Chapter 8

Results and Discussion

In earlier chapters we have discussed options for carrying out various sub-tasks of OCR for Gujarati script. In most of the cases we have also given results of testing the algorithms described in that chapter. Here, we collate those result to build OCR system. We have tested the end to end system on pages from books. The test results are used to propose an ideal work flow for Gujarati OCR with selection of algorithms at various stages. Text corpus generated from the scanned pages has been used to validate the output of the OCR and find the character level accuracy.

8.1 Preliminaries

Let us try to understand the edit distance which is used to compute the accuracy of OCR output.

8.1.1 Edit Distance

Edit Distance

Given two character strings s_1 and s_2 , the *edit distance* between them is the minimum number of edit operations required to transform s_1 into s_2 . The edit operations allowed for this purpose are: (i) insert a character into a string; (ii) delete a character from a string and (iii) replace a character of a string by another character; for these operations, edit distance is sometimes known as *Levenshtein distance* . For example, the edit distance between "cat" and "dog" is 3.

Levenshtein Distance

Implementation of this concept uses the dynamic programming algorithm , where the characters in s_1 and s_2 are given in array form. The algorithm fills the entries in a matrix m of the size $(|s_1|+1) \times (|s_2|+1)$. Where, $|s_1|$ and $|s_2|$ denotes the lengths of strings s_1 and s_2 respectively. The $(i,j)^{th}$ entry of the matrix will hold (after the algorithm is executed) the edit distance between the strings consisting of the first i characters of s_1 and first j characters of s_2.. The central dynamic programming step is depicted in Lines 8-10 of Figure 3.5 , where the three quantities whose minimum is taken correspond to substituting a character in , inserting a character in and inserting a character in .

Algorithm 8.1 To Computer Edit Distance Between Two Strings

Input: Strings s_1 and s_2.

Output: Edit distance between s_1 and s_2.

Process:

```
m[0,0] ← 0
for i = 1 to |s1| do
    m[i,0] ← i
end for
for j = 1 to |s2| do
    m[0,j] ← j
end for
for i = 1 to |s1| do
    for j = 1 to |s2| do
        if s1[i] = s2[j] then
            temp ← 0
        else
            temp ← 1
        end if
        m[i,j] ← min{m[i-1,j-1] + temp, m[i-1.j] + 1, m[i,j-1] + 1}
    end for
end for
return m[|s1|, |s2|]
```

8.2 OCR Testing

We would like to recall that this being the first effort for developing an end-to-end Gujarati OCR, we are targeting the documents with single column of text without graphics, images or any other special decorative fonts and formats. We are also assuming pages without broken and touching glyphs.

In chapter one we have described various subtasks of an OCR system. For

our experiment we have used pages from different books scanned at 300dpi resolution with 8-bit gray scale. That is total 256 gray levels. As mentioned earlier that this gray level image can have noise due to various reasons.

Mean and median filters can be used to remove such noise. As mentioned earlier mean filter may give rise to edge blurring which in turn can generate ladder effect in binarization. Hence we use edge preserving median filter to remove isolated noise pixels.

The next step is binarization to separate text from its background. We have tested ocr system by using both the binarization method presented earlier in this work and we will present the result while testing on pages from books.

Binarized image is then subjected to segmentation. As mentioned earlier that we have considered pages with single column text without any images in it. Hence we are not doing the step for text - image separation and directly proceed to the next step. As mentioned in chapter 2 we follow top down approach for segmentation. We use the algorithms mentioned earlier for line and word segmentation. It is also clear from the discussions that we need to carry out zone boundary identification and hence the next task is to do zone boundary identification.

There are different approaches described earlier for zone boundary identification. As mentioned in chapter 4 the zone boundary detection though the algorithm mentioned there has significant limitations. Hence, we are using **Algorithm** 3.2 for zone boundary detection for this testing.

It is clear from the statistics for classifier accuracy that fringe map features with nearest neighbor classifier gives best classification accuracy. Therefore, we have selected this combination for this testing.

Recognized glyphs are then used for text generation through the method presented in chapter 7 and unicode text with appropriate line and word breaks is given as output of the system.

Edit distance is used to measure the accuracy of the output. Unicode characters are used as bases for comparison. Text output of a page is considered a long string of Unicode characters,say S_r and edit distance of this string with the manually entered and proof read text (S_o) for that page is computed.

Error rate E in percentage, is computed as follows :

$$E = \frac{ED(S_r, S_o)}{NU}\% \tag{8.1}$$

where $NU =$ No of Unicode characters in proof read text.

We have used software developed as a part of consortia mode project of The Ministry of Communications and Information Technology to measure the accuracy.

8.3 Results and Discussion

Following tables give results of experiments done with the work-flow mentioned above.

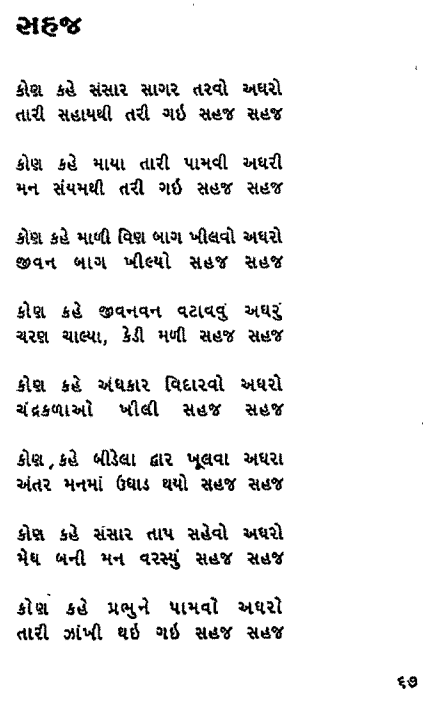
Table 8.1: % Error with Niblack's binarization

Book Name	No of Pages	% Error		
		Minimum	Maximum	Average
Prateeksha	76	4.10	24.8	14.89
Birbal in Bahoshi	254	7.08	25.5	13.46

Table 8.2: % Error with Sauvola and Pietaksinen s binarization

Book Name	No of Pages	% Error		
		Minimum	Maximum	Average
Prateeksha	76	4.57	24.0	14.02
Birbal in Bahoshi	254	4.57	27.3	12.25

It can be seen that the recognition accuracy varies greatly. There are several reasons to which these errors can be attributed to like error in zone boundary identification, classifier error and error in text generation etc.. Fig. 8.1 and Fig. 8.2 give output of OCR software developed as a part of this research work.



સહજ

કોણ કહે સંસાર સાગર તરવો અઘરો
તારી સહાયથી તરી ગઈ સહજ સહજ

કોણ કહે માયા તારી પામવી અઘરી
મન સંયમથી તરી ગઈ સહજ સહજ

કોણ કહે માળી વિણ બાગ ખીલવો અઘરો
જીવન બાગ ખીલ્યો સહજ સહજ

કોણ કહે જીવનવન વટાવવું અઘરું
ચરણ ચાલ્યા, કેડી મળી સહજ સહજ

કોણ કહે અંધકાર વિદારવો અઘરો
ચંદ્રકળાઓ ખીલી સહજ સહજ

કોણ, કહે બીડેલા દ્વાર ખૂલવા અઘરા
અંતર મનમાં ઉઘાડ થયો સહજ સહજ

કોણ કહે સંસાર તાપ સહેવો અઘરો
મેઘ બની મન વરસ્યું સહજ સહજ

કોણ કહે પ્રભુને પામવો અઘરો
તારી ઝાંખી થઈ ગઈ સહજ સહજ

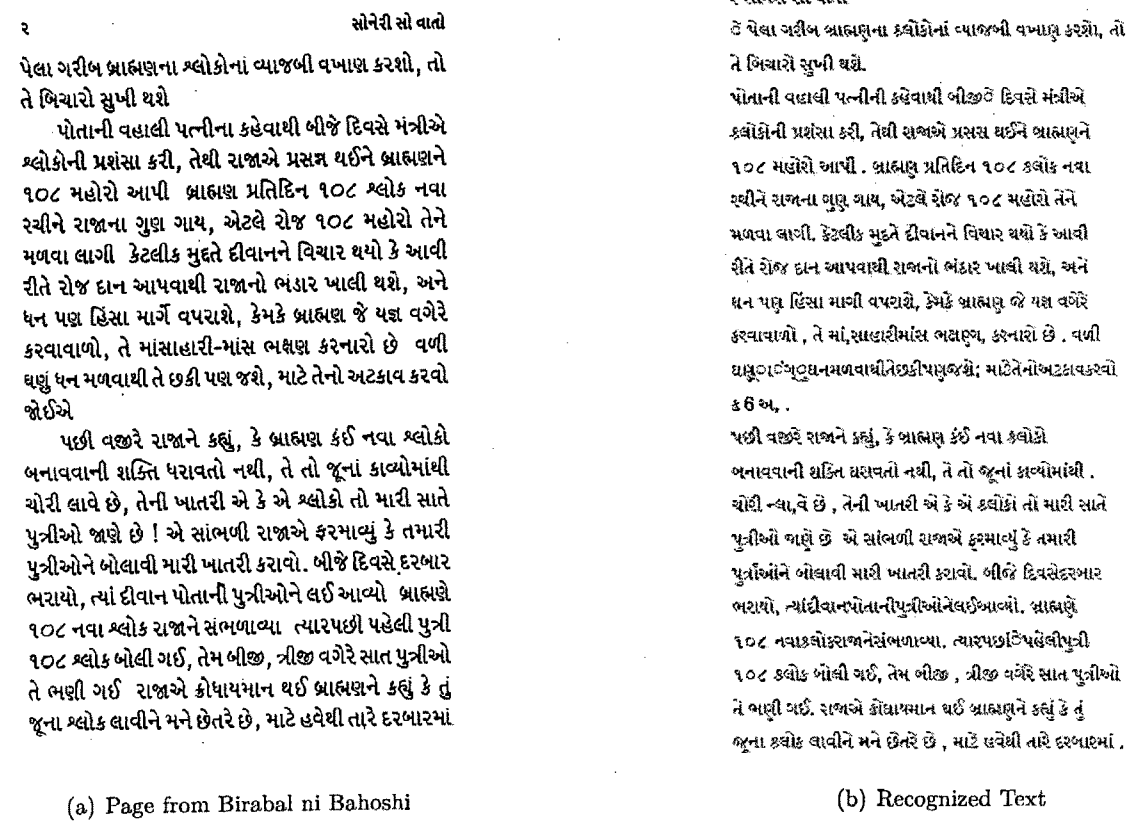
૬૭

(a) Page from Prateeksha

સ હજ
કોણ કહે સંસાર સાગર તરવો અઘરો
તારી સહાયથી તરી ગઇ સહજ સહજ
કોણ કહે માયા તારી પામવી અઘરી
મન સંયમાથી તરી ગઇ સહજ સહજ
કોણ કહે માળી વિણ બાગ ખીલવો અઘરો
જી વન બા ગ ખીલ્‌યો સ હજ સ હજ
કોણ કહે જીવનવન વટાવવું અઘરું
ચરણ ચાલ્યા, કેઠી મળી સહજ સહજ
કોણ કહે અંધકાર વિદારવો અઘરો
ચં દ્રકળાઓ ખીલી સહજ સહજ
કોણ , કહે બીઠેલા દ્વાર ખૂલવા અઘરા
અંતર મનમાં ઉથાઠ થયો સહજ સહજ
કોણ કહે સંસાર તાપ સહેવો અથરો
મેઘ બની મન વરસ્યું સહજ સહજ
કો ણ કહે પ્રભુને પામવો અઘરો
તારી ઝાંખી થઇ ગઇ સહજ સહજ
ઉલીઉઠી

(b) Recognized Text

Figure 8.1: Result of Gujarati OCR



૨ સોનેરી સો વાતો

પેલા ગરીબ બ્રાહ્મણના શ્લોકોનાં વ્યાજબી વખાણ કરશો, તો તે બિચારો સુખી થશે

પોતાની વહાલી પત્નીના કહેવાથી બીજે દિવસે મંત્રીએ શ્લોકોની પ્રશંસા કરી, તેથી રાજાએ પ્રસન્ન થઈને બ્રાહ્મણને ૧૦૮ મહોરો આપી બ્રાહ્મણ પ્રતિદિન ૧૦૮ શ્લોક નવા રચીને રાજાના ગુણ ગાય, એટલે રોજ ૧૦૮ મહોરો તેને મળવા લાગી કેટલીક મુદ્દતે દીવાનને વિચાર થયો કે આવી રીતે રોજ દાન આપવાથી રાજાનો ભંડાર ખાલી થશે, અને ધન પણ હિંસા માર્ગે વપરાશે, કેમકે બ્રાહ્મણ જે યજ્ઞ વગેરે કરવાવાળો, તે માંસાહારી-માંસ ભક્ષણ કરનારો છે વળી ઘણું ધન મળવાથી તે છકી પણ જશે, માટે તેનો અટકાવ કરવો જોઈએ

પછી વજીરે રાજાને કહ્યું, કે બ્રાહ્મણ કંઈ નવા શ્લોકો બનાવવાની શક્તિ ધરાવતો નથી, તે તો જૂનાં કાવ્યોમાંથી ચોરી લાવે છે, તેની ખાતરી એ કે એ શ્લોકો તો મારી સાતે પુત્રીઓ જાણે છે ! એ સાંભળી રાજાએ ફરમાવ્યું કે તમારી પુત્રીઓને બોલાવી મારી ખાતરી કરાવો. બીજે દિવસે દરબાર ભરાયો, ત્યાં દીવાન પોતાની પુત્રીઓને લઈ આવ્યો બ્રાહ્મણે ૧૦૮ નવા શ્લોક રાજાને સંભળાવ્યા ત્યારપછી પહેલી પુત્રી ૧૦૮ શ્લોક બોલી ગઈ, તેમ બીજી, ત્રીજી વગેરે સાત પુત્રીઓ તે ભણી ગઈ રાજાએ ક્રોધાયમાન થઈ બ્રાહ્મણને કહ્યું કે તું જૂના શ્લોક લાવીને મને છેતરે છે, માટે હવેથી તારે દરબારમાં

(a) Page from Birabal ni Bahoshi

૨ સોનેરી સો વાતો

તે પેલા ગરીબ બ્રાહ્મણના શ્લોકોનાં વ્યાજબી વખાણ કરશો, તો તે બિચારો સુખી થશે.

પોતાની વહાલી પત્નીની કહેવાથી બીજીજે દિવસે મંત્રીએ શ્લોકોની પ્રશંસા કરી, તેથી રાજાએ પ્રસન્ન થઈને બ્રાહ્મણને ૧૦૮ મહોરો આપી . બ્રાહ્મણ પ્રતિદિન ૧૦૮ શ્લોક નવા રચીને રાજાના ગુણ ગાય, એટલે રોજ ૧૦૮ મહોરો તેને મળવા લાગી. કેટલીક મુદ્દતે દીવાનને વિચાર થયો કે આવી રીતે રોજ દાન આપવાથી રાજાનો ભંડાર ખાલી થશે, અને ધન પણ હિંસા માર્ગે વપરાશે, કેમકે બ્રાહ્મણ જે યજ્ઞ વગેરે કરવાવાળો , તે માં,સાહારીમાંસ ભક્ષણ, કરનારો છે . વળી ઘણુંધનમળવાથીતેછકીપણજશે; માટેતેનોઅટકાવકરવો ક6અ, .

પછી વજીરે રાજાને કહ્યું, કે બ્રાહ્મણ કંઈ નવા શ્લોકો બનાવવાની શક્તિ ધરાવતો નથી, તે તો જૂનાં કાવ્યોમાંથી . ચોરી લા,વે છે , તેની ખાતરી એ કે એ શ્લોકો તો મારી સાતે પુત્રીઓ જાણે છે એ સાંભળી રાજાએ ફરમાવ્યું કે તમારી પુત્રીઓને બોલાવી મારી ખાતરી કરાવો. બીજે દિવસેદરબાર ભરાયો, ત્યાંદીવાનપોતાનીપુત્રીઓનેલઈઆવ્યો. બ્રાહ્મણે ૧૦૮ નવાશ્લોકરાજાનેસંભળાવ્યા. ત્યારપછીપહેલીપુત્રી ૧૦૮ શ્લોક બોલી ગઈ, તેમ બીજી , ત્રીજી વગેરે સાત પુત્રીઓ તે ભણી ગઈ. રાજાએ ક્રોધાયમાન થઈ બ્રાહ્મણને કહ્યું કે તું જૂના શ્લોક લાવીને મને છેતરે છે , માટે હવેથી તારે દરબારમાં .

(b) Recognized Text

Figure 8.2: Result of Gujarati OCR

8.4 Conclusions and Future Work

Gujarati script is considered to be one of the most difficult scripts from OCR perspective. It poses greater challenges in terms of segmentation in abscess of *shirorekha* and there by increasing the number of classes.

In this research we have constructed the first end to end OCR system for printed documents of Gujarati script. We have presented various subtasks for this and also presented mathematical techniques that can be used to carry out each of these subtasks. The approach selected for this work is a zone separation based approach wherein we are segmenting connected components in each of the three logical zones and recognize them separately. The recognized glyphs are then used to generate text using finite state machine based model. This being a pioneering effort in building complete OCR system for Gujarati script, results of the work-flow suggested here are found to be encouraging for the future developments.

This work is being extended now as a part of the same project funded by the Ministry of Communications and Information Technology. One of the major experiments that can be done in future is to avoid zone separation and check if the classifier can be designed for such a large number of classes.