Elimination of Splitting Errors in Printed Bangla Scripts

Md. Abul Hasnat and Mumit Khan
Center for Research on Bangla Language Processing, Department of Computer Science and Engineering, BRAC University, Dhaka, Bangladesh
mhasnat@gmail.com, mumit@bracu.ac.bd

Abstract

Accurate and robust character segmentation is a significant challenge in Bangla optical character recognition (OCR). The two main errors in segmentation are joining and splitting errors. To solve the problems of joining errors, several algorithms have been proposed in the literature, with varying degrees of accuracy. Few solutions have been proposed to handle the splitting error issue; however, the accuracy of these proposed solutions were not measured. In an actual implementation of the proposed techniques, we observe the presence of over segmented units. In this paper, we present a dissection based splitting error elimination method which solves the problem of over segmentation under a wide range of document images. Our methodology performs its tasks in two stages: we first concentrate on the careful clipping of the matraa (headline) and put our effort in keeping the pixel information of the units intact which are sensitive to splitting errors. In the second stage, we apply several rules based on the feature information of the units in a word. The combined performance of these two stages results in success rate of 99.93% in eliminating the splitting errors.

1. Introduction

In document image analysis systems, robust and accurate character segmentation is a necessary precondition in obtaining sufficient accuracy rate in recognition. From our experience, the rate of character recognition accuracy for the Bangla script goes down significantly in the presence of segmentation errors. For certain scripts, such as the Roman script, the problem of character segmentation has already been identified and solved using a variety of different approaches [1]. For the Bangla script however, the issue of character segmentation is yet to be solved in a satisfactory manner to produce high levels overall accuracy. The zone based approaches of Bangla character segmentation [2-9] cause lots of over- and under-segmentations which are detected in an implementation through observing the output. The idea behind the zone based Bangla character segmentation is to divide the word into three zones – upper, middle and lower zones – and then using the vertical projection profile of the middle zone the gaps between the characters are identified and used to segment the characters. Segmentation technique for the Devanagari script also followed similar approaches [10-13] at the preliminary stage. An example of this technique is shown in Fig. 1. The steps in zone based character segmentation are as follows:

1. Find out the matraa (headline) of the line/word.
2. Separate the upper zone from the middle and lower zone using the matraa.
3. Find out the baseline (separator line which divides the middle and lower zone).
4. Separate the middle and lower zone using the baseline.
5. Take vertical histogram in the middle zone for segmentation of characters.
6. Separate characters using the gaps identified in the histogram information.

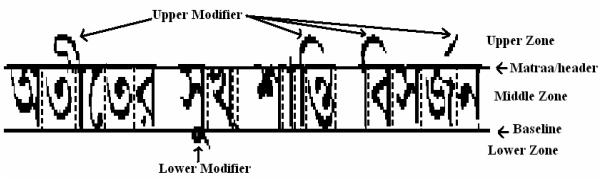

Fig. 1: Zone based approach of Bangla character segmentation. Dotted lines show the character boundary

An actual implementation of this zone based approach generates a number of segmentation errors and hence adversely affects the recognition rate [3-6, 8-14]. Segmentation in Bangla script following the zone based approach suffers from mainly two types of errors [14], joining error (segmenter fails to establish a boundary where there should be one) and splitting error (mistakenly introduces a boundary where there should be none). The joining error is further classified into two types, shadow character problem and touching character problem. An example of the erroneously segmented units is shown in Fig. 2. A related discussion about the joining errors in Bangla and Devanagari character segmentation and possible solutions is found in the literature [3-6, 8, 10-14]. The problem of the splitting errors and probable solution during segmentation is focused at [5, 6, 9 and 14]. However the actual implementation of both the proposed techniques left few over segmented units as well as introduces few under segmented units.

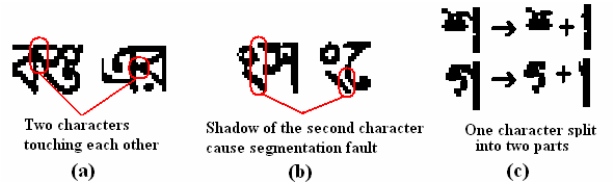

Fig. 2: Example of segmentation errors after applying zone based approach of segmentation. (a) Shows touching errors (b) Shows shadow character problem and (c) Shows splitting errors

In Bangla script the basic characters those suffers from splitting error are গ ণ প শ. The compound characters formed by these units are also split into more than one part in rare cases, however they do not break down as the compound part is attached with the second part of the broken unit where the shadow the first part falls over the second part. The reasons of the separation of these basic characters are caused by over threshold, removal of noise and matraa/headline deletion. These reasons cause the characters to break down into two parts where the breaking occurs at weak joining portion (just before the aa-kar (া) like portion) in the character. We observed in the output that the careful matraa deletion approach using the connected component analysis discussed at [9] does not always successfully handle the splitting issue. It fails often when the second part of the sensitive unit is connected with the header/matraa where the first part remains disconnected and hence it generates completely new units in few cases. This approach also fails to segment the character শ where the upper part of the character has almost same alignment with the matraa. The training based techniques proposed at [9 and 14] fails in the cases when the splitting causes the creation of new units which is not been trained and also the case when the second part attached a portion of the compound unit. Examples of these problems are shown in Fig. 3. To solve the splitting problem during segmentation, reference [6] uses the information that Bangla characters do extend up to the lower zone which is violated by the component above it. So, it must be a part of its nearest character. However the real practice of the solution introduces under segmentation (because of false combination of two characters) where the bottom location of a character fails to reach the baseline. An example of this problem is shown in Fig.4 where we can see that the bottom of the units does not extend up to baseline and hence eligible to combined to the nearest units, which will be under segmentation fault. However the concept of this approach helps us to think about the features of Bangla characters while solving the problem.

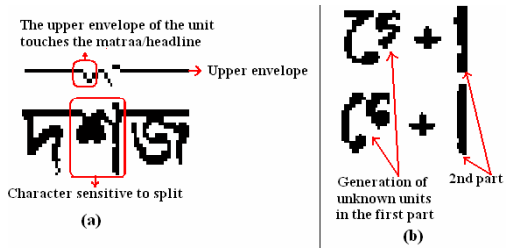

Fig. 3: (a) Unit where matraa clipping approach fails (b) Units where training based approach fail

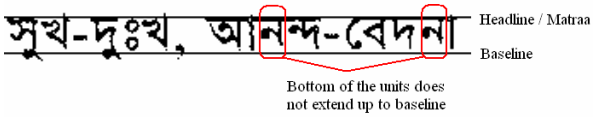

Fig. 4: Units which fails to reach the baseline

In our approach we solved the problem of splitting error in two steps. In the first step we resolved the problem of matraa clipping using connected component analysis on the envelope of the matraa. In the second step we performed rule based feature analysis on the extracted characters/units of each word. The first step is applied at the beginning of the segmentation process and the second step is performed in between the shadow character elimination and touching character segmentation process.

We briefly review the existing literature in section 2, then present our methodology in section 3, and then present our results with discussion, and then finally conclude.

2. Related work

There are several research works on Bangla character segmentation found in the literatures along with the solution of OCR system. Among the literatures there are few works which addressed the issue of character segmentation only. B. B. Chaudhuri and U. Pal have done a great amount of tasks in the area of Bangla OCR as well as segmentation [2-5]. At reference [2] they mentioned the zone based approach of character segmentation. They identified the problem of shadow character and proposed piecewise linear segmentation approach to solve that [3]. They address the problem of touching character errors at [4] and provide a solution of problem using multi factorial analysis considering five fuzzy factors. They performed recognition based segmentation technique to solve the problem of touching error. At reference [5] B. B. Chaudhuri addressed the problem of splitting error and uses information about Bangla character to solve the problem. The problem of splitting error was mentioned at reference [6] where they solved the problem by considering each broken part as a separate character and train them. They solved the shadow character problems they using DFS algorithm. The most recent task in character segmentation is found in [7] where the solution concerns about the reformation of the characters of upper and middle zone. Reference [8] consider the problem of shadow character only and proposed a piece-wise linear segmentation to solve the problem. A different solution of the splitting error is proposed at [9] where they used a special matraa clipping algorithm that do not rub off the matraa of the characters which are sensitive to be split.

The solution of the segmentation problem is found in several literatures for Devanagari characters. Among the literatures V. Bansal et. al. [10] presented a complete solution of the joining errors occurred in Devanagari character segmentation. In their approach they made attempt to segment the conjuncts also. H. Ma et. al. at [11] also addresses the same problems and focused on the joining errors. They considered few units as special units for training which breaks into more than one unit during segmentation. S. Kompalli et. al. at [12 and 13] provides a complete statistics about the occurrence of the different Devanagari characters in the scripts. The error rate mentioned for segmentation of the characters is 5.50% for the core characters. They used the zone based approach to perform the segmentation and they proposed recognition based segmentation to eliminate the errors. They proved that the error rate is reduced in the recognition based segmentation approach. Hasnat et. al. at [14] discussed the problems and provide a training dependent solution to reduce the segmentation error.

3. Methodology

The perquisite of our proposed approach is that the preprocessing steps before the stages of the splitting error elimination are perfect. We assumed the success of several tasks which includes image acquisition and binarization, noise elimination, skew angel detection and correction, text boundary extraction or page layout analysis, line and word segmentation and joining errors elimination. Our proposed technique performs its core operation in two steps as follows:

1. Clip the matraa/headline using histogram and connected component analysis.
2. Rule based feature analysis to identify the splitted units.

3.1. Clip the matraa/headline using histogram and connected component analysis

We apply this task at the beginning of the character segmentation task. To perform this task first we take horizontal run length histogram of the word image. First we identified the location of the headline/matraa which will be maximum valued location (maxValLoc) in the histogram. The matraa is a thick line and can not be represented by a single row of pixels; hence we identified the probable candidate rows as a part of matraa. For this we search certain number (depending on the height of the word) of rows upward and downward location of the preliminary detected row and store their run length value in a 1D matrix (MArr). Then we take the standard deviation (stdVal) of those run length values. Any row (i) in the matrix will be considered as a part of matraa if it satisfies the following equation:

$$MArr(maxValLoc) - MArr(i) <= stdVal*2$$

After the identification of the matraa rows we locate the start and end position of the matraa and separate the region of the matraa from the word image. Next we take the upper envelope of the identified matraa region and take the connected components from the upper envelope. We store the ratio of the component width

vs. word height after the matraa location (ratCcwdWht) which is actually the aspect ratio of the units and check this ratio against a threshold value (thVal = 0.5). This threshold value will ensure that we are keeping the matraa of a single character that is disconnected from the preceeding and following characters and do not affect the segmentation process of the characters using headline deletion process. We also take the bottom location (bottomLoc) of the connected component. We observed that characters which are sensitive to be broken into more than one part have the upper envelope which is connected to the bottom. We keep the pixels below those connected components that satisfy the rules given below.

1. ratCcwdWht < thVal
2. bottomLoc > Height of Matraa /2

An example of this step is illustrated in Fig. 5 where 5(a) shows a word which contains three characters which are sensitive to be broken in the zone based approach. 5(b) shows the conditions after the matraa deletion process using zone based approach where the sensitive characters breaks down into more than one part. 5(c) shows the upper envelope of the matraa region. 5(d) show the condition after matraa rubbed off using the technique discussed above. It also shows that the character শা remains broken into two parts as the upper region of this character has the same alignments with the matraa. The reason of this error is illustrated in Fig. 3(a).

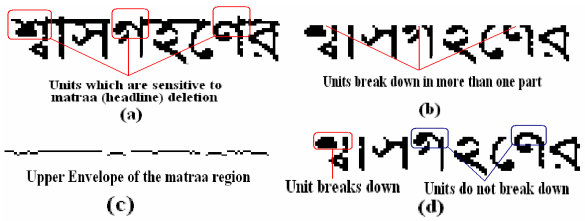

Fig. 5: Example of the matraa/headline deletion process

3.2. Rule based feature analysis to identify the splitted units

This step is necessary to detect merge the characters that break down into more than one part. This approach is applied after performing the shadow error elimination and before touching error elimination. So this is the second last stage of the entire segmentation process. To perform this task we take the necessary information related to the character which is character height, width, aspect ratio (aspRatio) and bottom location (bottomLocUnit). From these information we calculate the following feature measurement:

1. Average bottom location of the units (avgUnitsBottom)
2. Median of the bottom locations of the units (medianUnitsBottom)
3. Standard deviation of the bottom location of the units (stdUnitsBottom)
4. Maximum bottom location of the units (maxUnitsBottom)
5. Z Value of the bottom location of each unit (zValueUnit) is used for the preliminary selection of the first part of the broken unit.
6. Ratio of the present character height vs. next character height (ratPhtNht)

Using these feature measurements we set four rules to ensure solution of the splitting problem as well as ensure that no units get connected that are not splitted. We set four threshold values as follows:

1. Threshold for aspect ratio of the next unit (thUnitAspectRatio): This value is set to 0.4 to ensure that the unit is likely to an aa-kar (া).
2. Threshold for ratio of the height of present and next unit (thUnitsHtHtRatio): This threshold value is set to 0.75 to unsure the eligibility of merging between two units.
3. Threshold for the z value of the unit (thUnitZValues): this value is set as 0. This is required to select the preliminary candidate units to be merged with the next.
4. Threshold value of bottom of the units (thUnitBtLoc): The threshold value is calculated from the feature measurements of the units. This value is necessary for the ultimate selection of the units which can be merged with the next.

For each unit (i) of a line, we calculate the feature measurement and check those against the rules. A unit is marked as a valid unit to be merged with the next if it main the following rules:

Rule 1: zValueUnit(i) < thUnitZValues

Rule 2: bottomLocUnit(i) <= thUnitBtLoc

Rule 3: aspRatio(i+1) of next unit < thUnitAspectRatio

Rule 4: ratPhtNht(i) < thUnitsHtHtRatio

Rule 1 performs the preliminary selection of the units that ensures the rejection of the other units from being consideration. However it also selects non-splitted units also. Rule 2 is applied on the selected units which passed rule 1. This determines whether the candidate unit is eligible to merge with the next or not and it successfully eliminates those erroneously selected units passed from Rule 1. Rule 3 ensure that the second part of the next unit is quite similar with the broken part of the second unit. Finally rule 4 ensures the elimination of few units which has height likely to the height of the first part of the broken units. The above rules are made based on the knowledge of the characteristics of the units which are sensitive to over segmentation.

4. Algorithm

Algorithm: Splitting error elimination by feature analysis

1. Calculate avgUnitsBottom

   Calculate medianUnitsBottom

   Calculate stdUnitsBottom

   Calculate maxUnitsBottom

   Set thUnitZValues to 0

   Set thUnitAspectRatio to 0.4

   Set thUnitsHtHtRatio to 0.75

   IF maxUnitsBottom - medianUnitsBottom > stdUnitsBottom then

      set thUnitBtLoc_1 to avgUnitsBottom - stdUnitsBottom

   ELSE

      set thUnitBtLoc_1 to maxUnitsBottom - (avgUnitsBottom/5 + stdUnitsBottom);

   END

   set thUnitBtLoc_2 to maxUnitsBottom - (avgUnitsBottom/5 + stdUnitsBottom)

   set thUnitBtLoc to minimum(thUnitBtLoc_1, thUnitBtLoc_2)

2. For all the units extracted from the word

   Calculate aspRatio(i + 1)

   Calculate zValueUnit (i)

3. IF rules 1, 2, 3 and 4 (*at section 3.2*) are satisfied then merge this unit with the next unit.

5. Result analysis and discussion

To select the appropriate threshold values at different stage of our algorithm, we run our experiments on four different types of printed text document images. Those are Bangla old Book having congested text lines (BB1), Bangla books having well formatted text (BB3), Bangla official page document (BPD1) and Bangla Typewriting document (BT1). In the four type of document we found total number of middle zone / core units is 5365. We used regression technique to find out the accurate values for different threshold measurement. Here we measured the performance of the technique in each step of the entire process. The threshold values were chosen based on our experiments and observations of the results.

At first we performed character segmentation without our approach and observed that due to splitting error total number of splitted units is 90 and hence the rate of splitting error rate is 1.68%. Next we measured the performance of both steps separately and the error rate is 1.04% after applying careful matraa clipping approach only and the rate is 0.52% after performing rule based feature analysis. Next we applied our proposed approach and observed that only 4 units suffer from splitting error out of 5365 units. So, the error rate is reduced to 0.07%. The error rate at different approaches is presented in Table 1.

Table 1: Result of the splitting error elimination at different stage

Image Name	Total Units	Error rate (Generic Approach)	Error rate (After Matraa deletion)	Error rate (Feature based)	Error rate (Combined)
BB1	2026	1.97	1.09	1.09	0.10
BB3	2095	0.97	0.58	0.10	0.00
BPD1	917	2.4	1.53	0.44	0.22
BT1	357	2.24	2.24	0.00	0.00
Average		1.68	1.04	0.52	0.07

From the erroneous units we observed that the reason of the error is the presence of noise at the bottom of the first part of the splitted units that cause the rule based segmentation to fail. So, it is strongly recommended to filter the image and apply noise

elimination approach even before this stage for the image which is very much sensitive to noise.

6. Conclusion

In this paper we present a complete method to eliminate the splitting errors that occurred during segmentation. We applied the proposed technique in two stages. In the first stage we put our effort in preventing the breaking of the sensitive units and in the second stage we use a selection and merge approach to rebuild the splitted units into one character. Our approach shows significant improvement compared to the other proposed approaches to solve the splitting errors. Since the Bangla script is a derivative of the Brahmi script that is also the mother of many other Indian scripts, the methodology outlined here is applicable to Devanagari as well.

7. Acknowledgements

This work has been supported in part by the PAN Localization Project (www.PANL10n.net) grant from the International Development Research Center, Ottawa, Canada.

8. References

[1] Richard G. Casey and Eric Lecolinet, "A Survey of Methods and Strategies in Character Segmentation", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 18 No. Retrieved: (July,7, 1996).
Available:
http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=506792

[2] U. Pal, B. B. Chaudhuri, "OCR in Bangla: an Indo-Bangladeshi Language", Proc. of ICPR, pp. 269-274, Jerusalem, Israel, 1994. Available:
http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=576917

[3] B. B. Chaudhuri, U. Pal, "An OCR System to Read Two Indian Language Scripts: Bangla and Devnagari(Hindi)", Proc. of 4th ICDAR, Page(s): 1011 -1015 vol.2, Ulm, Germany, 1997.
Available:
http://ieeexplore.ieee.org/Xplore/login.jsp?url=/iel3/4891/13496/00620662.pdf?arnumber=620662

[4] U. Garain and B. B. Chaudhuri , "Segmentation of Touching Characters in Printed Devnagari and Bangla Scripts Using Fuzzy Multifactorial Analysis", IEEE Transactions on Systems, Man and Cybernetics, vol. 32, pp. 449-459, Nov., 2002.
Available:
http://ieeexplore.ieee.org/Xplore/login.jsp?url=/iel5/5326/26422/01176894.pdf?arnumber=1176894

[5] Bidyut B. Chaudhuri, "Digital document processing : major directions and recent advances", Springer, London, 2007.

[6] S.M. M Mahmud, N. Shahrier, A.S.M D. Hossain, M. T. M. Chowdhury, M.A. Sattar,"An Efficient Segmentation Scheme for the Recognition of Printed Bangla characters", Proceedings of ICCIT, pp 283-286, 2003.
Available:
http://research.banglacomputing.net/iccit/ICCIT_pdf/7th%20ICCIT-2004_123.pdf

[7] Md. Abdus Sattar, Khaled Mahmud, Humayun Arafat and A F M Noor Uz Zaman, "Segmenting Bangla Text for Optical Recognition", ", Proceedings of ICCIT, 2007.
Available:
http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=4579373

[8] Jalal Uddin Mahmud, Mohammed Feroz Raihan and Chowdhury Mofizur Rahman, "A Complete OCR System for Continuous Bangla Characters", IEEE TENCON-2003: Proceedings of the Conference on Convergent Technologies for the Asia Pacific, 2003.
Available:
http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=1273141

[9] A. A. Chowdhury, E. Ahmed, S. Ahmed, S. Hossain and C. M. Rahman, "Optical Character Recognition of Bangla Characters using neural network: A better approach". 2nd ICEE, 2002.

[10] V. Bansal and R. M. K. Sinha, "Segmentation of touching and fused Devanagari characters", Pattern Recognition, Volume 35, Number 4, pp. 875-893(19), April, 2002.
Available: http://www.iitk.ac.in/ime/veena/PAPERS/s.pdf

[11] H. Ma and D. Doermann, "Adaptive Hindi OCR Using Generalized Hausdorff Image Comparison", ACM Transactions on Asian Language Information Processing, Vol. 26, No. 2, pp198-213, 2003.
Available: http://portal.acm.org/citation.cfm?id=979875

[12] S. Kompalli, S. Setlur and V. Govindaraju, "Design and Comparison of Segmentation Driven and Recognition Driven Devanagari OCR", Proc. of the 2nd DIAL, pp. 96 – 120, 2006.
Available:
http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=506792

[13] S. Kompalli, S. Nayak, S. Setlur, V. Govindaraju, "Challenges in OCR of Devanagari Documents", Proc. of ICDAR, pp 327-333, 2005.
Available:
http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1575563

[14] Md. Abul Hasnat, S M Murtoza Habib and Mumit Khan. "A High Performance Domain Specific OCR for Bangla Script", Int. Joint Conf. on Computer, Information, and Systems Sciences, and Engineering (CISSE), 2007.
Available:
http://www.springerlink.com/content/l75400676825p373/