

Urdu Qaeda: Recognition System for Isolated Urdu Characters

Nabeel Shahzad

Sketch Recognition Lab
Texas A&M University
Mail Stop 3112,
College Station, TX 77843
nskhan@cs.tamu.edu

Brandon Paulson

Sketch Recognition Lab
Texas A&M University
Mail Stop 3112,
College Station, TX 77843
bpaulson@cs.tamu.edu

Tracy Hammond

Sketch Recognition Lab
Texas A&M University
Mail Stop 3112,
College Station, TX 77843
hammond@cs.tamu.edu

ABSTRACT

This paper presents an online system for recognizing isolated, hand-sketched Urdu characters drawn on a Tablet PC. Attributes of Urdu characters are analyzed to define a set of features which are then trained and classified using a weighted, linear classifier. As a proof of concept, we have integrated our recognition algorithm into an application used to help people learn the Urdu language. Preliminary results obtained from our studies showed an accuracy of 92.8% for native Urdu writers.

Author Keywords

Urdu, text, sketch recognition, online character recognition.

ACM Classification Keywords

H.5.2 Information Interfaces and Presentation: User Interfaces - *Input devices and strategies*

INTRODUCTION

The recognition of characters from different languages has been an active area of research in sketch recognition. Tael & Hammond developed a recognition system for Chinese characters using the geometric properties of the characters [8]. This system was implemented using the geometric-based sketching language, LADDER [3], combined with Sezgin's low-level recognizer [7]. Such character recognition systems improve man-machine communication while giving the ability to process large volumes of text through computer automated systems. In general, however, the recognition of cursive scripts is a challenging problem because of the variations in the style of writing and the dynamic behavior of characters in cursive form.

Urdu is a cursive script widely used in the Indian sub-

continent and is also written in a similar form as part of the Arabic and Farsi language. Little research has been done to recognize hand-drawn (or even computer-printed) Urdu scripts. To our knowledge, no significant research has been directed towards the online recognition of the Urdu language.

Learning the basic grammar and structure of a language begins first with learning the basic characters of the language. The traditional teaching mechanism of character learning is to first make the students identify each character and memorize its written form through repeated writing of the character. Associating a word with a picture also helps students to memorize the character. This is goal of our proof-of-concept application, Urdu Qaeda.

In this paper we will analyze the geometric and visual properties of Urdu characters and define a set of features which can distinguish one character from another. We will then introduce our algorithm for recognizing hand-sketched Urdu characters. We then conclude our paper by presenting our proof-of-concept application, Urdu Qaeda, which is a system for associating written Urdu characters with pictures. Such a system helps to teach the Urdu language while providing better interactivity and feedback.

PREVIOUS WORK

Text recognition has been an active area of research in the last decade, and researchers have produced solutions with positive results for many non-cursive scripts. However, with cursive scripts it is still considered as an open problem. Researchers have focused mainly on offline, image-based recognition techniques for cursive scripts.

Because the Arabic language shares many similarities in writing with the Urdu language, research in Arabic character recognition provides a starting point to working with Urdu characters. Although recognition of computer-generated Arabic characters (similar to OCR research) has been an area of active research, very little work has been done in online hand-sketched recognition of Arabic characters.

El-Wakil & Shoukry developed an online sketch recognition system of handwritten Arabic characters and achieved an accuracy of 84% [2]. They identified a set of

features which can distinguish between the different Arabic characters and trained a recognizer from a set of sample data. We take a similar approach, but have identified a set of features which are more appropriate in classifying Urdu characters rather than Arabic characters.

El-Shaikh & El-Taweel also developed a system for real-time, handwritten character recognition [1]. The system achieves an accuracy of 98%, but was tested by data provided from a single user. Since the system uses a decision tree for recognition, the thresholds set by the authors may not work well for other users with different writing styles.

Pal & Sarkar developed a system for recognizing printed Urdu script using optical character recognition (OCR) techniques [4]. The accuracy achieved by their system is 97.8%, but it will not handle the messiness and variation of handwritten characters.

The system we propose in this paper is (to our knowledge) the first online, handwritten character recognition system for the Urdu language.

IMPLEMENTATION

In order to achieve the ultimate goal of recognizing handwritten Urdu language, one must first start by recognizing its basic components, Urdu characters. To this end, we first analyze the visual and geometric characteristics of the different Urdu characters, and then define a set of features which are specific to disambiguating Urdu characters. We then define the classification strategy we used to recognize each character.

Urdu Characters

The Urdu language consists of 38 basic characters (Figure 1), each of which has an associated sound. Most are multi-stroke characters and share a common structure. The prominent factor which distinguishes one character from another is the number and position of “dots” and “toyeins” (ٲ). For example, characters like ‘ب’, ‘پ’, ‘ت’, ‘ث’, and ‘ٲ’, are very similar to each other but differ only in the number and positioning of dots. Also, characters such as ‘ٲ’ have a superscript toyein (ٲ) which will differentiate it with other characters.

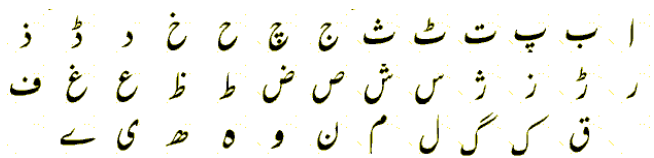


Figure 1. Urdu characters.

Each character has different forms when written in a cursive style. A character can take up to a maximum of four different forms when written in cursive, depending on the positioning of the character in the word. Characters can

have different forms when they are at the start, at the end, or in the middle of the cursive word. Each character can also be written in isolation, which can be considered another different form. Figure 2 shows the four different forms the character ‘Ayen’. Currently, our system does not handle the non-isolated, cursive forms of a character.

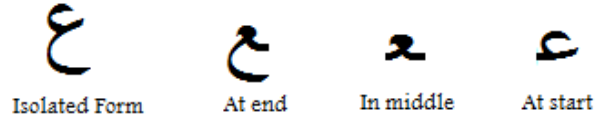


Figure 2. Different forms of the character 'Ayen'.

Another property of the Urdu language is that most characters are multi-stroke (as seen in Figure 3). Typically, each character can be seen as having a single “primary” stroke and zero or more “secondary” strokes. For example, the letter ‘Sheen’ (ش) is a multi-stroke character which consists of four strokes. The primary stroke is ‘س’ with three secondary dot strokes. Figure 4 shows some example characters separated into primary and secondary strokes.



Figure 3. Multi stroke characters in Urdu language.

Feature Set

We want to choose a set of features which takes the multi-stroke nature of many Urdu characters into consideration. For this, we first divide strokes of each character into two categories: 1) primary strokes and 2) secondary strokes. Although the primary stroke is usually written before the secondary strokes, we do not want to restrict the user to this constraint. Therefore, we instead use relative stroke length to distinguish primary from secondary strokes.

Primary Stroke Features

To recognize the different types of primary strokes, we first looked at feature sets of previous works. The features used by Rubine are commonly used to recognize simple, single-stroke gestures [6]. Although the set of features defined by Rubine work well for gesture recognition, we found that some features impede recognition when applied to Urdu characters. We had to remove stroke-time related features mentioned by Rubine because people draw characters at different speeds depending upon their fluency of the written script. Non-native writers can be very slow in drawing characters because of their curvy nature. We also removed the start-angle related features because most Urdu

characters have similar starting angles and these features add to ambiguity in recognition. After carefully analyzing the aspects measured by each feature we empirically selected a subset of Rubine features which are more relevant to Urdu characters. The primary stroke features we used to train our classifier are:

1. The length of the bounding box diagonal.
2. The angle of the bounding box diagonal.
3. The distance between the first and last point.
4. The cosine of the angle between the first and last point.
5. The sine of the angle between the first and last point.
6. The total length of the primary stroke.
7. The total angle traversed.
8. The sum absolute value of angle at each point.
9. The sum of the squared value of those angles.

Character	Primary Stroke	Secondary Stroke
ظ Zoye	ط	•
چ Chay	ح	••
ق Qaaf	و	••
گ Gaaf	د	==

Figure 4. Characters divided into primary and secondary strokes.

Secondary Stroke Features

A set of secondary features are calculated to differentiate between similar characters. The secondary stroke features we chose are:

1. The number of secondary strokes.
2. Total length of secondary strokes.
3. Positioning of the secondary strokes.
4. Number of dots in secondary strokes.

The number of secondary strokes differs from character to character. Some characters like ‘Alif’ (ا) are written with

only a single stroke while characters like ‘Chey’ (چ) can take up to four strokes to be written.

The total length of the secondary strokes also varies among characters. Secondary strokes such as ‘ط’ have significantly greater stroke length than the dots in some other characters.

The positioning of the secondary strokes is also an important feature in distinguishing between different characters. Figure 5 shows characters with the same primary stroke, but which have a different positioning of the secondary strokes. The positioning of the stroke can either be above, inside, or below the primary stroke. Positions are calculated by performing a bounding box overlap function, and numerical values are assigned such that above = 1, inside = 2, and below = 3.

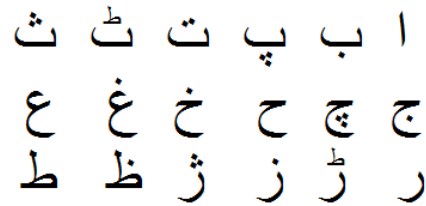


Figure 5. Characters with similar primary strokes but different positioning of secondary strokes.

The number of dots in the secondary strokes also varies between characters. For example, the character ‘Bay’ (ب) is similar to four other characters but differs only in the total number of dots. To recognize dots we use an extension of the low-level, geometric recognizer, PaleoSketch [5], which has recently added a filled-in dot primitive. Dots are recognized by analyzing a “density” feature which is calculated by dividing the total stroke length by the area of the bounding box of the stroke.

Classification

We use all the calculations mentioned above (9 primary + 4 secondary) as our feature set. We use the weighted, linear classifier presented in [6] to perform recognition.

RESULTS

We first conducted a preliminary user study with two participants who could fluently write Urdu characters. The linear classifier was first trained by data provided by the author using five examples of each of the 38 Urdu characters. Testing data was taken from the two participants; four samples of each character. With this testing data, our linear classifier was able to correctly classify 92.8% of characters while the Rubine features alone could only classify 56.7% of the characters correctly.

A separate user study was also conducted with a participant who had never seen or written the Urdu language. Two examples of each character were taken as testing data from the non-native user. The classifier was only able to classify

31% of the characters correctly in this case. The reason for this initial, poorer accuracy was that the user was unfamiliar with the Urdu characters and had drawn characters which sometimes were missing the important features of the characters such as the dots. After explanation and demonstration of the important features of the characters, the user was again asked to write each character at least two times. This time the linear classifier was able to classify 73% of the characters correctly.

DISCUSSION

The preliminary results obtained from the two native participants were encouraging. Figure 6 shows some of the characters that were correctly classified.

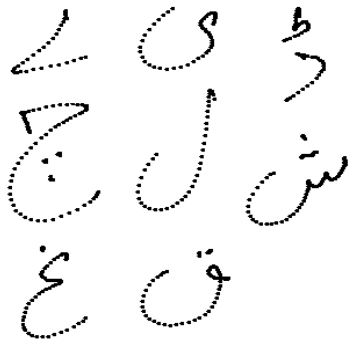


Figure 6. Examples of characters which were correctly classified.

Some characters, however, were not correctly recognized as they shared more similarity in writing. Figure 7 shows two particular characters which were commonly misrecognized. Due to their similarity in written forms humans also misrecognize these letters.



Figure 7. Characters 'Tday' (left) and 'Dhaal' (right) were repeatedly misrecognized.

Another important observation we made was that the non-native participant had significantly lower accuracy rates than the native participants, even after being shown the correct way to draw the characters. We believe the biggest reason for this discrepancy is due to cultural differences in writing. The traditional way of writing Urdu script is from right to left. Western cultures, however, tend to write characters from left to right. This can have a significant

effect on features like the sine/cosine of the angle between start and end points, because the start and end points are now different.

PROOF-OF-CONCEPT APPLICATION

As a proof-of-concept, we have embedded our Urdu character recognizer into an application called Urdu Qaeda. 'Qaeda' is an Arabic word which means a set of rules and regulations. 'Qaeda' is also generally referred to as elementary class books which teach students and children the basic characters of the language.

Urdu Qaeda provides a "training mode" to allow the users to define their own set of training examples for better recognition results. This mode serves the dual purpose of having users repetitively draw examples of each character, similarly to the way in which traditional methods begin teaching a language by having the students continuously draw each character in order to become familiar with its written structure.

Once the application is trained with the set of examples the user can then move to the "learning mode". In this mode, the user can write a single character and have the system recognize the character. The recognized result is shown in pop-up window containing the printed image of the character, a word in Urdu which uses the character. A picture of the meaning of the word and the English translation of the word also accompany (see Figure 8).



Figure 8. Example screenshot of Urdu Qaeda; recognition result for the character 'Sheen'.

FUTURE WORK

We found a number of problems during our preliminary study with the three participants. When the users drew a shape which was not a true character, the recognizer would still classify the input as a character. The system should be able to reject the input as an invalid character so the user could learn and draw the character correctly. To improve on this problem we plan to implement a rejection mechanism so that the input strokes which don't form a character are rejected.

As mentioned in the discussion section, non-native writers may sketch the characters differently from the traditional form. This indicates the possible need of a new set of

features which allows for variations in starting and ending position.

For cursive scripts, the recognition and learning of basic characters is only a very small portion of the understanding of the entire language. In order to achieve a full Urdu language understanding system, much work still exists for recognizing full words or sentences. Unlike other non-cursive scripts, which enable people to write complete words from the basic characters alone, we must also concern ourselves with the multiple character forms found in the Urdu language.

We plan to extend our system to recognize these various cursive forms. This will require a segmentation methodology to segment each word into isolated cursive characters. Our goal is to eventually yield a system that can recognize complete words and can be used in different Urdu language applications.

CONCLUSION

In this paper, we presented a method for recognizing handwritten Urdu characters. We analyzed the defining characteristics and introduced a set of features based on the primary and secondary strokes that make up each Urdu character. The accuracy of our preliminary results was found to be 92.8% for native Urdu writers, while a non-native participant only achieved 73% accuracy. We believe this discrepancy is due to cultural differences in the way in which characters are traditionally written.

ACKNOWLEDGMENTS

This research is supported by the NSF IIS Creative IT Grant #0757557 Pilot: Let Your Notes Come Alive: The SkRUI Classroom Sketchbook.

REFERENCES

1. T.S. El-Sheik and S.G. El-Taweel. Real-time arabic handwritten character recognition. *Pattern Recognition*, 23(12): 1323-1332, 1990.
2. M. El-Wakil and A. Shoukry. On-line recognition of handwritten isolated arabic characters. *Pattern Recognition*, 22(2): 97-105, 1989.
3. T. Hammond and R. Davis. Ladder, a sketching language for user interface developers. *Computers & Graphics*, 29(4): 518-532, 2005.
4. U. Pal and A. Sarkar. Recognition of Printed Urdu Script. In *ICDAR '03: Proceedings of the Seventh International Conference on Document Analysis and Recognition*, pages 1183-1187, Los Alamitos, CA, USA, 2003. IEEE.
5. B. Paulson and T. Hammond. Paleosketch: accurate primitive sketch recognition and beautification. In *IUI '08: Proceedings of the 13th international conference on Intelligent user interfaces*, pages 1-10, New York, NY, USA, 2008. ACM.
6. D. Rubine. Specifying gestures by example. In *SIGGRAPH '91: Proceedings of the 18th annual conference on Computer graphics and interactive techniques*, pages 329-337, New York, NY, USA, 1991. ACM.
7. T.M. Sezgin, T. Stahovich, and R. Davis. Sketch based interfaces: early processing for sketch understanding. In *PUI '01: Proceedings of the 2001 Workshop on Perceptive User Interfaces*, pages 1-8, New York, NY, USA, 2001. ACM.
8. P. Taelle and T. Hammond. Using a geometric-based sketch recognition approach to sketch chinese radicals. In *AAAI '08: Proceedings of the 23rd AAAI Conference on Artificial Intelligence*, pages 1832-1833, Menlo Park, CA, USA, 2008. AAAI.