# An Encoding Scheme to Support Efficient Searching and Linguistic Sorting for Bengali Texts

Tareque Mohmud Chowdhury
CSE Department
Islamic University of Technology
Dhaka, Bangladesh

M. A. Mottalib
CSE Department
Islamic University of Technology
Dhaka, Bangladesh

## ABSTRACT
Most of the known encoding schemes for Bengali language have a common drawback. That is characters order in the encoding scheme is different than the linguistic order. As a result, sorting of Bengali texts as per encoded value does not sort them in correct linguistic order. Even if Bengali characters are encoded in linguistic order, because of special properties of Bengali conjunct character, Bengali text can not be sorted directly using only traditional sorting algorithms. In this paper we proposed an encoding scheme for Bengali script which supports sorting of texts by sorting them as per encoded value. Thus the new encoding scheme can save significant amount of processing time for sort operations over large volume of Bengali texts.

## General Terms
Natural Language Processing

## Keywords
Bengali Character Encoding, Bengali Text Linguistic Sort, Bengali Text Search.

## 1. INTRODUCTION
A Script is a collection of letters and other written signs used to represent textual information in one or more writing systems. Bengali script is used to write Bengali, Assamese, Syloti Nagri, Manipuri, Garo, and Mundari Languages. Alphabets of Bengali script are adopted in Unicode from Indian Standard Code for Information Interchange (ISCII) - 1988 encoding system. Besides there are some other encoding system exists for Bengali which are used in various commercial software. One common and major disadvantage of all of these encoding schemes is that sorting of texts based on encoded value does not yield them sorted in linguistic order. We shall discuss problem details, existing solutions and our proposed solutions with performance evaluation.

Searching and sorting plays a vital role in every text processing system. Because of the extreme importance of those operations in almost all computer algorithms and database applications, a great deal of effort has been expended in the development and analysis of efficient searching and sorting algorithms. The elementary operation for both searching and sorting is comparison. String comparison needs character by character comparison of texts. If characters encoded values are not in linguistically sorted order then string comparison does not yield proper linguistic ordering or texts.

## 2. PROBLEM DESCRIPTION
Comparison of two characters is the basic operation to perform searching and sorting of texts. Character by character comparison is required to compare two text strings to find their proper order. TABLE 1 shows character by character comparison of two Latin text string "tester" and "testing". Both of the texts share same character upto index 4; at index 5, "e" < "i" yields the result "tester" < "testing". For texts in Latin script this comparison works perfectly because characters in Latin scripts are placed as per their linguistic order.

**Table 1. Compare two texts**

| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|
| t | e | s | t | e | r | |
| = | = | = | = | < | | |
| t | e | s | t | i | n | G |

But characters in Bengali script are not encoded as per their linguistic order. Table 2 shows linguistic order of Bengali character blocks of independent vowels, dependent vowels and consonants suggested by Bangla Academy [11]. In linguistic order numeral block is placed before other character blocks.

**Table 2. Bengali characters in linguistic order**

| **Block** | **lexeme** |
|---|---|
| Numerals | ০, ১, ২, ৩, ৪, ৫, ৬, ৭, ৮, ৯ |
| Independent vowels | অ,আ,ই,ঈ,উ,ঊ,ঋ,ঌ,এ,ঐ,ও,ঔ |
| Consonants | ং,ঃ,ঁ,<br>ক,খ,গ,ঘ,ঙ,চ,ছ,জ,ঝ,ঞ,<br>ট,ঠ,ড,(ড়),ঢ,(ঢ়),ণ,ত,(ৎ),থ,দ,ধ,ন,<br>প,ফ,ব,ভ,ম,য,(য়),র,ল,শ,ষ,স,হ |
| Dependent vowels | া,ি,ী,ু,ূ,ৃ,ে,ৈ,ো,ৌ |

Table 3 represents all characters in Unicode Bengali script in encoded order. It also shows some characters (ৰ,ৱ,৲,৳,৻ etc.) which are not used in Bengali language but are used in other languages which are using Bengali script as mentioned earlier.

Table 3 also clearly shows that order of character block is not maintained as suggested by Bangla Academy. Independent vowel, dependent vowel, and consonant blocks are scattered and overlapped. Numeral block is placed at the end which supposed to be placed at the beginning of code space.

To present the deficiency of Unicode we can see an example to compare two words "কয়লা" and "কলস". As "য়" < "ল" in linguistic order linguistic sort yields "কয়লা" < "কলস". But in Unicode "য়" is encoded after "ল". So sorting based on Unicode value yields "কলস" < "কয়লা". As both search and sort operations depends of string comparisons and string comparison depends on character comparison. Texts written in Unicode

Bengali script is not sortable using any sort function provided by operating systems, database systems or programming languages.

**Table 3. Bengali Characters order in Unicode [1]**

| Block | lexeme |
|---|---|
| Da'ri | ।,॥ |
| End consonants | ং,ঃ,ঁ, |
| Independent vowels | অ,আ,ই,ঈ,উ,ঊ,ঋ,ঌ,এ,ঐ,ও,ঔ |
| Consonants | ক,খ,গ,ঘ,ঙ,চ,ছ,জ,ঝ,ঞ,<br>ট,ঠ,ড,ঢ,ণ,ত,থ,দ,ধ,ন,<br>প,ফ,ব,ভ,ম,য,র,ল,শ,ষ,স,হ |
| Dependent vowels | ◌ৃ, ◌া, ◌ি, ◌ী, ◌ু, ◌ূ, ◌ৃ, ◌ৄ, ◌ে, ◌ৈ, ◌ো, ◌ৌ |
| Hasanta (joiner) | ◌্ |
| Consonants | ৎ, ড়, ঢ়, য় |
| Independent and dependent vowels | ৠ, ◌ৢ, ◌ৣ |
| Consonants | ৰ, ৱ, |
| Special symbols | ৲, ৳, ৴, ৵, ৶, ৷, ৸, ৹, ৺ |
| Numerals | ০, ১, ২, ৩, ৪, ৫, ৬, ৭, ৮, ৯ |
| Other | ZWJ (Zero Width Joiner), ZWNJ (Zero width Non Joiner) |

Besides Bengali scripts uses conjunctive characters which makes the sorting scenario much more complicated. For example ক্ষ is a conjunctive character. Its memory presentation is ক ◌্ ষ where ◌্ is hasanta. ক্ষ্ণ is another character whose memory presentation is ক ◌্ ষ ◌্ ণ. ক্ষ্ণ is not a single character and not stored in text as a single encoded value. Rather in text exactly ক ◌্ ষ ◌্ ণ is stored (without space) and the software which will display the text is responsible to render ক ◌্ ষ ◌্ ণ to display ক্ষ্ণ. In the same way all dependent vowels are placed after consonants or conjunctive consonants and display software will render them to display in proper order. For example কে and ক্ষৌ are presented in texts as ক+◌ে and ক্ষ+◌ৌ without + sign. We use + sign here in between consonants and dependent vowels to force not to render.

Dependent vowels of Bengali scripts can appear left, right, top or bottom of a consonant or conjunctive character. But in Unicode encoded text always dependent vowels are placed after consonants or conjunctives. The responsibility to render display is left to the displaying software.

In Unicode two special character ZWJ and ZWNJ for two special type of situations. Using ZWNJ we can manage following situations-

র + ◌্ + য = র্য

র + ZWJ + ◌্ + য = i¨

In above example ZWNJ is used to separate র from ◌্ + য. Then ◌্ + য will form ¨ and joined with earlier র to form i¨. ZWNJ is used to force hasanta not to work as a joiner but a simple hasanta. See the example-

ব + ◌া + হ + ◌্ + ব + ◌া = বাহ্বা

ব + ◌া + হ + ◌্ + ZWNJ + ব + ◌া = বাহ্‌বা

In second line ZWNJ is used to force hasanta to work as a hasanta other than it's default function as a joiner.

# 3. PREVIOUS WORKS

Comparison problem for Bengali texts under Unicode Bengali script can be solved in two different ways. Firstly, by using intermediate codes. Secondly, designing of a new encoding scheme that supports direct linguistic sort.

A. Using intermediate code: In this method it is needed to write a specialized sort function that will generate intermediate codes for each of the text strings. Then sort the intermediate codes using any existing sort functions. Remap the sorted intermediate codes back to the corresponding texts. We found a number of research paper based on this method.

1. M. R. Amin et al. [2] proposed an efficient Unicode based sorting algorithm for Bengali words.

2. MA Rahman and MA Satter [3] proposed a method which can sort Bengali words in a faster way. Each word is converted into a weighted bit string as intermediate code. Then sort the intermediate codes. One major drawback of the method is that they consider sorting only words but arbitrary long texts.

3. ME Imrul and MM Ali [4] proposed a method in a paper to sort Bengali text using ancillary maps. In their proposed method during conversion some data loss may occur for long words thus unexpected result may appear as a result.

4. M Murshed and M Keykobad [5] shows in their paper that It is not possible to define a Bengali coding scheme which either follows the linguistic order completely or embeds rules so that the complete linguistic order can be derived from the partially ordered scheme. They conclude this based on the scenario that Bengali dependent vowels can be present left, right, top, and bottom of a consonant. Even some dependent vowel can present at two side of a consonant. For example ◌ো and ◌ৌ appeared at both left and right side of a consonant. This problem resolved in Unicode where all dependent vowels are placed right side of a consonant in memory representation of texts. And software render the visual display properly while display on screen.

5. A number of other research works [6][7][8][9] presented ways to sort Bengali texts but all of them are almost similar to any of the first three paper discussed here.

The generalized flow chart of the intermediate code method is shown in Fig.1. This method has three major drawbacks. (i) It takes additional time to generate intermediate codes. (ii) It takes additional memory space to keep intermediate codes. For operation on large volume of texts the additional resource uses will be significant. (iii) Language specific sort function is required for Bengali texts. Various operating systems, programming languages, database systems have to develop and integrate unique sort function for Bengali texts. Otherwise every programmer will have to write their own sort function for Bengali texts.

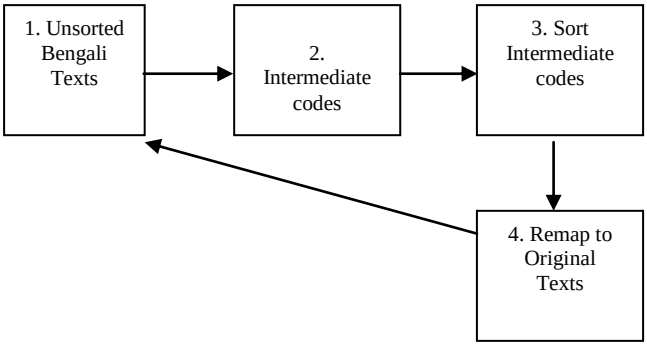


**Fig 1. Sort Bengali text by generating intermediate codes**

B. New encoding scheme that supports direct sorting of Bengali texts: We do not find any existing encoding scheme for Bengali where characters are placed in such a way that sorting of texts based on encoded value yields sort them in linguistic order directly. In this paper we are proposing such a new encoding scheme where sorting can be achieved by sorting the texts using any existing sorting algorithm without using any intermediate code.

## 4. PROPOSED ENCODING SCHEME

We should bear in mind that Bengali script is not only used to write Bengali writing systems but also used to write a number of other writing systems like Assamese, Syloti Nagri, Manipuri, Garo, and Mundari etc. There are few symbols exists that are used in one writing and may not used in others. For example ঋ, ঐ, ঔ, ঞ, ণ, ষ, শ are not used in Syloti Nagari language. Similarly, ৰ and ৱ are used in Assamese language but not in Bengali language. We set the character order in proposed encoding scheme which supports linguistic order for all of the languages using Bengali script. The proposed encoding order is presented in Table 4.

**Table 4. Proposed encoding order for Bengali characters**

| **Block** | **lexeme** |
|---|---|
| Special joiner and Da'ris | $ZWNJ_b$ , $ZWJ_b$ , ।,॥ |
| Special Symbols | ৲, ৳, ৴, ৵, ৶, ৷, ৸, ৹, ৺ |
| Numerals | ০, ১, ২, ৩, ৪, ৫, ৬, ৭, ৮, ৯ |
| Independent vowels | অ,আ,ই,ঈ,উ,ঊ,ঋ,ঌ,ঌ ,এ,ঐ,ও,ঔ |
| Consonants | ং,ঃ,ঁ,<br>ক,খ,গ,ঘ,ঙ,চ,ছ,জ,ঝ,ঞ,<br>ট,ঠ,ড,(ড়),ঢ,(ঢ়),ণ,ত,(ৎ),থ,দ,ধ,ন,<br>প,ফ,ব,ভ,ম,য,(য়),ৰ ,র,ল, ৱ ,শ,ষ,স,হ |
| Dependent vowels | া,ি,ী,ু,ূ,ৃ, ৄ, ৢ ,ে,ৈ,ো,ৌ |
| Hasanta (joiner) | ্ |

In the proposed encoding scheme numerals, independent vowel, consonant and dependent vowel block are placed in linguistic order. Also within each block characters are placed in proper linguistic order. Da'ris (single and double) are presenting end of a sentence and having the least weight. Special symbol block is placed after Da'ri block but before vowel and consonant block similar to Latin script. Hasanta is assigned the highest value possible within code space. ZWNJ and ZWJ is global for Unicode used for all scripts. But in our proposed scheme for Bengali script unique $ZWNJ_b$ and $ZWJ_b$ are required to manage the linguistic ordering and are placed at the beginning of the code space. Other special symbols #, !, <, >. [, ], {, }, (, ) etc. and operators +, -, *, / etc. will be used from Latin scripts with respective encoded value.

If Bengali characters are encoded as per proposed encoding scheme, texts can be searched/ sorted by using any of the standard search/sort algorithms. Thus it will remove necessity of steps 2 and 4 of existing Bengali text sorting algorithms (see fig. 1) and will reduce about 70-75% of time required (see section 6).

## 5. TECHNICAL ANALYSIS

In proposed encoding system Da'ris, special symbols, numerals, independent variables, consonants have got proper linguistic ordered value. So, comparing any two character yields their respective order in code space.

Memory presentation of texts for dependent vowels and conjunctive consonants remain same as Unicode. All dependent vowels will be placed after consonants of conjunctive consonants in texts and application software will be responsible to render display dependent vowels' and conjunctive characters.

**Table 5. Character জ followed by various types of characters in linguistic order**

| | |
|---|---|
| জ | জ |
| জই | জ + ই |
| জর | জ + র |
| জা | জ + া |
| জি | জ + ি |
| জ্ঞ | জ+ ্ + জ |
| জ্ঞা | জ+ ্ + জ + া |
| জ্ঞ্র | জ+ ্ + জ + ্ + ব |

As hasanta is placed at the end of code space, it confirms that for a character its conjunctives will appear after the character itself and character followed by independent vowel, consonant, and dependent vowel. Consonant জ followed by various types of characters are listed linguistically in the following table which is achieved direct sort by encoded value in the proposed system.

$ZWNJ_b$ and $ZWJ_b$ is placed at the beginning of code space which confirms that the respective characters (joined or forced to not join) will be placed in proper linguistic order.

**Table 6. Linguistic sort with ZWNJ**

| | |
|---|---|
| বাহবা | ব + া + হ + ব + া |
| বাহ্‌বা | ব + া + হ + ্ + ZWNJ + ব + া |
| বাহ্বা | ব + া + হ + ্ + ব + া |

Da'ris will get lower ordered values in code space. Other punctuation marks (comma, semi-colon, colon etc.) will be used from Latin script. That ensures their lower value than any character from Bengali text. Table 7 shows order of comma, colon, semi-colon and da'ri.

**Table 7. Punctuation symbols sort order**

| | |
|---|---|
| বাংলা, | ব + া + ং + ল + া + , |
| বাংলা; | ব + া + ং + ল + া + ; |

| | |
|---|---|
| বাংলা: | ব + া + ং + ল + া + : |
| বাংলা। | ব + া + ং + ল + া + । |
| বাংলাঃ | ব + া + ং + ল + া + ঃ |
| বাংলাদেশ | ব + া + ং + ল + া + দ + ে + শ |

If experts on language thinks comma, colon, and semi-colon are not placed in their proper linguistic order then new code for these three punctuation mark need to be introduced in Bengali with proper linguistic order.

## 6. PERFORMANCE EVALUATION

We assign numbers to each of the Bengali characters such that their encoding has the order shown in Table 4. Then we write a program to sort Bengali texts using traditional Bengali sorting method shown in Fig.1. This implementation interface is available online at [10]. We apply the sorting algorithm on four sets of Bengali words randomly retrieved from Bengali dictionary. These four sets contain 100000, 150000, 200000, and 250000 words respectively. Various sections of the sorting algorithm are (i) Bengali text to intermediate code conversion, (ii) integrate remap information to data structure, (iii) perform the sort operation, here we use PHP sort() function with SORT_STRING as second parameter, and finally (iv) remap the sorted code to original text array to have the data sorted. Table 8 represent time requires by various sections of the algorithm.

**Table 8. Results and performance analysis**

| **Words** | **100000** | | **150000** | |
|---|---|---|---|---|
| i) Conversion time | 0.4053 | 33.75% | 0.6145 | 32.99% |
| ii) Integrate remap info. | 0.0833 | 6.94% | 0.1242 | 6.67% |
| ***iii) Sort time*** | ***0.3205*** | ***26.68%*** | ***0.5292*** | ***28.42%*** |
| iv) Remap time | 0.3920 | 32.63% | 0.5945 | 31.92% |
| **Total time** | **1.2011** | **100.00%** | **1.8624** | **100.00%** |
| **Words** | **200000** | | **250000** | |
| i) Conversion time | 0.9271 | 34.56% | 1.1728 | 33.94% |
| ii) Integrate remap info. | 0.1680 | 6.27% | 0.2205 | 6.38% |
| ***iii) Sort time*** | ***0.7678*** | ***28.62%*** | ***1.0203*** | ***29.52%*** |
| iv) Remap time | 0.8194 | 30.55% | 1.0423 | 30.16% |
| **Total time** | **2.6823** | **100.00%** | **3.4559** | **100.00%** |

The outcome of the program shown in Table 7 depicts that actual sort time is 26-30% of total time required. Remaining 70-74% times are used by pre-sort and post-sort data processing. Hence, using our proposed character encoding scheme for Bengali will save roughly 70% time for Bengali text sort operations. Thus it will save computing time and increase performance of both the search and sort operations on Bengali texts.

## 7. CONCLUSION

If the proposed encoding scheme is adopted in Unicode to encode Bengali characters, it will resolve the difficulties of Bengali data searching and sorting issue. Presently, developers need to write specialized functions to search and sort Bengali texts. But having this encoding scheme implemented, developers and users can apply any of the existing searching and sorting algorithms on Bengali text data. It is also noted that the proposed encoding method can also be applied to many other Indian languages to resolve the searching and sorting deficiency similar to Bengali texts. These languages are Hindi, Gujarati, Gurumukhi, Kannada, Telugu, Malayalam, Marathi, Oriya, Tamil, etc. These set of languages are originated from the ancient Brahmi language and having many similarity in their characters, character pronunciation, and positions in the Alphabet. So the proposed encoding scheme has the potential to contribute a lot to the South Asian Language Processing and Computing.

## 8. REFERENCES

[1] Bengali script code block, Unicode consortium

http://www.unicode.org/charts/PDF/U0980.pdf

[2] M. R. Amin, A. M. Samir, M. Chakraborty, and M. M. Rahman, *"An efficient Unicode based Sorting Algorithm for Bengali Words"*, International Journal of Computer Applications (0975-8887), volume 24-No.7, Jun 2011.

[3] M. A. Rahman and M. A. Sattar, "A New Approach to Sort Unicode Bengali Text", *Proceedings of 5th International Conference on Computer and Information Technology, ICECE 2008*, Dhaka, Bangladesh, pp. 628-630.

[4] S. M. Emrul Islam and M. M. Ali, "An Approach to Sort Unicode Bengali text using ancillary maps", *Asian Journal of Information Technology, 4(10)* pp: 890-894, 2005

[5] M. M. Murshed and M. Kaykobad, "Linguistically Sorting Bengali Texts: A Case Study of Multilingual Applications", *Proceedings of the 9th International Conference of the Information Resources Management Association*, Boston, Massachusetts, USA, pp.795—797, 1998.

[6] M A. Mottalib, "Development of a Bengali Word/Information Processor - A First Approach", *M. Sc. Thesis Work, Asian Institute of Technology*, 1984

[7] M. F. Zibran, A. Tanvir, R. Sammi and M. A. Sattar, "Computer Representation of Bangla characters and Sorting of Bangla words". *Proceedings of International Conference on Computer and Information Technology*, Dhaka, 2002, pp. 191-195.

[8] M. H. Khan, S. M. R. Haque, M. S. Uddin, R. Khan, and A. B. M. T. Islam, "An Efficient and Correct Bangla Sorting Algorithm", *Proceedings of 7th International Conference on Computer and Information Technology*, Dhaka, 2004, pp. 125-129.

[9] M. S. Rahman and M. Z. Iqbal, "Bangla sorting algorithm: A linguistic approach". *Proceedings of International Conference on Computer and Information Technology*, Dhaka, 1998, pp. 204-208.

[10] *Accuracy and Performance evolution of proposed encoding scheme*, http://www.banglacomputing.info

[11] Bangla Abhidhan, Bangla Academy, Bangladesh (ISBN 984-07-4642-1)