Parts of Speech Tagging of Bangla Sentence

Md. Shahnur Azad Chowdhury, Nahid Mohammad Minhaz Uddin, Mohammad Imran, Mohammad Mahadi Hassan and Md. Emdadul Haque

Department of Computer Science & Engineering
International Islamic University Chittagong
E-mail: tipu_iiuc@yahoo.com, minhaz_iiuc@yahoo.com, m_imran610@yahoo.com mahadi_cse@yahoo.com, emdad_74@yahoo.com

Abstract

The recent studies on natural language have shifted their attention from rationalistic approach to empirical approach, where people are more concerned in applying corpus-generated data-base in language study than applying their intuitive assumptions. This change of attitude adds a new dimension to language study hitherto unknown to the linguistic world. Keeping this scenario at background, we examine how corpus can be used for studying Bangla language, and how new information and data obtained from corpus that can be used in Natural Language Processing (NLP) in Bangla. Our main goal is to tag the parts of speech of a Bangla word. The work presented here on a morphological processor for Bangla, working from the context free to the context bound level, is justifiable, especially as this will provide a basis for future work on the automatic processing of Bangla. Reliable morphological processing is a good building block for the development of further language processing systems. This paper presents the first steps taken towards an automated morphological analysis of Bangla language. We have considered the limitation and the advantages of suffixes finding the status of the word. We have developed some algorithms and shown different efficient usage of this technique.

Keywords

Lexicon, tag, suffix, tag vector, Parts of speech, Morphology and Morphological rules.

INTRODUCTION

Words in Bangla and other human languages are subjected to complex morphological processes. These processes result in the component parts of a word not being susceptible to analysis by a simple, general, algorithm. Such morphological processing is, however, an important part of Natural Language Processing (NLP), as it can yield important information to a linguistic analysis. In order to achieve such an analysis, two obvious strategies present themselves - working at either the word level, or above the word level. At the word level, a context free analysis is possible which can automatically extract information such as part-of-speech, tense, aspect, case, person, number, gender, particle, honorification etc. However, a context free approach may lead to ambiguous analyses. Above the word level, a context bound approach is possible, using, for example, co-occurrence information or grammatical rules working at the sentence level to achieve disambiguation. Needless to say, word level and context bound analysis need not be independent, and information gathered from word level processing can be used for context bound analysis, with context being applied to achieve disambiguation. The results of such a disambiguated analysis can then, in turn be used to generate further levels of analysis such as word sense disambiguation or information extraction.

MORPHOLOGY OF BANGLA WORDS

Bangla is not morphologically very complex. There are three different types of morphologies are recognized in Bangla.

Inflectional morphology[6] produces or derives words from another word form acquiring certain grammatical features but maintaining the same part of speech or category. Ex: করা,করিতেছে, করছে, করছি.

Derivation[6] is the combination of a word with an affix. Ex: পরাজিত, অ/ পরাজিত

Compound words are difficult for morphological analysis but the rules are needed to generate fluent Bangla from interlingua. Some forms of compounding[9] are discussed below:

- Noun + noun: example, শিক্ষক সমিতি বাড়ি-ঘর,নদী-তীর. This form of compounding can be analyzed as N+ N + morphological features. Again ‘কলেজের গেট, বইয়ের দোকান’, the morphological rule for this may be N + Biv + N+ morphological features. In other form the first noun indicates the place and time of the second one. For example ‘রাস্তার দোকান,বগুড়ার দই,গরমের ছুটি’
- Noun + verb : example ‘ফুলতলা,গাড়ী চালানো পাখিশিকার’. The morphological rule in this case may be N + V + morphological features.

METHODS OF AUTOMATIC WORD FORM RECOGNITION

We want to discuss three methods of automatic word form recognition [6]. They are as follows:

1. Word form method

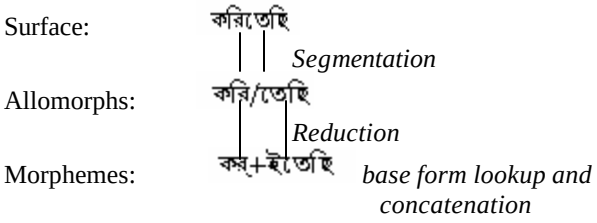
2. Morpheme method
3. Allomorph method

Word Form Method

Surface word is the lexically collected word. When lexicon read a word from surface, the word is then simply tried to find in the lexical entry without considering the suffixes, prefixes or any rules. This method assumes that all possible surface words would be present in lexical entry.

Morpheme Method

The morpheme method is based on a lexicon of analyzed morphemes. The analyzing method are given below:

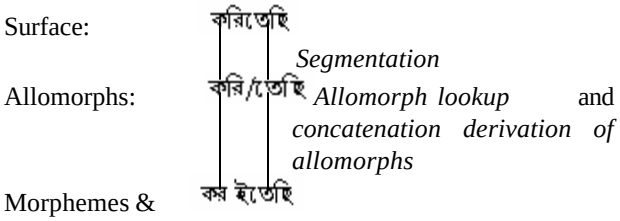


The main advantages are uses the smallest analysis lexicon, neologisms may be analyzed and recognized during run-time using a rule-based segmentation and concatenation of complex word forms into their elements. But, a great disadvantage is to have a maximally complex recognition algorithm.

Allomorph Method

Allomorph method is based on a lexicon of elementary concatenated base forms, from which a lexicon of analyzed allomorphs is derived. An algorithm of finding allomorph takes a lexical entry as input and maps it into one or more allomorphs.

The analyzing method is describing below:



For analyzing simplicity, we have used here the allomorph method.

THE LEXICON

Lexicon means dictionary, where various entries of words of any language are included.

To implement the system, two types of lexicon has been formed, namely –

a. A root lexicon.
b. A suffix lexicon.

During the lookup of a word in the lexicon, the root lexicon is searched first. If the word is found there, the corresponding tag probability vector is returned. If it fails, the suffix lexicon is searched next and for the rest part the word, root lexicon is searched. This is done until reach to the beginning of the word from ending point. If none of the previous step had been successful, the default entry of the lexicon is returned. The root and suffix lexicons were created from a tagged training corpus. [7]

The second part of the lexicon the suffix lexicon, is formed by applying hash function.. The key of the hash function was generated from the summation of the ASCII values of each character of every suffixes. Using this key, probable position was generated to store the suffix. Then that probable position of the array was directly accessed. If there was previously stored any suffix in that position ,then collision occurred. In case of collision the next free space was searched and the suffix was stored in that position. We found the collision is very minimum here.[10]

STRUCTURE OF TAG VECTOR

For tagging any word with its various aspects we have used a sixteen-bit tag vector[7]. Where parts of speech (POS). Person, Mode, Tense number and emotion are put in different length.

Three bits are kept for parts of speech. In POS there are noun, pronoun, adjective, verb and preposition. Noun is divided into proper noun and dictionary word whereas adjective is divided into proper adjective and modal adjective. Verb is divided into finite verb and infinite verb whereas infinite verb is divided into gerund and participle (Illustrated in Figure 2(a)). For person identification, it is divided into first, second and third person (Illustrated in Figure 2(b)). The mode in Bangla is mainly of two-type প্রাণীবাচক (*Pranibachok*) and অপ্রাণীবাচক *(Opranibachok*). The প্রাণীবাচক *(Pranibachok)* may be general, honor, disgrace and deictic. The অপ্রাণীবাচক *(Opranibachok)* are disgrace and general (Illustrated in Figure 2(c)). Tense are mainly three types: present, past and future where present tense may be divided into present indefinite, present continuous and present perfect. Past tense can be divided into past indefinite, past continuous and past perfect. The future tense is only one type, which is future indefinite (Illustrated in Figure2 (d)). In Bangla number is two types: Singular and Plural. We reserved three bits in tag vector to represent emotional state of sentence. So the tag vector is defined by sixteen bits data.

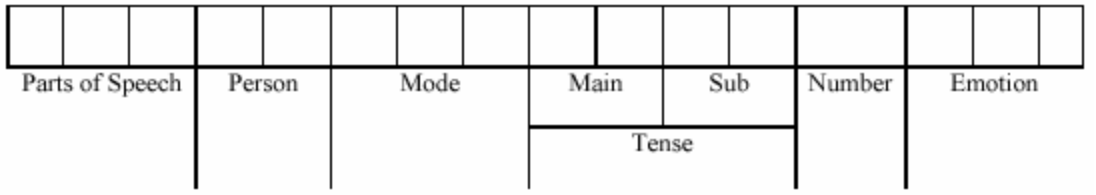


Figure1: 16-bit data representation

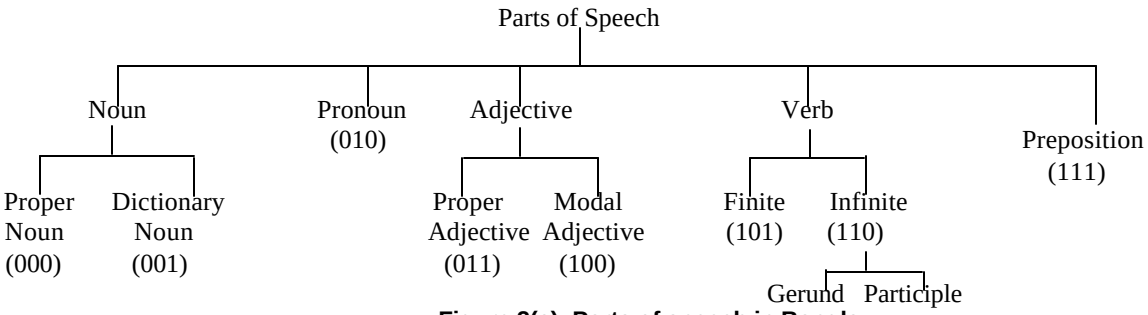


Figure 2(a): Parts of speech in Bangla

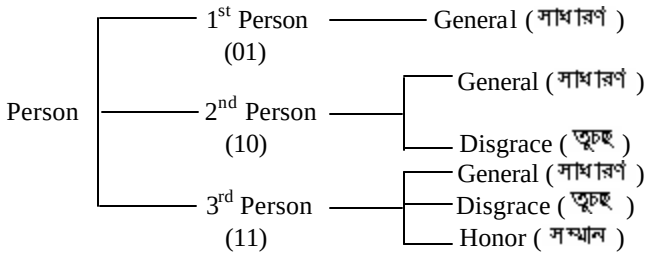


Figure 2(b): Person and mode

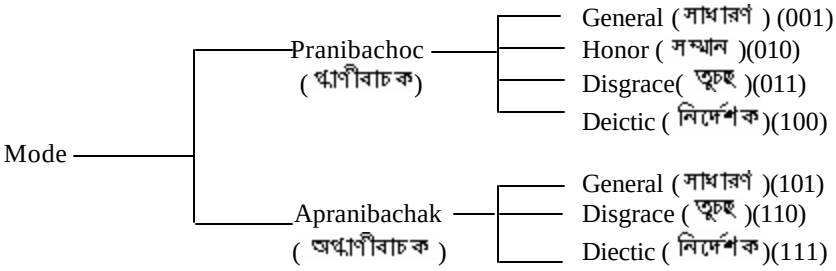


Figure 2(c): Mode in Bangla

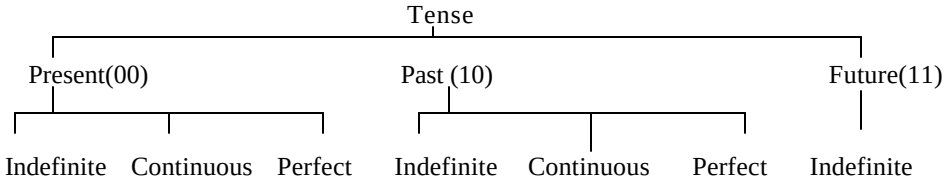


Figure 2(d): Tenses in Bangla

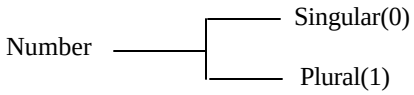


Figure 2(e): Number in Bangla

Table 1 Assigning Values to Some Roots and Suffixes

	Parts of speech			Person		Mode			Tense				Num	Emotion			
									Main		Sub						
আমি	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	18688
কর	1	0	1	1	0	0	0	1	0	1	0	1	0	0	0	0	45392
গণ	0	0	1	1	1	0	1	0	0	0	0	0	1	0	0	0	14856
শ্রেণী	0	0	1	1	1	1	0	1	0	0	0	0	1	0	0	0	15624
ইতেছেন	1	0	1	1	0	0	1	0	0	1	1	0	0	0	0	0	45664

MORPHOLOGICAL RULES

In this section, we have presented some Bangla morphological[8] rules for regular inflections, derivations and compounding with additional explicit rules for irregular inflection, derivation and compounding

Rules for Parts of Speech

Bangla parts of speech conversion rules are mainly for noun ? adjective. Some conversion rules are also done for indeclinable ? noun and indeclinable ? adjective and there are some exceptions also.

Rule 1: The conversion that does not follow any particular formula.

Rule 2: For noun ? adjective, add 'ি'with the last character and after that, add 'ত'(আনন্দ-আনন্দিত,আমোদ-আমোদিত)

Rule 3: For noun ? adjective, add 'ী' with the last character(আগমন-আগমনী)

Rule 4: For noun ? adjective, add 'টে' after the last character(ঝসড়া-ঝসড়াটে)

Rule 5: For noun ? adjective, add 'ী' with the last character and after that, add 'য়'(নাটক-নাটকীয়)

Rule 6: For noun ? adjective, if the last character is ন/ণ/য় then drop the last character and add 'ি' with the previous character and after that, add 'ত'(সাধন-সাধিত,আহরণ-আহরিত,পরিচয়-পরিচিত)

Rule 7: For noun ? adjective, add 'উ'after the last character (ঢাল+উ=ঢালু)

Rule 8: For noun ? adjective, add 'া' with the last of the word (ভোগ+া=ভোগ্য)

Rule 9: For noun ? adjective, add 'উক' at last (পেট+উক=পেটুক)

Rule 10: For noun ? adjective, add 'আ'at last (বাঘ+আ=বাঘা)

Rule 11: For noun ? adjective, add ইয়া/উরে' at last (শহর+ইয়া/উরে=শহারয়া/শহুরে)

Rule 12: For noun ? adjective, add 'চে' at last (লাল+চে=লালচে)

Rule 13: For noun ? adjective, add 'তো' at last (কুপা+তো=কুপাতো)

Rule 14: For adjective ? noun, add 'আই' at last (মিঠা+আই=মিঠাই)

Rule 15: For adjective ? noun, add 'ইমা ' at last (নীল+ইমা=নীলিমা)

Rule 16: For adjective ? noun, add 'পনা' at last (দুরন্ত+পনা=দুরন্তপনা).

Rule 17: For adjective ? noun, add 'আমি' at last (দুষ্ট+আমি=দুষ্টামি)

Rule I8: For adjective ? noun, add 'গিরি' at last (বাবু+গিরি=বাবুগিরি)

Rule 19: For adjective ? noun, add 'তা' at last (এক+তা=একতা)

Rule 20: For indeclinable ? noun, drop 'া' and add 'া' (তথা-তথ্য)

Rule 21: For indeclinable ? adjective, drop 'ঃ' and add 'ি' (বহিঃ-বহি)

Rules for Number

Here are few rules for the translation of Bangla Number:

Rule 1: Adding টি, টা, খানা, খানি with the main word represents Singular word (গরু-গরুটি, খাতা-খাতাখানা)

Rule 2: Adding রা,এরা,গণ,বৃন্দ,মন্ডলী,বর্গ with the main word represents the high class living things in plural number (ছাত্র-ছাত্ররা,মা-মায়েরা,জন-জনগণ)

Rule 3: Adding কুল,সকল,সব,সমূহ with the main word represents the low class living things and non-living things in plural number (কবি-কবিকুল,পর্বত-পর্বতসকল)

Rule 4: Adding গুলা,গুলো,গুলি,গুচ্ছ,দাম,রাশি with the main word represents the non-living things and dumb things in plural number (আম-আমগুলি. বালি-বালিরাশি).

THE ALGORITHM

We have proposed an algorithm to split the words. The steps of the algorithm are as follows:

Traverse characters from last toward first
Step 1: Take the last character
Step 2: Concatenate the character with suffix
Step 3: If character is post modifier then concatenate another character in suffix,
    Take another character
Else if character is not pre modifier then
        Take another character
        If the character is pre modifier then
    Concatenate the character with suffix
            Take another character
        End if
  End if
Step 4: Find suffix in suffix lexicon
Step 5: If suffix found then
        Root = the rest character starts from first
        Find the root in root lexicon
        If found then
          Go to step 6
        Else go to step 2
        End if
    End if
Step 6: Exit.

Example

An example word is shown here splitted by the algorithm.

Let us consider the word 'খাইতেছে'.

Let the algorithm start traversing from the last character towards the first character of the word and take the last character which is 'ছে',concatenate the character with the array named suffix and initialized empty. Test to see whether 'ছে' is a post modifier. As it is not a post modifier, now test whether it is not a pre modifier, as it is it not a pre modifier then another character is taken from the word 'খাইতেছে',which is 'ে'.As 'ে' is a modifier so it is concatenated with the array suffix, whose previous content was 'ছ'.So the content of the array suffix is now 'ছে' . Now take another character, which is 'ত'.Before inserting this character into the array suffix, test the array suffix whose content is 'ছে' whether it is in the suffix lexicon. If it is available in the lexicon, assign the rest part 'খাইতে' to the array named root and initialized empty. Test whether the root lexicon contain any entry as 'খাইতে', if found return the tag, otherwise go to the first step of the algorithm and follow the later steps for the rest part of the word *i.e.* for 'খাইতে'.

Complexity of the Algorithm

One can show that the complexity of a binary search algorithm is given by :

$$C(n)=log_2 n$$

We have stored the roots in the root lexicon by sorting them and searched them by using binary search technique.. If N_r is the number of root entry in the lexicon then the complexity of finding root is

$$C(N_r)=log_2 N_r.$$

Let n, which is very small in size, is the number of character in the pattern to be tagged. In case of splitting the pattern for finding suffix, we traversed the pattern from last to first linearly. If we consider here even in the worst case the complexity will be

$$C(n)=n.$$

We used Linear Hash function to store the suffixes and retrieved them accordingly. If we denote the complexity of finding suffix as N_S.

Hence the total complexity of our proposed algorithm is

$$C(n)= N_S * n\ (log_2 N_r)$$

N_S is almost 1 (one) in practical cases although memory wastage is not minimized. So,

$N_S * n\ (log_2 N_r) \approx n\ (log_2 N_r)$. Which is considerably small.

So the algorithm is efficient.

CONCLUSION

We have extracted correct neologism successfully by allomorphic method with allo-rules. As the word description are extracted successfully, it can be used in many other application beyond POS tagging such as spell checker, machine translation, emotion analysis and almost every area of natural language processing of Bangla.

REFERENCES

[1] Dr. Shahjahan Monir, "বাঙ্গলা ব্যাকরণ" (Bangla Bayakaron), eight edition, Dhaka, 1994.

[2] Monir Chowdhury, Mofazzal Haider Chowdhury, "বাংলা ভাষার ব্যাকরণ", Bangla Bhasar Bayakaron, Dhaka, 1983.

[3] Globe library private limited, "উচ্চতর বাংলা ব্যাকরণ ও রচনা", Uccatar Bangla Bayakaron O Racana, Dhaka 1977.

[4] Kamal Hasan Chy., Shudarshan Chy., Mohammad Hurunur Rashid Chy., Dulal Kishor Bhattacharjya, Md. Ahidul Islam, "ভাষা বিচিন্তা ", (Bhasa Bicinta) – Vhasha Prokashani, 1999.

[5] Humayun Azad "বাক্যতত্ত্ব", Bakyatatta. The University of Dhaka, 1994.

[6] Md. Hanif Siddiqui, A.K. Mohammad Shohel Rana, Abdullah Al Mahmud and Taufique Sayed, "Morphological Analysis of Bangla Words". 6th International Conference on Computer and Information Technology (ICCIT) 2003. Jahangirnagar University, Dhaka, Bangladesh, pp.313-316,2003.

[7] Md. Hanif Siddiqui, A.K. Mohammad Shohel Rana, Abdullah Al Mahmud and Taufique Sayed, "Parts of speech Tagging using Morphological Analysis in Bangla". 6th International Conference on Computer and Information Technology (ICCIT) 2003. Jahangirnagar University, Dhaka, Bangladesh, pp.374-379,2003.

[8] Tamanna Hoque Nipa, Mohammad Harun-or- Rashid, Salauddin Mohammad Masud, "Bangla and English Morphological rules for Machine Translation Dictionary", 6th International Conference on Computer and Information Technology (ICCIT) 2003. Jahangirnagar University, Dhaka, Bangladesh, pp.391-394, 2003

[9] M M Asaduzzaman and Muhammad Masroor Ali, "Morphological Analysis of Bangla Words for Automatic Machine Translation", 6th International Conference on Computer and Information Technology (ICCIT) 2003. Jahangirnagar University, Dhaka, Bangladesh, pp.265-270,2003

[10] Dr. Seymour Lipschutz "Data Structures", International Edition 1986.