Online Bangla Word Recognition Using Sub-Stroke Level Features and Hidden Markov Models

Gernot A. Fink, Szilárd Vajda
TU Dortmund University
44221 Dortmund, Germany
{Gernot.Fink,Szilard.Vajda}@udo.edu

Ujjwal Bhattacharya, Swapan K. Parui, Bidyut B. Chaudhuri
Indian Statistical Institute
Kolkata 700108, India
{ujjwal,swapan,bbc}@isical.ac.in

Abstract—For automatic recognition of Bangla script, only a few studies are reported in the literature, which is in contrast to the role of Bangla as one of the world’s major scripts. In this paper we present a new approach to online Bangla handwriting recognition and one of the first to consider cursive written words instead of isolated characters. Our method uses a sub-stroke level feature representation of the script and a writing model based on hidden Markov models. As for the latter an appropriate internal structure is crucial, we investigate different approaches to defining model structures for a highly compositional script like Bangla. In experimental evaluations of a writer independent Bangla word recognition task we show that the use of context-dependent sub-word units achieves quite promising results and significantly outperforms alternatively structured models.

Keywords—online handwriting recognition; Bangla script; sub-stroke level features; hidden Markov models;

I. INTRODUCTION

With the availability of electronic tablets at a cost affordable by common Indians, online handwriting recognition for Indian scripts has gained enough significance. On the other hand, both for standard and portable miniature computing devices (such as palmtop, PDA, mobile phone) equipped with pen-based input technology, non-keyboard based methods for data entry have additional importance particularly in the context of Indian scripts having large symbol sets.

Significant research works on online unconstrained handwriting recognition can be found in the literature (cf., e.g., [1]–[3]). However, the same is not true for Indian scripts. Most of the available online handwriting recognition works on an Indian script deal with isolated character symbols only [4]–[7]. To the best of our knowledge, there exists only one prior work [8] in the literature, which deals with unconstrained cursive online handwriting in Bangla. On the other hand, Bangla is the second most popular language and script in the Indian sub-continent and the fifth most popular of the world. It is also the official language / script of the country Bangladesh, and is used by more than 240 million people around the world.

Automatic recognition systems for unconstrained handwriting can be broadly categorized with respect to the degree of segmentation performed prior to recognition (cf. [9]). In analytic approaches an input word is generally segmented into smaller components such as pseudo-characters, graphemes, or allographs (cf. e.g. [10]). These smaller components are classified first before final recognition of the input word is performed. In contrast, no segmentation is performed in holistic approaches and input words are classified directly (cf. [11]). Consequently, holistic methods can only be applied for tasks with a limited lexicon, which, however, is quite common in automatic handwriting recognition. Recognition methods based on Hidden Markov Models (HMMs) can be considered as hybrids between analytic and holistic techniques, as they follow the analysis-by-synthesis approach. This means that an HMM-based writing model can be constructed analytically from smaller modeling units (e.g. characters). However, a segmentation into these units is not required prior to recognition. Rather, segmentation and classification is performed in an integrated scheme, which led to the term segmentation-free being used for HMM-based approaches (cf. [12]).

In this paper we present an HMM-based approach to online Bangla word recognition. It is based on a sub-stroke level feature representation of the script. For building the writing model we considered different model structures as this is a crucial aspect in HMM-based methods. Especially for a script like Bangla, which has been studied rather little up to now, this choice is far from obvious. We will show that the use of appropriate sub-word units for constructing the writing model is important and that context-dependent sub-word units significantly outperform context-independent ones on our writer independent Bangla word recognition task.

II. RELATED WORK

For online handwriting recognition many different approaches were proposed in the literature and a wide variety of pattern recognition methods were applied (cf., e.g., [2]). Many successful methods are based on HMMs (cf., e.g., [13], [14]) – a technique also widely used in the field of offline handwriting recognition (cf. [15]).

The success of all recognition methods heavily relies on an appropriate pre-processing of the script and the computation of a powerful feature representation. For online handwriting
recognition pre-processing usually comprises normalization of slant, baseline orientation, and writing speed (cf. [16]). The computed feature representations try to capture local shape properties of the pen trajectory together with dynamic information about the writing process (cf., e.g., [17]). The context considered within the pen trajectory can either be defined as a sliding window of fixed size or by pre-segmenting the pen trajectory into stroke-like segments (cf. [18]).

In the earliest available work on segmentation of handwritten cursive Bangla words [19], offline word images were considered. In this work, a recursive contour following approach was proposed. In [20], a certain water reservoir technique was used for segmentation of handwritten Bangla word images, where the “water reservoirs” are considered being the cavities between the different letter segment. The features derived from these notions are the centroid of the reservoirs, the size of the reservoir, etc. A fuzzy feature based segmentation technique for Bangla word images was proposed in [21]. In another related work [22], segmentation of touching characters of printed Bangla and Devanagari scripts were considered. Reports on online isolated Bangla character/numeral recognition are found in [5]–[7]. In an early work on online Bangla handwriting recognition [6], isolated basic characters of Bangla were considered. In [7], an HMM was used for recognition of online handwritten isolated Bangla numerals. A direction code based feature vector was used in [5] for recognition of online handwritten Bangla basic characters.

A similar attempt has been proposed in [4] for recognizing online handwritten Tamil characters. Instead of following the natural course of writing, the authors prefer to match first the different strokes using a flexible string matching algorithm and then a horizontal block is considered to reconstruct the character. As stated in the paper, while this horizontal block is obvious in Tamil, for other Indian scripts the grouping of strokes becomes more complicated.

III. ONLINE BANGLA WORD RECOGNITION

For online Bangla word recognition we adopt the HMM-based approach. It works on a stroke-level feature representation of the script that will be explained in the following. Afterwards, we will describe different approaches to build HMM-based writing models for Bangla script, which differ in the internal structure imposed and the basic modeling units used, respectively.

A. Preprocessing and Sub-Stroke Segmentation

Extraction of sub-strokes from an input character sample is preceded by a sequence of preprocessing operations and these include size normalization, smoothing and re-sampling. We normalize the height of the input word to 100 points and its aspect ratio is kept the same as the original one. Then we apply moving average method to smooth the sample. Finally, it is re-sampled so that the curve distance between two consecutive re-sampled points is 7.0.

In the present approach, we divide each stroke of the pre-processed word sample into several sub-strokes as in [23]. In fact, whenever a stroke has complex arc pattern, it is divided into sub-strokes of simpler shapes. The details are described below.

Suppose the sequence of points in a stroke is $P_0, P_1, P_2, \ldots, P_N$, where $N+1$ is the total number of points in the stroke. The algorithm describing the segmentation process is given in Fig. 1.

We compute the length of each sub-stroke obtained by segmenting a stroke. If the length of such a sub-stroke is less than 5% of the height of the pre-processed sample, then we do not consider the same for further processing. Since we normalize the height of the input sample at 100, each sub-stroke of length less than 5 is discarded.

An example of the sub-strokes obtained for an input sample is shown in Fig. 2.

B. Feature Extraction

From each sub-stroke $S$, 8 scalar feature values representing its shape, size and relative position are computed. Let the sub-stroke $S$ consist of the sequence of $n + 1$ points $P_0, P_1, P_2, \ldots, P_n$ with $l_i$ as the length of the line segment joining $P_{i-1}$ and $P_i$ and $(x_i, y_i)$ be the co-ordinates of $P_i$. The sub-stroke here is represented as a polyline which is a sequence of straight line segments. The length of the sub-stroke $S$ is given by $l = \sum_{k=1}^{n} l_k$. We next find six equidistant
Thus, there are two extreme approaches for defining basic units of writing models for Bangla script. First, the structure of the character shapes can be decomposed to its full extent into elementary units leaving only basic characters and modifiers as elementary models. Second, the compositional structure of the script can be completely preserved. In such a model basic units will be used, which correspond to all potential character shapes that can be built by grouping basic characters into compounds and adding modifier characters to basic or compound character shapes.

The first approach has the obvious advantage of using a rather small inventory of basic units only. Therefore, it can be expected that samples for all basic shapes will be sufficiently available in the training data and robust parameter estimates for the associated models can be obtained. However, such an approach is unable to capture mutual contextual influences between basic shapes that are combined into a more complex character shape. Such local context is more adequately described by the second approach. However, the inventory of basic models will be quite large. Consequently, several of the basic units considered will be quite rarely observed in the training data – a fact which will lead to poor parameter estimates for the associated models.

The dilemma that capturing contextual influence by enlarging modeling units and thus deteriorating parameter estimates can partly be resolved by using context-dependent sub-word units – a technique that was invented and pioneered in the field of automatic speech recognition. There is evidence in the literature that the technique is also beneficial for online handwriting recognition tasks [13], [14] but offers little advantage for offline recognition [24].

For the definition of context-dependent sub-word units for Bangla we started from the fully decomposed model. Units were defined by including one symbol of context to the left and right of every elementary character unit. By introducing context dependence the total number of sub-word units is considerably enlarged. Therefore, appropriate techniques have to be applied that allow the grouping of similar units which share parameters within the model and therefore, can be trained robustly. As in our work on context-dependent modeling for offline recognition we apply a greedy clustering approach to the intermediate state-space defined by the overall writing model without any grouping applied [24]. Compared to alternative methods this technique has the advantage of determining to-be-shared parameter sets within a complex model in a purely data-driven manner not requiring any expert knowledge.

1In order to reflect the special temporal characteristic of some modifiers that sort-of surround the modified character in the appearance of the script (see, e.g., modifier \textit{au} in Fig. 5), we split up the respective models into two basic components.
IV. EVALUATION

In order to evaluate our approach to online Bangla word recognition we performed experiments on a writer independent recognition task.

A. Dataset of Online Bangla Handwritten Words

We used a database of 14,073 Bangla online unconstrained handwritten samples of 50 Indian city names. These samples were written by 163 writers. Each of them wrote 1-4 times each of the 50 city names. A few samples written by these writers are not considered in the database due to improper capture of the signal by the data capturing devices. We used three different devices for collection of the samples. These include WACOM Intuous 2, Genius G-note 7000 and HP tc4400 tablet PC. The writers include different sections of the population with mother tongue Bangla. No restrictions were imposed on the writing style. Every writer wrote several words per page which were automatically segmented into individual words. Each segmented word sample is stored in UNIPEN format including additional information about the writers such as their age, sex, profession, writing skill, writing habit etc. A few samples from this database are shown in Fig. 6.

On this database a writer independence recognition task was defined by splitting the data into disjoint training and test sets. In this task data from 87 writers that provided 7,557 samples of handwritten city names in total are available for model training. Though all the training data is annotated on word-level, detailed manual annotations on sub-character level is available for only a tiny fraction of this data (97 samples, i.e. approximately two samples per city name considered).

The independent test set consists of data provided by the remaining 76 writers (6,516 word samples in total).

B. Recognition System

Our online Bangla word recognition system was built using the Open-Source development environment ESMERALDA [25]. It is based on the sub-stroke level feature representation described in section III-B and uses semi-continuous HMMs just like our previous offline handwriting recognition systems (cf. e.g. [26]).

We compared four different structures of the writing model – the three model variants discussed in section III-C and a holistic model. In the latter a single HMM without any further internal structure corresponds directly to each city name in the lexicon. All the other models apply the analysis-by-synthesis paradigm and, consequently, are based on the use of sub-word units. The most compact model is obtained by using only the most elementary pseudo-characters that define Bangla writing, i.e. basic character shapes and modifiers. For our task we obtain 52 basic units (39 basic characters, 9 atomic and 2 discontinuous modifiers with 2 partial models each).

A considerably larger writing model results from using all combinations of basic characters and modifiers in potential combined character shapes. We obtain 93 basic units in total for the lexicon considered. The largest potential model results from considering each pseudo-character in the context of its left and right neighbor as a distinct unit. However, due to limited variations in the actual contexts present only 345 context-dependent units are required for our task.

All elementary models have Bakis topology (i.e. the skipping of single state in a linear sequence is allowed) and share a codebook of 1k Gaussians with diagonal covariance matrices. The number of model states is automatically determined depending on the length of the respective unit in the initialization data. The parameters of the combined writing model were estimated in two phases. For initialization a codebook was estimated on the training data by applying the k-means algorithm. Using this set of densities, the remaining HMM parameters were initialized on the small set of labelled data. Then joint re-estimation of the model parameters was performed by applying standard Baum-Welch training (20 iterations). Using this first model a forced alignment of the complete set of training data was computed. In the second phase of model training, the above procedure was repeated using the large set of labelled data for initialization purposes. For estimating context dependent
units, the greedy state-clustering in the virtual state space of 1,714 states was performed after the second phase of model training was completed.

C. Experimental Results

The recognition results obtained in our writer independent online Bangla word recognition task are summarized in Tab. 1. Together with the actual recognition rates achieved, the model complexity is given by specifying the number of basic units and the number of independent HMM states within the overall model. As expected, the worst results are achieved by applying the holistic writing model, which treats all word samples separately. Interestingly, the model using all 93 potential combined character shapes as basic units, performs only slightly (but not significantly) better. The best result for context-independent sub-word units is achieved when only the basic character shapes and the modifiers are considered as basic pseudo-character units. However, by differentiating these pseudo-character units depending on their left and right context and defining context-dependent sub-word units, a further significant improvement can be obtained and a quite satisfactory recognition rate of more than 93% is achieved.

V. Conclusion

In this paper we presented a new approach to online Bangla handwritten word recognition. It is based on hidden Markov models, which can be considered the major modeling paradigm used for both online and offline word recognition. Our approach combines a sub-stroke level feature representation of Bangla script and an HMM-based writing model using context-dependent sub-word units. In experimental evaluations on a writer independent online Bangla handwriting recognition task, this model achieved quite promising results and significantly outperformed models using alternative sub-word unit definitions, as, e.g., pseudo-character models.

ACKNOWLEDGMENTS

Part of this work was supported by the German Research Foundation (DFG) within project F1799/3.

The authors of this paper thankfully acknowledge Mr. Tanmay Mondal, CVPR Unit, Indian Statistical Institute, Kolkata for his unconditional help in the preparation of the database and the development of code in C for feature computation.

<table>
<thead>
<tr>
<th>Writing Model</th>
<th>Complexity units</th>
<th>Complexity indep. states</th>
<th>Rec. Rate [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>holistic</td>
<td>50</td>
<td>1,645</td>
<td>88.0</td>
</tr>
<tr>
<td>combined characters</td>
<td>93</td>
<td>876</td>
<td>88.5</td>
</tr>
<tr>
<td>pseudo-characters</td>
<td>52</td>
<td>286</td>
<td>91.0</td>
</tr>
<tr>
<td>context-dependent</td>
<td>293</td>
<td>362 (1,714)</td>
<td>93.1</td>
</tr>
</tbody>
</table>

REFERENCES


