ABSTRACT

Date is a useful information for various application (e.g. date wise document indexing) and automatic extraction of date information involves difficult challenges due to writing styles of different individuals, touching characters and confusion among identification of numerals, punctuation and texts. In this paper, we present a framework for indexing/retrieval of Bangla date patterns from handwritten documents. The method first classifies word components of each text line into month and non-month class using word level feature. Next, non-month words are segmented into individual components and classified into one of text, digit or punctuation. Using this information of word and character level components, the date patterns are searched. First using voting approach and then using regular expression we detect the candidate lines for numeric and semi-numeric date. Dynamic Time Warping (DTW) matching of profile based features is used for classification of month/non-month words. Numerals and punctuations are classified using gradient based feature and SVM classifier. The experiment is performed on Bangla handwritten dataset and the results demonstrate the effectiveness of the proposed system.

1. INTRODUCTION

Date is useful and important information that could be used as key for searching and indexing of handwritten documents in administrative documents, historical archives, postal mails, etc. Some available OCR engines [1] do not work well in understanding handwritten documents. Standard OCR engines are not suitable for such handwritten Bangla documents. The commercial OCRs mainly used in Latin scripts cannot be used for date extracting compilers because of poor recognition result in such documents.

Recently, we proposed a method for date field extraction from handwritten English document [2] and using similar concept this paper deals with Bangla date extraction. To the best of our knowledge, there is no work on Bangla date extraction in printed/handwritten documents.

Date pattern detection and interpretation in handwritten documents is a challenging task due the unconstrained handwriting styles of different individuals. Alpha-numeric characters that represent date are sometimes touching and recognition confusion between numerals and alphabet makes the task more challenging. We have shown two examples of handwritten documents containing Bangla date information in Fig.1. It is to be noted that, the date patterns appear in different format in documents. Some of these formats of a single date are dd/mm/yyyy (e.g. ১৫/০৭/১৯৯০), dd month yyyy (e.g. ১লা বেশি ধুরো,২রা বেশি ধুরো, ৩রা বেশি ধুরো, ৪রা বেশি ধুরো, ৫ই বেশি ধুরো,২২শে বেশি ধুরো, ৬রা জামাইল ১৩৯০, ৭রা ১৩৯০), dd-mm-yy (e.g. ১৬-০৫-০৪) etc. Automatic searching of such different date patterns from the documents is difficult.

![Figure 1. Two examples of Bangla handwritten documents containing date fields. Numeric and semi-numeric date fields are marked with red rectangles.](image-url)
has been applied over the handwritten text lines. Thomas et al. [3] proposed an HMM based classification model for alpha-numerical sequence recognition. Chatelain et al. [5] proposed an approach to locate numerical sequence using a segmentation-driven recognition. To extract the desired numerical sequence, a syntactical analysis has been performed on each line of text. Most of the papers mentioned above deals with alpha-numeric string extraction. This paper moves a step further in document interpretation and uses the recognition labels of alpha-numeric characters of Bangle text to locate the date fields in these documents.

A block diagram of our proposed system is shown in Fig. 2. A three-stage approach has been proposed here for date field extraction. In the first stage, month and non-month handwritten word blocks are separated. For this purpose, Run Length Smoothing Algorithm (RLSA) has been applied on the document image. A component analysis has been done on the RLSA applied image to get the words from the documents. The segmented word blocks are classified into month and non-month classes using word block level feature analysis. The second stage performs component analysis for each non-month handwritten word blocks. Isolated digits, punctuations are identified using component level feature analysis. The components with low recognition confidence are analyzed further for touching segmentation [9]. We have used Dynamic Time Warping (DTW) based word matching with profile based features for month/non-month text identification. Numerals and punctuations are classified using gradient based feature and SVM classifier. Finally, in the third stage, numeric and semi-numeric (contains month field as text string) date patterns are searched from the sequence of the labeled components. To do so, candidate lines are selected first using voting approach and next a regular expression analysis is used to detect the date patterns.

Figure 2. Block diagram of the proposed system

The organization of the rest of this paper is as follows: In Section 2, we have discussed the date component extraction and classification techniques. Section 3 details proposed approach for searching a date pattern. Experimental results and performance analysis are discussed in Section 4. Finally in Section 5, the paper is concluded.

2. COMPONENTS OF DATE FIELD EXTRACTION

2.1 Preprocessing

The handwritten document images are in gray tone and we used global histogram-based Otsu binarization [10] method to convert them into two-tone images (0 and 1). The binarized image may contain noise pixels and irregularities, leading to undesired effects on the system. We smoothed [11] the image to remove some of the above noises. Our date retrieval approach searches the date patterns in text line images. Hence, the binary document is segmented into individual text lines using a line segmentation algorithm [5].

Horizontal Run Length Smoothing Algorithm (RLSA) [6] is applied on each text line to get individual words as a component. A connected component labeling is applied to find the bounding box of the word patches in the line. Next, using these patch masks, the words are segmented from the binary image.

Thus, each text lines of a handwritten document are segmented into separate words based on the inter-word and inter character space analysis. As mentioned earlier, date field in any document contains different components such as digit (numerals used for day, month, year), string (text representing month), contraction symbols (such as ০, ত, ই, ২, ৬), punctuation marks etc. To identify the complete date filed, our approach is to extract these separate components first and later use a regular expression analysis using these components. The process of extraction of different date field components are described in the following subsections. Different component levels (words, digits, punctuations and letters) are later used for date searching.

2.2 Month /Non-Month Identification

To identify the words as month, we have used a conventional profile based feature extraction and classification process. Here, the system performs a classification of two-class problem: month and non-month word blocks identification. To train the classifier, a data set is used with different types of month format that appear in date pattern of Bangla document. For example, “আকাঙ্ক্ষা”, “অষ্টাদশ”, “বিষ্ণু”, “অষ্টাদশ”, etc. The features and matching steps used for this month identification purpose are explained as follows.

Profile features: We have computed our word level features based on word profile. The distance between the lower boundary and the closest foreground pixel is considered here for lower profile and the distance between upper boundary and the closest foreground pixel is considered for upper profile features (See Fig 3). Projection profile and vertical transitions are also considered here as features in word level feature extraction procedure.

Classification of Month words: Dynamic time warping (DTW) [12] based model is used for identifying month/non-month words in this work. DTW is used for measuring similarity between two sequences. The sequences are “warped” non-linearly in the time dimension to determine a measure of their similarity independent of certain non-linear variations in the time dimension. This technique has been widely used in many applications like speech, signatures, robotics etc.
2.3 Digit and Character Component Identification

The words which are classified as non-month in earlier stage are considered here. A connected component analysis is employed to segment the non-month words into different components and component-wise classification is done to extract the character/digit/punctuation components from these non-month words. For this purpose, connected components are fed to component level classification stage. We have used 400 dimensional gradient based feature and Support Vector Machine (SVM) classifier for identification of date components (digit, punctuation etc.).

Feature Extraction and Classification: We compute 400 dimensional gradient based feature [7] of each component images using the following steps.

- The component is normalized into 126x126 sizes and converted to gray-scale image by applying a set of mean-filtering.
- Next the resultant gray image is segmented into 9x9 blocks. Roberts filter is then applied on the image to obtain gradient image.
- The direction of gradient is quantized into 16 directions and the gradient strengths are accumulated with each of the quantized direction.
- Histograms of the values of 16 quantized directions are computed in each of 9x9 blocks.
- Finally, 9x9 blocks are down sampled into 5x5 by a Gaussian filter. Thus, we get 5x5x16 = 400 dimensional feature.

We feed this 400 dimensional gradient feature into a SVM classifier [8] (Gaussian kernel with Radial Basis Function) for classification of character components in a word. By the above classification process, the components are mainly classified into “punctuation”, “digit” or “contraction” level (See Fig. 5). There are some components which might be touching and these components could not be classified properly in this stage. Hence, the components with high recognition confidence are accepted and directly considered for date pattern matching. The rest of the components with low confidence are selected for touching component segmentation analysis. The confidence threshold is selected as 0.5 according to the experimental results. If some isolated characters are not identified properly, the touching character segmentation step (explained in next section) will identify it.

2.4 Touching Character Segmentation

There may exist touching digits/characters in a component. We apply a Connected Component (CC) labeling to the word image and extract individual components. For each component, we
compute the recognition confidence for all character class models using our SVM classification process and rank the confidence scores in descending order. If we recognize a component with a very high accuracy, we assign it as a non-touching character. If the difference between top two recognition scores of a component is high, it is also considered as non-touching character. The rest of the components are considered as touching and we process them for segmentation. Here, we use a dynamic programming based touching character segmentation scheme [9].

First, we find the cavity regions formed between touching characters. The cavity regions are obtained using Water Reservoir concept [6]. We use Top-Bottom reservoir analysis to find the cavity regions in a touching component. A set of candidate segmentation points is obtained from these regions using cavity region analysis. Next, the touching component is segmented into these candidate points to find different sub-images. Using dynamic programming, the recognition confidence of sub-images is analyzed and optimum segmentation path is found. Finally, based on the segmentation lines, the touching component is segmented. This approach segments touching digits in most of the cases. In Fig.6, the circle shows segmentation result of two touching digits $\lambda$ and $\Lambda$.

![Figure 6. A segmentation result of touching digits.](image)

3. SEARCHING OF DATE PATTERN

Text lines with its four different types of recognized components (month, digit, punctuation, text) are considered here for date pattern detection. We have divided this approach into two parts: candidate line selection and date pattern searching.

3.1 Candidate Line Selection

The text lines that contain labeled months, digits and punctuation are selected here. For this purpose, we compute the total number of digits, punctuation marks, month string of a text line. Depending on the value of the individual counters we decide to search the date patterns in that text line in the next stage. Valid date patterns in our experimental data set contain at least 6 elements. If the total number of date elements in a line is greater or equal to 6 then we consider this line as candidate line and used for searching date pattern.

3.2. Date Pattern Matching

The components in each candidate text line are sorted in left to right direction using the CG (Centre of Gravity) and the positions of punctuation, digit and month text are noted. The date patterns are searched next using the sequence of labeled components. In our approach, we consider two different date patterns for searching, namely: numeric and semi-numeric patterns.

3.2.1. Numeric date matching

A date field consisting of only digits and punctuation is considered as numeric date field in our approach, e.g. (২০০৮/৫/৫৮৪, ৫৯/৫, ১৬, ২০০৩/২০১২ etc.). For numeric date extraction we match sub-sequence of components with the following date regular expression:

$$(dd)(/.,\cdot)(dd)(/.,\cdot)(dd)$$

where, $d$ represents digit and we are considering five types of punctuation in the date syntax. A complete numeric date field consists of single digit or double digits date information, single digit or double digit and month information and double digits or four digits year information. In our searching algorithm we first find the position of the two punctuation marks. If we get one or two consecutive digit in the left of left punctuation, one or two digits in the right of right punctuation, we consider this sequence as a valid numeric date field.

3.2.2 Semi-Numeric date matching

Other date fields that consist of textual month, digits and contraction (ো, ঠ, ই, ল, প) are considered as semi-numeric date. (e.g. ২১ আগস্ট, ২০১০, জুলাই ২৬, ২০১০, ১৫ ই (বেসামত, ৮৪১৩). For semi-numeric date field extraction we are searching the following regular expressions:

$$(\text{mdldd})(/.,\cdot)(dd)$$

and

$$(\text{dd})(\text{contraction})(/.,\cdot)(dd)$$

where, $m$ represents a month field. There are two types of sequence for semi-numeric date fields. We find the entire pattern in the sequence of line components for matching with any pattern. In a semi-numeric date pattern textual month information may be in the front or in the middle of the sequence. We are accepting a labeled text as contraction followed by numeric date digits if we find a month field between date field and year fields.

4. RESULT AND DISCUSSION

4.1 Data Set

To the best of our knowledge, there exists no standard database to evaluate date sequence extraction methods. For our experiment, we have collected 1050 Bangla handwritten text lines from different individuals of different profession. The dataset contains date sequence of valid patterns.

To train our classifier for detecting month blocks on handwritten document, we have used 4000 non-month samples and 2520 handwritten Bangla month samples of different months collected from different individuals for month and non-month classification. For the classification of numerals, punctuation and contraction we have considered 3651, 904, and 640 samples, respectively in the present work.

4.2 Results

We have computed precision (P) and recall (R) measure to evaluate the quantitative performance of the system. Depending on the ground truth of the date, extracted sequence is considered to be valid date sequence or not.

Line selection test: 1050 handwritten Bangla text lines are tested for date field spotting. A filtering process is used to remove the lines without date pattern matching. The digits, punctuation and month components are counted in a line. Depending on the value of this counter we are eliminating lines for next level processing. A
complete Bangla date information must have at least 6 elements (examples ৮-৭-১০, ৮ই মাঘ, ১৮৬২) . It is noted that 87.61% (820 out of 1050) lines have been eliminated by keeping the counter to 6. Fig.7 shows the overall result of line selection procedure according to the number of date components.

5. CONCLUSION

In this paper, we have proposed a methodology for date pattern extraction from Bangla handwritten documents for date based indexing. To the best of our knowledge there is no work available which focused on Bangla date pattern extraction. Component labeling and Dynamic Time Warping (DTW) based method have been used here for the month component extraction from handwritten Bangla documents. Gradient based features and Support Vector Machine (SVM) are used for classification of other date elements such as digit, punctuation and contraction (৪, ৫, ৬, ৭, ৮, ৯). Finally, sub-sequence of labeled text lines is matched to validate date with regular expression. We have obtained encouraging result from the experiment.

6. REFERENCES