Bengali Writer utilities

Abhijit Das
Department of Computer Science and Automation
Indian Institute of Science, Bangalore 560 012
E-mail: abhij@csa.iisc.ernet.in

Abstract

Bengali Writer (BW) is a set of utilities for computerized typesetting in Bengali. In this article, we describe the components of BW. We also discuss about a Roman-to-Bengali transliteration scheme provided by BW. Some components of BW have already been ported to work with Devnagari fonts. Porting to other Indian languages is also possible.

1 Introduction

The advent of digital computers has revolutionarized the art of typesetting and publishing articles, documents, books etc. Anybody having a personal computer or a work-station and a printer can produce neatly typeset documents whose quality is no worse than the quality of books available from presses. In addition, computerized typesetting is easy to learn and documents thus produced are readily amenable to quick and effective maintenance and archival. Many presses are now publishing books by photocopying camera-ready copies available from the authors who typeset the materials using their favorite computerized document preparation systems.

Computerized typesetting can be classified into two broad categories. The first one is called WYSIWYG (What you see is what you get). In this case, a user, while keying in the document, can directly see on the computer screen how the document is going to look when a hard-copy is taken from a printer. Some examples of WYSIWYG text processors include Microsoft Word, Framemaker, Word perfect, Ventura etc. In the second paradigm, the user writes a program with hypertext markings and then compiles the program to get a file that can be fed to the printer. Examples of such non-WYSIWYG text processors are Troff, TEX, LATEX etc. Both these types of softwares produce the same quality of output. However, whether a user would prefer WYSIWYG or non-WYSIWYG systems depends on the user's personal choices and experience with computers.

Unfortunately, all these text processing softwares and the interfaces provided are suitable for typesetting in Roman fonts. They can be used for Indian languages too, but the mechanism is cumbersome and error-prone. Specific softwares written for Indian languages can make the process easier and much more user-friendly. The availability of computers in India at reasonably cheap cost is making computers more and more popular nowadays. This, in turn, raises the demand for text processing softwares tailored to Indian languages.

In this paper, we report the implementation of a set of text processing tools for Bengali. These tools are collectively known as the Bengali Writer or BW in short. BW includes fonts in various formats, that define the Bengali alphabet and a set of utilities to handle these font files. Typesetting in Bengali is made possible in two ways: by invoking the individual glyphs of the Bengali fonts or by using a transliteration mechanism for converting Romanized text to Bengali. Though BW is specific to Bengali, it can be ported to work for any other Indian language. Indeed a large part of BW has been ported for typesetting in Devnagari (Hindi, Marathi and Sanskrit).

In the next section, we talk about the various components of BW, namely the fonts and the interfacing softwares. We also list the encoding of the Bengali alphabet in the font files. In Section 3, we describe the Roman-to-Bengali transliteration algorithm implemented in BW. A list of similar softwares for Bengali and other Indian languages available at present is provided in Section 4. We conclude the paper in Section 5.

2 Components of BW

BW provides a Bengali font in several formats and a set of utility softwares for easy use of these fonts for typesetting text in Bengali. In this section, we describe about these fonts and the interfacing softwares. The text editor provided by BW has been converted to work for Devnagari. We also describe this Devnagari editor (called dvedit) in this section.

2.1 The Bengali font files

BW provides a Bengali font in the following formats. Upright and slanted versions of this font are available.

Format	Purpose	Platform
METAFONT	typesetting with TeX or LaTeX	Any
Bitmap distribution format (BDF)	viewing with X windows applications	Unix work-stations
PostScript type 1 (GSF)	taking print-outs	Any
GIF images	viewing in web browsers	Any
Text format	printing Bengali banners in text terminals	Any

All these font files are 256-character font files that provide the same encoding of the Bengali character set. We call this encoding the BW encoding. This encoding is listed in the following table. To get the ASCII code for a particular character just add the headings of the row and the column to which the character belongs. For example, the character ছ is assigned the ASCII value $64 + 3 = 67$ (which is the value for “C” in the ASCII Roman alphabet).

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0		ই	ঈ	উ	ঊ	ঋ	এ	ঐ	ও						ʃ	‿
16	স্থ	স্ন	স্প	স্প্ল	স্ফ	স্ব	স্ম	স্ল	হ্ন	রু	রূ	হু	হূ	গু	শু	হৃ
32		!	”	#	/-	%	হ্ব	’	(	)	*	+	,	-	।	/
48	০	১	২	৩	৪	৫	৬	৭	৮	৯	:	;	॥	=	হ্ব	?
64	ন্ধ	া	ৎ	ছ	ড	ৈ	ষ	ঘ	ঃ	ী	ঝ	খ	হ্ম	ং	ণ	ৗ
80	খ্ঞ	ঞ	ড়	শ	ট	ু	ঢ়	ু	ঢ	য	ঠ	[	“	]	´	`
96	‘	অ	ব	চ	দ	ে	ফ	গ	হ	ি	জ	ক	ল	ম	ন	ঔ
112	প	ঙ	র	স	ত	ূ	ভ	ঁ	ধ	য়	থ	–	—	হ্ল	~	ড়া
128	ক্ক	ক্ট	ক্ত	ক্ব	ক্ম	ক্র	ক্ল	ক্ষ	ক্ষ্ণ	ক্ষ্ম	ক্স	গ্ধ	গ্ন	গ্ব	গ্ম	গ্ল
144	ঘ্ন	ঙ্ক	ঙ্ক্ষ	ঙ্খ	ঙ্গ	ঙ্ঘ	ঙ্ম	চ্চ	চ্ছ	চ্ছ্ব	চ্ঞ	জ্জ	জ্জ্ব	জ্ঝ	জ্ঞ	জ্ব
160		ঞ্চ	ঞ্জ	ঞ্ঝ	ট্ট	ট্ব	ড্ড	ণ্ট	ণ্ঠ	ণ্ড	ণ্ণ	ণ্ম	ত্ত	ত্ত্ব	ত্থ	ত্ন
176	ত্ব	ত্ম	ত্র	ত্রু	দ্গ	দ্ঘ	দ্দ	দ্ধ	দ্ব	দ্ভ	দ্ভ্র	দ্ম	ধ্ন	ধ্ব	ন্ট	ন্ড
192	ন্ত	ন্তু	ন্ত্ব	ন্ত্র	ন্থ	ন্দ	ন্ধ	ন্ন	ন্ব	ন্ম	ন্স	প্ট	প্ত	প্ন	প্প	প্ল
208	প্স	ফ্ল	ব্জ	ব্দ	ব্ধ	ব্ব	ব্ল	ভ্র	ভ্ল	ম্ন	ম্প	ম্ফ	ম্ব	ম্ভ	ম্ভ্র	ম্ম
224	ম্ল	ল্ক	ল্গ	ল্গু	ল্ট	ল্ড	ল্প	ল্ব	ল্ম	ল্ল	শ্চ	শ্ছ	শ্ন	শ্ম	শ্ল	শ্ব
240	ষ্ক	ষ্ক্র	ষ্ট	ষ্ঠ	ষ্ণ	ষ্প	ষ্ফ	ষ্ম	স্ক	স্ক্র	স্ক্ল	স্খ	স্ট	স্ত	স্তু	স্ত্র

The Bengali alphabet comprises of vowels, vowel forms (when combined with consonants), consonants, conjunct consonants, digits and punctuation symbols. The font files define all these characters. Every individual conjunct consonant is defined in the font files (instead of a set of primitives for the component consonants).

2.2 The interfacing softwares

2.2.1 The Web interface

This is a utility that converts a text written in Roman to a text in Bengali. The program puts images one after another corresponding to the desired letters. The output can be viewed in any web browser. Files can be printed after saving them in PostScript. A Roman-to-html converter (called bn2html) is also available.

2.2.2 The TEX interface

A style file for easy access of Bengali characters in the METAFONT files is provided. This helps the user type Bengali characters at 'bad' font positions, that is, at ASCII positions not supported by the user's keyboard. It also provides a set of customized definitions for many characters. For example, ছ (which is छ in Devnagari) can be typed as `C` or as `\chh`. A Roman-to-Bengali converter (bn2tex) for TEX and LATEX is also under preparation.

2.2.3 A text editor

A Bengali text editor called 'bwedit' is designed. The editor runs on any X11 window system having the Tcl/Tk shell wish. It allows two modes for typesetting in Bengali. In the first mode, one keys in Bengali text directly from the keyboard. Characters at 'bad' ASCII positions can be keyed in by special short-cut key sequences or by clicking the buttons in an auxiliary keyboard map window. In the second mode of editing, one uses a Roman editor for typing transliterated Bengali text in a Roman font. An on-line view shows the equivalent Bengali text for the current line being edited. The current version of bwedit supports opening of multiple editor windows for each type of editing.

The editor incorporates some hypertext features, that is, inclusion of text in subscript, superscript etc., changing the font style (slanted or upright) and size, etc. It is also capable of exporting Bengali text to HTML, LATEX and PostScript. It provides facilities to send Bengali documents through e-mail.

2.2.4 A banner utility

This is a utility (called bnbanner) for displaying large Bengali banners in a text-only terminal. This is similar to the Unix command 'banner'.

2.2.5 A sort program

This is a program (called bwsort) for sorting Bengali text files written according to the BW font encoding. It reads multiple text files and sorts the concatenated list line-wise. 'bwsort' supports two different ordering schemes of the Bengali character set. The first one is based on the alphabetic positions of the Bengali characters in the Bengali alphabet. The second one is based on the character ordering which many Bengali dictionaries follow.

2.3 Porting to other languages

An older version of the Bengali text editor 'bwedit' has been ported to a Devnagari text editor 'dvedit'. Like bwedit, it works on X11 windows with the Tcl/Tk shell wish. It supports exporting text to HTML, LaTeX and PostScript and sending Devnagari text through e-mail.

There are METAFONT and/or BDF files for many other Indian languages (for example, Gurumukhi, Kannada, Malayalam, Oriya, Tamil and Telugu). 'bwedit' can be ported to editors of these languages. But I know neither of these languages. So I need help from people who speak, read and write these languages.

3 A Roman-to-Bengali transliteration scheme

When one communicates in Bengali through, say, e-mails, but using the Roman font, one uses a certain form of transliteration scheme. This scheme is usually very flexible, since it is meant for a human reader. For example, one might use 'a' to stand for both অ (अ) and আ (आ), as in 'katha' to imply কথা (कथा). The reader makes sense out of it using the database of Bengali words (s)he knows and often taking help of the context of the word. A transliteration scheme for computers, on the other hand, is meant for a machine which does not *know* Bengali, and is, therefore, expected to be more rigorous than the ad hoc schemes mentioned above. This section discusses about the Roman-to-Bengali conversion rules provided by BW. In the discussion we refer to an *encoding file* which defines the encoding for Bengali vowels, consonants and conjunct consonants.

- To start with one fixes a *vowel alphabet* and a *consonant alphabet* as two disjoint sets of Roman letters. Elements of the vowel alphabet and the consonant alphabet are called *vowel symbols* and *consonant symbols* respectively. One also fixes a Roman character not in the above two alphabets as a *syllable separator*. Any other Roman letter is called a *non-alphabetic symbol*. As an example, BW works with the following default choices:

  vowel alphabet = $\{$a, A, e, E, i, I, o, O, u, U$\}$

  consonant alphabet = $\left(\{\hat{}\} \cup \{\text{ A} - \text{Z }\} \cup \{\text{ a} - \text{z }\} \right) \setminus$ (vowel alphabet)

  syllable separator = _

- A *vowel code* is a *maximal* sequence of (contiguous) vowel symbols. A vowel code is said to be associated with a Bengali vowel and/or vowel form when the encoding file specifies a font character for that vowel code. Otherwise the vowel code is said to be undefined.

- A *consonant code* is a *maximal* sequence of (contiguous) consonant symbols. A consonant code is said to be associated with a Bengali consonant (or conjunct consonant) when the encoding file specifies a font character for that consonant code. Otherwise the consonant code is said to be undefined. Note that conjunct consonants are treated exactly the same way as the pure consonants.

- A *token* is either a vowel code or a consonant code or the syllable separator or a non-alphabetic symbol.

- BW reads the input Roman text sequentially and generate tokens one after another as they appear in the input file.

- When BW finds a non-alphabetic symbol, it simply transfers that to the output.

- When BW finds a vowel code not (immediately) preceded by a consonant code, it outputs the Bengali vowel encoded by the vowel code if the encoding exists, else it outputs nothing.

- When BW finds a consonant code not (immediately) followed by a vowel code, it outputs the corresponding Bengali consonant character if it is defined, else it outputs nothing.

- When BW finds a consonant code (immediately) followed by a vowel code, it outputs the corresponding encoded consonant character surrounded by the appropriate vowel form(s). If the consonant code is undefined, it outputs a space for it. If the vowel code is undefined, it outputs nothing (i.e. no vowel forms).

Let us exemplify the rules with a simple example. Consider the input

```
bhaarater ek bhaassaa.
```

BW separates the following tokens:

`bh`	consonant code
`aa`	vowel code
`r`	consonant code
`a`	vowel code
`t`	consonant code
`e`	vowel code
`r`	consonant code
`[space]`	non-alphabetic symbol
`e`	vowel code
`k`	consonant code
`[space]`	non-alphabetic symbol
`bh`	consonant code

`aa`	vowel code
`ss`	consonant code
`aa`	vowel code
`.`	non-alphabetic symbol

It then applies the above rules to generate

ভারতের এক ভাষা। (भारतेर एक भाषा।) (This means ‘a language of India’.)

Of course the story does not end here. Suppose we want to get একটি (एकटि) and input `ekTi`. This gives এক্টি (एक्टि). This is because `kT` is the encoding of ক্ট (क्ट). We have to use the syllable separator (_) to separate `k` and `T`. Since _ is not a consonant symbol, the maximal sequence of consonant symbols that BW finds is `k`. Since _ is also not a vowel symbol, it simply outputs the character corresponding to `k`, that is ক (क). BW outputs nothing for the syllable separator. Then it finds the consonant code - vowel code combination `Ti` and outputs টি (टि). So the rule here goes like this:

- Whenever one wants to put two successive consonant characters without a vowel sound appearing between them, one should separate their codes by the syllable separator.
- Whenever one wants to put two successive vowel characters without a consonant sound appearing between them, one should separate their codes by the syllable separator.

The second rule is exemplified by দুইটি (दुइटि). This requires the input `du_iTi` and not `duiTi`, because the vowel code `ui` is not defined at all.

This completes the description of the transliteration algorithm. There are further subtleties in the algorithm. We are not going to deal with them, but refer the reader to the on-line or off-line documentation available with the BW packages. We end this section by a more elaborate example:

বাংলা পূর্ব ভারতের একটি বহুল প্রচলিত ভাষা। (बांला पूर्ब भारतेर एकटि बहुल प्रचलित भाषा।)

which means ‘Bengali is a widely used language of east India.’ The transliteration encoding for this sentence is:

```
baaM_laa pUrba bhaarater ek_Ti bahul prachalita vAssA.
```

4 Other resources for Bengali and other Indian languages

In this section, we list a set of utilities other than BW, that can be used for typesetting in Bengali and various other Indian languages. This list is by no means exhaustive. The reader may refer to the web-page

`http://theory.tifr.res.in/bombay/history/people/language/`

for a detailed listing of the resources for Indian languages.

CDAC, Pune has developed text processing softwares for many Indian languages. These work on Microsoft windows, Mac and various Unix work-stations and are commercial softwares. The directory

`http://www.tex.ac.uk/tex-archive/languages/`

at the CTAN archive refers to many freely downloadable utilities for TeX-related word processors. In particular, it contains METAFONT files for Bengali, Devnagari, Gurumukhi, Malayalam, Oriya, Tamil and Telugu. A popular Roman-to-TeX converter known as ITRANS for these fonts is also available from the CTAN archive. For more information on ITRANS, look at the page maintained by its author Avinash Chopde:

`http://www.aczone.com/itrans/`

Avinash has also written a commercial package known as the ‘Indian Language Keyboard Program’. The page for this is

`http://www.aczone.com/ilkeyb/`

Angshuman Pandey has developed a Roman-to-TeX converter similar to ITRANS but more tailored to Bengali. Subhabrata Biswas’s program called IWRITE32 can be used on Microsoft windows platforms. These two packages are available for free downloading. The former can be downloaded from the CTAN archive mentioned above. For IWRITE32, visit the page

`http://members.tripod.com/~sbiswas/IWrite32/IWrite32.html`

Some other softwares that support document preparation in Indian languages, are: InProS, Vijay Patel’s true type fonts, Wordmate etc. All of these work on Microsoft windows and are commercial.

5 Conclusion

In this paper, we describe a set of utilities for typesetting text in Bengali. These utilities have been developed by the author in the Indian Institute of Science, Bangalore, and can be downloaded and redistributed free of charge for all sorts of academic and research purposes. We refer to the webpage

`http://www.geocities.com/SiliconValley/Lab/6024/bengali.html`

for more information on BW. For the Devnagari editor dvedit, look at the page

`http://www.geocities.com/SiliconValley/Lab/6024/dvedit.html`

In addition to incorporating new features to the existing BW utilities, there are scopes for further developments in this area. Here are some directions:

- Development of a full-fledged text processor (bwedit is a text editor with limited text processing capabilities.)
- Compiling an on-line Bengali dictionary (or word list, at least) for spell-checking and other purposes.
- Development of translation aids between Bengali and English and/or other Indian (and European) languages.

Before we end this paper, we point out a problem associated with typesetting in Indian languages. Since the character set and the associated encoding of the same has not been standardized yet, different fonts use different encodings of the characters. This causes incompatibility between different fonts and interfaces. To alleviate this difficulty we badly need a standard encoding scheme for all Indian languages.

Acknowledgements

I am grateful to the users of BW, who provided me feedback in various aspects. Special thanks go to P. A. Meyer for his valuable suggestions regarding the Roman-to-Bengali transliterator, Probal Sengupta for sharing his views about Bengali fonts, Angshuman Pandey for adding two Assamese characters to BW font files, Jaijeet Roychowdhury for writing the ITRANS interface of BW METAFONT files and Rahul Athale for helping me convert bwedit to dvedit. I also thank all of my colleagues and friends for encouraging me to develop and upgrade the BW utilities.