Online handwritten Tamil word recognition using segmentation, bigram models and verification

A G Ramakrishnan and Bhargava Urala
MILE Laboratory, Dept. of Electrical Engineering, Indian Institute of Science, Bangalore

The term 'online' refers to the fact that the handwritten data is a series of (x, y) co-ordinates captured by a digitizer (Tablet PC) using a pen-like interface. There are 'pen-down' and 'pen-up' signals, which can be used to separate the captured online data into strokes. The co-ordinates are stored in chronological order as opposed to 'offline' handwritten data, which has the nature of an image with no chronological order of points available.

Preprocessing

This consists of 3 steps: (i) Smoothing – reduces the amount of high frequency noise in the input resulting from the capturing device or jitters in writing; (ii) Normalising - eliminates variability due to size differences; (iii) Resampling - obtains a constant numberof data points, and makes the data independent of local and global variations in writing speed. Figure 1 (a) shows a raw character and (b), after it is preprocessed.

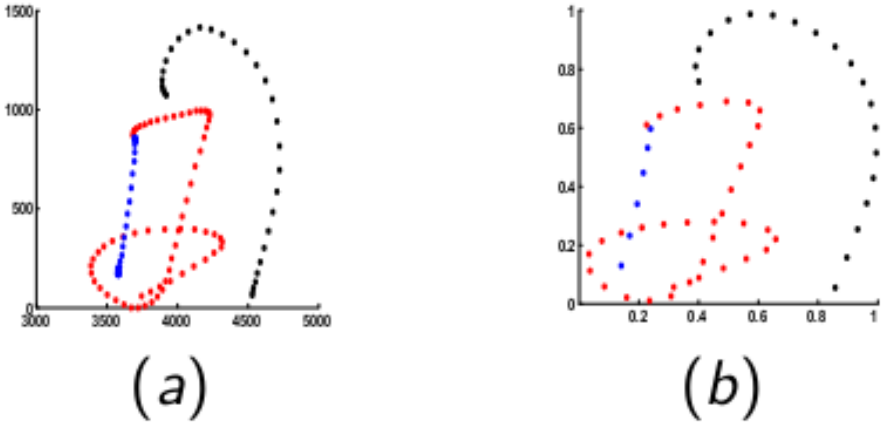

Fig. 1. (a) Original acquired sample of /ki/. (b) its preprocessed version.

Segmentation

The segmentation of individual Tamil symbols from the words (a sequence of strokes) is accomplished by two successive steps – dominant overlap criterion segmentation (DOCS) and attention-feedback segmentation (AFS) .

The DOCS module segments using the degree of horizontal overlap between bounding boxes of consecutive strokes. If the measured overlap is greater than an experimentally determined appropriate hreshold, the strokes are merged to be part of the same stroke group; otherwise, they are split into different stroke groups.

The horizontal overlap between successive strokes is determined as the maximum of the two ratios obtained by dividing the x-overlap by the x-ranges of each stroke separately. The overlap is obtained as $x_{max}^{S_k} - x_{min}^{c}$, whereas the x-range of S_k and c are obtained as $x_{max}^{S_k} - x_{min}^{S_k}$ and $x_{max}^{c} - x_{min}^{c}$, respectively, where x_{max}^{Sk}, $x_{min}^{Sk,}$ x_{max}^{c}, x_{min}^{c} denote the mamimum and minimum x-values of the previous stroke group and the current stroke, respectively. In this method, depending on the way the characters are written in a word, cases of oversegmentation (i.e. a stroke group being a part of a valid symbol) and undersegmentation (i.e a stroke group being a merger of two or more valid symbols) can arise occsionally. Figure 2 (a), (b) and (c) show one example each of correct segmentation, oversegmentation and undersegmentation, respectively performed by dominant overlap criterion segmentation module.

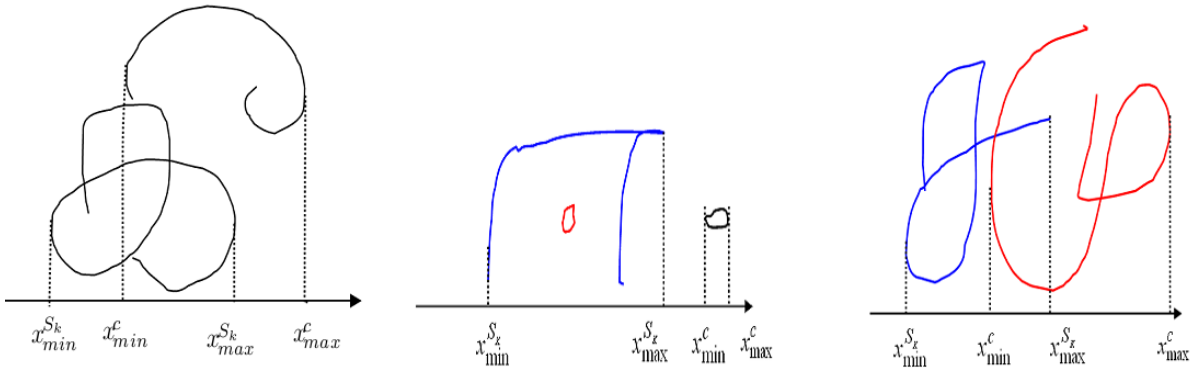

Fig. 2. Correct segmentation, over-segmentation and under-segmentation by DOCS.

Possible segmentation errors are detected by paying attention to specific features such as pen displacement, bounding box to stroke displacement and the number of dominant points (the minimum number of points to be retained to represent the character's shape) . After detecting these errors, recognition likelihood from the main SVM (support vector machine) classifier and certain statistical features measured over large data such as inter-stroke displacement are used as feedback to correct the segmentation. This is known as attention-feedback segmentation (AFS) . The overall AFS scheme is shown in Fig. 3.

Main Classifier

Support Vector Machine with a radial basis function kernel is used as main classifier. It is trained on preprocessed (x, y) coordinates of the data from IWFHR database.

Bigram Models

Bigram statistics are generated from a large Tamil text corpus (Emille-CIIL corpus and MILE OCR corpus), which consists of about 14 million words. The Unicode sequence of each word is converted to class label sequence and the following statistics are generated:

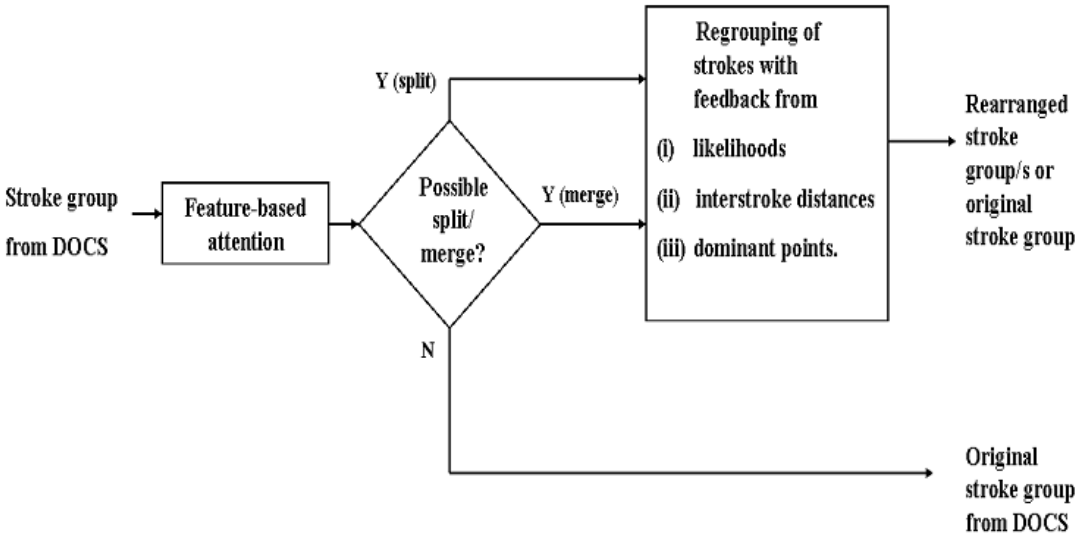

Fig. 3. Principle behind attention-feedback segmentation scheme.

N_T: Total count of all the words in the entire text corpus.

N_1 (s_i): Count of number of occurrences of the single symbol s_i in the corpus.

N_2 (s_1, s_2): Count of joint occurrences of the two symbols s_1 and s_2 contiguously.

Any handwritten word W can be considered to be a first-order Markov process, where any symbol depends only on the previous symbol. Joint symbol (bigram) probabilities can be obtained as,

$$P(s_i|s_{i-1}) = N_2\ (s_{i-1}, s_i)\ /N_1\ (s_{i-1})$$

Probabilities of a symbol being at the start or end of a word are obtained as,

$$P_b(s_i) = N_2\ (b, s_i)\ /N_T$$

$$P_e(s_i) = N_2\ (s_i, b)\ /N_T$$

where b denotes 'space'. In order to effectively use bigram models, we consider the 3 class labels with the topmost SVM recognition scores for every symbol in the given word X. Let W represent the set of all possible words. The most likely symbol sequence W^* to represent the unknown word is obtained as that which maximizes P (W|X) .

Using Bayes' rule we get, $W^* = arg\ \max_W\{P(X|W)P\ (W)\ |P\ (X)\ \}$

where, $P(X|W) = \prod_{j=1}^{k} P\ (x^{S_j}|w_j)$ represents the likelihood of the handwritten word given by the SVM and P (W) is obtained from bigram statistics previously computed. Neglecting P (X), the equation can be rewritten in terms of logarithms as follows.

$$W^* = arg \max_W \{\log P(X|W) + logP\ (W)\ \}$$

The most likely sequence W^* is obtained by backtracing the lowest cost path using the Viterbi algorithm.

Lexicon based correction

A lexicon is constructed by extracting all the unique words from the corpus of Tamil text. Total size of the lexicon is 2.4 lakh words, which is divided into several smaller lexicons, depending upon the number of symbols contained in each word.

In this postprocessing, the number of symbols k in the input word is noted down, as reported by the attention-feedback module. Then the class label sequence of the input word is obtained using the main classifier. The lexicon Lk, containing all words with k number of symbols, is loaded into the memory. Using Levenshtein distance metric, the distance di between the recognized sequence of symbols and every word Wpi in the lexicon is computed. If the distance to any entry in the lexicon is zero, it represents an exact match and the lexicon search is terminated. Otherwise, the minimum distance dm = min (di) over all the entries of the lexicon Lk is found out, and the corresponding lexicon word is taken as the recognized word.

Verification

In Tamil script, there are many pairs or sets of symbols that are visually similar, which are often confused by the main classifier. Some of these symbol sets are /mu/ and /zhu/; /na/, /La/ and /ai/; /ni/ and /Li/; /ki/ and /chi/; /la/ and /va/. In order to disambiguate between these frequently confused sets, an expert classifier is trained on the discriminating features between the set of confused symbols we use a technique called reevaluation where. Every time the AFS module assigns a class label corresponding to any one of the confused pair symbols we use the expert classifier to decide its final class label. We have examined and chosen 6 possible confusion pairs in Tamil.

Results

To see the real potential of the bigram language models and lexicon on the recognition accuracy, 2000 words with quality lables B and C are taken. These are relatively badly written words, with unusual number of strokes, unexpected overlaps, etc. The raw symbol recognition accuracy is 75.4% and the word recognition accuracy is 34.9%. With bigram models, these numbers improve to 78.8% and 42.5%.The accuracies using the lexicon are 74.1% and 46.6%, respectively. It was analyzed that the limited improvement obtained by the use of lexicon is due to the fact that nearly 25% of the test words were not found in the 2.4 lakh vocabulary, due to the morphological richness of Tamil language. On the other hand, only by using verification of 5 confusion sets, the symbol and word recognition accuracies improved to 76.9% and 45.1%, respectively. A combination of bigram models, verification and lexicon based postprocessing could lead to a more significant improvement.