imp. Pres.	n/a	n/a	ুন [un]	ø	ø	n/a	n/a
Imp. Fut.	n/a	n/a	েন [ēn]	ো [ō]	িস [is]	n/a	n/a

***Legends**: Ger. - Gerund, Inf. - Infinitive, Gen. - Genitive, Pres. - Present, Spl. - Simple, Cont. - Continuous, Hbl. - Habitual, Perf. - Perfect, Part. - Participle, Imp. - Imperative, Fut. - Future*

Table 1: Suffix table for the Bengali verb কর [kar] ‘do’

	Inanimate		Animate		Human		Elite	
	Singular	Plural	Singular	Plural	Singular	Plural	Singular	Plural
Nominative	ø/টা [ṭā] /খানা [khānā] /খানি [khāni]	গুলো [gulō] /টুকু [ṭuku]	ø/টা [ṭā]	গুলো [gulō]	ø/টা [ṭā]	রা [rā]	ø	রা [rā] /গণ [gaṇa] /বৃন্দ [bṛinda]
Objective	ø/টা [ṭā] /খানা [khānā] /খানি [khāni]	গুলো [gulō] /টুকু [ṭuku]	কে [kē] /টাকে [ṭākē]	গুলোকে [gulōkē]	কে [kē] /টাকে [ṭākē]	দেরকে [dērkē]	কে [kē]	কে [kē] /গণকে [gaṇkē] /বৃন্দকে [bṛindakē]
Genitive	র [ra] /ের [ēr] /য়ের [yēr] /টার [ṭār] /খানার [khānār] /খানির [khānir]	গুলোর [gulōr] /টুকুর [ṭukur]	র [ra] /ের [ēr] /য়ের [yēr] /টার [ṭār]	গুলোর [gulōr]	র [ra] /ের [ēr] /য়ের [yēr] /টার [ṭār]	দের [dēr]	র [ra] /ের [ēr] /য়ের [yēr]	দের [dēr] /গণের [gaṇēr] /বৃন্দের [bṛindēr]
Locative	য়ে [yē] /য় [y] /তে [tē] /টায় [ṭāy] /খানায় [khānāy] /খানিতে [khānitē]	গুলোয় [gulōy] /টুকুতে [ṭukutē]	n/a	n/a	n/a	n/a	n/a	n/a

Table 2: Suffix table for Bengali noun inflections

```
<pardef n="ফ/ের__vblex">
  ... ... ...
  <e>
    <p>
      <l>িরছিল</l>
      <r>ের<s n="vblex"/><s n="past"/><s n="cnt"/><s n="p3"/><s n="infml"/></r>
    </p>
    <par n="enclitic"/>
  </e>
  <e>
    <p>
      <l>িরে</l>
      <r>ের<s n="vblex"/><s n="pcnd"/></r>
    </p>
    <par n="enclitic"/>
  </e>
  ... ... ...
</pardef>
```

Figure 1: Part of paradigm definition for ফ/ের__**vblex**

```
<pardef n="enclitic">
  <!-- pass-through -->
  <e>
    <p>
      <l />
      <r />
    </p>
  </e>
  <!-- Enclitic ই, (only) -->
  <e>
    <p>
      <l>ই</l>
      <r><j />ই<s n="adv" /></r>
    </p>
  </e>
  <!-- Enclitic ও, (also) -->
  <e>
    <p>
      <l>ও</l>
      <r><j />ও<s n="adv" /></r>
    </p>
  </e>
</pardef>
```

Figure 2: A paradigm definition for Bengali enclitics

of this approach is we cannot directly use existing dictionary from other resources, but the benefits outweighs this in most cases.

As mentioned earlier, initially we took advantage of the open-source English to Bengali machine translation tool called Anubadok. Anubadok served as a starting point for creating the rules section for this project, but we eventually realised that more extensive rules would be needed to build a high quality morphological analyser and generator. Although Anubadok is a functional English to Bengali machine translation system, morphological analysis was never its main focus. So we took the most basic approach, that is consulting the grammar books. Most of the inflection rules come from Klaiman (2009), Chowdhury et al. (2000) and Ray et al. (1966) as well as Wikipedia.[6]

Most of the part-of-speech tagged data came from Anubadok, but Anubadok uses the Penn Treebank tagset[7], so a converter was needed to convert the tags to our proposed tagset which is a superset[8] of Apertium's normal tagset. Also, each of our each lexical category carries more lexical data such as gender, animacy of nouns, mood and aspect of tense; we had to manually tag them. We wanted to remain faithful to Zipf's Law,[9] which Anubadok does not follow. So a lot of high frequency words that were not in Anubadok's dictionary had to be manually tagged. The frequency list of the words were obtained from CRBLP.[10] The list was created at CRBLP as per their *Prothom-alo lexicon project* , the analysis done on the frequency of Bengali words by crawling *Prothom-alo*,[11] one of the most circulated Bengali newspapers in Bangladesh. We were able to get a list of most frequently used 20,000 Bengali words from the project.

3 Development

The development of the XML dictionary was done using open source tools like Python, PHP, MySQL and shell-scripting. At first we focused on the open category words starting with verbs and then moving onto nouns, pronouns and adjectives etc. We first populated our database with lemmas, then made additional changes to the tables with proper animacy, gender, number tags. Then we put down the inflection rules in our python scripts. In some cases, we took the help of another intermediate format called *Speling format*[12] for ease of use. The scripts with the inflection rules generate the semi-colon delimited files which are then used to create the final Apertium dictionary. We then use lttoolbox to compile the text dictionary to binary format. The Speling format does not currently support enclitics, so it cannot be used when the words might take on additional enclitic, e.g. verbs and nouns. In these cases, the monodix is directly generated by the script.

Figures 3 and 4 show the pseudo code for generating the inflections of verbs. In line 2 of figure 3 we calculate the effective length of the verb root by removing ‘◌ঁ’ (*chandrabindu*) and ‘◌্’ (*hashant*). Then depending on the length of the verb, we choose appropriate function to generate the actual inflection. Figure 4 shows how

[6] http://en.wikipedia.org/wiki/Bengali_grammar

[7] http://www.cis.upenn.edu/~treebank/

[8] http://wiki.apertium.org/wiki/Bengali_and_English/TagSets

[9] http://en.wikipedia.org/wiki/Zipf's_law

[10] http://crblp.bracu.ac.bd

[11] http://www.prothom-alo.com

[12] http://wiki.apertium.org/wiki/Speling_format

Part of speech	Number of entries
Noun	2,035
Adjective	866
Proper noun	800
Adverb	432
Verb	136
Numeral	123
Post-position	46
Determiner	45
Pronoun	32
Conjunction	8
	4,523

Table 3: Number of entries in the lexicon for the main parts of speech

we are generating the inflected forms. First in line 1 and 2, we generate two type of umlauts for this particular kind of verb root. Normally this is a regressive vowel harmony, a phonological phenomenon attributive to SCB. But for some verbs like যা [ẏā], আছ [ācha], it is rather a morphological phenomenon (e.g. যা [ẏā] - গেল [gēla], আছ [ācha] - থাকত [thākta]). In the following lines we concatenate the root or the umlauts with the suffixes. It is to be noted that the actual code for these two procedure involves some complex regular expression matching (e.g. detection of *hashant* and *chandrabindu*, unicode normalisation, handling irregular verbs), something which was removed from the pseudo-code for simplicity and clarity.

A preliminary version of the verb conjugator can be found online[13] which generates all the forms for a particular verb. The current morphological analyser can also be accessed online.[14]

4 Evaluation

As table 3 shows, the number of currently tagged parts of speech is 4,523. This is not a very high number. However, as we shall later see, our dictionary puts emphasis on the most frequently used words, therefore achieving a higher effective coverage.

Doing an evaluation poses some major challenges for us. As we have stated before, this implementation of Bengali morphological analyser was created keeping in mind only the SCB (Standard Colloquial Bengali). However there are few digital SCB text resources available in the Internet, apart form *Wikipedia* and the *Prothom-alo* website. Therefore, our evaluation was also confined to corpus developed from *Wikipedia* and *Prothom-alo*. The data was collected by running a web crawler on these two web sites.

First we calculate the *naïve* coverage according to following formula:

$$\text{Coverage} = \frac{\text{No. of words with at least one analysis}}{\text{No. of words}}$$

Table 4 shows the naïve coverage for wikipedia and prothom-alo. The statistics clearly show optimization nature of our analyser. Since the frequency list was derived from prothom-alo, we get a higher coverage (80.35%) of prothom-alo than of wikipedia (68.21%).

Site	File size	Total	Recognised	Naïve
	(MB)	words	words	coverage
Wkipedia	27.1 MB	1,730,745	1,180,542	68.21%
Prothom-alo	23.4 MB	1,572,601	1,263,661	80.35%

Table 4: Naïve coverage for wikipedia and prothom-alo

For calculating *recall* and *precision* we also took two sets of data. First one is the list of most frequently used 1000 words (in their surface form) from CRBPL (Prothom-alo corpus). The second one is randomly selected 1000 word text excerpt from the web-crawler data of Prothom-alo. The formulas for calculating recall and precision are as follows:

$$\text{Precision} = \frac{\text{Number of correct analyses}}{\text{Number of analyses retrieved}}$$

$$\text{Recall} = \frac{\text{Number of correct analyses}}{\text{Total number of analyses}}$$

Table 5 shows the *recall* and *precision* values for the two sets of data. Precision is high in both cases. Recall value falls in the latter case (random excerpt), but this is still a high value.

[13] `http://xixona.dlsi.ua.es/~fran/bengali/conj/`

[14] `http://xixona.dlsi.ua.es/~fran/bengali/analysis.php`

```
PROCESS-VERB($verbList$)
for each $verb$ in $verbList$
    do $length \leftarrow$ length of $verb$ discarding ‘ঁ’ [chandrabindu] and ‘্’ [hashant]
       if $length > 2$                    ▷ Rule for verbs that are more than 2 characters long
         then if $verb$ ends with a marker preceded by a consonant
                then GET-INFLECTION-DO($verb$)          ▷ করা,পড়া etc.
              if $verb$ ends with a consonant preceded by a ‘ে’ or ‘ো’
                then GET-INFLECTION-WRITE($verb$)       ▷ লেখ, খেল etc.
              if $verb$ ends with a consonant preceded by a marker
                then GET-INFLECTION-SHAKE($verb$)       ▷ পাড়, নাড়, ছুট etc.
       elseif $length = 2$                ▷ Process 2 letter verbs
         then if $verb$ ends with a consonant preceded by an ও
                then GET-INFLECTION-WRITE($verb$)       ▷ ওঠ etc.
              if $verb$ ends with a consonant preceded by an vowel or a consonant
                then GET-INFLECTION-SHAKE($verb$)       ▷ আস, আন, আঁক etc.
              if $verb$ is ‘যা’
                then GET-INFLECTION-GO($verb$)
              if $verb$ ends with ে or ো preceded by a consonant
                then GET-INFLECTION-TAKE($verb$)        ▷ নে, দে etc.
              if $verb$ ends with a marker preceded by a consonant
                then GET-INFLECTION-EAT($verb$)         ▷ খা, পা, শু, নি etc.
         else                             ▷ Process single letter verbs
              if $verb$ ends with a consonant
                then GET-INFLECTION-EAT($verb$)         ▷ ক, হ etc.
```

Figure 3: Pseudo code for the procedure PROCESS-VERB

```
GEN-INFLECTION-TAKE($verb$)
$uml \leftarrow$ replace ‘ে’ with ‘ি’ and ‘ো’ with ‘ু’ in $verb$
$uml2 \leftarrow$ replace ‘ে’ with ‘া’ in $verb$
$I[Ger.] \leftarrow verb +$ ওয়া
$I[Inf.] \leftarrow uml +$ তে
$I[Gen.] \leftarrow verb +$ বার
$I[Pres.Spl.] \leftarrow verb +$ ই , $verb +$ ন , $uml2 +$ ও , $uml +$ স , $verb +$ ন , $verb +$ য়
$I[Pres.Cont.] \leftarrow uml +$ চ্ছি , $uml +$ চ্ছেন , $uml +$ চ্ছ , $uml +$ চ্ছিস , $uml +$ চ্ছেন , $uml +$ চ্ছে
$I[Fut.Spl.] \leftarrow verb +$ ব , $verb +$ বেন , $verb +$ বে , $uml +$ বি , $verb +$ বেন , $verb +$ বে
$I[Past.Spl.] \leftarrow uml +$ লাম , $uml +$ লেন , $uml +$ লে , $uml +$ লি , $uml +$ লেন , $uml +$ ল
$I[Past.Hbl.] \leftarrow uml +$ তাম , $uml +$ তেন , $uml +$ তে , $uml +$ তি , $uml +$ তেন , $uml +$ ত
$I[Past.Cont.] \leftarrow uml +$ চ্ছিলাম , $uml +$ চ্ছিলেন , $uml +$ চ্ছিলে , $uml +$ চ্ছিলি , $uml +$ চ্ছিলেন , $uml +$ চ্ছিল
$I[Perf.] \leftarrow uml +$ য়েছি , $uml +$ য়েছেন , $uml +$ য়েছ , $uml +$ য়েছিস , $uml +$ য়েছেন , $uml +$ য়েছে
$I[PluPerf.] \leftarrow uml +$ য়েছিলাম , $uml +$ য়েছিলেন , $uml +$ য়েছিলে , $uml +$ য়েছিলি , $uml +$ য়েছিলেন , $uml +$ য়েছিল
$I[Part.Past] \leftarrow uml +$ য়ে
$I[Part.Cond.] \leftarrow uml +$ লে
$I[Imp.Pres.] \leftarrow verb +$ ন , $uml2 +$ ও , $verb, verb+$ ন , $uml +$ ক
$I[Imp.Fut.] \leftarrow verb +$ বেন , $uml +$ ও , $uml +$ স , $verb+$ বেন , $uml +$ বে
```

Figure 4: Pseudo code for procedure GEN-INFLECTION-TAKE

Type	Correct	Retrieved	Total	Precision	Recall
Top 1000	1,626	1,723	1,631	99.6%	94.37%
Random 1000	1,328	1,504	1,330	99.8%	88.29%

Table 5: Precision and Recall

5 Discussion

Currently the analyser is in preliminary stage and there are lots of room for improvement. Firstly, the coverage needs to be expanded. This will require manual tagging of many words. Secondly, the verb section faces difficulty in treating multi-word verbs and the negative form are not well recognised. This is because several forms of the verb like infinitives and participles demand a negative particle before the verb while finite forms require the particle to follow the verb and in some cases as enclitic. Matters get complicated in the former case, multi-word verbs require the negative particle to be in the middle. Table 6 shows the negative forms of a multi-word verb কাজ কর 'work' clearly identifying the problem. This can be partially taken care of by a nested paradigm, but this makes the analyser slow, we need to come up with a better solution to solve this.

One possibility is porting this analyser (the python scripts) to a more robust architecture such as *foma* (Huldén, 2009), an open-source implementation of the Xerox finite-state tools (Beesley and Karttunen, 2003). It should be noted that there has been similar attempts to create finite-state technology based morphological analyser for Bengali with PC-KIMMO (Dasgupta and Khan, 2004) and JKimmo (Islam and Khan, 2006). However, PC-KIMMO is not Unicode compliant, it cannot be directly used for Bengali morphological analysis. On the other hand JKimmo requires a transliteration scheme. The solution we present here is Unicode compliant and requires no transliteration scheme. Furthermore, *foma* is fully able to handle Unicode characters, thus maintaining our Unicode compliance should we choose to port to it.

As mentioned earlier, this morphological analyser/generator was created as a part of a new language pair `bn-en` for Apertium. We are on our way of creating a functional English to Bengali translation system. The morphological analyser could also be used as a stemmer for any search engine for Bengali language. It could also double as a spell checker. In fact works are in the way to create a new Bengali dictionary for Firefox and OpenOffice.Org using this morphological analyser by Ankur.[15]

To our knowledge, this is the first open-source attempt in creating a fully-functional wide-coverage morphological analyser and generator for Bengali that is publicly available to everyone. All the tools that were used in this project are open source and the output, both the linguistic data and the toolset is available under the GNU GPL license.

Acknowledgments

Development was funded as part of the Google Summer of Code programme[16]. We are also very grateful to the personnel from CRBLP, specially Dr. Mumit Khan for allowing us to use their the most frequently used word-list. And special thanks to co-mentor Kevin Donnelly, who helped us a lot during the initial phase of the project and to Dr. G. M. Hossain, the author of the Anubadok system which was used during the development.

References

Beesley, K. R. and Karttunen, L. (2003). *Finite State Morphology*. CSLI Publications.

Chowdhury, S. M., Chowdhury, S. M. H., Khalil, M. I., Muhammad, D. K. D., and Lahiri, S. (2000). বাংলা ভাষার ব্যাকরণ *(A grammar of Bengali Language)*. National Curriculam and Textbook Board (NCTB).

Dasgupta, S. and Khan, M. (2004). Morphological parsing of Bangla words using PC-KIMMO. *Proceedings of the 7th International Conference on Computer and Information Technology (ICCIT2004, Dhaka, Bangladesh)*.

Huldén, M. (2009). Foma: a finite-state compiler and library. *EACL 2009*, pages 29–32.

Islam, M. Z. and Khan, M. (December 2006). Jkimmo: A multilingual computational morphology framework for pc-kimmo. *Proc. of*

[15] `http://ankur.org.bd/`

[16] `http://socghop.appspot.com/student_project/show/google/gsoc2009/apertium/t124021625239`

কাজ কর [kāz kar]		First person	Second person			Third person / Impersonal	
			Polite	Familiar	Informal	Polite	Familiar
Ger.	কাজ না করা [kāz nā karā]	n/a	n/a	n/a	n/a	n/a	n/a
Inf.	কাজ না করতে [kāz nā kartē]	n/a	n/a	n/a	n/a	n/a	n/a
Gen.	কাজ না করার [kāz nā karār]	n/a	n/a	n/a	n/a	n/a	n/a
Pres. Spl.	n/a	কাজ করি না [kāz kari nā]	কাজ করেন না [kāz karēn nā]	কাজ কর না [kāz kara nā]	কাজ করিস না [kāz karis nā]	কাজ করেন না [kāz karēn nā]	কাজ করে না [kāz karē nā]
Pre. Cont.	n/a	কাজ করছি না [kāz karchi nā]	কাজ করছেন না [kāz karchēn nā]	কাজ করছ না [kāz karcha nā]	কাজ করছিস না [kāz karchis nā]	কাজ করছেন না [kāz karchēn nā]	কাজ করছে না [kāz karchē nā]
Fut. Spl.	n/a	কাজ করব না [kāz karba nā]	কাজ করবেন না [kāz karbēn nā]	কাজ করবে না [kāz karbē nā]	কাজ করবি না [kāz karbi nā]	কাজ করবেন না [kāz karbēn nā]	কাজ করবে না [kāz karbē nā]
Past. Spl.	n/a	কাজ করলাম না [kāz karlām nā]	কাজ করলেন না [kāz karlēn nā]	কাজ করলে না [kāz karlē nā]	কাজ করলি না [kāz karli nā]	কাজ করলেন না [kāz karlēn nā]	কাজ করল না [kāz karla nā]
Past. Hbl.	n/a	কাজ করতাম না [kāz kartām nā]	কাজ করতেন না [kāz kartēn nā]	কাজ করতে না [kāz kartē nā]	কাজ করতি না [kāz karti nā]	কাজ করতেন না [kāz kartēn nā]	কাজ করত না [kāz karta nā]
Past. Cont.	n/a	কাজ করছিলাম না [kāz karchilām nā]	কাজ করছিলেন না [kāz karchilēn nā]	কাজ করছিলে না [kāz karchilē nā]	কাজ করছিলি না [kāz karchili nā]	কাজ করছিলেন না [kāz karchilēn nā]	কাজ করছিল না [kāz karchila nā]
Perf.	n/a	কাজ করিনি [kāz karini]	কাজ করেননি [kāz karēnni]	কাজ করনি [kāz karani]	কাজ করিসনি [kāz karisni]	কাজ করেননি [kāz karēnni]	কাজ করেনি [kāz karēni]
Plu Perf.	n/a	কাজ করিনি [kāz karini]	কাজ করেননি [kāz karēnni]	কাজ করনি [kāz karani]	কাজ করিসনি [kāz karisni]	কাজ করেননি [kāz karēnni]	কাজ করেনি [kāz karēni]
Part. Past.	কাজ না করে [kāz nā karē]	n/a	n/a	n/a	n/a	n/a	n/a
Part. Cond.	কাজ না করলে [kāz nā karlē]	n/a	n/a	n/a	n/a	n/a	n/a
Imp. Pres.	n/a	n/a	কাজ করবেন না [kāz karbēn nā]	কাজ করবে না [kāz karbē nā]	কাজ করবি না [kāz karbi nā]	কাজ করবেন না [kāz karbēn nā]	কাজ করবে না [kāz karbē nā]
Imp. Fut.	n/a	n/a	কাজ করবেন না [kāz karbēn nā]	কাজ করো না [kāz karō nā]	কাজ করিস না [kāz karis nā] /কাজ করিসনে [kāz karisnē]	কাজ করবেন না [kāz kar nā]	কাজ করবে না [kāz karbē nā]

Table 6: Example of negative inflections of a compound verb

9th International Conference on Computer and Information Technology (ICCIT 2006), Dhaka, Bangladesh.

Islam, M. Z., Uddin, M. N., and Khan, M. (2007). A light weight stemmer for Bengali and its use in a spelling checker. *Proceedings of the 1st International Conference on Digital Communications and Computer Applications (DCCA2007, Irdbid, Jordan).*

Klaiman, M. H. (2009). *The World's Major Languages*, volume 23. Routledge, 2nd edition.

Ortiz-Rojas, S., Forcada, M. L., and Ramírez-Sánchez, G. (2005). Construcción y minimización eficiente de transductores de letras a partir de diccionarios con paradigmas. *Procesamiento del lenguaje natural*, (35):51–57.

Ray, P. S., Chakravarti, P. N., Chatterjee, S., Jahan, R., and Ahmed, M. (1966). *A Reference Grammer of Bengali.* The University of Chicago.

Samit Bhattacharya, Monojit Choudhury, S. S. and Basu, A. (2005). Inflectional morphology synthesis for bengali noun, pronoun and verb systems. *In Proc. of the National Conference on Computer Processing of Bangla (NCCPB 05), Dhaka, Bangladesh.*