M.Tech. (Computer Science) Dissertation Series
2004-2005

DEVELOPMENT OF A WINDOWS-BASED SOFTWARE FOR BUILDING OF BENGALI DATABASE FROM SCANNED DOCUMENTS (SOFT COPY OF BENGALI DICTIONARY)

A dissertation submitted in partial fulfillment of the requirements
for the M.Tech.(Computer Science) degree of the
Indian Statistical Institute

BY

TILAK KUMAR ADHYA

Under the supervision of

Dr. PALASH SARKAR
Applied Statistics unit (ASU)

INDIAN STATISTICAL INSTITUTE
203, Barrackpore Trunk Road
Calcutta - 700035

Acknowledgements

I take pleasure in thanking Dr. Palash Sarkar for his valuable guidance throughout the dissertation period. His encouraging words have always kept my spirits up.

I express my deep gratitude to Dr. D.P.Mukherjee of ECSU and Dr. Mandar Mitra of CVPRU for their valuable guidance, suggestions and inspiration. I would like to thank Dr. Abhijeet Das, IIT Kharagpur, for his valuable advice at the initial stage of our work. I am very grateful to Dr. Arnab Chakraborty, ISI, for his valuable advice and providing me his Bengali OCR program. Finally I take this opportunity to thank my classmates and friends for their help, support and lighter moments throughout the course.

Last but not the least I am grateful to all the teachers and staff of ISI.

Tilak Kumar Adhya

To Whom It May Concern

This is to certify that **Mr. Tilak Kumar Adhya** of **M.Tech. (Computer Science)**, second year student of **Indian Statistical Institute, kolkata**, has done his dissertation titled **Development of a Windows based Software for Building of a Bengali Database from Scanned Documents(Soft Copy of Bengali Dictionary)** under my guidance. This dissertation partially fulfills the requirement of M.Tech. (Computer Science) curriculum.

Palash Sarkar.

Date:
Place: Kolkata

(Dr. Palash Sarkar)
Applied Statistics Unit
Indian Statistical Institute (Kolkata)

Contents

1 Introduction

To learn a language we always need the help of a dictionary in that language. Even for writing a document in any language it will be advantageous to have a dictionary. In the 21st century we are not using pen and paper for writing something. We are always typing in computer for preparing any document. This is not only true for English, but also for any other languages. All languages have their own dictionaries, but these are printed version in the form of a book. Now there is a need for soft copy of dictionary in each subject. The soft copy of dictionary can be linked to the editor of that language.

In this dissertation we try to build a software, which will serve the purpose of a soft copy of Bengali Dictionary like the soft copy of Oxford Dictionary available in English. A small GUI will be displayed on the screen where the user are supposed to write the word whose meaning he/she wants to know. Then if the user press the 'search' button of the GUI, the meaning will be displayed to the user. There will be a option in the GUI such that the user can hear the pronounciation of the word he/she typed in the text area. The software must provide the facility of cross linking of words i.e. if the user wants to see the meaning of another word which is displayed as a meaning of other word then simply by clicking the mouse over the word he can find the meaning of that word. So there will be no need to write the word again.

Till today no soft copy of Bengali Dictionary is available to the best of our knowledge. That's why we are trying to make such a dictionary.

1.1 Design of Work

To make a dictionary we've to first built a Bengali word database. Typing all the words from a bengali dictionary is cumbersome. So our idea is to use a software which can automate the process. For this reason we take help of Optical Charecter Recognition (*OCR*) software for reocognition of Bengali words. So we thought that preparing for a Bengali Dictionary database would be very easy. Just giving the scanned page of Bengali Dictionary as the input of the OCR software, will give us the soft copy of that page. In this way we can have whole of the dictionary as a soft copy in our machine. But the output of the OCR gives the unstructured $\TeX$ representation of the scanned dictionary pages. Now we've developed a program, depending upon the pattern of the $\TeX$ output, which will put the $\TeX$ output of the

OCR into the database in a structured way(i.e. segmenting into different parts). So using OCR we've been able to make a Bengali database.

Now comes the second phase of our project i.e. making of a GUI where the user can type his Bengali words to find the meaning of the word which is already in the database and displaying the meaning of the word in bengali fonts to the user. The user write the word in romanized bengali. We have converted this to the corresponding LaTeX representation and then applying a search query in the database extracted the meaning of the word. Then TeX representation of the meaning of the word is converted to the coresponding Bengali fonts and displayed on the screen. User can also hear the pronounciation of his word from this dictionary which will be one of the advantages over hard copy of dictionary.

2 Software Description

After running the software a window will appear where the user can write the Bengali word, whose meaning he wants to know, in romanized Bengali and then press the search button. Then a new DVI file named diss1.dvi(diss1 - program name) will be created in the directory where the software lies and that dvi file contains the meaning of the Bengali word. In this way user can change the word and press search to find the meaning of his desired words. We take the Bengali words and their meaning from the well-known Bengali Dictionary published by 'Sangsad'. In making of this software we have built the Bengali word database by running an Optical Character Recognition program on the scanned Bengali dictionary pages. The output of the OCR program is unstructured TEX representation of scanned Bengali Dictionary pages. These unstructured documents are then entered into the database in a structured way. The following example will clear the above idea

Output of OCR (for a single word) :-

Anacar(A'na'car) ib shaes/tRr ba
pRitiSh/Tht riiitr ibrud/dh AacrN ba kajkr/m;
mn/d AacrN,inn/dniiy AacrN. 'sNNG n
AacarNNG tt//'.ibN Anacrii

In our database we have 5 column for each word entry i.e. we break each word into 5 segments. For example the above word can be segmented as follows:-

1) Anacar;
2) A'na'car
3) ib shaes/tRr ba
pRitiSh/Tht riiitr ibrud/dh AacrN ba kajkr/m;
mn/d AacrN,inn/dniiy AacrN.
4) sNNG n
AacarNNG tt//
5) ibN Anacrii.

Each of the above parts is entered into the table. So the Bengali words are stored into the database in the TEX format. Now when a user writes a Bengali word and press search button to know its meaning, firstly the word is converted to corresponding TEX representation, then a search query is run to the database for the TEX representation of the word. If the word is available in the database, then the query returns the meaning of the word in the TEX format. We have converted the TEX representation of the word to the actual Bengali fonts and display it in a 'dvi' file.

3 Software Specification

3.1 System Requirements

Windows98/XP, j2sdk-1.4.1, MiKTeX, Oracle, gcc compiler for windows(win_gcc), BangOCR(Optical Charecter Recognition for Bengali words).

3.2 Software Installation

3.2.1 BangOCR

Unzip the zip file in a directory, say C:BangOCR.
Unzip the bfonts.tar file available with that in the same directory.

3.2.2 win_gcc

To install the gcc compiler for windows see http://www.delorie.com/djgpp/zip-picker.html for a form-based guide to what to download. In general, download the binary distributions only. To build C programs, you'll need djdev203.zip, gcc*b.zip and bnu*b.zip. For C++, also get gpp*b.zip.

Installation

a. Create a directory for DJGPP, say C:DJGPP. (WARNING: do NOT install DJGPP in a directory C:DEV or D:DEV etc, or in any of their sub-directories: it will not work!). Do not use long directory name or one with spaces or special characters.
b. Unzip all the zip files from the directory, preserving the directory structure. For example:

```
pkunzip -d djdev203
Or
unzip32 djdev203
```

c.After unzipping all the zip files, set the DJGPP environment variable to point to the file DJGPP.ENV in the main DJGPP installation directory and its BIN subdirectory to your path. The exact way that these variables should be set depends on your operating system:

For Windows 98 systems:

- Click START;
- Choose Programs ; Accessories ; System Tools ; System Information;
- Click tools in the menu-bar, then choose "System Configuration";

- Use the tab provided there for editing your AUTOEXEC.BAT as explained below.

For Windows 2000 or Windows XP systems:

- Right-click "My Computer", then select "Properties";
- Select the "Advanced" tab, then click "Environment Variables" button;
- Edit the PATH system variable to add the DJGPP bin subdirectory; (if you are not the administrator, add the DJGPP bin directory to the user PATH variable - or create one with only this directory since it is added to the system path).
- Add a new variable DJGPP and set its value to the full path name of the DJGPP.ENV file as explained below.

Assuming your DJGPP installation is rooted at C:, the values of the two environment variables DJGPP and PATH should be set like this:

```
Set DJGPP=C: DJGPP DJGPP.ENV
Set PATH=C: DJGPP BIN;
```

d.Reboot. This makes sure the two lines added to AUTOEXEC.BAT will take effect. (On Windows NT/2000/XP, the changes take effect immediately, so you don't need to reboot there, but you do have close and reopen the DOS box windows.)

e. Run the go32-v2.exe program without arguments:

```
Go32-v2
```

It should report how much DPMI memory and swap space can DJGPP use on your system.

Compilation

GCC is a command-line compiler, which you invoke from the DOS command line. To compile and link a single-file C program, use a command like this:

```
gcc myfile.c -o myfile.exe -lm
```

The -lm links in the lib/libm.a library (trig math) if needed.

To compile a C source file into an object file, use this command line:

```
gcc -c -Wall myfile.c
```

This produces the object file myfile.o. The '-Wall' switch turns on many useful warning messages, which are especially beneficial for new users of GCC.

One can also combine the compilation and link steps, like this :

```
gcc -Wall -o myprog.exe mymain.c mysub1.c mysub2.c
```

Further information of DJGPP will be available in readme page of DJGPP.

3.3 Database Design

We have stored the TEXrepresentation of Bengali dictionary pages in a database. For this reason we have used Oracle8i as our database system. All the Bengali words are stored in a table containing 5 coloumns.

- First coloumn contains the Bengali word.
- Second coloumn contains the pronounciation of that word.
- Third coloumn contains all the meaning of that word.
- Fourth coloumn contains the grammar of that word.
- Fifth coloumn contains the related forms of that word.

The primary key of the table is the first coloumn which contains the word because any word is present in the dictionary only once. While inserting the word into the database and finding the meaning of a particular word, we have interacted with the database.We have used the following code to connect the database with our java program.

```
  class Connection1{
    public Connection returnCon(){
      String driver="sun.jdbc.odbc.JdbcOdbcDriver";
      String url="jdbc:odbc:tilak";
      Connection con ;
    try{
      Class.forName(driver);
      con= DriverManager.getConnection(url,"scott","tiger");
      return con;
    }catch (Exception e){
      System.out.println ("Database not connected/problem in sql");
      e.printStackTrace();
      return null;
      }
    }
}
```

Here we have used Sun Jdbc Odbc Driver. Our global database name is 'tilak'. We have used SQL for querying the databse. The following SQL is used to insert the records of a single word.

```
stmt.executeUpdate("INSERT INTO dic2 VALUES ('"+str1+"','"+str2+"
','"+str3+"','"+str4+"','"+str5+"')");
stmt.executeUpdate("COMMIT");
        where, Statement stmt = conn.createStatement();
                'str1' contains the record of first coloumn etc.
                'dic2' is the table name.
```

To find the meaning of a word the user writes the word in a text field. Suppose the string "f" contains the word and corresponding TEX is stored in the string "str2". Then applying a search algorithm to the database with the string "str2" we will get the record of that word stored in the database. The following SQL query will do this job.

```
SELECT * FROM dic2 WHERE c1=('"+str2+"')
        where, c1 represents the name of the first coloumn of the "dic2" table.
```

3.4 How to Start the Software

Assuming DJGPP is installed in the directory C: DJGPP, execute the following commands in the DOS mode to get the dictionary.

1. Entered into the directory C: DJGPP gcc342b bin
2. javac diss_front1.java
3. java diss_front1

NOTE : diss_front1.java is our java program.
Now a separate window will open . The user will write the word and press 'search' to get the meaning of the word in a separate 'dvi' file which will be in the C:/DJGPP/gcc342b/bin directory.

4 Algorithms

In making of this software we have mainly used two different algorithms, one for inserting the LaTeX representation of Bengali words into the database and the other one is to build the user interface for searching the words. Let us first describe the first one in an algorithmic way given below.

4.1 Algorithm For Inserting Bengali Words(TeX format) into the Database

Suppose ‘in’ points to the input file containing the LaTeX representation of Bengali words , i.e. ‘in’ points to the output file obtained by running OCR program on the scanned Bengali dictionary pages.

```
while (in.available() !=0)
{
byte b = (byte)in.read();
    String str=””;
    while(b==’ ’ OR b==’ t’ OR b==’ n’)
    {
        b = (byte)in.read();
    }
    while(b !=’(’ )
    {
        str=str + (char)b;
        if(b ==’ n’)
        break;
        if(in.available() ==0 )
        break;
        b = (byte)in.read();
    }
    System.out.println(str + ”-1”); //Segment 1
    str=””;
    while(b !=’)’)
    {
        str=str +(char)b;
        if(b ==’ n’)
        break;
    if(in.available() ==0 )
    break;
    b = (byte)in.read();
```

```
}
str=str +(char)b;
System.out.println(str + ”-2”); //Segment 2
str=””;
b=(byte)in.read();
while(b !=’‘)
{

    str=str +(char)b;
if(in.available() ==0 )
break;
b = (byte)in.read();
}
System.out.println(str + ”—3”); //Segment 3
str=””;
b=(byte)in.read();
while(b!=”’)
{
str = str + (char)b;
if(in.available() == 0)
break;
b = (byte)in.read();
}
System.out.println(str + ”——4”); //Segment 4
str = ””;
b = (byte)in.read();
}
b = (byte)in.read();
while(b==’ ’OR b==’ t’OR b==’ n’)
{
    b = (byte)in.read();

}
str = ””;
if(b==’i’)
{
    while(b!=’.’)
{
    str = str + (char)b;
    b = (byte)in.read();
```

```
}
System.out.println(str + ”——5”); //Segment 5
}
```

4.2 Algorithm For Finding Word from the Database

The user writes the Bengali word in Romanised Bengali, but in the database the words are stored in LaTeX format. So first we have to change the word to LaTeX representation, then running a search query we extract the word from the database and then we have to convert it to Bengali fonts. So the algorithm consists of the following three steps.
1. Convert the User written Bengali word to corresponding LaTeX representation.
2. Run a search query with the LaTeX representation of the word.
3. Convert the records with the word available in the database from LaTeX to Bengali fonts and display it.

5 Future Goal

Our future goal is to incorporate the following features into our software,
a) In the present software user writes the Bengali word in the GUI in romanized Bengali. It will be better if he/she can write it in Bengali fonts in the GUI.
b) Rather than displaying the meaning of the word in a separate dvi file it will be advantageous if the meaning can be displayed in the same GUI where the user wrote the word to know the meaning.
c) Cross-linking of words, i.e. if the user clicks over any Bengali word on the dictionary then our dictionary should show the meaning of that word. This important aspect of a soft copy of a dictionary can be incorporated to this software.
d)Pronunciation can be attached with each word, so that user can properly pronounce the word in future.

6 Conclusion

To make the Bengali word database we have used OCR program. But we have faced a tough situation while using the OCR. We have tested for different OCR but no one is better than the other. Using the OCR we are not able to make the Bengali word database at all. Research is going on to make a good OCR for Bengali words.

In the present version of the software the output is displayed in a separate dvi file. We are not able to keep the TEX information with each Bengali word displayed in that dvi file. This creates the problem for cross-linking of words.

We have tested the software for different words by mannually typing the words instead of using OCR. It is working fine for almost all the words.

7 Appendix

7.1 Dictionary Database

Due to failure of OCR the dictionary contains only the following Bengali words.
anAlOchit; anAchAr; anAs+thA; anAhAr; anik+et; anAdi; anir+dhArit; anir+bAn; anir+bApit; kkh+Ir; kkh+IrA; kkh+uN+N; kkh+ud+ra; kkh+ub+dha; anavilAS; anaman+Iy; anashan; analas; anAmikA; adhar+ma; ashiS+T; ash+It+ipar; ash+In; ash+uch+i; ash+ud+dha; ash+uva; ash+eS; ash+O; ash+Odh+it; ash+Ovan.

7.2 Sample OCR Output

This is the corresponding OCR TEX output of the Bengali Dictionary page given in the next page.

```
AishSh/T (A'ishsh'eTa) ibN AbhdR;ruicHiin;
Ash/liil (Aish AacrN).'sNNG n ishSh/T.
 n tt//'ib AishSh/Tta.
Ashiiitpr (A'ish'it'pr)ibN Jar bys
Aaish ba tar ebish.Ait brRd/dh, buerha
(Ashiiitpr brRd/dh).'sNNG Ashiiit Aaish
pr'.
Ashiin (A'ishn) ibN buid/dhman,ebadhJ.'sNNG
n shiin'.
Ashuic (A'shu'ic) ibN ApibtR,ApirSh/kar,
Jaek echaya Jay na ba Eirhey clt Hy
(etar kuuepr bair Ashuic rbiin/dR)'sNNG n
shuicNNG tt//'. ib Ashuicta.
Ashud/dh (A'shud'edha)ibn shud/dh ba inr/bhul ny
Emn,bhul(Ashud/dh banan); pibtR ny
Emn.ApibtR.'sNNG n shud/dhNNG tt//'
ib Ashud/dhta.
Ashubh (A'shu'ebha) ibN shubh ba mNG/gljnk ny
Emn (Ashubh gRH,Ashubh lkKn).
ib AmNG/gl AklYan Ashubh ghTna.'sNNG n
shubhNNG tt//'.
AeshSh (A'eshSh)ibN Jar eshSh enI,AineshSh
(AeshSh Aasha, AeshSh trRip/t
pRcur;siimaHiin.'sNNG n eshSh'.
```

Aesha (A'eshak) ibN pRaciin bhartR mur/J sr/maT. ibN Jar eshak ba dukh enI.'sNNG n eshak.'.
Aeshaidht (A'esha'idh'eta) ibN eshadhn ba sNNGeshadhn kra ba sNNGs/kar kra Hyin Emn; (Aeshaidht etl); Amair/jt.'sNNG n eshaidht. n tt//'.
Aeshabhn (A'esha'ebhan) ibN sun/Dr ny ba ruicsm/mt ny Emn;kut//ist;ebmanan (Aeshabhn epashak).'sNNG n eshabhn. n tt//'

অশিষ্ট (অ'শিশ'টো) বিণ অভদ্র;রুচিহীন; অশ্লীল (অশি আচরণ)।'সং ন শিষ্ট। ন তৎ'বি অশিষ্টতা।

অশীতিপর (অ'শি'তি'পর)বিণ যার বয়স আশি বা তার বেশি।অতি বৃদ্ধ, বুড়ো (অশীতিপর বৃদ্ধ)।'সং অশীতি আশি পর'।

অশীন (অ'শিন) বিণ বুদ্ধিমান,বোধয।'সং ন শীন'।

অশুচি (অ'শু'চি) বিণ অপবিত্র,অপরিষ্কার, যাকে ছোয়া যায় না বা এড়িয়ে চলত হয় (তোর কূপের বারি অশুচি রবীন্দ্র)'সং ন শুচিং তৎ'। বি অশুচিতা।

অশুদ্ধ (অ'শুদ'ধো)বিন শুদ্ধ বা নির্ভুল নয় এমন,ভুল(অশুদ্ধ বানান); পবিত্র নয় এমন।অপবিত্র।'সং ন শুদ্ধং তৎ' বি অশুদ্ধতা।

অশুভ (অ'শু'ভো) বিণ শুভ বা মঙ্গলজনক নয় এমন (অশুভ গ্রহ,অশুভ লক্ষন)। বি অমঙ্গল অকল্যান অশুভ ঘটনা।'সং ন শুভং তৎ'।

অশেষ (অ'শেষ)বিণ যার শেষ নেই,অনিশেষ (অশেষ আশা, অশেষ তৃপ্তি) প্রচুর;সীমাহীন।'সং ন শেষ'।

অশো (অ'শোক) বিণ প্রাচীন ভারত্র মুর্য সর্মাট। বিণ যার শোক বা দুখ নেই।'সং ন শোক।'।

অশোধিত (অ'শো'ধি'তো) বিণ শোধন বা সংশোধন করা বা সংস্কার করা হয়নি এমন; (অশোধিত তেল); অমার্জিত।'সং ন শোধিত। ন তৎ'।

অশোভন (অ'শো'ভোন) বিণ সুন্দর নয় বা রুচিসম্মত নয় এমন;কুৎসিত;বেমানান (অশোভন পোশাক)।'সং ন শোভন। ন তৎ'