Improving Nastalique-Specific Pre-Recognition Process for Urdu OCR

Sobia Tariq Javed and Sarmad Hussain
Center for Research in Urdu Language Processing
National University of Computer and Emerging Sciences, Pakistan
sobia.tariq@nu.edu.pk, sarmad.hussain@nu.edu.pk

Abstract

Urdu language is written using Arabic script in Nastalique writing style. Nastalique script is highly cursive, context sensitive and is hard to process as only the last character in its ligature sits on the baseline. In addition, it exhibits character and ligature level spatial overlap. Due to these factors, the placement of dots and other diacritics is also highly contextual and variable. There is now increasing amount of work to process and recognize Nastalique script to develop Urdu OCR. This paper proposes improvements to these methods. The paper focuses on Nastalique specific pre-processing methods which can be employed before the text recognition process. The recognition and post recognition processes will be addressed separately.

Introduction

Optical Character Recognition refers to the process of converting the text image, such as scanned document, electronic fax file, pictures of documents taken from cameras, into a text file. The text in the image is non-editable. Therefore, aim of the OCR system is to imitate the human ability to extract text from such images. Once the text is extracted, it is editable making it useful for further processing including searching and information retrieval. OCR systems have many applications, including reading filled forms, postal addresses, retrieving and archiving data etc.

Broadly, OCR process can be divided into three main processes, Preprocessing, Recognition, and Post Processing, as shown in Figure 1.

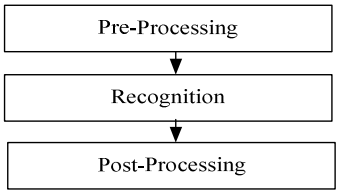


Fig. 1. OCR process

Briefly, after the image is acquired, it first goes through noise and distortion removal process. Once the cleaned image is available, it goes through script specific pre-processing, in which the image is broken into its constituents. These constituents may be letters, combination of letters or parts of letters. These segments from the pre-processed image are then recognized using different classifiers and then the text is recreated from these images. Results of recognition are improved by using linguistic processing in the post-processing phase.

There has been much work in this area for a variety of scripts. There has also been some initial work on Urdu in this context [14, 15, 16, 17, 18, 20, and 21]. The current work builds and improves on the existing script-specific pre-processing work done for Urdu.

Urdu Writing System

Urdu is the national language of Pakistan. Urdu is written using Arabic script in Natalique writing style. Urdu words are written from right to left and numbers are written left to right so it is bidirectional as shown below [3].

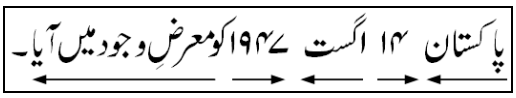


Fig. 2. Bidirectional Urdu script

Urdu uses extended Arabic character set, given below [11, 12]. These letters join together to form words of the Urdu language. By nature Nastalique style is very context sensitive, that is, characters change their shapes depending upon the characters preceding and succeeding it. For proper pronunciation of constituent word diacritics are used [13]. The diacritics may appear above or below a character. These characters and diacritics are given in Figure 3.

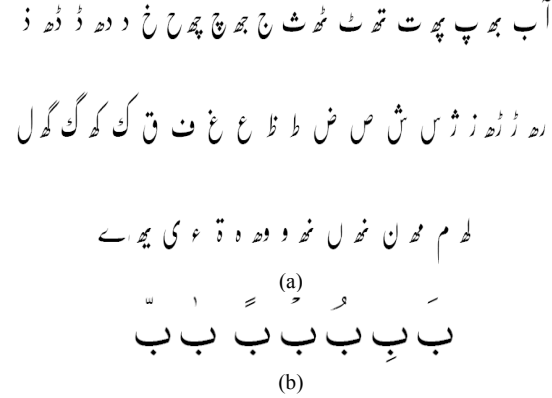


Fig. 3. Urdu (a) character set and (b) diacritical marks

Nastalique Script

Nastalique is a combination of two different fonts, Naskh and Taleeq, created by Mir Ali Tabrezi. It is a complex script as it based not only on the pre-defined rules but also on the aesthetic sense of the calligrapher. It is highly cursive and context sensitive in nature. Some characteristics of Nastalique are as follows [2]:

1. It is written diagonally from top right to bottom left. This means that all the ligatures are tilted at an angle. The angle is variable, depending on the letters being written. The diagonal nature was invented to conserve writing space, but in turn makes the writing system much more complex compared with other styles like Naskh. The example in Figure 4 shows how Nastalique is written diagonally and consumes less horizontal space as compared to Naskh.

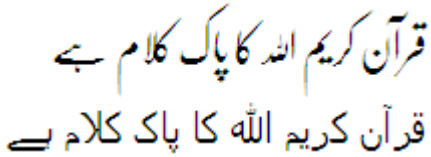


Fig. 4. Nastalique style (top) takes less horizontal space than Naskh style (bottom) for the same text

2. Due to diagonality only the last letter sits on the baseline. This results in complex mark placement mechanism [3]. As the marks are squeezed in horizontal space, they have to be moved vertically to avoid collisions. This can be seen from the example in Figure 5, where for Naskh font the *Nuqtas* (or dots) for letter *Pay* remain unchanged, but in Nastalique script the *Nuqtas* for *Pay* have to moved from to allow space for *Nuqtas* of *Chay*.

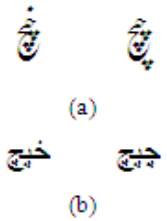


Fig. 5. Two different placements of dots of Chay

3. Overlapping problem is present in characters and ligatures (portion of connected letters). The ligature overlapping is needed to avoid unnecessary white space. For example, in Figure 6, *Kaf* of the word کر is overlapping *Tay* of word بات.

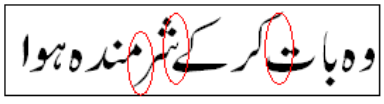


Fig. 6. Overlap between the ligatures

4. Nastalique letters take up different shapes depending on the context in which it is written. Figure 7 shows different shapes of letter *Bay* [3].

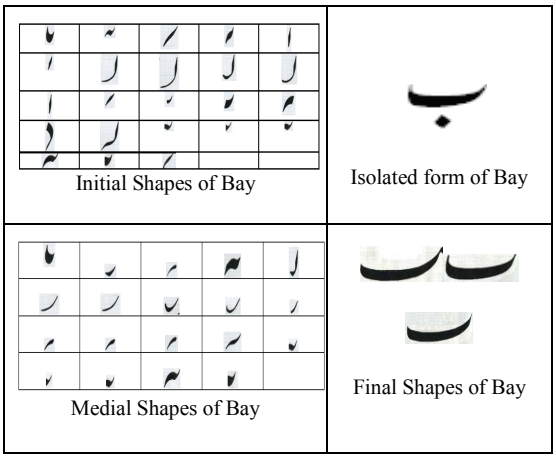


Fig. 7. Different form of letter Bay depending on the context.

Methodology

There has been considerable work on Arabic OCR. However, all that work is based on Naskh style. As indicated above, Nastalique style for Urdu presents much more challenges and thus a very different OCR challenge. In summary, the challenges include much more cursiveness, diagonality, mark placement and significantly more contextual shaping. This entails that though the work on Arabic language is relevant, these algorithms need to be further evolved for Urdu. The current paper looks into these challenges. The work presented is part of larger project to develop Urdu OCR. However, the current we only present the pre-processing stage, as highlighted in Figure 8 and explained below.

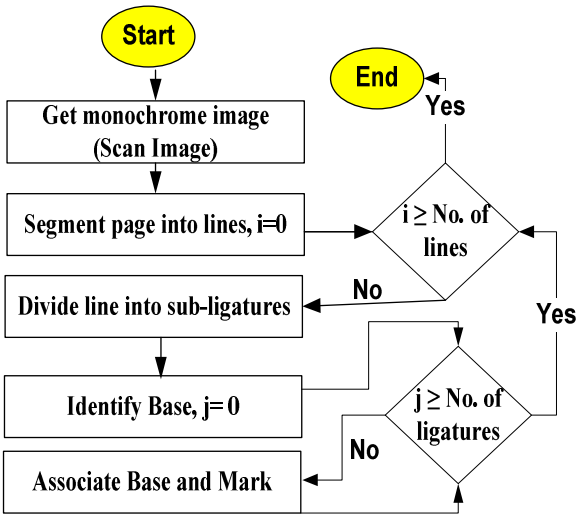


Fig. 8. Flow Chart of Methodology

A printed Urdu page is scanned with 150 dpi to get a monochrome image. After the image is acquired, the following processes are conducted to segment the page into textual components.

1. Page Segmentation into Lines

After binarization, the image is passed on to the module which automatically detects and separates individual text lines from the image using horizontal and vertical projection of pixels [1]. A projection profile is a histogram giving the accumulated sum of black pixels along each row. The trough between two consecutive peaks in the horizontal profile marks the boundary between two text lines. This is shown in Figure 9.

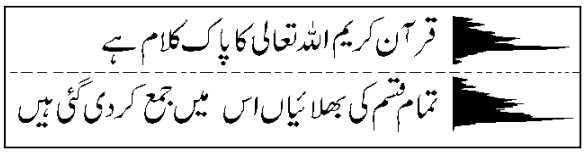
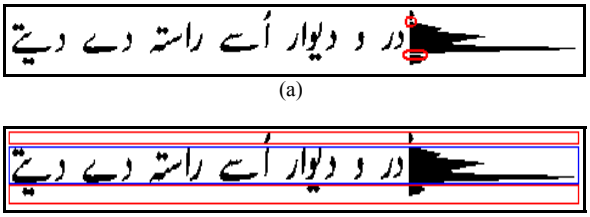


Fig. 9. Histogram of horizontal projection to separate lines in the text

However, this method is not robust. Sometimes ligatures in a line are arranged such that there is a minimum in the histogram between the main bodies and the dots (and other diacritical marks) above and/or below the main bodies. As a result, the single text line is mis-segmented as two or three separate lines as shown in Figure 10.



(a)

(b)

Fig. 10. (a) Histogram giving zero values between main bodies and diacritics, and (b) Text line mis-segmented into three lines due to zero values

In order to overcome such problems a threshold value is determined based on the height of the lines. If the trough between the two consecutive peaks is less than threshold value, then lines are not separated. In such cases, the line with diacritic(s) is associated with the upper or lower line based on the vertical distance from adjacent lines. The diacritical portion is associated with the line which is closer based on the vertical distance. After the separation, the boundaries of the lines are defined using vertical histogram. The resulting boundary is shown in Figure 11.

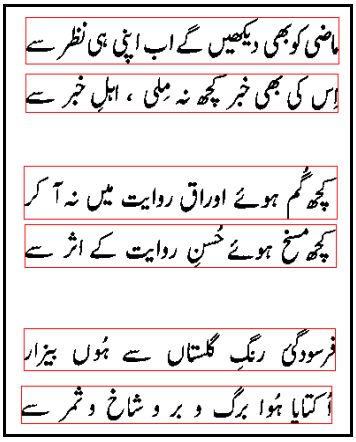


Fig. 11. Boundary marked for lines

Similarly, after horizontal separation, vertical projection of each line is used to mark the right and left ends of the line. This is also indicated in Figure 11.

2. Line segmentation into ligature base and diacritics

Once lines are separated, they are further segmented into ligatures. This process requires identification of base forms and identification of dots and other marks, and then also associating the marks with the appropriate base forms.

A. Line division into sub-ligatures

Many of the existing systems use projection profile method, which computes the vertical histogram of text line and segments where the histogram has zero values (e.g. [4,5,6,7]). But this method cannot be applied to Nastalique, where the ligatures would overlap both in horizontal and vertical projections. A more sophisticated method is to find the connected components, which checks eight neighbors of each black pixel and adds all neighboring pixels to a component, repeating the process for all additional pixels incorporated through the process until all the pixels have been checked (following the method proposed by Elms 1994 [8]). The output of this method is given in Figure 12, which shows that all ligature-components are separated (and shaded differently).

Fig. 12. Output of connected component method

B. Base Identification

Although this method finds the sub-ligatures, including dots, marks and base forms, it cannot separate the base ligature from dots and marks. Nastalique is written such that the last character of each ligature rests on a horizontal line called

baseline. The diacritics do not normally rest on or cross the baseline. This feature of Nastalique can be used to distinguish ligature base from its diacritics. Baseline calculation is based on horizontal projection of pixels. Row that contains the maximum number of pixels is a candidate for the baseline[1]. This is shown in Figure 13 (also see [9]).

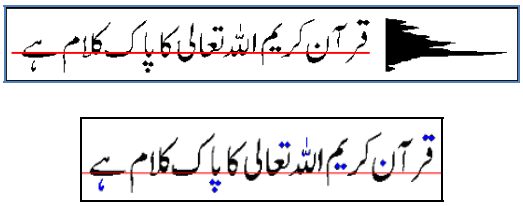


Fig. 13. Base and diacritic separation by using horizontal projection histogram

However, sometimes false baseline may be set towards the top of line, as shown in Figure 14 (a). Such errors can be avoided by checking a couple of heuristics. First, as per the writing rules of Nastalique, every ligature should touch baseline. Second, the baseline must be in the lower half of the line, and any maxima in the top half of the histogram should be ignored. Using these heuristics, the false identification is reduced, as is given in Figure 14 (b).

(a) (b)

Fig. 14. (a) False baseline, and (b) Corrected baseline with the use of heuristics

Issues are still found as the fonts may not follow the Nastalique style specifications accurately (to the last few pixels), and so having a precise single pixel baseline can be ineffective. For example, analysis showed that some ligatures like letter *Alif* did not touch the baseline as shown in Figure 15 (a), even though this letter should touch the baseline as per the writing rules. This issue is resolved by using a more practical band of pixels to mark the baseline shown in Figure 15 (b), which then touches all the base forms. The thickness of the band is normalized with respect to the line height.

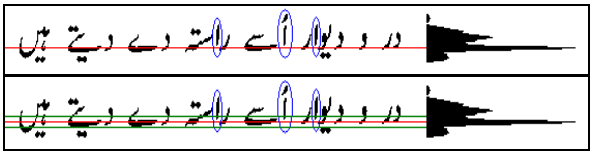


Fig. 15. (a) Baseline using horizontal projection histogram, and (b) Adjustment of baseline into a band to touch all base forms

[1] This computational baseline may differ from the calligraphic baseline.

Another issue which arises due to the diagonal nature of the writing style is that some diacritics lie in the range of baseline and are thus misclassified as main body, as shown in Figure 16.

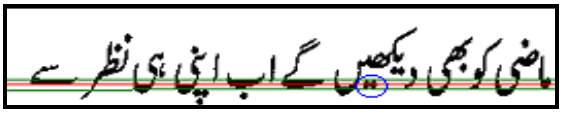


Fig. 16. Encircled are the dots which are crossing the baseline

To overcome such cases, a threshold value is computed for the size of the main body versus the diacritics, as the latter are much smaller in size. All the connected bodies crossing baseline but having the size smaller than the threshold are still considered as diacritics.

C. Base and Mark Association

The techniques discussed above identify the components and mark them as base or a diacritic. The dots and marks also need to be associated with the relevant base forms. This has been attempted by Husain et al. [10], who propose calculating centroid of each shape and then use centroid-to-centroid distance to associate base forms with diacritics. Initially a variation of this method is used in the current work. Centroid is calculated for each mark and dot, and is projected vertically to form such associations. Sample results are shown in Figure 17.

Fig. 17. Vertical Projections of the centroids of diacritics to the bases

Though this method works reasonably well, there are cases where this method does not give accurate results. Sometimes the centroid of the diacritics does not project onto the right base, as the dots of a letter may be shifted left or right due to context. This is shown in Figure 18, in which the first set of dots of letter *Tay* (read from right to left) are shifted right because of the next letter *Kaf*, and are projecting on the previous ligature.

Fig. 18. Centroid of diacritic of *Tay* projecting on previous ligature

To address such issues, an alternative process is concurrently applied in which the complete horizontal span is taken for each diacritic and is projected to the base form. In case of complete overlap, the decision is straightforward. However, in

case the diacritic overlaps more than one base forms, it is associated with the one on left side[2]. Position of left-most pixel of diacritic with respect to the main body of ligature is checked. If a diacritic lies within the boundary of certain main body, the diacritic is associated with it. Where there is a complete overlap with multiple base form, the diacritic is associated with the one with which it has the lesser distance. Also, if the diacritic does not lie within the boundary of any main body then centroid to centroid distance method is used. Some of these scenarios are shown in Figure 19.

(a) (b) (c)

Fig. 19. Overlap of diacritics with base forms: (a) overlap within single base form, and (b) overlap across multiple base forms, and (c) overlap across a single base form

3. Results

The processes discussed above were implemented and tested. The process to segment page into lines discussed in Section 2 was tested by running the enhanced algorithm on 20 pages (10 lines each) scanned from three different books on Urdu poetry, published using Noori Nastalique font. The pages are scanned at 150 dpi and the font size varied from 34-38 point-size. All 200 hundred lines are accurately segmented giving 100% accuracy.

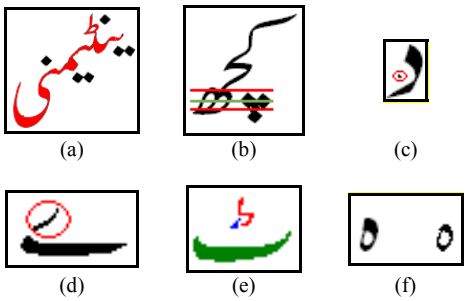
The same 200 lines used for verifying whether the algorithm in Section B correctly marks the baseline. All samples are correctly marked with the baseline, again giving 100% accuracy for this process.

The method for identification for the base ligature and marks is also tested. A total of 1282 unique ligatures are extracted from the 5000 high frequency words in a corpus-based dictionary[3] [19]. It is also confirmed that all Urdu letters are used in these ligatures in a variety of contexts. For analysis purpose three or more samples of each ligature are generated to form the text. These pages are printed in Noori Nastalique font at font size 36. The pages are then scanned at dpi 150 and then separated back into ligatures. A total of 3655 ligatures are tested and 3436 ligatures are accurately separated with proper association of marks, giving an accuracy of 94%.

4. Discussion

Testing and further analysis shows the following types of errors in the OCR pre-processing, which need further improvement.

1. Due to noise, or placement, the diacritic connects with the base, and is thus not separable. The example in Figure 20 (a) shows that the combining mark *Small Toay* is colored the same as base as it connects with it.
2. Due to noise (e.g. spreading of ink on lower quality paper), the three dots join, and also are in the region of the base line. In this case they have enough area to overcome the threshold and qualify as a base character. Such an example is shown in Figure 20 (b).
3. Sometimes extra noise is introduced which is miss-classified as a dot or a mark. This is shown in Figure 20 (c).
4. Poor printing quality can also cause main bodies to break at fine junctions and introduce false diacritics, as shown for *Bari Yay* in Figure 20 (d), where the connected body has been disconnected. The upper portion is identified as diacritic and lower part as main body. Such issues also occur within diacritics. Figure 20 (d) shows that the diacritic is broken into two portions due to printing error and is identified as two different diacritics.
5. Finally, again due to printing quality, the smaller isolated letter can some times get confused with diacritics (based on the thresholding value) and be mis-classified as diacritics instead of base forms. This original size and badly printed instance is shown in Figure 20 (f) for comparison.



(a) (b) (c)

(d) (e) (f)

Fig. 20. : Different error scenarios in Nastalique OCR

5. Conclusion

The Nastalique style of Arabic script used to write Urdu language is complex due to its diagonal, context sensitive and cursive nature. The mark placement and movement rules, in addition to the other factors, make the pre-processing of Nastalique for OCR very challenging. The current paper addresses various steps involved in the pre-processing phase, using and improving existing methods for Nastalique. The results show that though much of the pre-processing can be successfully undertaken, there are some printing related issues, which need to be further addressed in the future. This work is part of a larger project which looks into development of a complete OCR system for Urdu.

[2] This follows from the property of Nastalique which only extends characters rightward (backward) and not leftward (forward).

[3] This is done to ensure that valid Urdu ligatures are used.

Reference

[1]. Pal, U. and Sarkar, A. “Recognition of Printed Urdu Text,” in the Proceedings of the Seventh International Conference on Document Analysis and Recognition (ICDAR), 2003.

[2]. Hussain, S. “www.LICT4D.asia/Fonts/Nafees_Nastalique,” in the Proceedings of 12th AMIC Annual Conference on E-Worlds: Governments, Business and Civil Society, Asian Media Information Center, Singapore, 2003.

[3]. Wali, A. and Hussain, S. “Context Sensitive Shape-Substitution in Nastaliq Writing system: Analysis and Formulation,” in the Proceedings of International Joint Conferences on Computer, Information, and Systems Sciences, and Engineering (CISSE), 2006.

[4]. Pechwitz, M. and Maergner, V. “HMM based approach for handwritten Arabic word recognition using the IFN/ENITdatabase,” in the Proceeding of Seventh International Conference on Document Analysis and Recognition (ICDAR), 2003.

[5]. Lu, Z., Bazzi, I., Kornai, A. and Makhoul, J. “A Robust, Language-Independent OCR System,” in the 27th AIPR Workshop: Advances in Computer Assisted Recognition, SPIE, 1999.

[6]. Bojovic, M. and Savic, M. D. "Training of Hidden Markov Models for Cursive Handwritten Word Recognition," in the Proceedings of the15th International Conference on Pattern Recognition (ICPR) vol.1, 2000.

[7]. El-Hajj, r., Likforman-Sulem, L. and Mokbel, C. "Arabic Handwriting Recognition Using Baseline Dependant Features and Hidden Markov Modeling," in the 8th International Conference on Document Analysis and Recognition (ICDAR), South Korea, 2005.

[8]. Elms, A.J., "A Connected Character Recognizer Using Level Building of HMMs," in the Proceedings of 12th International Conference on Pattern Recognition, 1994.

[9]. Shah, Z. and Saleem, F. “Ligature Based Optical Character Recognition of Urdu, Nastaliq Font,” in the Proceedings of International Multi Topic Conference, Karachi, Pakistan, 2002.

[10]. Husain, S.A. and Amin, S.H. “A Multi-tier Holistic approach for Urdu Nastaliq Recognition,” in the Proceedings of International Multi Topic Conference, Karachi, Pakistan, 2002.

[11]. Hussain, S. and Durrani, N. “Urdu,” in A Study on Collation of Languages from Developing Asia, Center for Research in Urdu Language Processing, NUCES, Pakistan, 2007.

[12]. Hussain, S. and Afzal, M. “Urdu Computing Standards: UZT 1.01”, in the Proceedings of the IEEE International Multi-Topic Conference, Lahore, Pakistan, 2001.

[13]. Hussain, S. “Letter to Sound Rules for Urdu Text to Speech System”, In the Proceedings of Workshop on Computational Approaches to Arabic Script-based Languages, COLING 2004, Geneva, Switzerland, 2004.

[14]. Ahmad, Z., Orakzai, J. K. , Shamsher, I. and Adnan, A. “Urdu Nastaleeq Optical Character Recognition,” in the Proceedings of World Academy of Science, Engineering and Technology 26, 2007.

[15]. Shafait, F., Hasan, A., Keysers, D. and Breuel, T. “Layout analysis of Urdu document images” in Proceedings of IEEE Multitopic Conference (INMIC 06), 2006.

[16]. Safabakhsh, R. and Abidi, P. "Nastaaligh Handwritten Word Recognition Using a Continuous-Density Variable-Duration HMM", The Arabian Journal for Science and Engineering, April 2005.

[17]. Shamsher, I., Ahmad, Z., Orakzai, J. K. and Adnan, A. “OCR for Printed Urdu Script Using Feed Forward Neural Network”, in the Proceedings of World Academy of Science, Engineering and Technology 23, 2007.

[18]. Malik, S.; Khan, S.A., “Urdu online handwriting recognition”, Emerging Technologies, 2005. Proceedings of the IEEE Symposium on Volume, Issue, 17-18 Sept. 2005

[19]. Ijaz, M., Hussain, S. “Corpus Based Urdu Lexicon Development”, In the Proceedings of Conference on Language Technology (CLT07), University of Peshawar, Pakistan, 2007.

[20]. Razzak,M., Hussain,A., Sher,M., and Khan,Z. "Combining Offline and Online Preprocessing for Online Urdu Character Recognition",Proceedings of the International MultiConference of Engineers and Computer Scientists 2009 Vol I, IMECS 2009, March 18 - 20, 2009.

[21]. Hussain, A., Anwar, F., and Sajjad, A. “Online Urdu Character Recognition System.” MVA2007 IAPR Conference on Machine Vision Applications, 2007.