EURASIP Journal on Image and Video Processing

RESEARCH

Open Access

Segmentation-free optical character recognition for printed Urdu text

Israr Ud Din[1], Imran Siddiqi[1*], Shehzad Khalid[1] and Tahir Azam[2]

Abstract

This paper presents a segmentation-free optical character recognition system for printed Urdu Nastaliq font using ligatures as units of recognition. The proposed technique relies on statistical features and employs Hidden Markov Models for classification. A total of 1525 unique high-frequency Urdu ligatures from the standard Urdu Printed Text Images (UPTI) database are considered in our study. Ligatures extracted from text lines are first split into primary (main body) and secondary (dots and diacritics) ligatures and multiple instances of the same ligature are grouped into clusters using a sequential clustering algorithm. Hidden Markov Models are trained separately for each ligature using the examples in the respective cluster by sliding right-to-left the overlapped windows and extracting a set of statistical features. Given the query text, the primary and secondary ligatures are separately recognized and later associated together using a set of heuristics to recognize the complete ligature. The system evaluated on the standard UPTI Urdu database reported a ligature recognition rate of 92% on more than 6000 query ligatures.

Keywords: Optical character recognition, Printed Urdu text, Ligature, Hidden Markov models, Clustering

1 Introduction

With the tremendous advancements in computation and communication technologies, the amount of information available in the digital form has increased manifolds over the recent years. Consequently, an increased tendency to digitize the existing paper documents in the form of books, magazines, newspapers, and notes has also been observed over the last decade. With this, the need to have efficient Optical Character Recognizers (OCRs) to convert the digitized images into text has increased. OCR is one of the most researched pattern classification problems. Today, commercially mature OCRs are available realizing high recognition rates on a number of scripts, those based on Latin and Chinese alphabets for instance [1, 2]. Despite these developments, OCRs for many languages are yet either to be developed or are in very early stages, and cursive Urdu being one of such example is investigated in our study.

The alphabet of Urdu is a super set of Arabic, borrows some characters from Pashto, and comprises a total of 39 characters. Unlike Pashto and Arabic which are mostly scripted in the Naskh style, Urdu generally employs the Nastaliq script which runs diagonally from right to left. The major challenges offered by Urdu document images include non-uniform inter- and intra-word spacing, overlapping of neighboring and partial words, filled or false loops, and no fixed baseline [2–4]. OCR finds applications in a wide range of problem areas including printed (religious, poetry and literature books, newspapers, passports, utility bills), handwritten (office records, historical manuscripts, input to mobile devices) and mixed (traffic challans, bank checks, driving licenses, hand-filled forms) documents. Other applications include text-guided autonomous vehicle navigation and text-to-speech, text-to-text, and speech-to-text recognition for people having sight, speaking, and hearing disabilities, respectively [2].

This paper presents a statistical features-based holistic (segmentation-free) OCR for Urdu Nastaliq font. Ligatures, which ideally correspond to connected components (Fig. 1), are employed as recognition units in the proposed technique. A semi-automatic algorithm is selected for clustering multiple occurrences of high-frequency ligatures (HFLs) extracted from the well-known Urdu

*Correspondence: imran.siddiqi@gmail.com
[1] Bahria University, Islamabad, Pakistan
Full list of author information is available at the end of the article

© The Author(s). 2017 **Open Access** This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (http://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

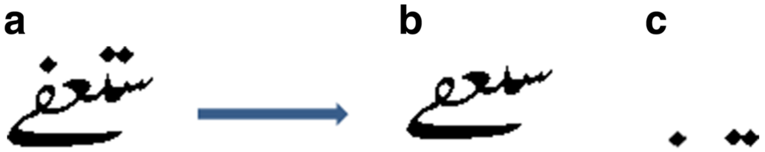


Fig. 1 **a** Complete ligature. **b** Primary/main ligature. **c** Secondary ligatures (dot/diacritic)

Printed Text Images (UPTI) dataset [5]. Hidden Markov Models (HMMs) are employed for recognition, a separate HMM is trained for each ligature cluster. The recognition of primary ligatures and dots/diacritics is carried out separately which are associated later to form the complete/true ligature. As opposed to conventional methods that either carry out recognition of detached characters [6–9] or work with single font size [5, 10], the developed methodology recognizes ligatures irrespective of the font size. The training data prepared through the sequential clustering algorithm is expendable making the platform scalable for incorporation of more ligatures.

This paper is organized as follows. Section 2 reviews some promising techniques proposed for recognition of text in Urdu and other languages based on similar scripts. Section 3 presents the proposed methodology with in-depth discussion on the training and recognition modules. In Section 4, experiments carried out to validate the proposed technique are described while conclusions are drawn in the last section.

2 Literature review

Character recognition is one of the most investigated pattern classification problems. Recognition systems for Urdu and text in similar languages, however, are yet to mature as opposed to other scripts. Most of the work carried on Urdu either deals with individual characters [6–9, 11, 12] or employs separate recognition of primary and secondary ligatures [5, 10, 13]. Traditionally, recognition techniques can be categorized into analytical and holistic approaches as discussed in the following.

2.1 Analytical approaches

Analytical (segmentation-based) methods rely on segmentation of ligatures into characters either explicitly [14, 15] or implicitly [16–18]. Among these methods, Javed and Hussain [19] propose a recognition system for Urdu Noori Nastaliq font extracting discrete cosine transform (DCT) features from skeletonized and already segmented characters. Classification is carried using HMMs and less than 20 classes are considered in their study. The system evaluated on 1692 Urdu ligatures achieved

Table 1 A summary of notable contributions on recognition of Urdu text

Study	Database	Classifier	Results	Recognition unit	Approach
Pal and Sarkar [6]	Custom	–	97.8%	Isolated characters	Analytical
Shamsher et al. [7]	Custom	Neural networks	98.3%	Isolated characters	Analytical
Tariq et al. [8]	Custom	Neural networks	97.43%	Isolated characters	Analytical
Sardar and Wahab [9]	Custom	–	97.12%	Isolated characters	Analytical
Nawaz et al. [12]	Custom	–	89%	Isolated characters	Analytical
Ahmed et al. [11]	Custom	Neural networks	93.4%	Segmented characters	Analytical
Hussain et al. [21]	CLE Urdu	HMM	87.76%	250 graphemes	Analytical
Hassan et al. [16]	UPTI	BLSTM	86.4%/95.8%	Characters	Analytical
Ahmed et al. [22]	UPTI	BLSTM	89%	Characters	Analytical
Naz et al. [18]	UPTI	MDLSTM	96.40%	Characters	Analytical
Hussain et al. [32]	Custom	–	95%	Spotting ligatures	Holistic
Sabbour and Shafait [5]	UPTI	KNN	91%	10,000 primary ligatures	Holistic
Javed and Hussain [10]	CLE Urdu	HMM	92%	1282 unique primary ligatures	Holistic
Akram et al. [13]	CLE Urdu	Modified tesseract	97.87%	1475 unique primary ligatures	Holistic
Akram et al. [28]	CLE Urdu	Modified tesseract	86.15%	Unique ligatures	Holistic
Javed et al. [19]	CLE Urdu	HMM	92.73%	1692 Unique ligatures	Holistic
Khattak et al. [29]	CLE Urdu	HMM	97.93%	2028 Unique ligatures	Holistic

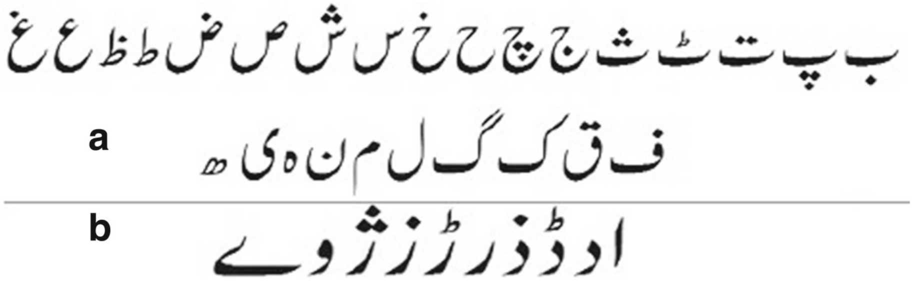


Fig. 2 Urdu character set with **a** joiners and **b** non-joiner separated

a recognition rate of 92.7%. Malik and Faheim [15] presented a line and character segmentation scheme that mainly relies on projection profiles. The under- and over-segmentations are handled through a set of heuristics. Segmentation accuracy of 99% is reported on a custom dataset of Urdu Batool font. In a similar work, Uddin et al. [20] presented a novel technique for segmentation of overlapped and joined text lines with subsequent complete ligature extraction from printed Urdu document images. The system evaluated on 30 document images reports line and ligature segmentation accuracies of 98.79 and 92.49% respectively. Hussain et al. [21] proposed a segmentation-based OCR for printed Urdu Nastaliq ligatures. The original grapheme shape classes are increased from 47 to 250 in order to achieve better recognition. A window is slided over the contour of a segmented grapheme for extraction of DCT low-frequency coefficients. The coefficients serve to train separate HMMs for 250 grapheme classes each having 30 instances. A query ligature is separated into main and secondary bodies and then segmented into graphemes for individual recognition. Once recognized individually, graphemes are joined to form the main body ligatures that are associated with the secondary bodies using a lookup table. System evaluation on 93,018 words (from the Center of Language Engineering (CLE) database) in font size 14 realize 87.76% ligature recognition rate.

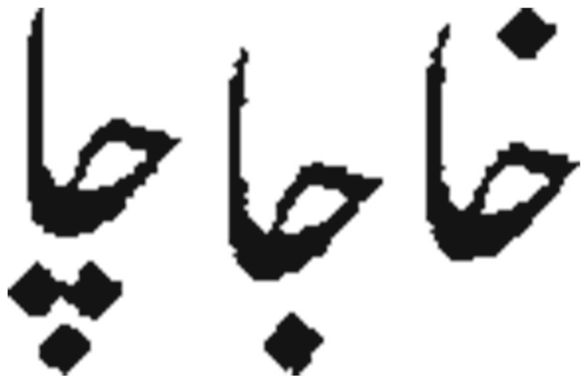


Fig. 3 Complete ligatures with the same primary and different secondary ligatures

A number of recent studies [16, 22] have employed deep learning-based implicit segmentation techniques for recognition of cursive text leaving it to the classifier to implicitly find segmentation cue points of characters. Ahmed et al. [22] applied Bidirectional Long Short-Term Memory (BLSTM) classifier to the character recognition of Urdu. Using raw pixels, the proposed approach realized 89% accuracy on the UPTI dataset. In a similar study, Naz et al. [17] employ the more advanced multi-dimensional LSTM (MDLSTM) with Connectionist Temporal Classification (CTC) layer. System trained using set of statistical features extracted from normalized gray scale images report character recognition rate of 96.4% for the clean text part of the UPTI dataset.

Segmentation-based approaches have the advantage of reduced number of training classes that is same as the number of characters (having different context-based letter shapes) in the alphabet. However, segmenting (Urdu and alike) cursive scripts into characters is a challenging task in itself. Recently, implicit segmentation using deep learning has been successfully investigated for recognition of Urdu text [23–26]. These techniques, however, require large training data and employ characters as units of recognition rather than ligatures or words.

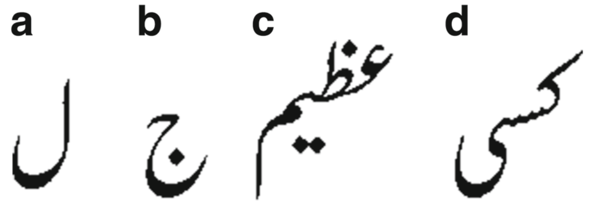


Fig. 4 Complete ligatures comprising: **a** Single-character primary ligature, **b** single-character primary and secondary ligature, **c** Multi-character primary and secondary ligatures, and **d** Multi-character primary ligature

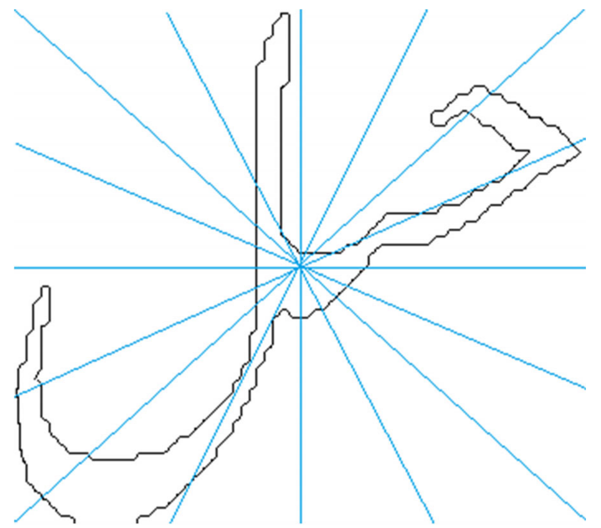


Fig. 5 Ligature image (after detection of Sobel edges) divided into 16 sectors around center

Table 2 Summary of features employed for clustering of ligatures using DTW

Features	Description	Dimension
*f*1-*f*16	Sum of horizontal edges in each sector	16
*f*17-*f*32	Sum of vertical edges in each sector	16

2.2 Holistic approaches

Holistic (segmentation-free) approaches employ partial words (ligatures) as units of recognition rather than characters. The ligatures themselves have to be extracted from text lines but since they are not further segmented, these techniques are termed as segmentation-free. Holistic techniques are known to be more robust for Urdu text as reported in a number of studies [5, 9, 10, 12, 19, 27–32].

Among notable holistic approaches, Javed et al. [10] present a holistic approach for recognition of Urdu text. DCT features extracted from sliding windows are used to train HMM-based classifiers for 1282 high-frequency Urdu ligatures. System evaluation reports 92% recognition rate for 3655 ligatures. In a similar work [28], modified tesseract OCR engine is adapted for Urdu Nastaliq font. The system with a reduced search space realizes around 97% recognition rate for primary ligatures in font sizes of 14 and 16. Sabbour and Shafait [5] contributed a large database of Urdu text line images, UPTI, now considered a benchmark for the evaluation of Urdu OCR systems. The recognition methodology relies on extracting shape descriptors from contours of ligature images. Classification using *k*-nearest neighbor classifier achieved a recognition rate of 89% for 10,000 primary ligatures.

A major drawback of most of the ligature-based systems is the sensitivity to font size and inability to handle dots and diacritics. Among the systems handling these issues, Khattak et al. [29] proposed a holistic system for separate recognition of primary and secondary ligatures irrespective of font size. Projection profile of edges, concavity, and curvatures features are extracted by sliding the windows right-to-left over the ligature image. Features are fed to the HMM classifiers, training separate HMMs for 2028 unique ligatures. System evaluation carried on 6084 query ligatures reports a recognition rate of 97.93%. The system, however, does not associate primary and secondary ligatures after recognition. Likewise, in continuation of their previous work [13], Akram et al. [28] employ the open source tesseract for recognition of multiple font-sized ligatures. The primary and secondary ligature recognition is carried out individually that are associated later to form the complete ligatures. The system evaluated on 224 documents achieves 86.15% end-to-end ligature recognition accuracy. The main drawback of the system is the need of separate training for each font size.

A summary of recent contributions to recognition of Urdu text is presented in Table 1. It can be seen from Table 1 that among the recognition systems that work on complete text lines, implicit segmentation-based methods have shown impressive recognition rates. A major issue with these implicit segmentation-based recognition techniques is that a large amount of training data is required. Moreover, recognition rates are reported at character level rather than the more natural unit of Urdu text—the ligatures.

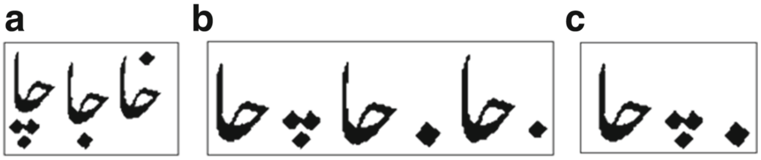


Fig. 6 a Complete ligatures having same primary and different secondary ligatures. **b** Separated as primary and secondary ligatures. **c** Candidate ligature clusters

Table 3 Character classes with members and respective Unicodes

Class name	Class image	Class members	Class unicode
Alif	ا	ا, آ	0627
Bay	ب	ب, پ, ت, ٹ, ث, ن, ہ, ی	0628
Jeem	ج	ج, ح, خ, چ	062C
Daal	د	د, ڈ, ذ	062F
Ray	ر	ر, ڑ, ز, ژ	0631
Seen	س	س, ش	0633
Suad	ص	ص, ض	0635
Toyen	ط	ط, ظ	0637
Ain	ع	ع, غ	0639
Fay	ف	ف, ق	0641
Qaaf	ق	ق	0642
Kaaf	ک	ک, گ	06A9
Laam	ل	ل	0644
Meem	م	م	0645
Noon	ن	ن, ں	0645
Waw	و	و	0648
Aik-chashmi-hay	ہ	ہ	06C1
Do-chashmi-hay	ھ	ھ	06BE
Choti-yaye	ی	ی	06CC
Bari-yaye	ے	ے	06D2

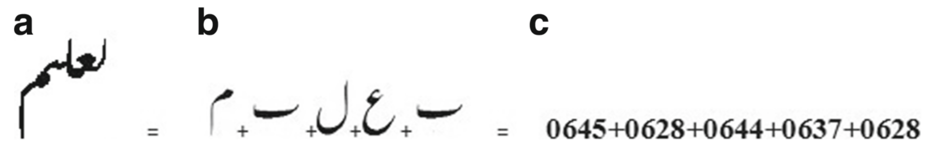


Fig. 7 **a** Sample primary ligature. **b** Constituent character classes. **c** Respective Unicodes

2.3 Motivation

A critical analysis of the literature on Urdu OCR systems reveals that the problem has attracted significant research attention during the last 10 years. While the initial endeavors primarily focused on recognition of isolated characters [6–8], a number of deep learning-based robust solutions [17, 23–26] have been proposed in the recent years. These methods mainly rely on implicit segmentation of characters and report high recognition rates. However, as discussed earlier, such systems ignore recognition of complete ligatures and require large amount of training data. Ligatures represent the most natural unit of recognition for cursive scripts like Urdu Nastaliq. From the viewpoint of end-to-end recognition systems, ligature recognition rates are more significant as compared to character recognition rates. Urdu has a huge set of unique complete ligatures summing up to around 26,000 [33]. Most of these complete ligatures, however, are very rarely used. Moreover, many complete ligatures only differ by position and the number of dots while the primary component of the ligature remains the same. Splitting ligatures into primary and secondary ligatures results in a significant reduction of the number classes. It has been shown that more than 99% of the complete Urdu corpus can be covered with around 2300 unique primary and secondary ligatures only [33]. Association of secondary ligatures with the primary ligatures after recognition, however, is a very challenging task and has been mostly ignored in ligature-based studies [13, 28, 29] on Urdu OCR systems. The present research aims to develop a robust technique for recognition of primary and secondary ligatures and their association allowing recognition of the complete ligatures.

Table 4 Table of dots/diacritics with example images and corresponding number values

Dot/Diacritic	Example image	Value	Description
No dots	–	−1	Absence of dot/diacritic
One dot above		01	One dot above baseline
One dot below		02	One dot below baseline
Two dots above		03	Two dots above baseline
Two dots below		04	Two dots below baseline
Three dots above		05	Three dots above baseline
Three dots below		06	Three dots below baseline
Hey stroke		07	Secondary stroke of “Aik-chashmi-hey” when used as joiner
Gaaf kash		08	Secondary stroke of “Gaaf”
Madda		09	Secondary stroke appearing with “Alif” forming “Alif-mad-aa”
Hamza-e-izafat		10	Secondary stroke appearing with “Bay” class when used as joiner
Shadda		11	Arabic like Thashdid
Full stop		12	Full stop
Choti toyen		13	Secondary stroke of “Thay,” “Rday,” and “Dhaal”
Hamza		14	Secondary stroke with “Bay” class, some times appears in isolation
Comma		15	Comma
Question mark		16	Question mark

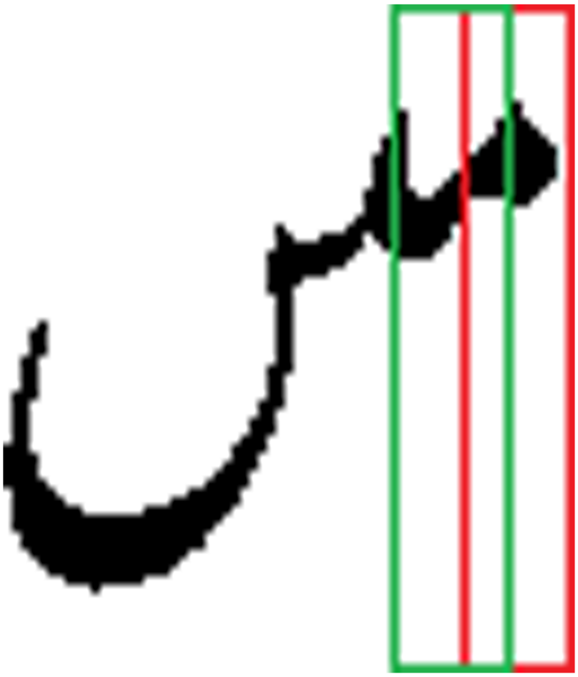


Fig. 8 A sample Urdu ligature with overlapped sliding windows

Table 5 Summary of frame features employed in our study

Features	Description	Dimensionality
f1	Hu's moments	7
f2	Horizontal projection of Zernike energy	32
f3	Vertical projection of Zernike energy	32
f4	Mean of horizontal projection of Zernike energy	1
f5	Mean of vertical projection of Zernike energy	1
f6	Horizontal projection of FFT energy	32
f7	Vertical projection of FFT energy	9
f8	Mean of horizontal projection of FFT energy	1
f9	Mean of vertical projection of FFT energy	1
	Total	116

2.4 Contributions

The key contributions of the present study include the following.

- A scale invariant, statistical features-based holistic OCR system for Urdu Nastaliq font is proposed that employs ligatures as units of recognition.
- A semi-automatic and scalable sequential clustering technique is presented to group ligatures into clusters to prepare the training data.
- Separate recognition of primary and secondary ligatures is carried through HMMs and recognized ligatures are combined to form the complete ligatures using a comprehensive reassociation technique.
- High-ligature recognition rates are realized on a benchmark dataset of Urdu text lines.

3 Proposed methodology

This section presents the details of the proposed methodology for recognition of Urdu Nastaliq text. A segmentation-free approach is adopted for recognition of complete ligatures. A complete ligature may comprise of one or more characters joined together through joiner rules of Urdu language. The list of joiner and non-joiner characters is illustrated in Fig. 2.

In many cases, multiple complete ligatures consist of the same primary ligature shape but differ only in the number, type or position of secondary ligatures as shown in Fig. 3. The ligatures are extracted from text line images. Each ligature is represented by a feature vector, and instances of the same ligatures are grouped into clusters. These clusters are employed to train a separate HMM for each ligature class. Once the system is trained, a document image presented to the system is first divided into ligatures, primary and secondary ligatures are separately recognized through HMMs and later associated (through assignment of dots and diacritics to the respective characters in the primary ligature). The details of the training and recognition modules are presented in the following.

3.1 Training

Training is carried out to make the models learn to discriminate between different ligature classes. The first 3000 text lines from the UPTI database are used as the training set in our study. The key steps in training involve extraction of ligatures from text lines, clustering of ligatures and training of hidden Markov models on clusters of ligatures as detailed in the following.

3.1.1 Ligature extraction

Urdu is a highly cursive language where lines and words are collections of ligatures that are similar to 'Parts of Arabic Words' (PAW). A complete ligature either comprises a single character or multiple characters joined (using joiner rules) together. Moreover, a ligature may only be primary ligature or a combination of primary and secondary ligatures (dots and diacritical marks), different examples being illustrated in Fig. 4. We employ connected component labeling on each line of text to extract the ligatures irrespective of type (primary or secondary). These ligatures are then grouped into clusters as discussed in the following.

3.1.2 Ligature clustering

Training a classifier to recognize classes requires labeled ligature classes. Manually generating and labeling the

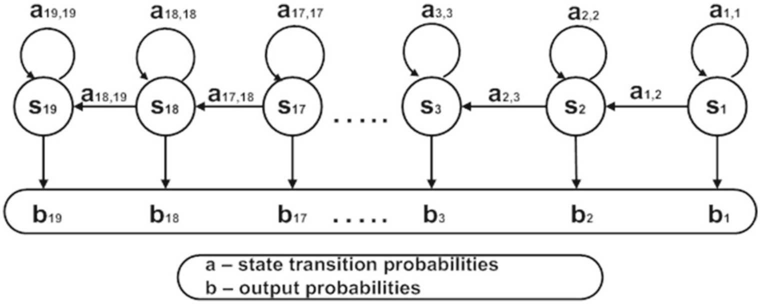


Fig. 9 19-state HMMs employed for classification

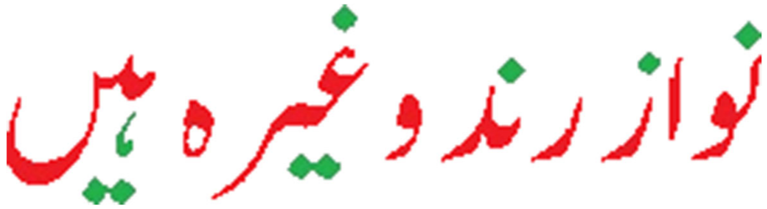


Fig. 10 Sample text line with ligatures identified as primary (red) and secondary (green) ligatures

training data is naturally an expensive solution in terms of time and effort. We, therefore, carry out a semi-automatic clustering of extracted ligatures to generate the training data for the classifiers. Errors in the generated cluster classes are then corrected through visual inspection in order to make clusters error free to serve as training data.

We adapt a sequential algorithm [1, 34, 35] that employs dynamic time warping (DTW) [1] for clustering of ligatures. The algorithm does not require the number of desired classes in advance. *Sobel* edge detector is first applied to the binarized ligature image in order to find the horizontal and vertical edges in the ligature. The image is next divided into 16 sectors around its center (Fig. 5) that are eventually used for extraction of effective and computationally inexpensive 32 dimensional features as summarized in Table 2.

Initially, a ligature is randomly chosen and is assumed as mean of the first cluster. Each of the remaining ligatures is then picked one by one and the distance from the mean of each cluster is computed using DTW. If the distance to the nearest cluster is below a predefined threshold, the current ligature is added to the respective cluster and the cluster mean is updated. Otherwise, a new cluster is created with the present ligature as its center. The sequential clustering algorithm does not impose any constraints on the number of clusters and hence the system remains scalable to add more clusters. The algorithm, however, is sensitive to the order in which the ligatures are presented to it. Nevertheless, it should be noted that the idea is to generate an approximate set of clusters which are corrected by visual inspection prior to training the models (as they form training data for the classifiers). To keep only the high-frequency ligature (HFL) clusters, the Unicode associated with each cluster is compared with those in the standard frequency list of HF ligatures compiled by the Center of Language Engineering (CLE) [21]. A cluster that finds a match in the HFL list is kept in the database while the remaining clusters are discarded. The process produced a total of 1525 HFL clusters each containing at least 10 instances.

As discussed earlier, the primary and secondary ligatures are treated as separate clusters in our study. This allows reducing the total number of unique clusters. The idea is illustrated in Fig. 6 where three ligatures segmented into primary and secondary components are shown along with the unique ligature clusters to be generated.

Once the clusters are produced, each cluster is assigned its respective Unicode. Each character has a unique Unicode in Urdu, and ligatures having multiple characters are assigned a code that is a combination of the Unicodes of the characters constituting the ligature. Similar to ligature level, at character level, the number of unique character classes is reduced to 20 after removing the dots and diacritics. The different forms (isolated, initial, middle, end) of characters which appear similar in the absence of dots/diacritics are assigned a single class label. The idea is illustrated in Table 3 where groups of characters are identified that have the same visual appearance in the absence of dots and diacritics. A ligature is then represented by the combination of the Unicodes of the character classes' representatives as shown in Fig. 7. Unlike primary ligatures, the secondary ligatures (and punctuation marks) are labeled by unique values from 01–16 as described in Table 4.

Once the primary and secondary ligatures are grouped into clusters and each cluster is assigned the respective Unicode, we proceed to training models to learn the ligature classes. We have selected to employ hidden Markov

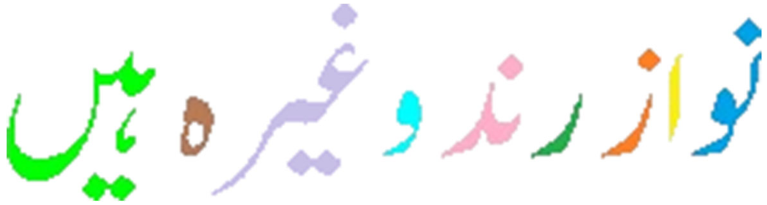


Fig. 11 Primary ligatures with respective secondary ligatures association (same color) forming complete ligatures

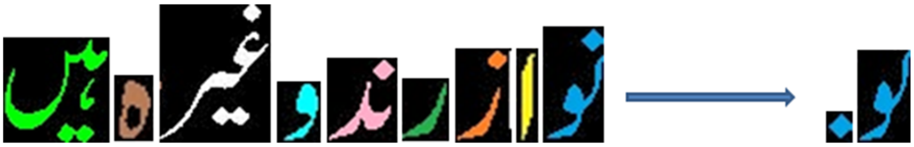


Fig. 12 Text line segmented as complete ligatures showing sample separation into primary and secondary ligatures

models which have been successfully applied to a number of diverse problems including gesture recognition [36–38], speech recognition [39], handwriting recognition [38, 40], musical score recognition [41], and optical character recognition [10, 42–44]. The steps of feature extraction from ligature clusters and subsequent training are discussed in the following subsections.

3.1.3 Feature extraction

Features play a vital role in any recognition system. Features are broadly categorized into structural and statistical features. While structural features are a rich and intuitive representation of the objects under study, classification using structural features, in general, is computationally intensive. Statistical features which represent certain statistics computed from the objects or shapes under study can be compared using a wide range of classifiers. In our study, statistical features are computed by sliding an overlapped window over the height-normalized image of each ligature. Height normalization preserves the aspect ratio of the ligature image and also ensures that the computed features are not dependent upon the size of ligature. We normalize the height of each ligature to 32 pixels and employ a sliding window of 32 × 9 with an overlap of 4 pixels as illustrated in Fig. 8.

The set of features extracted from each window/frame includes Hu's moments [45], horizontal and vertical projections with respective mean values of two-dimensional fast fourier transform (FFT) energy and Zernike moments [46] energy. These features are briefly described in the following.

Hu's moments: For an $M \times N$ binary image *f*, the regular moment is defined as:

$$u_{pq} = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} x^p y^q f(x,y)$$

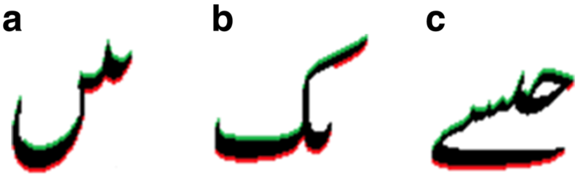


Fig. 13 Primary ligatures with upper (green) and lower (red) profiles as reference baselines

where $f(x,y)$ is the intensity of the pixel at pixel coordinates (x,y) and $p+q$ is the order of the moment. For translation invariance, the center of gravity of the image $(\bar{x},\bar{y})$ is used to define the central moments:

$$u_{pq} = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} (x-\bar{x})^p (y-\bar{y})^q f(x,y)$$

Hu further derived a set of seven rotational invariant moments $M_0 \ldots M_7$ for an effective representation of the shapes under study. In our implementation, the seven Hu's moments are computed from each frame and are employed as features. The computational details of these moments can be found in [45].

Zernike energy features: Zernike moments of order four [47, 48] are computed from each frame that is resized to 32 × 32. The computation of Zernike moments comprises three main steps, computing the radial polynomial, computing the basis function of Zernike and computing Zernike moments. Zernike moments for discrete image of symmetric size $N \times N$ are computed as follows.

$$\begin{aligned} Z_{n,m} &= n+1/\lambda_N \sum_{c=0}^{N-1}\sum_{r=0}^{N-1} f(x,y) V^*_{m,n}(x,y) \\ &= n+1/\lambda_N \sum_{c=0}^{N-1}\sum_{r=0}^{N-1} f(x,y) R^*_{m,n}(\rho_{xy}) e^{-jm\theta_{cr}} \end{aligned}$$

Where λ_N is the normalization factor and $0 \le \rho_{xy} \le 1$. n represents the order of the radial polynomial and is a non-negative integer. m is an integer satisfying the constraints

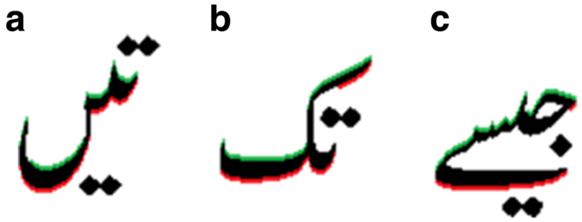


Fig. 14 a, **b** Primary ligature(s) without loop and **c** with loop showing secondary ligatures positions

Table 6 Lookup table for character classes—possible occurrences of dots/diacritics

Character class	Character class shape	Possible dot/diacritic number
Alif	ا	9
Bay	ٮ	1, 2, 3, 4, 5, 6, 7, 10, 11, 13, 14
Jeem	ح	1, 2, 6
Daal	د	1, 13
Ray	ر	1, 5, 13
Seen	س	5
Suad	ص	1
Toyen	ط	1
Ain	ع	1
Fay	ڡ	1, 3
Qaaf	ٯ	3
Kaaf	ک	8
Laam	ل	11
Meem	م	−1
Noon	ں	1
Waw	و	−1
Aik-chashmi Hey	ہ	−1
Dochashmi-Hey	ھ	−1
Choti-Yaye	ى	−1
Bari-Yaye	ے	10, 14

$n - |m| = even$ and $m \leq n$, representing the repetition of the azimuthal angle. $R_{n,m}$ is radial polynomial, and $V_{n,m}$ is a 2-D Zernike basis function [47]. The amplitude of Zernike moments is calculated from moment's absolute values that are used for computing Zernike energy normalized by the frame size.

$$Zernike_{energy} = (abs(ZernikeMoments)).^{2}/(size_{frame})$$

The Zernike energy is then used to find the horizontal projection (the sum of each row in the energy matrix; dimension, 32), vertical projection (the sum of each column in the energy matrix; dimension, 32), and mean values of the two projections (dimension, 2) forming a 66 dimensional Zernike Energy feature vector for each frame.

Two-dimensional FFT energy features: Fourier transform converts an image from the spatial to the frequency domain. The two-dimensional discrete Fourier transform of an image is computed as follows.

$$F(u,v) = \sum_{x=0}^{N-1}\sum_{y=0}^{M-1} f(x,y)e^{-j2\pi(ux/M+vy/N)}$$

where $u = 0, 1, 2, \ldots M - 1$ and $v = 0, 1, 2, \ldots, N - 1$. For each frame, we compute the 2D-FFT and find the absolute value of each FFT coefficient. These absolute values are then employed to compute the (normalized) FFT energy profile.

$$FFT_{energy} = abs(FFT_{frame}).^{2}/size_{frame}$$

The dot operator with the power of 2 represents the element-wise square of absolute values while division by size of frame ($frame_{size}$) normalizes the FFT energy. The sum of each row of FFT energy matrix gives its horizontal projection while the sum of each of column gives a vertical projection of energy. The horizontal and vertical projections and their mean values form a 43 (32+9+1+1) dimensional FFT energy-based feature vector.

Each frame is represented by a single 116-dimensional feature vector as summarized in Table 5.

Table 7 Sample character class with respective dot/diacritics values forming characters

Character class	Dot/diacritic class	Form-able character
Jeem(ح)	(◆=02), (∵=06), (−1), (◆=01)	Jeem(ج) Chay(چ) Hay(ح) Khay(خ)

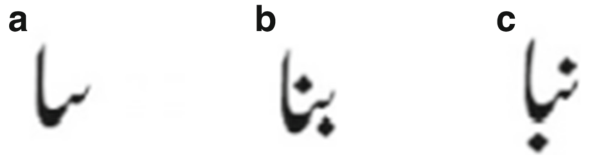


Fig. 15 a The same primary ligature forming different complete ligatures **b** 'Bana' and **c** 'Naba', due to the same dots (having the same *position* information) with different *access order*

3.1.4 Hidden Markov Model (HMM) training

A separate HMM is trained for each of the 1525 high-frequency ligature clusters. Each cluster comprises atleast 10 images of the respective ligature. For training, the features extracted from each ligature using right-to-left sliding windows are employed. Since HMMs are discrete, the extracted features are quantized to a 100-symbol codebook and a 19-state right-to-left HMM (Fig. 9) is trained on ligature images in each cluster using the standard Baum-Welch algorithm. Each trained HMM is associated with the Unicode of the respective ligature. Once the models are trained, we proceed to recognition of query ligatures as presented in the following section.

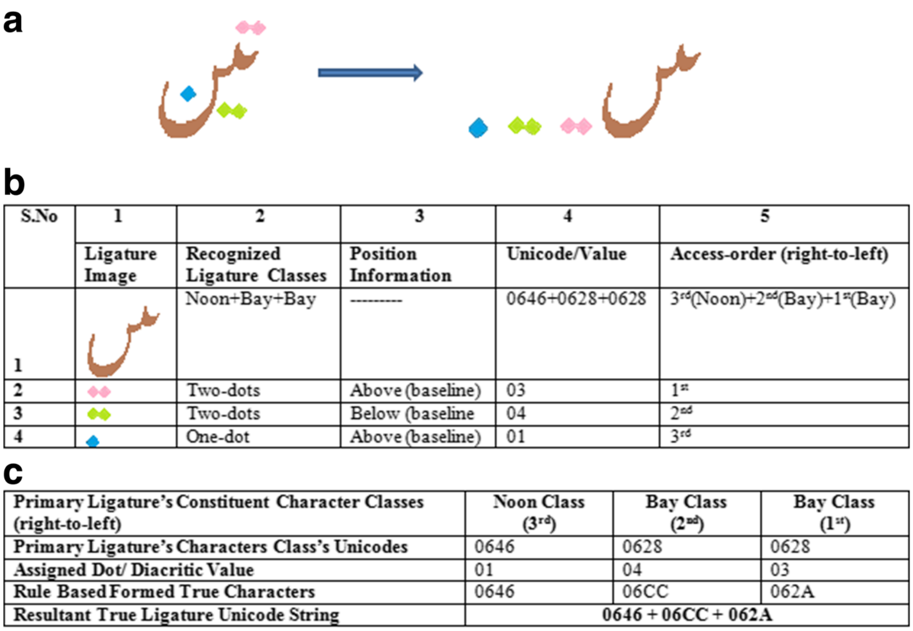
3.2 Ligature recognition

For recognition, text lines fed to the system are scanned from right to left. Six *handcrafted* features reported in [49] with additional features (width, height, area, loops) are extracted for identification of ligatures as primary or secondary, as indicated in Fig. 10. Once identified, secondary ligatures are associated with their respective primary ligatures forming complete ligatures as shown in Fig. 11. Details of the ligature identification and association method can be found in [49].

Prior to recognition, the *position* and *access-order* information of secondary ligatures with respect to the upper and lower profiles of primary ligature is extracted and stored. Recognition is carried out by feeding the feature sequences extracted from a query ligature to all the trained models. The model that reports the maximum probability characterizes the query ligature. The *position* and *access-order* information are later used for association of secondary ligatures with their primary ligature to form the complete ligature, which is eventually validated from a dictionary and is written to a text file.

3.2.1 Secondary ligature position and access-order information

As discussed earlier, the input text lines are scanned from right to left for extraction of complete ligature that is split into primary and secondary ligatures (Fig. 12), if any. In order to retrieve the information on the position of secondary ligature, the upper and lower profiles of the primary ligature are used as two reference baselines as shown (by green and red colors, respectively) in Fig. 13. A primary ligature may or may not contain loop(s); hence, scanning from right to left, the position information of a secondary ligature with respect to the (primary ligature's) reference baselines is found as follows.



S.No	1	2	3	4	5
	Ligature Image	Recognized Ligature Classes	Position Information	Unicode/Value	Access-order (right-to-left)
1		Noon+Bay+Bay	---------	0646+0628+0628	3rd(Noon)+2nd(Bay)+1st(Bay)
2	◆◆	Two-dots	Above (baseline)	03	1st
3	◆◆	Two-dots	Below (baseline	04	2nd
4	◆	One-dot	Above (baseline)	01	3rd

Primary Ligature's Constituent Character Classes (right-to-left)	Noon Class (3rd)	Bay Class (2nd)	Bay Class (1st)
Primary Ligature's Characters Class's Unicodes	0646	0628	0628
Assigned Dot/ Diacritic Value	01	04	03
Rule Based Formed True Characters	0646	06CC	062A
Resultant True Ligature Unicode String	0646 + 06CC + 062A		

Fig. 16 a Splitting a complete ligature as primary and secondary components. **b** Recognized primary and secondary ligatures with the position and access-order information of secondary ligatures. **c** Assignment of recognized secondary ligatures to character classes of primary ligature to determine true characters

1. *Primary ligature with no loop(s)*
   - Secondary ligature is considered *above* the baseline if it is above the upper profile.
   - Secondary ligature is considered *above* the baseline if it is in between the upper and lower profiles.
   - Secondary ligature is considered *below* the baseline if it is below the lower profile.
2. *Primary ligature with loop(s)*
   - Secondary ligature is considered *above* the baseline if it is above the upper profile and *below* otherwise.

The complete process for retrieval of position and access information of dots/diacritics is presented in Algorithm 1. The dot/diacritic positions of the two types of primary ligatures (without loop and with loop) are shown in Fig. 14. The position and access-order information is computed for all secondary components considered in our study. The position information, however, matters only for *one dot* and *two dots* while the access-order information is required for all the secondary components (Table 4).

Algorithm 1 : Algorithm for secondary ligature's *position* and *access-order* information retrieval

```
Require: Complete ligature image $I$
Ensure: Array of secondary ligatures $position$ and $access - order$ information $infoArray$
  $index \leftarrow 1$
  $infoArray \leftarrow null$
  //Find labeled components L and is its total number N
  $[LN] \leftarrow Labeled(I)$
  //Extract primary ligature pLigature and its label pLabel
  $[pLigaturepLabel] \leftarrow getPrimary(I, L, N)$
  //Find baseline from primary ligature
  $baseline \leftarrow getBaseline(pLigature)$
  //Extract info. on of position and access-order secondary ligs
  For($i \leftarrow 1toN$)
  If($i \neq pLabel$)
  //Extract secondary ligature's dimensions
  $dimension \leftarrow getDimension(L, i)$
  //Extract position information
  $position \leftarrow getPosition(dimension, baseline)$
  //Store position sequentially
  $infoArray[index] \leftarrow position$
  $index \leftarrow index + 1$
  End
  End
```

Table 8 Recognition rates of (primary, secondary, and punctuation ligature) clusters

Ligature type	Number of ligatures	Correctly recognized	Recognition rate
Primary ligatures	5969	5685	95.24%
Secondary ligatures	3591	3342	93.07%
Punctuations	218	218	100%
Total ligatures	9778	9245	94.55%
Total complete ligatures	6187	5708	92.26%

3.2.2 *Primary and secondary ligature recognition*

After retrieval of *position* and *access-order* information, the complete ligature's primary as well as secondary ligatures are individually fed to the trained HMM classifiers. Recognition is carried out by the HMM producing the highest probability for the queried ligature, the corresponding label is returned as output. These labels are not the complete ligature Unicodes but represent a collection of corresponding character classes' Unicodes (Table 3) for primary ligatures and the numeric codes (Table 4) for secondary ligatures. A postprocessing of the association of secondary ligatures with the primary ligature and subsequent assignment of each secondary ligature to the character classes of primary ligature is carried to form complete characters and is described as follows.

3.2.3 *Secondary ligature association and complete ligature formation*

Once recognized, the dots/diacritics (secondary ligatures) are associated with the respective primary ligature using

Table 9 Complete ligatures (per number of characters) with respective recognition rates

Characters in ligature	Total ligatures	Correctly recognized	Recognition rate
Isolated characters	2705	2657	98.22%
Two character ligatures	2391	2287	95.65%
Three character ligatures	733	483	65.89%
Four character ligatures	106	49	46.22%
Five character ligatures	31	14	45.16%
Six character ligatures	2	0	0.00%
Seven character ligatures	1	0	0.00%
Punctuations	218	218	100%
Total true ligatures	6187	5708	92.26%

the pre-computed *position* information. A list is maintained for numeric values of secondary ligatures with respective Unicode value (composed of single or multiple character class's Unicodes) of primary ligature. Next, the *access-order* information comes in to play for rule-based assignment of secondary ligatures (if any) to the character classes that constitute the primary ligature in order to form the characters. Starting from right to left, the possible occurrence of the first secondary ligature (in *access-order*) with the first character class (of the primary ligature) is verified from a lookup table (Table 6) and assigned to from a true character. Otherwise, the process is repeated for the subsequent classes till the last character class of the primary ligature. The idea is illustrated in Table 7 for character class "Jeem" of Table 6 (row number 3) where the recognized character class "Jeem" may either have one of the following secondary ligatures: one-dot-above, one-dot-below, three-dots-below, or may not have any secondary ligature. This leads to the formation of true characters "Khay," "Jeem," "Chay," or "Hay" as illustrated in Table 7.

Once assigned, a dot/diacritic can not be assigned to any other character class. Similarly, the second dot/diacritic is verified and assigned, and so forth. The *access-order* information effectively resolves the assignment issue of the same secondary ligatures with the same position information to the same primary ligatures as summarized in Fig. 15.

The overall recognition process of a query ligature is summarized in Fig. 16. The splitting of true ligature into primary and secondary ligatures is shown in Fig. 16a. Figure 16b shows the individually recognized ligature classes, *position* information of secondary ligatures with respect to reference baselines, Unicodes of primary ligatures, and number values and the *access-order* information of secondary ligatures. Likewise, Fig. 16c elaborates the idea of assigning the recognized dots/diacritics to the character classes of the

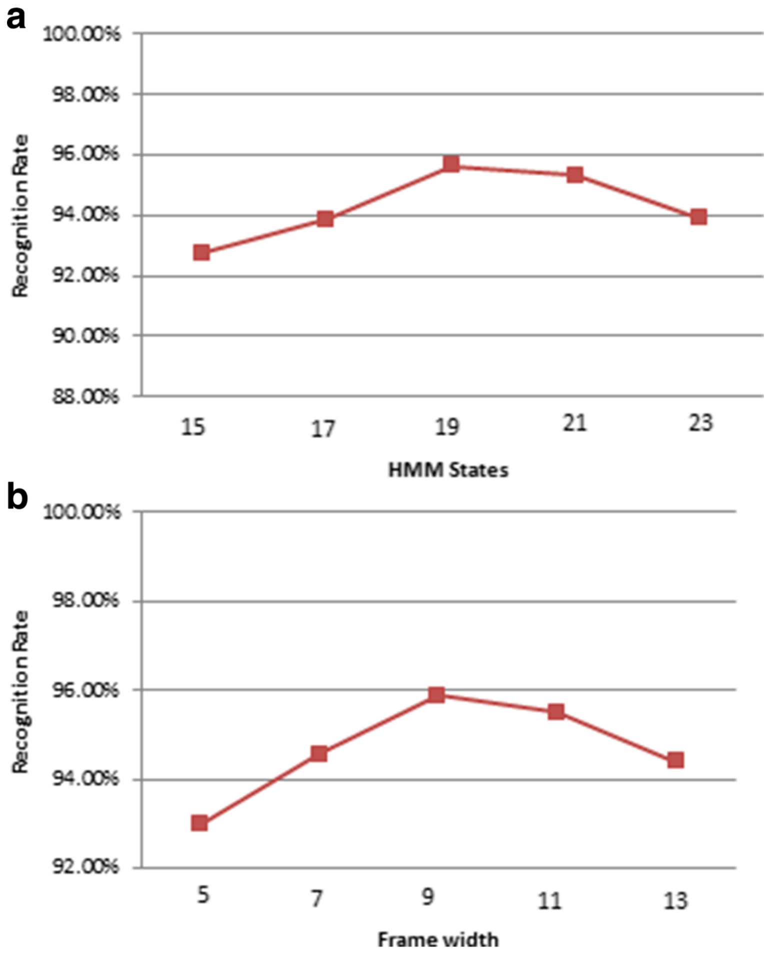


Fig. 17 Classification rates (500 ligatures) as a function of **a** HMM states (window width = 9) **b** window width for feature extraction (HMM states = 19)

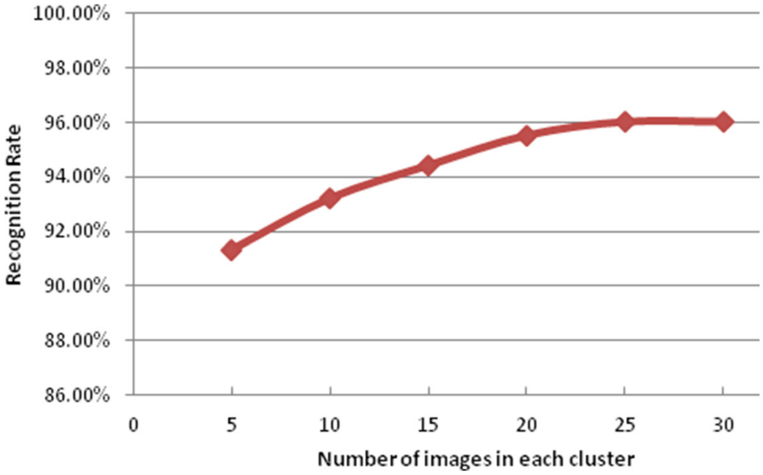


Fig. 18 Classification rates (500 ligatures) as a function of number of images in each ligature cluster

recognized primary ligature to form the complete ligature Unicode string.

Once recognized, the Unicode strings of the true characters are concatenated to form the Unicode string of the complete ligature which is written to a text file in UTF-8 after verification from a lexicon.

4 Results and analysis

This section presents the details of the experiments carried to assess the effectiveness and validity of the proposed technique. Recognition rates on primary and secondary ligatures are computed separately as well as after the association of the two. The achieved results are also compared with the recent Urdu OCR systems reported in the literature. The system is trained on 1525 high-frequency ligature clusters among which 16 clusters represent dots and diacritics. Evaluation is carried on 6187 complete/true ligatures comprising 1 to 7 characters per ligature. These test complete ligatures have been extracted from text lines selected randomly from the 7063 text lines (which were not used for system training). When distinguished into primary and secondary ligatures, the 6187 complete ligatures result in a total of 9778 primary and secondary ligatures. An overall recognition rate of 94.55% is achieved on these 9778 ligatures. After the association of primary and secondary ligatures, the recognition rate on complete ligatures (partial words) reads 92.26%. These results are summarized in Table 8. It can be seen that the individual recognition rates on primary and secondary ligatures are comparatively high (95.24% for primary ligatures, 93.30% for secondary ligatures, and 100% for punctuations) and drop (to 92.26%) once both are associated together. This

Table 10 Comparison of proposed method with notable studies

Study	Dataset	Recognition unit	Accuracy	Complete ligature recognition
Ahmed et al. [22]	UPTI	Character	89.00%	–
Hassan et al. [16]	UPTI	Character	87.40%/ 94.85%	–
Naz et al. [17]	UPTI	Character	96.40%	–
Naz et al. [18]	UPTI	Character	94.97%	–
Javed et al. [10]	CLE	Ligature	92.00%	No
Akram et al. [13]	CLE	Ligature	97.87%	No
Javed and Hussain [19]	CLE	Ligature	92.73%	No
Khattak et al. [29]	CLE	Ligature	97.93%	No
Sabbour and Shafait [5]	UPTI	Ligature	91.00%	No
Akram et al. [28]	CLE	Ligature	86.15%	Yes
Hussain et al. [21]	CLE	Ligature	87.76%	Yes
Proposed	UPTI	Ligature	92.26%	Yes

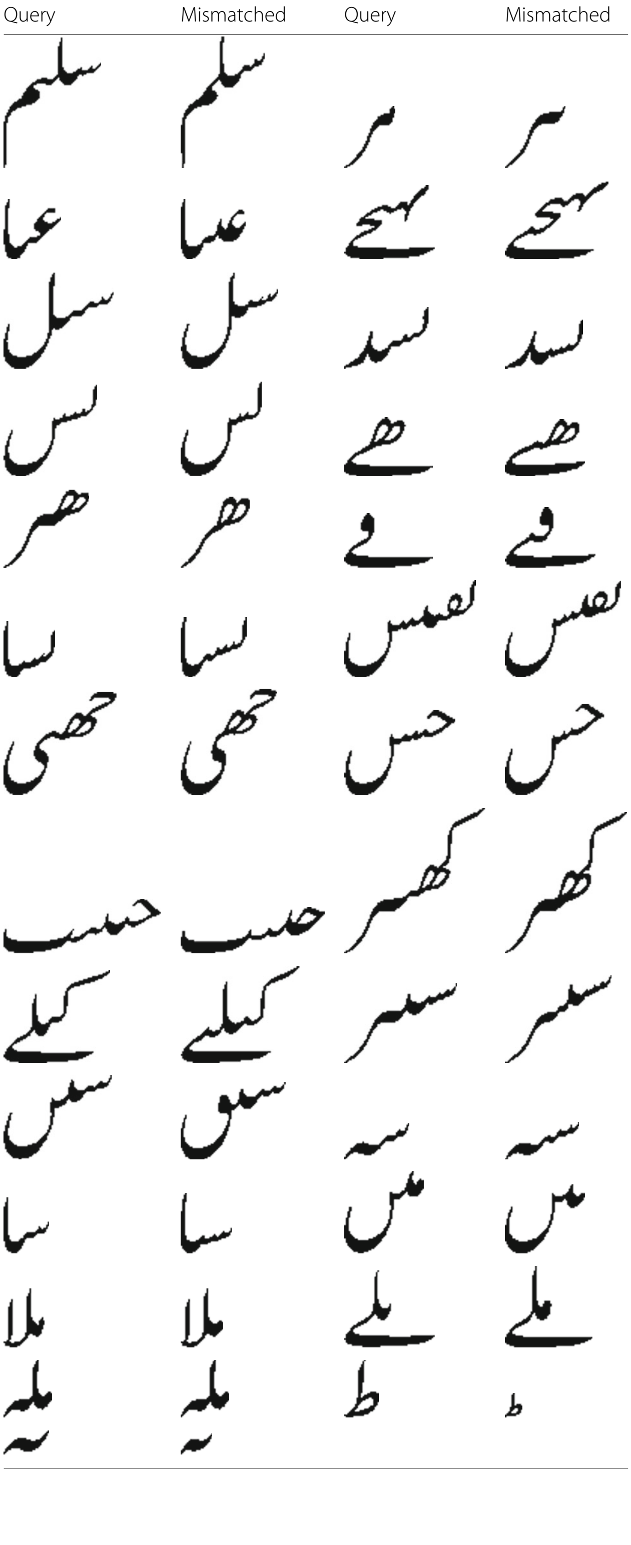
drop is very much natural as the association of dots and diacritics with the primary ligatures is a challenging task. The highly cursive nature of the script makes it difficult to correctly associate all secondary ligatures with the respective primary ligatures. Considering the challenges it offers, a recognition rate of around 92% on more than 6000 ligatures is very much promising.

We also computed recognition rates of complete ligatures as a function of complete ligature length (number of characters in the complete ligature). A summary of these results is presented in Table 9. It can be seen that recognition rates gradually decrease with the increase in the number of characters per complete ligature. This observation can be attributed to the increasing number of dots/diacritical marks with the increasing length of ligature which makes recognition more challenging.

We also carried experiments to study the performance of the system as a function of system parameters including the frame size and number of hidden states in the HMMs. First, 500 ligatures of the dataset were selected for conducting these experiments. The recognition rates for different frame sizes and the number of HMM states are illustrated in Fig. 17. It can be seen that the recognition rates are more sensitive to smaller frame widths in a way that smaller windows are unable to capture distinguishing characteristic of ligatures. Likewise, an increase in recognition rate is observed by increasing the number of HMM states. The recognition rates start to stabilize at 17–21 states. Similarly, we study the performance evaluation as a function of the number of images in each ligature cluster. The number of images in each training cluster is varied from 5 to 30, and the realized classification rates are summarized in Fig. 18. It can be seen from Fig. 18 that a promising recognition rate of around 91% is achieved with only five training images in each cluster. It can also be noticed that the recognition rates stabilized from 20 images per cluster onwards, a manageable number of training the system.

We also carried a comparison of our system performance with those of the notable recent techniques on the same problem. As discussed earlier, most of the work on Urdu OCR considers either isolated characters [6–8] or separate recognition of primary and secondary ligatures [13, 29] without associating the two. In most cases, the results have been reported on custom-developed databases making it difficult to objectively compare different systems. Comparison of our system can be carried out with [5, 16–18, 22] and [5, 10, 13, 19, 21, 28, 29] on the basis of database used (UPTI) and recognition unit (ligatures), respectively. The recognition rates of these studies are summarized in Table 10. It should however be noted that we employ ligatures as basic units of recognition as opposed to characters in [16–18, 22]. Consequently, we report the ligature recognition rates while the implicit segmentation-based studies listed in Table 10 report character recognition rates. The character recognition rates are provided for completeness, and the two rates are not directly comparable as in case of ligatures, an error in a single character leads to rejection of complete ligature

Table 11 Examples of similar ligatures resulting in false matches


Query
Mismatched
Query
Mismatched

representing a more challenging scenario. Among studies evaluated on ligatures [5, 10, 13, 19, 21, 28, 29], only [28] and [21] consider the association of primary and secondary ligatures. These studies, however, are evaluated on a different dataset (CLE). A meaningful comparision of our work is possible with [5] where the authors employ ligatures as units of recognition and use the same UTPI database. A recognition rate of 91% is reported in [5] without associating primary and secondary ligatures together (hence complete ligatures are not recognized). Our proposed technique achieves a recognition rate of 92.26% complete ligature recognition rate (after the association of primary and secondary ligatures) demonstrating its effectiveness.

In an attempt to investigate the reasons for recognition errors, we analyzed the classification errors in the recognition and re-association phases. It was observed that the errors mainly resulted due to the following factors.

- Ligatures with visually similar shapes
- False joining of secondary ligatures with the respective or neighboring primary ligature
- False re-association of dots/diacritics with the primary ligatures

Table 11 illustrates examples of few of the mismatched ligatures where it can be seen that the queried ligature have close resemblance with the falsely recognized ligature leading to classification errors. Likewise, in some cases, the secondary ligatures are joined with their own or an adjacent primary ligature and cannot be extracted (Fig. 19) causing false recognition. In a similar fashion, in situations where consecutive character classes have the same dots/diacritics according to the lookup table (Table 6), the postprocessing step may result in false associations. Example errors of false association are illustrated in Table 12.

5 Conclusions

We presented a holistic optical character recognition system for printed Urdu Nastliq font using statistical features and Hidden Markov Models employing ligatures as units of recognition. The developed system is trained on 1525 unique high-frequency Urdu ligature clusters from the standard UPTI database. The complete ligatures are first split into primary and secondary ligatures and are recognized separately. The secondary ligatures are then associated with the primary ligature using a set of heuristics to recognize complete ligature. The system evaluated through a number of interesting experiments achieved high recognition rates which are comparable to the recent studies on this problem.

In our further study on the subject, we intend to incorporate the entire set of Urdu HFLs (around 2300) to cover almost complete (99%) Urdu vocabulary. Likewise, we presently consider 16 frequently occurring dots and diacritics and this number can be enhanced as well. The postprocessing which associates secondary components with the respective primary components can be further improved to reduce the recognition errors when classifying the true ligatures.

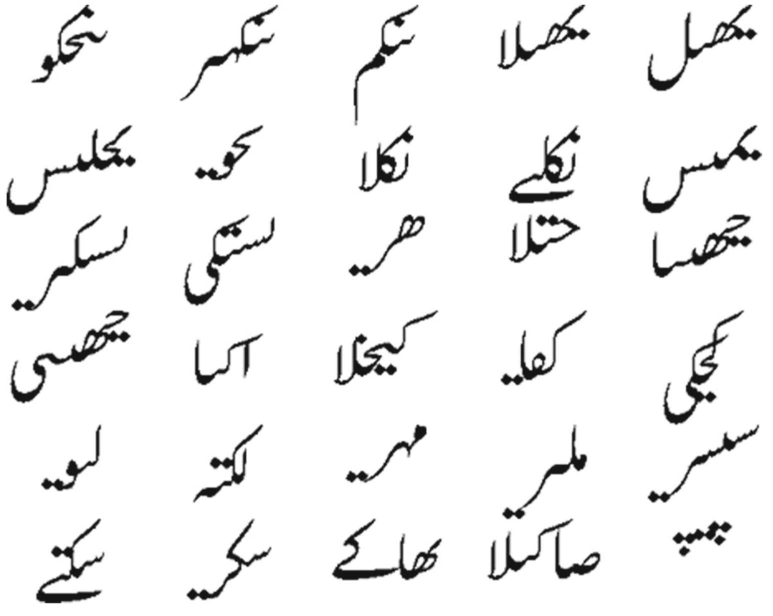


Fig. 19 Examples of joined ligatures resulting in a mismatch

Table 12 Examples of incorrectly postprocessed complete ligatures (after correct recognition of primary and secondary ligatures) due to incorrect re-association

Query complete ligature	Falsely formed complete ligature
عظم	غطم
عظیم	غطیم
جعفر	جغفر
حسنہ	خسٮہ
تحفظا	تخفطا
تحفظ	تخفط
بعض	بغص
ستعفے	ستغفے
ستعفی	ستغفی
لطف	ڶطف
محنت	مختٮ

Funding
The presented research is not a part of any funded project and is carried out during the PhD research work of Israr Ud-Din.

Authors' contributions
IS and SK contributed to the algorithmic development while IUD and TA contributed to the implementation and paper writing. All authors read and approved the final manuscript.

Competing interests
The authors declare that they have no competing interests.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Author details
[1]Bahria University, Islamabad, Pakistan. [2]Federal Directorate of Education, Islamabad, Pakistan.

Received: 25 February 2017 Accepted: 17 August 2017
Published online: 06 September 2017

References

1. S Shabbir, I Siddiqi, Optical character recognition system for Urdu words in nastaliq font. Int. J. Adv. Comput. Sci. Appl. **7**(5), 567–76 (2016)
2. S Naz, AI Umar, SB Ahmed, SH Shirazi, MI Razzak, I Siddiqi, in *Multi-Topic Conference (INMIC), 2014 IEEE 17th International*. An OCR system for printed nasta'liq script: a segmentation based approach (IEEE, Pakistan, 2014), pp. 255–259
3. ST Javed, Investigation into a segmentation based OCR for the nastaleeq writing system, Master's thesis, National University of Computer and Emerging Sciences Lahore, Pakistan (2007)
4. DA Satti, Offline Urdu nastaliq ocr for printed text using analytical approach, upublished master's thesis, Quaid-i-Azam University Islamabad, Pakistan (2013)
5. N Sabbour, F Shafait, in *IS&T/SPIE Electronic Imaging*. A segmentation-free approach to Arabic and Urdu OCR (International Society for Optics and Photonics, USA, 2013), pp. 86580–86580
6. U Pal, A Sarkar, in *proceedings of the 7th International Conference on Document Analysis and Recognition (ICDAR'03)*. Recognition of printed Urdu Script, (UK, 2003), pp. 1183–1187
7. I Shamsher, Z Ahmad, JK Orakzai, A Adnan, OCR for printed Urdu script using feed forward neural network. Proc. World Acad. Sci. Eng. Technol. **23**, 172–175 (2007)
8. J Tariq, U Nauman, MU Naru, in *Computer Engineering and Technology (ICCET), 2010 2nd International Conference On*. Softconverter: a novel approach to construct OCR for printed Urdu isolated characters, vol. 3 (IEEE, China, 2010), pp. V3–495
9. S Sardar, A Wahab, in *Information and Emerging Technologies (ICIET), 2010 International Conference On*. Optical character recognition system for Urdu (IEEE, Pakistan, 2010), pp. 1–5
10. ST Javed, S Hussain, A Maqbool, S Asloob, S Jamil, H Moin, Segmentation free nastalique Urdu OCR. World Acad. Sci. Eng. Technol. **46**, 456–461 (2010)
11. Z Ahmad, JK Orakzai, I Shamsher, A Adnan, in *Proceedings of World Academy of Science, Engineering and Technology*. Urdu nastaleeq optical character recognition, vol. 26 (Citeseer, 2007), pp. 249–252
12. T Nawaz, S Naqvi, H ur Rehman, A Faiz, Optical character recognition system for Urdu (naskh font) using pattern matching technique. Int. J. Image Process. (IJIP). **3**(3), 92 (2009)
13. QUA Akram, S Hussain, A Niazi, U Anjum, F Irfan, in *Document Analysis Systems (DAS), 2014 11th IAPR International Workshop On*. Adapting tesseract for complex scripts: an example for Urdu nastalique (IEEE, France, 2014), pp. 191–195
14. Z Ahmad, JK Orakzai, I Shamsher, in *Computer Science and Information Technology, 2009. ICCSIT 2009. 2nd IEEE International Conference On*. Urdu compound character recognition using feed forward neural networks (IEEE, China, 2009), pp. 457–462
15. H Malik, MA Fahiem, in *Visualisation, 2009. VIZ'09. Second International Conference In*. Segmentation of printed Urdu scripts using structural features (IEEE, 2009), pp. 191–195
16. A Ul-Hasan, SB Ahmed, F Rashid, F Shafait, TM Breuel, in *2013 12th International Conference on Document Analysis and Recognition*. Offline printed Urdu nastaleeq script recognition with bidirectional LSTM networks (IEEE, USA, 2013), pp. 1061–1065
17. S Naz, AI Umar, R Ahmad, SB Ahmed, SH Shirazi, I Siddiqi, MI Razzak, Offline cursive Urdu-nastaliq script recognition using multidimensional recurrent neural networks. Neurocomputing. **177**, 228–241 (2016)
18. S Naz, AI Umar, R Ahmad, SB Ahmed, SH Shirazi, MI Razzak, Urdu nastaliq text recognition system based on multi-dimensional recurrent neural network and statistical features. Neural Comput. Appl. **28**(2), 1–13 (2015)
19. ST Javed, S Hussain, in *Iberoamerican Congress on Pattern Recognition*. Segmentation based Urdu nastalique OCR (Springer, Cuba, 2013), pp. 41–49
20. Line and ligature segmentation in printed Urdu document images. J. Appl. Environ. Biol. Sc. **6**(3S), 114–120 (2016)

21. S Hussain, S Ali, QU Akram, Nastalique segmentation-based approach for Urdu OCR. Int. J. Doc. Anal. Recognit. (IJDAR). **18**(4), 357–374 (2015)
22. SB Ahmed, S Naz, MI Razzak, SF Rashid, MZ Afzal, TM Breuel, Evaluation of cursive and non-cursive scripts using recurrent neural networks. Neural Comput. Appl. **27**(3), 603–613 (2016)
23. MR Yousefi, MR Soheili, TM Breuel, E Kabir, D Stricker, in *Document Analysis and Recognition (ICDAR), 2015 13th International Conference On*. Binarization-free OCR for historical documents using LSTM networks (IEEE, France, 2015), pp. 1121–1125
24. A Ul-Hasan, SS Bukhari, A Dengel, in *2016 12th IAPR Workshop on Document Analysis Systems (DAS)*. Ocroract: a sequence learning OCR system trained on isolated characters, (Greece, 2016), pp. 174–179
25. R Messina, J Louradour, in *Document Analysis and Recognition (ICDAR), 2015 13th International Conference On*. Segmentation-free handwritten Chinese text recognition with LSTM-RNN (IEEE, France, 2015), pp. 171–175
26. A Ray, S Rajeswar, S Chaudhury, in *Advances in Pattern Recognition (ICAPR), 2015 Eighth International Conference On*. Text recognition using deep BLSTM networks (IEEE, India, 2015), pp. 1–6
27. M Akram, S Hussain, in *Proceedings of the 8th Workshop on Asian Language Resources*. Word segmentation for Urdu OCR system (Beijing, 2010), pp. 88–94
28. Q Akram, S Hussain, F Adeeba, S Rehman, M Saeed, in *the Proceedings of Conference on Language and Technology. (CLT 14)*. Framework of Urdu nastalique optical character recognition system (Karachi, 2014)
29. IU Khattak, I Siddiqi, S Khalid, C Djeddi, in *Document Analysis and Recognition (ICDAR), 2015 13th International Conference On*. Recognition of Urdu ligatures-a holistic approach (IEEE, France, 2015), pp. 71–75
30. MW Sagheer, CL He, N Nobile, CY Suen, in *Pattern Recognition (ICPR), 2010 20th International Conference On*. Holistic urdu handwritten word recognition using support vector machine (IEEE, Turkey, 2010), pp. 1900–1903
31. SA Sattar, S Haque, MK Pathan, in *Proceedings of the 46th Annual Southeast Regional Conference on XX*. Nastaliq optical character recognition (ACM, USA, 2008), pp. 329–331
32. R Hussain, HA Khan, I Siddiqi, K Khurshid, A Masood, in *2015 11th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*. Keyword based information retrieval system for Urdu document images (IEEE, Thailand, 2015), pp. 27–33
33. GS Lehal, in *Proceeding of the Workshop on Document Analysis and Recognition*. Choice of recognizable units for Urdu OCR (ACM, India, 2012), pp. 79–85
34. A Bensefia, T Paquet, L Heutte, A writer identification and verification system. Pattern Recognit. Lett. **26**(13), 2080–2092 (2005)
35. I Siddiqi, N Vincent, Text independent writer recognition using redundant writing patterns with contour-based orientation and curvature features. Pattern Recognit. **43**(11), 3853–3865 (2010)
36. CW Ng, S Ranganath, Real-time gesture recognition system and application. Image Vis. Comput. **20**(13), 993–1007 (2002)
37. J Triesch, C von der Malsburg, Classification of hand postures against complex backgrounds using elastic graph matching. Image Vis. Comput. **20**(13), 937–943 (2002)
38. HS Yoon, J Soh, YJ Bae, HS Yang, Hand gesture recognition using combined features of location, angle and velocity. Pattern Recognit. **34**(7), 1491–1501 (2001)
39. XD Huang, Y Ariki, MA Jack, *Hidden Markov Models for Speech Recognition*, vol. 2004. (Edinburgh university press, Edinburgh, 1990)
40. E Kavallieratou, E Stamatatos, N Fakotakis, G Kokkinakis, in *International Conference on Pattern Recognition*. Handwritten character segmentation using transformation-based learning, vol. 15, (Spain, 2000), pp. 63–637
41. B Pardo, W Birmingham, in *Proceeding of the National Conference on Artificial Intelligence*. Modeling form for on-line following of musical performances, vol. 20, (USA, 2005), p. 1018
42. T Plotz, GA Fink, Markov models for offline handwriting recognition: a survey. Int. J. Document Anal. Recognit. (IJDAR). **12**(4), 269–298 (2009)
43. A Khemiri, AK Echi, A Belaid, M Elloumi, in *Document Analysis and Recognition (ICDAR), 2015 13th International Conference On*. Arabic handwritten words offline recognition based on HMMS and DBNS (IEEE, France, 2015), pp. 51–55
44. E Chammas, C Mokbel, L Likforman-Sulem, in *Document Analysis and Recognition (ICDAR), 2015 13th International Conference On*. Arabic handwritten document preprocessing and recognition (IEEE, France, 2015), pp. 451–455
45. M-K Hu, Visual pattern recognition by moment invariants. IRE Trans. Inf. Theory. **8**(2), 179–187 (1962)
46. D Yu, H Yan, Separation of touching handwritten multi-numeral strings based on morphological structural features. Pattern Recognit. **34**(3), 587–599 (2001)
47. A Tahmasbi, F Saki, SB Shokouhi, Classification of benign and malignant masses based on Zernike moments. J. Comput. Biol. Med. **41**(8), 726–735 (2011)
48. F Saki, A Tahmasbi, H Soltanian-Zadeh, SB Shokouhi, Fast opposite weight learning rules with application in breast cancer diagnosis. J. Comput. Biol. Med. **43**(1), 32–41 (2013)
49. GS Lehal, in *Document Analysis and Recognition (ICDAR), 2013 12th International Conference On*. Ligature segmentation for Urdu OCR (IEEE, USA, 2013), pp. 1130–1134

Submit your manuscript to a SpringerOpen journal and benefit from:

- ► Convenient online submission
- ► Rigorous peer review
- ► Open access: articles freely available online
- ► High visibility within the field
- ► Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com