Hidden Markov Models for Online Handwritten Tamil Word Recognition♦

Bharath A, Sriganesh Madhvanath
HP Laboratories India
HPL-2007-108
July 6, 2007*

hidden Markov models, online handwriting recognition, Tamil word recognition

Hidden Markov Models (HMM) have long been a popular choice for Western cursive handwriting recognition following their success in speech recognition. Even for the recognition of Oriental scripts such as Chinese, Japanese and Korean, Hidden Markov Models are increasingly being used to model substrokes of characters. However, when it comes to Indic script recognition, the published work employing HMMs is limited, and generally focussed on isolated character recognition. In this effort, a data-driven HMM-based online handwritten word recognition system for Tamil, an Indic script, is proposed. The accuracies obtained ranged from 98% to 92.2% with different lexicon sizes (1K to 20K words). These initial results are promising and warrant further research in this direction. The results are also encouraging to explore possibilities for adopting the approach to other Indic scripts as well.

* Internal Accession Date Only
♦ ICDAR'2007, 23-26 September 2007, Curitiba, Brazil

Approved for External Publication

Hidden Markov Models for Online Handwritten Tamil Word Recognition

Bharath A, Sriganesh Madhvanath
Hewlett-Packard Labs India
Bangalore
{bharath.a, srig}@hp.com

Abstract

Hidden Markov Models (HMM) have long been a popular choice for Western cursive handwriting recognition following their success in speech recognition. Even for the recognition of Oriental scripts such as Chinese, Japanese and Korean, Hidden Markov Models are increasingly being used to model substrokes of characters. However, when it comes to Indic script recognition, the published work employing HMMs is limited, and generally focussed on isolated character recognition. In this effort, a data-driven HMM-based online handwritten word recognition system for Tamil, an Indic script, is proposed. The accuracies obtained ranged from 98% to 92.2% with different lexicon sizes (1K to 20K words). These initial results are promising and warrant further research in this direction. The results are also encouraging to explore possibilities for adopting the approach to other Indic scripts as well.

1. Introduction

Tamil, the native language of a southern state in India has several million speakers across the world and is an official language in countries such as Sri Lanka, Malaysia and Singapore. As it is the case with all Indic scripts, Tamil has a large alphabet size and hence text entry through QWERTY keyboard is cumbersome. The penetration of Information Technology (IT) becomes harder in a country such as India where the majority read and write in their native language. Therefore, enabling interaction with computers in the native language and in a natural way such as handwriting, is absolutely necessary.

Indic script recognition poses different challenges when compared to Western, and Chinese, Japanese and Korean (CJK) scripts. When compared to Western scripts, Indic scripts exhibit a large number of classes, stroke order/number variation and two dimensional nature. Indic script recognition also differs from that of CJK in a few significant ways. In the case of CJK scripts, the shape of each stroke in a character is generally a straight line and hence stroke direction based features are often sufficient. But in the case of Indic scripts, the basic strokes are often nonlinear or curved, and hence features that provide more information than just the directional properties are required. Moreover, in CJK scripts, a word is generally written discretely and hence segmenting it into characters is much easier when compared to Indic scripts, where the most common style of writing is run-on. Due to these differences, the techniques employed for other scripts may not be readily applicable for Indic script recognition.

Hidden Markov Models are suitable for handwriting recognition for a number of reasons [3]. Since these are stochastic models, they can cope with noise and variations in the handwriting. The observation sequence that corresponds to features of an input word can be of variable length, and most importantly, word HMMs can solve the problem of segmentation implicitly. In this work, Hidden Markov Models, which are shown to be successful for western cursive recognition, and CJK script recognition to some extent, are applied to model Tamil words.

The remainder of the paper is organized as follows: Section 2 briefly reviews the prior work on online recognition of Tamil characters. Section 3 introduces the Tamil script, and the symbols we have used for word recognition. Section 4 describes the preprocessing and feature extraction stages of the system proposed. Tamil word modelling using HMMs and the dataset used for our investigation are explained in Sections 5 and 6. The results of our experiment are tabulated in Section 7 and finally, our future directions and some conclusions are mentioned in Section 8.

2. Literature Review

Even though there have been a few efforts in online Tamil character recognition, to the best of our knowledge, there is no published work on online recognition of handwritten Tamil words. In [12], the problem of high interclass similarity in the case of Tamil characters is addressed by finding appropriate features. Angle features, Fourier co-

efficients and Wavelet features are compared using a Neural Network classifier. In the absence of smoothing, angle features are susceptible to noise and may fail to capture the intra-class similarity. Fourier coefficients do not capture subtle differences between two similar-looking characters because a change in the values of x and y over a small interval of time gets nullified over the entire frequency domain. On the other hand, Wavelet features are shown to retain the intra-class similarity and inter-class differences, resulting in high recognition accuracy. A prototype-based approach using Dynamic Time Warping (DTW) is described in [10]. DTW distance is computed for both creating prototypes using agglomerative hierarchical clustering and testing. The work also proposes several rejection schemes for the DTW-based classifier.

The work published in [7] aims at writer-dependent recognition. Features such as normalized x-y, quantized slopes and dominant points (points of high change in writing angle) are compared to arrive at hybrid schemes (two-stage classification) to address the time-complexity involved in plain DTW matching. For instance, short-listing prototypes based on Euclidean distance in the first stage followed by DTW matching in the second stage is shown to perform well both in terms of recognition accuracy and time. In [5, 8] a subspace-based method using Principal Component Analysis (PCA) is applied for Tamil character recognition. Each class is modelled as a subspace, and for classification, the orthogonal distance of the test sample to the subspace of each class is computed. The effort published in [8] compares the performance of DTW and PCA for three modes of recognition: writer independent, writer dependent and writer adaptive. DTW is shown to outperform PCA in all the three modes of recognition. The work also proposes a classifier combination scheme for the two methods.

In [1] a generalized framework for Indic script character recognition is proposed and Tamil character recognition is discussed as a special case. Unique strokes in the script are manually identified and each stroke is represented as a string of shape features. The test stroke is compared with the database of such strings using the proposed flexible string matching algorithm. The sequence of stroke labels is then converted into "horizontal block" using a rule list and the sequence of horizontal blocks is recognized as a character (with its IISCI code) using a Finite State Automaton (FSA).

3. Tamil Script

Tamil script belongs to the family of syllabic alphabets [4] and consists of symbols for vowels and consonants. Each consonant has an implicit vowel which can be modified to another vowel by using special diacritical marks known as *matras*. A consonant can also be changed to its half form using the vowel-muting diacritic which eliminates the implicit vowel sound. A consonant and a vowel combine to give a composite character, which is referred to here as a *syllabic unit*. The constituents of a syllabic unit i.e, vowels, consonants or matras are loosely called *symbols* in this paper. Matras in Indic scripts can occur at several locations around the base consonant resulting in a two-dimensional nature much like CJK scripts. Figure 1 shows the set of symbols present in Tamil, and these form the basic building blocks of our recognition system. Symbols 0 to 10 correspond to vowels, 11 (*aytham*) is a special symbol, 12 to 33 correspond to consonants with implicit vowel sound, and symbols 72 to 80 correspond to matras. A consonant gets converted to its half form when symbol 72 (vowel-muting diacritic) is placed above it. Symbol 81 which always occurs with 75, and symbol 82 are compound characters (also known as conjuncts) formed as a result of combining a half-consonant, a consonant and a matra, and symbol 83 corresponds to the period. The rest of the symbols are distinct syllabic units formed by consonant-vowel combination where both the consonant and the matra lose their individual identities, and hence are best represented as unique symbols. A word in Tamil is normally written as a sequence of syllabic units one after another from left to right. In this paper, an HMM-based approach is proposed for writer-independent recognition of online handwritten Tamil word by considering the symbols described above as the fundamental units for recognition.

Figure 1. Tamil symbols for word recognition

4. Preprocessing and Feature Extraction

Preprocessing of captured ink involves two steps: noise elimination and normalization. The noise elimination in the system involves *removal of duplicate points* and *smoothing*. Duplicate points (successive points that have identical values of x and y) are redundant and do not contain any information. Hence these were removed from the captured ink before processing further. Smoothing of strokes is required to remove any noise in the trajectory due to erratic pen motion. A moving average filter of window size three was used for smoothing in our experiment. Normalization is required to compensate for the size, slant and rotation of the captured ink so that the patterns become comparable. As we did not notice any slant or rotation in the data samples, only *size normalization* was carried out. Normalizing size requires estimation of lower and upper core lines. To determine these reference lines, the mean value of y is first computed and then the strokes that intersect with the line $y = y_{mean}$ are identified. The mean values of y for the lower most and upper most points in the intersecting strokes correspond to the lower and upper core lines respectively. Figure 2 shows the estimated core lines for a word sample. To achieve size normalization, the distance between the two lines was fixed to 100 while retaining the aspect ratio. Once

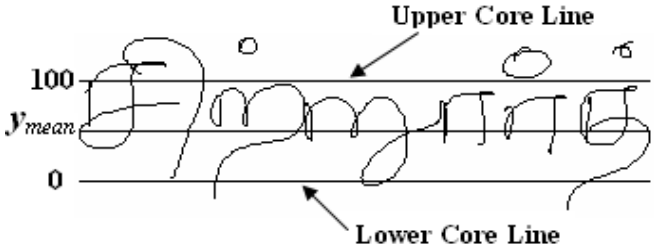

Figure 2. Reference lines and normalized word

the raw ink was preprocessed, it was passed to the feature extraction stage. The features employed for recognition are described below.

(i) Normalized Y - Once the input word size is normalized, the y value corresponds to the vertical position of each point with respect to the lower core line. The points below the lower core line have negative y values. The vertical component (y) also helps capture the relative position between symbols. For instance, both symbols 72 and 83 in Figure 1 are identical shapes representing dots, varying only in their vertical position in a word. Symbol 72 is a matra found above the upper core line, whereas symbol 83 is a period and is expected close to the lower core line.

(ii) Normalized Derivatives - The normalized derivatives proposed in [11] are shown to perform better than equidistant resampling. Normalized first and second derivatives capture the speed of direction change but lose the speed information, making it suitable for writer-independent recognition.

(iii) Angle Features - Angle features are widely used in word recognition systems due to their translation and scale invariant nature. The angle features employed in the system capture the writing direction and curvature of the trajectory as described in [6]. The writing direction was represented using the cosine and sine of the angle subtended by the line segment joining the neighboring points on either side with the horizontal line. The curvature at a point was represented by the cosine and sine of the angle formed by the line segments joining the point of consideration and its second neighboring points on either sides.

(iv) Pen-up/Pen-down Bit - In this system, every pen-up stroke was resampled to ten points by linear interpolation in order to simulate the continuous time varying nature of the signal. Pen-up/Pen-down bit is a binary-valued feature indicating whether a stroke is a pen-up stroke (value set to 1) or a pen-down stroke (value set to 0).

5. Word Modelling

The preprocessing and feature extraction stages of the input handwriting signal were explained in Section 4. In this section, building of word models using HMMs is explained in detail.

5.1. Symbol Modelling

The features extracted from the symbols were used to train a continuous density HMM for each symbol. For modelling a symbol using HMM, a simple left-to-right topology with no state skipping was adopted and the training was carried out using the Baum-Welch re-estimation procedure. The number of states per model was determined based on the shape complexity of the symbol and this has been shown to model the symbols better than having a fixed number of states for each symbol. The number of states was computed as a fraction of average length of the training observation sequences of the symbol. The fraction was empirically determined as 0.2, and similarly the number of Gaussians per state was set to two.

5.2. Pen-Up Stroke Modelling

The pen-up strokes within a symbol were implicitly modelled using the symbol models whereas the pen-up strokes between symbols were modelled explicitly. Once the annotated training word samples were normalized, the pen-up strokes between symbols in the word samples were extracted. Since the word samples do not contain all pairs of symbols occurring together, the possible pen-up strokes between a chosen symbol and the rest are not known. Therefore, common pen-up stroke models were built which were shared between any pair of symbols. These common pen-up stroke models were determined by clustering the inter-symbol pen-up strokes obtained from the word samples. The clustering was done by assigning each pen-up stroke to one of the eight directions shown in Figure 3. The samples falling into each cluster are used to train a two-state left-to-right *pen-up HMM* having 2 Gaussians per state. For

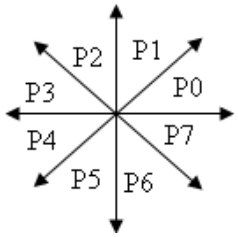

Figure 3. Inter-symbol pen-up strokes grouped based on writing direction

a given word in the lexicon, its word model was built by concatenating the constituent symbol models and having the parallel network of pen-up models inserted in between them as shown in Figure 4. A lexicon was represented as a network of word HMMs where each path in the network from the start node to the final node corresponds to a word. During evaluation, the best path in the network was determined by the standard Viterbi decoding.

6. Dataset Description

The word samples for this experiment were collected using an HP Tablet PC which has a sampling rate of 1200 points per second. The list of words used for data collection was selected from a text corpus by an Optimal Text Selection (OTS) program which applies the Set Cover Algorithm presented in [9] to identify a minimal set of words covering all the 84 symbols. The majority of the writers who participated in the data collection activity had Tamil as their native language and their profession involved writing in Tamil everyday. Totally 132 writers belonging to different age groups contributed their handwriting samples.

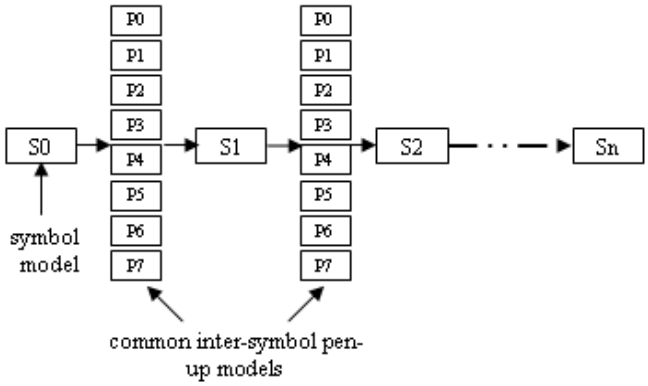

Figure 4. Word network with explicit inter-symbol pen-up models

Each writer provided two samples of 30 words, out of the 80 words selected by the OTS. The collected word samples were manually annotated at the symbol level by following the annotation process described in [2]. The annotated ink files were stored in UNIPEN format along with the truth of each sample. The dataset was then split into train and test sets. The train set consisted of word samples written by 112 writers (6,252 samples) and the remaining data (981 samples) written by 20 writers was used for evaluation. Since the approach aims at writer-independent recognition, samples of the same writer were not present in both the train and the test set.

7. Experimental Evaluation

The evaluation of the word recognition system was carried out on different lexicon sizes such as 1K, 2K, 5K, 10K and 20K words to assess the performance of the word models in terms of recognition accuracy. The lexicons were created from the EMILLE-CIIL text corpus that contain news articles on politics, sports, current affairs and cinema. The words extracted from the corpus were sorted based on their frequency of occurrence in the corpus. For instance, the lexicon of size 1K contained the most frequently occurring thousand words in the text corpus. Since the words in the corpus were in Unicode encoding, they had to be converted into the sequence of symbol IDs defined. Even though a Tamil word is normally written as a sequence of syllabic units, the writing order of symbols within a syllabic unit may change with writers. However, from the experience of data collection and by manual inspection of a few collected samples, it was observed that the majority write the base consonant first and then the matra (if any), except for matras 78, 79 and 80. These matras are written before writing the base consonant for two reasons: (i) these matras

Table 1. Accuracy of the system on different lexicon sizes

Lexicon Size	Accuracy % (Zero Rejection)
1k	97.96
2k	95.82
5k	94.49
10k	93.17
20k	92.15

are horizontally separate from the base consonant and occur on the left of the consonant and (ii) writing them after the base consonant will considerably interrupt the flow of writing. These facts were taken into account while determining the expected order of symbols for any given Unicode string. Manual inspection of the samples also revealed that handwritten Tamil words rarely suffer from the problem of delayed strokes when compared to western cursive writing. This alleviates the need to capture delayed strokes in the word model. During evaluation, it is ensured that the truth of an input test sample is always present in the lexicon in the expected order of symbols. Table 1 shows the accuracy of the system on different lexicon sizes. Relatively low accuracy in the case of 10K and 20K can be attributed to higher perplexity involved in recognition, and thus provides interesting directions for future research.

8. Conclusions and Future Work

In this work, a writer-independent online handwritten Tamil word recognition system that employs HMMs for word modelling was discussed. A symbol set consisting of 84 symbols was defined for the word recognition task. Each symbol was modelled using a left-to-right HMM. Inter-symbol pen-up strokes were modelled explicitly using two-state left-to-right HMMs to capture the relative positions between symbols in the word context. Since one cannot expect the training word samples to contain all pairs of symbols occurring together, the inter-symbol pen-up models were shared between any two symbols. Independently built symbol models and inter-symbol pen-up stroke models were concatenated to form the word models.

There are several possible improvements to the system. The relatively low performance in the case of high lexicon size can be improved by the use of statistical language models, which are commonly applied in Western cursive recognition. Even though real-time performance was not our objective, the response time for 10K and 20K was found to be more than 3 seconds on a machine with 256MB RAM and Pentium 4 processor, making it unsuitable for real-time applications. A Trie representation of the word network may be implemented instead of the linear list to improve the response time. When the confusion matrix of recognition was examined, a substantial number of confusions were between symbol 76 and 75, and 76 and 77. The distinction between these symbols is less evident usually and hence specific features that would help discriminate them are necessary, which will be another research direction for the future.

References

[1] H. Aparna, V. Subramanian, Kasirajan, V. Prakash, V. Chakravarthy, and S. Madhvanath. Online Handwriting Recognition for Tamil. *Proceedings of the 9th International Workshop on Frontiers in Handwriting Recognition*, 2004.

[2] A. S. Bhaskarabhatla and S. Madhvanath. Experiences in Collection of Handwriting Data for Online Handwriting Recognition in Indic Scripts. *Proceedings of the 4th International Conference Linguistic Resources and Evaluation*, 2004.

[3] H. Bunke, M. Roth, and E. G. Schukat-Talamazzini. Offline Cursive Handwriting Recognition using Hidden Markov Models. *Pattern Recognition*, 28(9):1399–1413, 1995.

[4] F. Coulmas. *The Blackwell Encyclopedia of Writing Systems*. Blackwell, Oxford, 1996.

[5] V. Deepu and S. Madhvanath. Principal Component Analysis for Online Handwritten Character Recognition. *Proceedings of the 17th International Conference on Pattern Recognition*, 2004.

[6] S. Jaeger, S. Manke, J. Reichert, and A. Waibel. Online Handwriting Recognition: The NPen++ Recognizer. *International Journal on Document Analysis and Recognition*, 3:169–180, 2001.

[7] N. Joshi, G. Sita, A. G. Ramakrishnan, and S. Madhvanath. Comparison of Elastic Matching Algorithms for Online Tamil Handwritten Character Recognition. *Proceedings of the 9th International Workshop on Frontiers in Handwriting Recognition*, 2004.

[8] N. Joshi, G. Sita, A. G. Ramakrishnan, and S. Madhvanath. Tamil Handwriting Recognition Using Subspace and DTW Based Classifiers. *Proceedings of the 11th International Conference on Neural Information Processing*, 2004.

[9] B. Kalika, A. G. Ramakrishnan, P. P. Talukdar, and N. S. Krishna. Tools for the Development of a Hindi Speech Synthesis System. *5th ISCA Speech Synthesis Workshop*, June 2004.

[10] R. Niels and L. Vuurpijl. Dynamic TimeWarping Applied to Tamil Character Recognition. *Proceedings of the 8th International Conference on Document Analysis and Recognition*, 2005.

[11] M. Pastor, A. Toselli, and E. Vidal. Writing Speed Normalization for On-Line Handwritten Text Recognition. *Proceedings of the 8th International Conference on Document Analysis and Recognition*, pages 1131–1135, 2005.

[12] C. S. Sundaresan and S. S. Keerthi. A Study of Representations for Pen based Handwriting Recognition of Tamil Characters. *Proceedings of the 5th International Conference on Document Analysis and Recognition*, 1999.