2009 10th International Conference on Document Analysis and Recognition

HMM-Based Handwritten Amharic Word Recognition with Feature Concatenation

Yaregal Assabie and Josef Bigun
School of Information Science, Computer and Electrical Engineering
Halmstad University, Halmstad, Sweden
{*yaregal.assabie, josef.bigun*}*@hh.se*

Abstract

Amharic is the official language of Ethiopia and uses Ethiopic script for writing. In this paper, we present writer-independent HMM-based Amharic word recognition for offline handwritten text. The underlying units of the recognition system are a set of primitive strokes whose combinations form handwritten Ethiopic characters. For each character, possibly occurring sequences of primitive strokes and their spatial relationships, collectively termed as primitive structural features, are stored as feature list. Hidden Markov models for Amharic words are trained with such sequences of structural features of characters constituting words. The recognition phase does not require segmentation of characters but only requires text line detection and extraction of structural features in each text line. Text lines and primitive structural features are extracted by making use of direction field tensor. The performance of the recognition system is tested by a database of unconstrained handwritten documents collected from various sources.

1. Introduction

Amharic is the official language of Ethiopia which has a population of over 80 million at present. It belongs to Afro-Asiatic language family, and today it has become the second most widely spoken Semitic language in the world, next to Arabic. Along with several other Ethiopian languages, Amharic uses Ethiopic script for writing. The Ethiopic script used by Amharic has 265 characters including 27 labialized (characters mostly representing two sounds, e.g. ሷ for ስዋ) and 34 base characters with six orders representing derived vocal sounds of the base character. The alphabet is written in a tabular format of seven columns where the first column represents the base characters and others represent their derived vocal sounds. Part of the alphabet is shown in **Table 1**.

Table 1. A sample of handwritten Ethiopic characters.

	Base Sound	Orders						
		1st (ä)	2nd (u)	3rd (i)	4th (a)	5th (e)	6th (ə)	7th (o)
1	h							
2	l							
3	h							
4	m							
5	s							
·	·	·	·	·	·	·	·	·
·	·	·	·	·	·	·	·	·
·	·	·	·	·	·	·	·	·
32	f							
33	p							
34	v							

Automatic recognition of unconstrained handwritten text is one of the most challenging pattern recognition problems due to the varying nature of the data. With respect to this inherent nature, various recognition methods have been proposed. Offline recognition of Latin, Chinese, Indian, and Arabic handwritten text has long been an area of active research and development [3], [4]. However, Ethiopic handwriting recognition in general, and Amharic word recognition in particular, is one of the least investigated problems. The difficulty in automatic recognition of Ethiopic script arises from the relatively large number of characters, their interclass similarity and structural complexity. In this paper, we present Amharic word recognition in unconstrained handwritten text using hidden Markov model (HMM).

978-0-7695-3725-2/09 $25.00 © 2009 IEEE
DOI 10.1109/ICDAR.2009.50

IEEE computer society

2. Description of the Recognition System

Originally applied to the domain of speech recognition, HMMs have emerged as a powerful paradigm for modeling pattern sequences in different areas such as online handwriting recognition. Inspired by the success in such fields, they have attracted a growing interest more recently in various computer vision applications including offline handwriting recognition [3], [4], [5].

2.1. Theoretical background

HMMs are doubly stochastic processes which model time varying dynamic patterns. The system being modeled is assumed to be a Markov process that is hidden (not observable), but can be observed through another stochastic process that produces the sequence of observations. The hidden process consists of a set of states connected to each other by transitions with probabilities, while the observed process consists of a set of outputs or observations, each of which may be emitted by states according to some output probability density function. HMMs are characterized by the following parameters [5]:

- **N**, the number of states in the model. Individual states are denoted as $S = \{S_1, S_2, \ldots, S_N\}$, where the state at time t is denoted as q_t .
- **M**, the number of distinct observation symbols per state, denoted as $V = \{v_1, v_2, \ldots, v_M\}$.
- **A**=$\{a_{ij}\}$, the state transition probability distribution where $a_{ij} = P[q_{t+1} = S_j | q_t = S_i]$, $1<i, j\leq N$.
- **B**=$\{b_j(k)\}$, the observation symbol probability distribution in state j, where $b_j(k)= P[v_k \text{ at } t \mid q_t = S_j]$, $1\leq j\leq N$, $1\leq k \leq M$.
- $\boldsymbol{\pi}=\{\boldsymbol{\pi}_i\}$, the initial state distribution, where $\boldsymbol{\pi}_i=P[q_1=S_i]$, $1\leq i\leq N$.

The above HMM is represented by a compact notation: $\lambda=\{\mathbf{A},\mathbf{B},\boldsymbol{\pi}\}$. Parameters of HMMs are estimated from training data samples that they intend to model. The maximum likelihood parameter estimation can obtained by the iterative procedure with multiple observation sequences. The *Bauch-Welch algorithm* is one of the powerful tools to estimate parameters with probabilistic approach. The most likely sequence of hidden states corresponding to a sequence of observations are also obtained by making use of the well known *Viterbi algorithm*.

2.2. Feature design

The traditional way of using HMMs for handwritten word recognition is by concatenation of HMMs of characters constituting the word. Input features are usually extracted by a moving a sliding window from left to right to produce a sequence of observations. When sliding the window a set of features are extracted after passing through image normalization procedures. In our proposed system the features vectors are computed from the structural features, i.e. primitive strokes and their spatial relationships, which are extracted in a sequential order based their spatial arrangements. Primitive strokes are formed from vertical and diagonal lines and end points of horizontal lines, whereas connectors are defined as horizontal lines between two primitives. Spatial relationship refers to the way two primitives are connected to each other with horizontal lines. Primitives are further classified hierarchically based on their orientation or structure, relative length with in the character, and relative spatial position. This classification scheme results in 15 types of primitives, which is summarized in **Table 2**, with numbers in brackets showing feature values. Thus a primitive stroke is represented by three feature values.

Table 2. Classification of primitive strokes.

Orientation/ Structure	Length	Position	Example Character
Vertical (8)	Long (9)	Top-to-bottom (8)	
	Medium (8)	Top (9)	
		Bottom (7)	
	Short (7)	Middle (6)	
Forward Slash (9)	Long (9)	Top-to-bottom (8)	
	Medium (8)	Top (9)	
		Bottom (7)	
	Short (7)	Middle (6)	
Backslash (7)	Long (9)	Top-to-bottom (8)	
	Medium (8)	Top (9)	
		Bottom (7)	
	Short (7)	Middle (6)	
Appendage (6)	Short (7)	Top (9)	
		Middle (6)	
		Bottom (7)	

A primitive can be connected to another at one or more of the following regions: *top* (1), *middle* (2), and *bottom* (3). A connection between two primitives is represented by ***xy*** where ***x*** and ***y*** are numbers representing connection regions for the *left* and *right* primitives, respectively. Between two primitives, there can also be two or three connections, and a total of 18 spatial relationships are identified, which are listed as: 11 (), 12 (), 13 (), 21 (), 22 (), 23 (), 31 (), 32 (), 33 (), 1123 (), 1132 (), 1133 (), 1232 (), 2123 (), 2132 (), 2133 (), 112232

(ቓ), and 112233 (ቘ). The classification and definition of these structural features is further exposed in [1]. A spatial relationship between two primitives is defined to have six feature values where a value of zero is padded at the beginning for those whose number of connections are two or less. For example, the feature value of a spatial relationship of the type 13 (ቛ) will be {0,0,0,0,1,3}. The sequential order of primitive strokes **A** and **B** is set as **AB** if **A** is spatially located at the left or top of **B**. However, if **A** and **B** are both connected to the right of another primitive stroke and **A** is located at the top of **B**, their sequential order is represented as **BA**. Each primitive is connected to another one to the left except the first primitive in each character, in which case it does not have any one to be connected to the left. In such cases, all the six feature values for such spatial relationship will be all zeros.

2.3. Training and recognition

The goal of the training phase is to estimate the parameter values of word models from a set of training samples, and the recognition phase decodes the word based on the observation sequence. In this work, Baum-Welch algorithm is used for training and Viterbi algorithm is used for recognition. A simplified flowchart of training and recognition procedures is shown in **Fig. 1**. The dotted-line box in the flowchart shows repetitive tasks for each word.

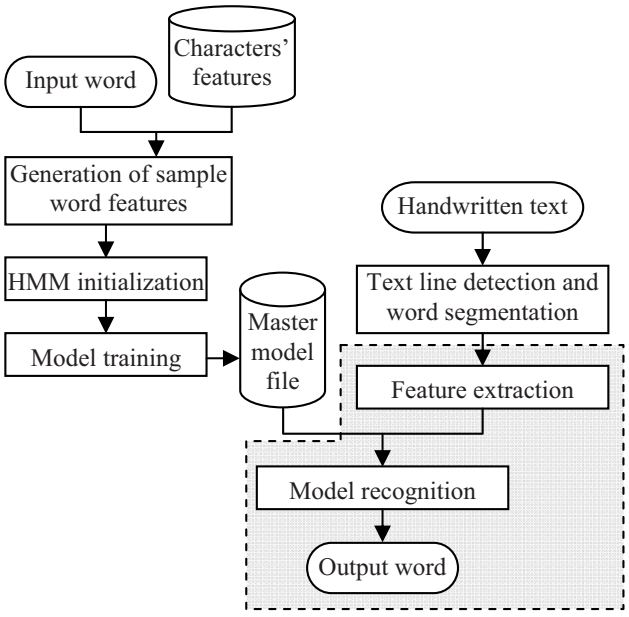

Figure 1. Training and recognition flowchart.

Training samples for a given word are generated from the stored feature lists of characters which contain possibly occurring sample features each Ethiopic character. A character can have many sample features stored as character feature list reflecting variations of writing styles. Suppose that the input word $\mathcal{W}$ has sequences of characters C_1, C_2, C_3,…,C_m, where m is the total number of characters making up the word. Then, sample features of the word are generated as all combinations of sample features of each character. **Figure 2** shows a sample feature for the word "አገር" generated from the character features. Each group in the rectangular box beneath characters represents sample features for the corresponding character, whereas each line represents a feature vector of primitives and their associated spatial relationships.

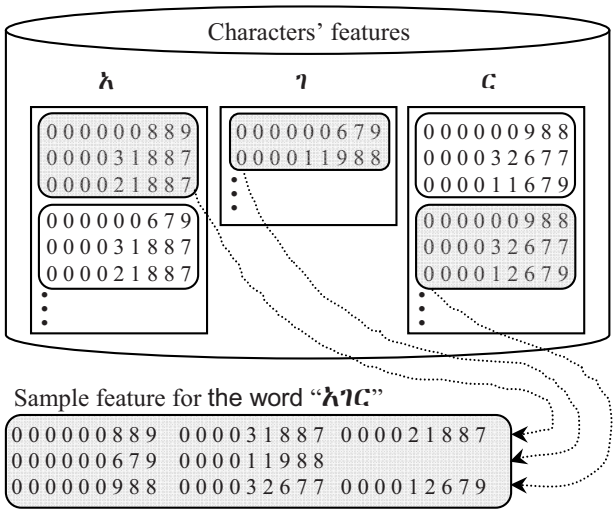

Figure 2. Generation of sample feature for "አገር".

After generating sample features for the input word, the next procedure is HMM initialization which sets a prototype for HMM of the word to be trained including its model topology, transition and output distribution parameters. Gaussian probability function that consists of means and variances is used to define the model parameters. The number of states of a word corresponds to the total number of primitive strokes in the word. The HMM topology of "አገር" which has eight primitive strokes is shown in **Fig. 3**. Once the HMM is trained with sample features of the word, the model is stored into a master model file which will be used later during the recognition phase.

In the recognition phase, handwritten text image is processed to detect lines and segment words. For each word in the text image, a sequence of primitive strokes and their spatial relationship is extracted. **Figure 4** shows primitive strokes and spatial relationships identified for the handwritten word "አገር". Then, the

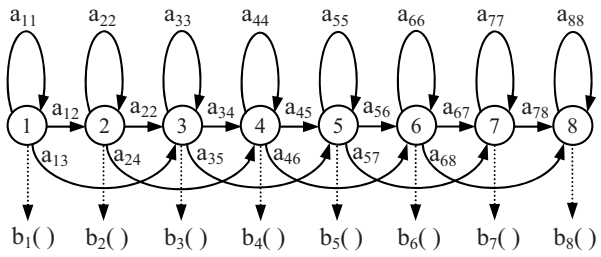

Figure 3. HMM topology for the word "ሕገር".

sequence is generated as: {{αA, βB, γΓ},{δΔ, εE}, {ζZ, ηH, μM}}, where the Greek capital letters represent primitive strokes and smaller letters represent associated spatial relationships. Note that α, δ, and ζ are not shown in the figure since they correspond to a spatial relationship of the first primitive stroke in each character, in which they do not have primitive to the left to be connected with. Once structural features are identified and classified, they are assigned with feature values as discussed in Section 2.2. Then, the extracted feature sequences are considered as observations which are used by the decoder for recognition.

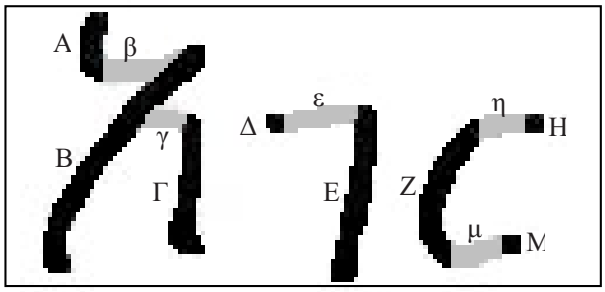

Figure 4. Structural features for the word "ሕገር".

3. Feature Extraction

The recognition system requires text lines, words and pseudo-characters to be segmented for analysis. We developed an algorithm for such segmentation tasks using direction field image. Pseudo-characters represent two or more physically connected characters, but hereafter we simply refer to them as characters.

3.1. Computation of direction field image

Direction field tensor $\boldsymbol{S}$ is a 2x2 matrix which computes the optimal direction of pixels in a local neighborhood of an image f [2]. It is computed as:

$$S = \begin{pmatrix} \iint (D_x f)^2 dxdy & \iint (D_x f)(D_y f) dxdy \\ \iint (D_x f)(D_y f) dxdy & \iint (D_y f)^2 dxdy \end{pmatrix} \quad \textbf{(1)}$$

The integrals are implemented as convolutions with a Gaussian kernel, and D_x and D_y are derivative operators. The local direction vector is the most significant eigenvector modulated by the error differences (the difference of eigenvalues). This vector field is also known as the *linear symmetry* (LS) vector field and can be obtained directly by use of complex moments. The latter are defined as:

$$I_{mn} = \iint ((D_x + iD_y) f)^m ((D_x - iD_y) f)^n dxdy \quad \textbf{(2)}$$

where m and n are non-negative integers. Among other orders, of interest to us are I_{10}, I_{11}, and I_{20} derived as:

$$I_{10} = \iint ((D_x + iD_y) f) dxdy \quad \textbf{(3)}$$

$$I_{11} = \iint \left| (D_x + iD_y) f \right|^2 dxdy \quad \textbf{(4)}$$

$$I_{20} = \iint ((D_x + iD_y) f)^2 dxdy \quad \textbf{(5)}$$

In a local neighborhood of an image, I_{10} computes the ordinary gradient field; I_{11} measures gray value changes (the sum of eigenvalues of $\boldsymbol{S}$); and I_{20} gives a complex value where its argument is the optimal direction of pixels in double angle representation and its magnitude is the local LS strength (the difference of eigenvalues of $\boldsymbol{S}$). Pixels with low magnitude are said to be lacking LS property. As shown in **Fig. 5**, I_{10} and I_{20} images can be displayed in color where the hue represents direction of pixels with the red color corresponding to the direction of zero degree.

Figure 5. (a) Handwritten ሕ, (b) I_{10}, (c) I_{20}, (d) I_{11} of ***a***.

3.2. The segmentation process

Segmentation and text line detection is done on the direction field image (I_{20}) in two passes. In the first pass, the image is traversed from top to down and pixels are grouped into two as *blocked* (character) and *open* (background) regions. A pixel is recursively classified as open if it:

- is in the first row of the direction field image,
- lacks LS and one of its immediate top and/or sideways neighborhood is open.

The remaining are grouped as blocked pixels. Boundaries of blocked regions produce segmented characters. In the second pass, the I_{20} image is traversed from left to right grouping each segmented character into appropriate text lines based on character's proximity along *global* (average direction of the text line) and *local* (direction at the head of the text line) directions. Segmented characters that do not fit into the existing text lines form a new text line. The directions of a text line help to predict the direction in

which the next member character is found during traversal, which is essential especially in skewed documents and non-straight text lines. **Figure 6** shows segmentation and text line detection for handwritten Amharic text skewed by 15°. Words are then segmented based on the relative gap R between characters within a text line, defined as $R_i = G_i - G_{i-1}$, where G_i is the horizontal gap between the i^{th} character and its predecessor. Although the horizontal gap between consecutive characters varies greatly, the relative gap suppresses those variations and a threshold segments words fairly well.

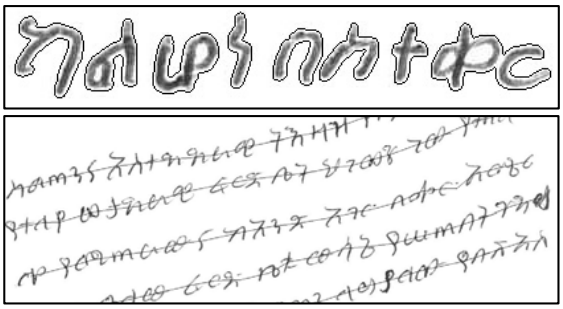

Figure 6. Results of (a) character segmentation, and (b) text line detection.

Pixels are grouped as parts of primitives and connectors based on their optimal direction. After halving the double angle of I_{20}, pixels having LS properties and directions of [0..60] or [120..180] degrees are set as parts of primitives whereas those with LS properties and having directions of (60..120) degrees are considered as parts of connectors. The extracted linear structures in the I_{20} image are mapped onto the I_{10} image to classify them into left and right edges of primitives. A primitive is then formed from the matching left and right edges. Primitives are then further classified using their direction, relative length, and spatial position.

4. Experiment

To test the performance of the system, a database of handwritten Amharic documents collected from 177 writers is developed. The writers were provided with Amharic documents dealing with various real-life issues and they used ordinary pen and white papers for writing. A total of 307 pages were collected and scanned at a resolution of 300dpi, from which we extracted 10,932 distinct words to build a list of words for training. For filtering operations of scanned texts, a symmetric Gaussian of 3x3 pixels was used. Training and recognition of HMMs were implemented by using the HTK toolkit. Recognition rates show variations due to the quality of the handwriting and number of training words. Documents are classified as good and poor based on their quality such as readability and connectivity of characters and words. The most frequent 10 and 100 words were also used for training and testing the system with different sizes. The result is summarized in **Table 3**.

Table 3. Recognition result.

Quality of text	Number of training words		
	10	100	10,932
Good	98%	93%	76%
Poor	85%	81%	53%

5. Discussion and Conclusion

HMM-based Amharic word recognition system for handwritten text is presented. Script-independent text line detection, and character and word segmentation algorithms are also presented. Our proposed method generates sample features of training words from a stored feature list of characters. The feature list stores a variety of sample features for each character reflecting different real-world writing styles. The advantage of this is that it is possible to produce real-world sample word features for any word without collecting sample text, which is turned out to be writer-independent recognition system. It also means that the system can be directly applied for other Ethiopian languages which use Ethiopic script for writing. Since we are encoding the relative size of primitive strokes, recognition system does not require size normalization. The recognition result can be further improved by working more on extraction of structural features and employing language models to HMMs. The database we developed can be used as a benchmark resource for further studies on recognition of Ethiopic script.

References

[1] Y. Assabie and J. Bigun, " Writer-independent offline recognition of handwritten Ethiopic characters", In: *Proc. 11th ICFHR*, Montreal, pp. 652-656, 2008.

[2] J. Bigun, *Vision with Direction: A Systematic Introduction to Image Processing and Vision*. Springer, Heidelberg, 2006.

[3] B. B. Chaudhuri (Ed.), *Digital Document Processing: Major Directions and Recent Adavances*, Springer, ISBN 978-1-84628-501-1, London, pp. 165-183, 2007.

[4] R. Plamondon, S.N. Srihari, "On-line and off-line hand writing recognition: A comprehensive survey", *IEEE Trans. PAMI*, 22(1): pp. 63-84, 2000.

[5] R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition", *Proc. IEEE* , 77(2), pp. 257-286, 1989.