A NEW APPROACH IN COMPUTER REPRESENTATION OF BANGLA WORDS AND BANGLA SORTING ALGORITHM

Md. Sharif Uddin, Rahat Khan, A.B.M Tariqul Islam, S.M. Rafizul Haque
Computer Science & Engineering Discipline, Khulna University, Khulna-9208, Bangladesh.
auni_ku@yahoo.com, rahatkhanr@yahoo.com, tariq_cse_ku@yahoo.com, rafizulku@yahoo.com

Abstract:
Development of Bangla based computer application is relatively complex due to the complexities of Bangla character set (for example computer representation of composite letters). This paper focuses on a new technique on internal representation of Bangla words in computer system along with a Bangla word sorting algorithm using that representation. Here, we propose a special technique which converts a Bangla word into a unique real number. Now, if the numbers corresponding to a given set of Bangla words are sorted using any of the familiar sorting algorithms then we get the sorted order of the words in that set which is simply the sorted order of the numbers that represents words. Our algorithm compares real numbers rather than characters to sort the words and thus decreases the difficulties of character comparing which exists in many of the current Bangla sorting algorithm.

1. INTRODUCTION

Bangla is a very rich language and approximately 10% of world's populations speak in Bangla [7]. Hence, the computerization of this language is the inevitable need today, but unfortunately we have advanced a very little in this regard. For the development of Bangla database systems an expedient, efficient, versatile sorting algorithm is a must. The word format used in various word processors is not suitable for sorting, matching etc. Because the way the character strings are stored in physical devices is not convenient for any mathematical computation such as sorting. In our previous paper [4] we have presented a word representation technique based on integer number which needs some pre-processing before sorting (a number of 0 has to be inserted at the end of some numbers that represents words, to make all of them of equal in size, see [4] for more details). In this paper we are proposing a method to represent Bangla words internally in the computer systems as a real number, which will provide the scope of efficient sorting of Bangla words and requires no preprocessing as in [4]. Our proposed method converts a Bangla word into a unique real number based on the characters it contains.

1.1. The Bangla language

In the written form of Bangla there are 11 vowels and 39 consonants. Moreover, there are 10 short forms of vowels called vowel modifiers (i.e. Kar), 7 short forms of consonants called consonant modifiers (i.e. Fala) [7]. Beside these, there are more than about 253 compound characters composed of 2,3 or 4 consonants (200 compound characters composed of 2 consonants, 51 compound characters composed of 3 consonants and 2 compound characters composed of 4 consonants) [6]. In accordance with the order of Bangla Academy standard [1], vowels and corresponding vowel modifiers and their placement within words are listed in Table 1.1.

Table 1.1: Vowels and vowel modifiers.

Vowels	Vowel Modifiers	Placement	Example
অ	None	None	none
আ	v	Right	সাবাশ
ই	ি	Left	নিহিত
ঈ	ী	Right	নীড়
উ		Below	বুনন
ঊ		Below	সর্য
ঋ	‹	Below	কৃষি
এ	ে	Left	পেঁপে
ঐ	ৈ	Left	শৈবাল
ও	ো	ে at left, া at right	কোমল
ঔ	ৌ	ে at left, ৗ at right	কৌশিক

According to the standard of Bangla Academy consonants are ordered as follows:

st KLMNOPQRSTUVWoXpYZ_`abcdefghqijklmn

Consonant modifiers (i.e. Fala) with their corresponding consonants are listed in Table 1.2 [2]. Besides the vowel, consonant and their modified form we have a special character Hoshonto (nm ÖÖ).

Table 1.2: Consonant modifiers.

Consonants	Consonant Modifiers
b	È
e	
g	§
h	¨
i	,
j	¬

Unlike English words, Bangla words are not only composed of individual characters placed one after another. In Bangla 2 or 3 or 4 consonants can be merged together to form a single compound character. Some examples are in Table 1.3.

Table 1.3: Compound characters.

Number Of Characters	Compound Character	Decomposed Form
2	›`	b+`
3	¾	R+R+e
4	š¨	b+Z+i+h

1.2. Sorting of Bangla text

English words are composed of individual alphabets and so the sorting of English words is quite simple. To sort two English words we start the comparison from the first letters of both the words and proceed towards the end of the words comparing characters pair by pair. On the basis of the first dissimilar pair of characters, a sorting decision is made. For example, the sorting of two English word "FARNANDEZ" and "FARNANDOS" is shown in Table 1.4.

Table 1.4: Sorting of English words.

Characters For First Word	Characters For Second Word	Action
F	F	PASS
A	A	PASS
R	R	PASS
N	N	PASS
A	A	PASS
N	N	PASS
D	D	PASS
E	O	END
Z	S	No need to compare

As we see from Table 1.4, when the pair of characters are same the action is to just "PASS" to the next pair of characters. The first dissimilar pair of characters in our example is 'E' and 'O'. So decision is to be made from the comparison of these two characters. In our example, "FARNANDEZ" is to be placed before "FARNANDOS".

In case of Bangla, the scenario is quite different. Bangla words cannot be sorted using such a simple algorithm. In Bangla words vowel and consonant modifiers are placed before, after, above or below any character. Moreover there are frequent uses of compound characters. Moreover, some modifiers such as ‡ v and ‡ Š are fragmented into ‡ + v and ‡ + Š respectively. Keystrokes are stored in the file following the same sequence. For example, in case of typing ‡Mvajx we first type ‡, then M, then v and so on. And in the same order the characters and modifiers are stored in the file. Here two modifiers ‡ and v are associated with M but actually there is a single modifier ‡ v with M. This results in inconsistency in sorting. Suppose two Bangla words Mgb and ‡Mvajx are to be sorted. This could be done as follows. Here M is first compared with ‡. Since ‡ precedes M, ‡Mvajx comes before Mgb in the sorted list. Obviously this sorting is not correct. Because in the word ‡Mvajx, M has the vowel modifier ‡ v but in case of Mgb, M has no modifier. Hence Mgb should precede ‡Mvajx in the sorted list if we are to follow the standard of Bangla dictionary.

2. PREVIOUS WORKS

2.1. Method 1: as described in [7]

In order to maintain proper sorting Rahman and Iqbal [7] have proposed an internal representation of Bangla words where a dummy character is placed after the character, which has no modifier. Moreover, it is also ensured that there would be no dummy character between the constituent parts of a compound character. Again, vowel modifiers are included in the character set and they can be typed before or after the characters but for internal representation every time they are to be shifted after the character. In case of compound characters, they are decomposed into their constituent components and stored accordingly. In Table 2.1 internal representation of few words are shown where @ represents the dummy character: For sorting the words the relative order in the character set are arranged in the following way-

Null modifier < Vowel Modifiers < Vowels < Consonants

Table 2.1: Internal representation of words in [7].

Word	Internal Representation
A¶vsk	A @ K l v s @ k @
¯vMZg	m e v M @ Z @ g @
Kgjv	K @ g @ j v
eM	E @ i M @
‡gvoK	g ‡ v o @ K @
KvK	K v K @

This method has the following shortcomings:

- Previously extra vowel modifiers had to be accommodated in the keyboard, which is not needed according to our opinion.
- Shifting of the vowel modifiers adds extra overhead. The keyboard interface has to be complex enough to do this job.
- In the keyboard mapping proposed by them, N is mapped to '[', O is mapped to '\', P is mapped to ']' and n is mapped to '{'. But these '[', '\', ']' and '{' symbols are used in Bangla. So they cannot be removed.

Due to use of the dummy character, a large amount of disk space is consumed to store Bangla words.

2.2. Method 2: as described in [9]

According to the proposal of Palit and Sattar [9], the keyboard will accommodate vowels, consonants and necessary symbols. In this proposal, a special key is used for link character. The words will be typed as they are spelled. The characters in the words are mapped to appropriate ASCII values. No link character is used. The vowel modifiers are assigned 10 distinct ASCII values higher than those of the consonants. The compound characters are divided into their constituent components and saved to file. The shape of those components will vary based on their relative position in the compound character. All the shapes are stored in the Video ROM and distinct codes are assigned to them. Internal representations of some words are shown in Table 2.2.

Table 2.2: Internal representation of words in [9].

Words	Internal Representations
‡mvbvjx	m ‡ v b v j x
mKvj	m K v j
mwP	m P w
mwPZv	m P w Z v
Aši	A b _ Z i
A›`i	A b _ ` i

For sorting, we will follow the same order as used in Bangla dictionaries:

Vowels < Consonants < Vowel Modifiers

This method has the following drawbacks:

- Due to use of the key used for link character, extra space is required to store Bangla words.

Since different codes are assigned to different shapes of the constituent parts of the compound character, a wide range of shapes and their corresponding codes are to be maintained.

3. PROPOSED METHOD

In our proposed method we have assigned some unique numbers to all Bangla letters. Any word then can be converted to a real number using our developed formula. Here we are going to represent the whole procedure.

3.1. Representation of Bangla letter as a number

At first we assign a two digit unique number for every letter of Bangla alphabet along with the vowel modifiers and the consonant modifiers. The letters and their corresponding numbers are given in Table 3.1. It is to be noticed that here ' Av ' is treated as a set of two characters that is ' A + v '. The consonant modifiers are having the same number as their original consonants.

Table 3.1: Numbers assigned to the letters of Bangla alphabet.

Letters	Assigned Number
A	11
B - J, s, t,	12 - 20, 21, 22, 23
K - O	25 - 29
P - T	30 - 34
U, V, W, o, X, p, Y	35, 36, 37, 38, 39, 40, 41
r	42
Z - b	43 - 47
c - g	48 - 52
h, q, i, , j, k, l, m, n	53, 54, 55, 56, 57, 58, 59, 60, 61
v, w , x, , , , †, ˆ, †v, ‡$	71, 72, 73, 74, 75, 76, 77, 78, 79, 80
compound character	99

3.2 Converting a Bangla Word into a Real Number

After assigning a two digit number to each of the letters and vowel and consonant modifiers, let us move forward to the process of converting a word into its corresponding real number. We call this real number *Word_value* which can be computed using the given formula.

$$Word\_value = \sum_{i=0}^{n-1} a_i \times d^{-i} \qquad (3.1)$$

where, *n* is the total number of letters in the word, *i* is the position of a letter in the word from left to right (starting from 0), a_i is the assigned number of a letter (from Table 3.1) positioned at location *i* in the word and *d* is a constant such that *d* = (maximum value of assigned number (from Table 3.1) +1).

The strategy is very simple. Each word is scanned from left to right and for each letter of the word; the corresponding number is taken from Table 3.1 and compute the *Word_value* using the Equation 3.1. For an example, let *d* = 100 (in the rest of paper we will use *d* = 100), then the word ‘Mgbô changes into the real number ’27.5247’. From Table 3.1 we find, for ‘M’ the number is ‘27’, for ‘g’ it is ‘52’ and for ‘b’ we found ‘47’. So, the final number we get from Equation 3.1 is 27 + 0.52 +0.0047 =27.5247. But, while scanning a word from left to right to get a letter, difficulties arise when we find vowel modifiers, compound characters (such as, ß, °, ¾ etc.) and consonant modifiers. Let’s see how we can handle those difficulties.

3.2.1. Handling vowel modifiers

When we are handling with vowel modifiers, one thing we have to always keep in mind is that, for vowel modifiers, wherever its position is (i.e. Left, Right or Down), it will always be scanned after the letter over which it was applied. For Example, the words ‘mvÔ, ÔmÔ, ÔwmÔ, Ô†mvÔ and Ô†mÔ change into numbers ’60.71’, ’60.74’, ’60.72’, ’60.79’ and ’60.77’. Here it is easy to notice that the position of the vowel modifiers is after the letter over which it was applied. Some examples with vowel modifiers are shown in Table 3.2.

Table 3.2: Some words with vowel modifiers.

Word	Converted Number
‡mvbvjx	60.7947715773
‡Mvajx	27.7946755773
‡M$ie	27.805550
wZwgi	43.72527255

3.2.2. Handling compound characters

In case of compound characters we have applied the similar method developed in our previous paper [4], which we call number stuffing. For Compound characters with 3 or 4 letters, a unique number is stuffed into every pair of consequent characters i.e., an extra character is considered between every pair of consequent character. Here, we stuff a special number ‘99’ between the corresponding numbers of the two letters by which the compound character is composed by without breaking the relative positions of individual characters in the compound character. For example, the compound character ‘°’ changes into ‘259925’ by this method as like in Figure 3.1. Table 3.3 shows some example of number conversion of words with compound characters.

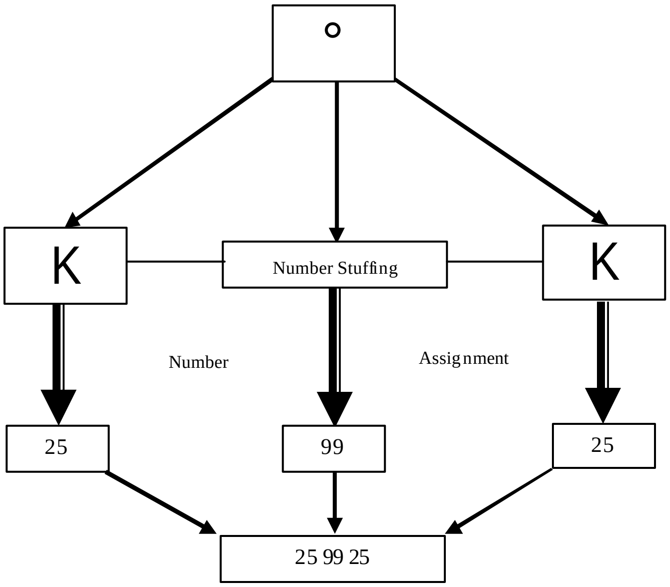

Figure 3.1: Number assignment on compound character.

Table 3.3: Some words with compound characters.

Word	Converted Number
mß	60.74489943
e›`K	50.4799457425
D¾j	14.329932995057
iw³g	55.2599437252

3.2.3 Handling consonant modifiers

For consonant modifiers strategy taken is all the same as we have taken to handle compound characters. Here, the number 99 is stuffed before the corresponding number of the consonant to which the modifier belongs to. For example, for the word ‘mn¨’ the converted number is ‘60619953’. Here. ‘99’ is stuffed before ‘53’ which is the assigned number for ‘h’. We can see from Table 1.2 that ‘¨’ is the modifier of the consonant ‘h’.

3.3. Sorting algorithm of proposed method

Here we formalize our sorting algorithm.

Input: A list *A* of *n* Bangla words.

Output: Sorted sequence of words in A.

```
1.  for i := 1 to n do
        B(i) := Word_value of A(i);
2.  Apply any conventional sorting algorithm on B.
3.  Output the words in A according to the sorted sequence of Word_values in B.
```

3.4. Sorting example using the algorithm

In this section we have given an example (using *d*=100) for the better understanding of our proposed algorithm. The sorting of five Bangla words ‡Mvajx, Mgb, mß, ü`q and ms¯wZ is given below. The *Word_value* of corresponding words are given in the Table 3.4.

Table 3.4: The numbers representing the Bangla words ‡Mvajx, Mgb, mß, ü`q and ms¯wZ.

Word	*Word_value*
‡Mvajx	27.7946755773
Mgb	27.5247
mß	60.74489943
ü`q	61.764554
ms¯wZ	60.21609925764372

Now, by applying any conventional sorting algorithm on the real numbers we get the sorted list of Bangla words. The sorted list of words is presented in the Table 3.5.

Table 3.5: The sorted list of the Bangla words ‡Mvajx, Mgb, mß, ü`q and ms¯«wZ.

Word	*Word_value*
Mgb	27.5247
‡Mvajx	27.7946755773
ms¯«wZ	60.21609925764372
mß	60.74489943
ü`q	61.764554

4. CONCLUSION

In this paper we have proposed a new approach in internal representation of Bangla words and an efficient way of sorting the Bangla words which will be convenient for sorting, matching and various other applications of Bangla processing. Although the internal representation is in real numbers that uses a little bit more memory than representation in integers, our proposed method increases the sorting efficiency by reducing all the pre-processing complexities aroused in our previous method [4].

5. REFERENCES

[1] *Bangla Academy Bengali-English Dictionar*y, First Edition June,1994, Bangla Academy, Dhaka, Bangladesh.

[2] Kazi Din Mohammad: *"Adhunik Bangla Byakoron O Rochona"*, First Edition, June,1999.

[3] Khan Ferdous, *"Haraf Shamashha*", Munir Chowdhuri Rachanabali, Vol.3, pp.551-553, 1984.

[4] Mafizul Haque Khan, S.M. Rafizul Haque, Md. Sharif Uddin, Rahat Khan, A. B. M. Tariqul Islam: "An Efficient And Correct Bangla Sorting Algorithm", Accepted in *7th ICCIT*, December 2004.

[5] Md. Ameer Ali, "Development of Bangla Keyboard". *B.Sc.Engg.Thesis*, Department of Computer Science and Engineering, BUET, August 2001.

[6] Md. Salahuddin Masum, "Study of Bangla Conjunctive Characters for Recognition", *B.Sc.Engg.Thesis*, Department of Computer Scince and Engineering, BUET, August 2001.

[7] Md. Shahidur Rahman and Md. Zafar Iqbal, "Bangla Sorting Algorithm: A Linguistic Approach", Proceedings of *ICCIT'98*, December 1998, pp. 204-208.

[8] Mozammel Haq Azad Khan, "Optimal Realization of Bengali Keyboard and Character Encoding for Computer Application", *M.Sc Engg Thesis*, Department of Computer Science and Engineering, BUET, 1986.

[9] Rajesh Palit, Md Abdus Sattar, "Representation of Bangla Characters in the Computer Systems". *Bangladesh Journal of Computer and Information Technology*, Vol. 7, No. 1, December 1999.

[10] Thomas Cormen, Charles Leiserson, and Ronald Rivest: *"Introduction to Algorith*m", Prentice – Hall of India Private Limited, 1999.