A Novel Short Merged Off-Line Handwritten Chinese Character String Segmentation Algorithm Using Hidden Markov Model

Zhiwei Jiang, Xiaoqing Ding, Changsong Liu, Yanwei Wang
State Key Laboratory of Intelligent Technology and Systems
Department of Electronic Engineering, Tsinghua University
Beijing, P. R. China
{jiangzw, dxq, lcs, wangyw}@ocrserv.ee.tsinghua.edu.cn

***Abstract*—Hidden Markov model (called "HMM" for short) has been a widespread method to segment sequential data in speech recognition and DNA sequence analysis. According to the same principle, it can be also used in segmenting short merged off-line handwritten Chinese character strings, which is a tough issue but often met in practice. Because HMM is still not a common method in this field nowadays, in this paper, we will introduce a novel algorithm using HMM for the segmentation issue above. Eventually, this segmentation algorithm can achieve an applicable performance even when 3755 character classes are compressed into similar characters classes with only 1% amount of original ones, and it also shows an enormous potential of segmenting long text lines.**

Keywords: HMM, merged handwritten Chinese characters, string segmentation, merging similar characters.

I. INTRODUCTION

The segmentation of off-line handwritten character string in a paragraph is an extremely challenging issue in the field of OCR research, because every person has specific writing style and occasional merged handwritings. As Chinese characters are concerned, not only the amount of them is exceedingly marvelous (the amount of common ones reaches more than 3755 already), but their constructions are also absolutely more complicated than any other characters. Since the performance of merged off-line handwritten Chinese character string segmentation affects the performance of recognition directly, this issue attracts many researchers' attention.

For the purpose of overcoming this difficulty, Hong advances a method based on projection in [1] and Liu makes use of connected components in [2]. Besides these segmentation algorithms mainly using appearance information, Fu optimizes redundant segmentation result with multiple confidences in [3] and Su accomplishes segmentation indirectly during recognition with hidden Markov models (HMM) in [4] and [5]. Most solutions above show good performance in the situation of neat handwriting without any characters touching or overlapping, but can hardly do anything about handwriting containing merged characters.

In this paper, we focus on the segmentation of short merged off-line handwritten Chinese character string using HMM, for the merging phenomenon usually occurs among a few characters in a long text line. Compared with hard boundary provided by classic character string segmentation algorithm, HMM is a widespread method to determine soft boundary statistically in the period of decoding and its power has been proved in the field of automatic speech recognition and DNA sequence analysis. At first, we study the influence of HMM parameters on segmentation performance. Here HMM is trained by single character samples, on a small-scale test sample set only with 40 Chinese character classes. Meanwhile, we design a segmentation evaluation criteria based on position and put forward two improvements to improve performance. Then we introduce a strategy, merging similar characters, into our segmentation algorithm based on HMM and evaluate the performance of new algorithm on both previous test sample set and a large-scale one with 3755 common Chinese character classes totally. The final results imply that through these effective improvements, our algorithm can achieve an applicable segmentation performance even when we compress 3755 Chinese character classes into similar characters classes with only 1% amount of original.

The rest of this paper is organized as follows. In the next section, an overview of our algorithm is provided and its fundamental principle is demonstrated as well. The parameters of HMM are discussed in Section III and some improvements are shown in Section IV. Section V presents how to combine merging similar characters with existing segmentation algorithm based on HMM and Section VI shows the performance of segmenting long text line with our algorithm as an extra attempt. Finally, Section VII is a summary of whole paper.

II. ALGORITHM OVERVIEW

Our segmentation algorithm using HMM mainly consists of three procedures and they are pretreatment, training and decoding as shown in Fig. 1.

In pretreatment, both training and test samples are observed through a sliding narrow window with fixed aspect ratio and step length following the direction of writing, from left to right, and the gradient feature vectors of all slice images observed from one sample are extracted as a series of sequential data frames. Since some elements of a feature vector may not contain any useful information, to reduce the dimension of feature vectors by PCA is essential to make sure that the following training procedure can be successful.

1520-5363/11 $26.00 © 2011 IEEE
DOI 10.1109/ICDAR.2011.140

The rest two procedures are both about HMM. As we know, HMM is a classical model-based method in pattern recognition. In essence, it is a state sequence modeled statistically and augmented with state-specific outputs of the model [6]. Going a step further, only if a model is composed of some sequential sub-models and every sub-model can be described by respective HMM, the HMM of whole model equals the connection of HMMs of all sub-models (we can call them sub-HMMs) in turn.

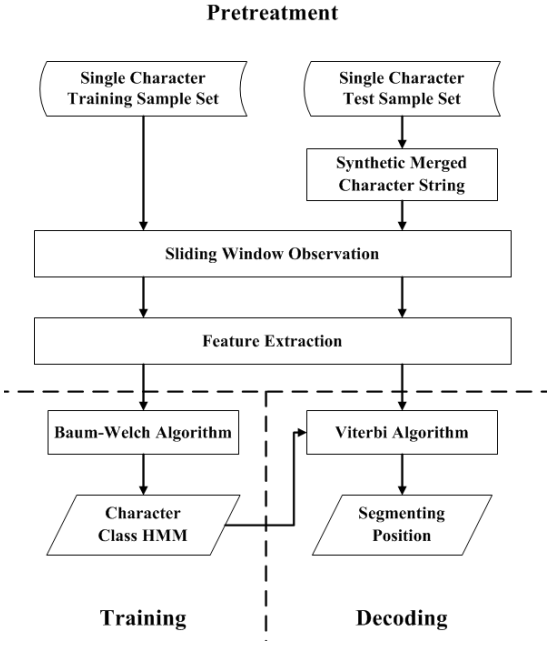


Figure 1. Arcitechture of segmentation algorithm based on HMM.

As our segmentation algorithm is concerned, an HMM modeled for a Chinese character class is trained by standard Baum-Welch Algorithm [6] only with samples of this character class itself and the simply Bakis architecture as shown in Fig. 2 is selected for controlling states transition. When all HMMs of Chinese character classes have been trained separately and successfully, they are considered as sub-HMMs and a new large HMM that combined with them will be generated for modeling the whole Chinese character system.

Figure 2. Simply Bakis architecture with 6 states.

In this new HMM, states are just the ones of all sub-HMMs but allocated a new ordered label. What is more, the first states of every sub-HMM have the same opportunity to be the beginning of a state-transfer trace corresponding to a character string sample and only the last states of every sub-HMM are permitted to transfer to the first ones of all sub-HMMs with the same opportunity. As this new HMM represents complete character system, the observation of a character string sample can be directly decoded by Viterbi Algorithm [6] with it and the result contains which class every character belongs to and where is the most probable segmenting position between two neighbor characters.

Furthermore, there will be two improvements and one merging similar characters strategy assembled into our algorithm in the rest sections, so they are not mentioned here redundantly.

III. FUNDAMENTAL PARAMETERS OF HMM

Since the performance of our segmentation algorithm is extremely sensitive to parameters of HMM, especially the number of states and Gaussian mixtures, it is necessary to compare the segmentation performances under HMMs with different parameter combinations.

A. Segmentation Evaluation Criteria

According to the form of segmentation results, evaluation criteria can be designed based on character block or segmenting position. The former is fit for evaluating quality of final result after adjusting segmenting positions, while the latter is inclined to evaluate performance of segmentation algorithm itself through raw results.

Similar to the evaluation criteria based on block designed by Su [4] in appearance, segmentation success rate (SSR) and segmentation accurate rate (SAR) based on segmenting positions are defined to evaluate our algorithm as follows.

$$SSR = \frac{N_t - D_e}{N_t} \tag{1}$$

$$SAR = \frac{SSR}{\frac{I_e}{\lambda N_t} + 1} \tag{2}$$

In (1), N_t is the number of real segmenting positions and D_e is the error number of correct segmenting positions deleted in results. So SSR evaluates the performance of algorithm only through the ratio of successful segmentation. Then in (2), I_e is the error number of incorrect segmenting positions inserted in results and λ is a factor standing for the tolerance of this error. So SAR represents the comprehensive performance of algorithm. In this paper, λ is set 20%. That is to say, if 20% of N_t incorrect inserted segmenting positions exist, the value of SAR will decrease to half of SSR.

B. Experiments and Results

For the sake of studying the influence of HMM parameters and building relationship with latter merging similar characters strategy, we just choose 40 character classes and extract some short merged Chinese character strings in real text lines written by ourselves only with character classes above as a small-scale test sample set. According to these 40 character classes, the training samples with 100 ones per character class come from THOCR-HCD1

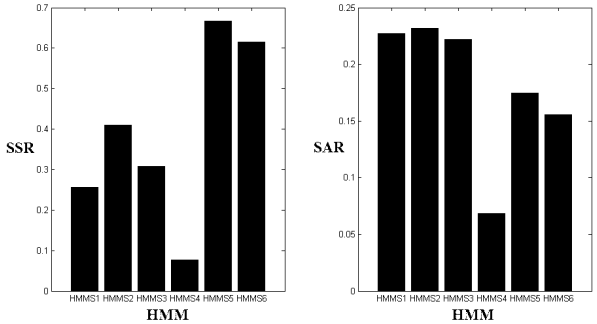
database. By the way, THOCR-HCD (called "HCD" for short) databases are collected by State Key Laboratory of Intelligent Technology and Systems of Tsinghua University and contain 3755 level- one Chinese characters in GB2312-1980.

We select 6 combinations of parameters with different numbers of HMM states and Gaussian mixtures as shown in Table 1. Then, HMMs are trained separately.

TABLE I. COMBINATIONS OF PARAMETERS

	HMM S1	HMM S2	HMM S3	HMM S4	HMM S5	HMM S6
states	9	6	6	6	4	4
Gaussian mixtures	1	1	2	3	1	2

Results of all experiments are collected in the form of bar graph and showed in Fig.3.



(a) SSR performance (b) SAR performance

Figure 3. SSR and SAR of algorithm under different parameters

Firstly, taking a note of the results of HMMS1, HMMS2 and HMMS5 shown in Fig.3, we can find that more states make results conservative at risk of less segmenting while fewer states make results progressive at risk of over segmenting. As known, if more states we use, one character will be divided into more parts different from each other along the observation direction. But test sample may not fit every state of this model completely or accurately. In contrast, too few states are not enough to illustrate a character and bring error to algorithm as well. By the reason of that, Feng gives a method to identify the best number of states of an HMM taking advantage of stroke permeability number in [7].

On the other hand, the results of HMMS2, HMMS3 and HMMS4 (or comparing HMMS5 and HMMS6) tell us that more Gaussian mixtures modeling a state of HMM together cannot bring more enhancements, because the samples for training every state of an HMM are obviously similar and one or two Gaussian mixtures are just enough. Moreover, too many Gaussian mixtures will not only make covariance matrix of them singular for lack of enough samples, but also cost much more training time than the fewer ones.

IV. IMPROVEMENTS

Since accomplishing segmentation only with original HMM is an extremely difficult work, some specialized improvements customized to Chinese characters segmentation algorithm are necessary and imperative. Here we introduce two improvements to make the performance better and they are global height normalization and enhanced decoding.

A. Global Height Normalization

As probable space at the top or bottom of one character sample will affect the ratio of actual observation during pretreatment, the height normalization is essential to make observations in accordance with corresponding trained models. Since merged handwriting phenomenon usually appears among 2~4 characters as our test samples, it is sufficient to take global height normalization during pretreatment of test samples.

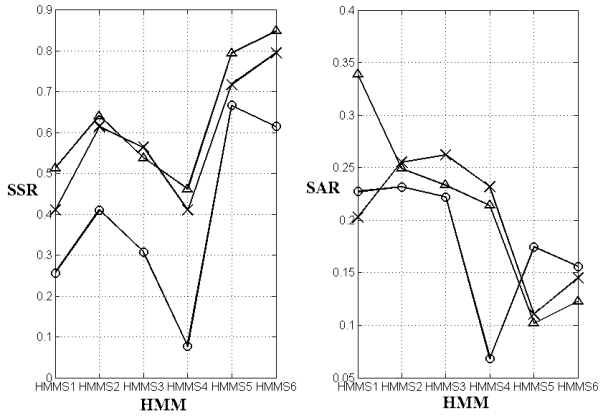
In addition, this improvement will be also used before the next improvement.

B. Enhanced Decoding

Enhanced decoding is to connect blank images before and behind one test sample and to insert a blank HMM into existing segmentation algorithm. If the last observation of one test sample belongs to an intermediate state of an HMM, the decoding result will not be a complete connection of some HMMs and that is impossible absolutely. So the purpose of this improvement is to avoid that situation.

C. Experiments and Results

We use the same training and test samples in experiments as the previous section. Results, as shown in Fig. 4, are collected in the form of line graph and the ones without any improvements are also added.



(a) SSR performance (b) SAR performance

Figure 4. SSR and SAR of algorithm with and without improvements

In Fig. 4, circles represent original results in previous section, crosses represent global height normalization results, and triangles represent enhanced decoding results.

SSR curves imply that two improvements can both increase SSR performances of algorithm under all

combinations of HMM parameters. Generally, enhanced decoding can bring an extra benefit to the algorithm with global height normalization.

From SAR curves, it is seen that global height normalization increases SAR performance of algorithm with 6 states. Since there are not enough states to model all characters in algorithm with 4 states, too many redundant segmenting positions decrease SAR performance. Meanwhile, SAR of algorithm with 9 states is also decreased because there exists more risk that decoding procedure stops at an intermediate state in algorithm with more states than in one with fewer states. This inference can be proved with the obvious advance of SAR in algorithm with 9 states after enhanced decoding is executed. However, enhanced decoding will not make effective improvement of algorithm with fewer states.

V. MERGING SIMILAR CHRACTERS

In experiments above, some characters have been segmented correctly, but sometimes the classes of them are not the real ones they belong to. The reason why that appears is there are so many similar characters in Chinese that they are usually mistaken for each other.

Originally, discriminating similar characters is such a hard problem in recognition field that many researchers have been making every effort to solve it. However, as our goal is only to segment every two neighbor characters but not to recognize which class they belong to, merging similar characters strategy, which is usually used in multi-level Chinese recognition system [8] [9], becomes a potential way to confront the interference among similar characters and compress decoding space of Viterbi Algorithm [6] consumedly in segmentation algorithm based on HMM.

A. Similar Characters List

Confusion matrix is quite important for generating similar characters list. Firstly, we train 3755 models of common Chinese character classes with all HCD training samples (from HCD1 to HCD10 except HCD4 and HCD9), using traditional recognition system with 392D gradient feature and LDA compressing to 128D. And then, record the top 10 choices when every training sample is recognized by this trained system. Finally, a 3755D square matrix only with zero elements is used to collect recognition results and if character class j appears in the top 10 choices of character class i, the element at (i, j) will plus one. That is confusion matrix we need.

Then a clustering algorithm as shown in Fig. 5 is executed and the output is a similar characters list. Here we set 40 as the number of similar characters clusters and it means only 1% amount of original Chinese character classes.

B. Large-Scale Test Sample Database

With the aim of testing our algorithm more accurately, we need a test sample database with more short merged character string samples and containing all 3755 common Chinese character classes. Synthetic sample mentioned in [10] is a good way to generate samples massively. Here we use HCD2 as an element set and generate 2000 synthetic merged samples of 2, 3 or 4 characters separately. In a synthetic merged sample, all elements are connected at the same baseline and the distance between two characters is a random value from zero to negative 10% of average character size just like merging phenomenon in a handwritten string.

Given a confusion matrix M, its dimension N and desired number of similar character clusters K.

1. **Initialization**
   for all labels i (i = 1, 2, …, N):
   $$M(i, i) = 0$$
2. **Iteration**
   a) **Searching**
   for all combinations of i and j (i, j = 1,2, …, N) except the ones that i equals j:
   calculate
   $$M(i, j) + M(j, i)$$
   and find i* and j* (i* < j*) making the sum maximum.
   b) **Merging**
   add the information of character class i* to the one of j*:
   $$M(:, i^*) = 0.5*M(:, i^*) + 0.5*M(:, j^*)$$
   $$M(i^*, :) = 0.5*M(i^*, :) + 0.5*M(j^*, :)$$
   [note: symbol ":" means all elements in specific dimension]
   then, delete the j*th row and column of M.
   c) **Recording**
   set
   $$N = N-1$$
   and record character class i* is similar to character class j*.
3. **Termination**
   if N does not reach K,
   continue with step 2
   otherwise Stop!

Figure 5. Algorithm for generating similar characters list

C. Experiements and Results

HCD1 is still used as training sample set in this section, but every HMM of similar characters classes is trained with 1000 samples, which averagely come from samples of all character classes in the corespoding similar characters class. And only the parameters in HMMS1 and HMMS2 are selected because of their good performance before. The same training and test procedure with two improvements above are executed and the results are shown in Table 2. In addition, Fig. 6 shows some test samples decoded in HMMS1 as examples.

TABLE II. SSR AND SAR PERFORMANCE OF ALGORITHM WITH MERGING SIMILAR CHARACTERS

	2 characters		3 characters		4 characters	
	SSR	SAR	SSR	SAR	SSR	SAR
HMMS1	0.7875	0.2039	0.7620	0.2200	0.7512	0.2265
HMMS2	0.8125	0.1175	0.8120	0.1418	0.8052	0.1549

From Table 2, we can infer that more than 75% segmenting positions are detected successfully, and the more characters there are in a string, the better performance of algorithm is. So merging similar characters is an effective way to improve performance and our algorithm can solve short merged off-line handwritten Chinese character string segmentation problem preliminary.

Then, on the previous small-scale test sample set, Table 3 shows a comparison between the best results in last section and results through current algorithm in this section.

TABLE III. PERFORMANCE COMPARISON ON SMALL-SCALE TEST SAMPLE DATABASE

	Previous		Current	
	SSR	SAR	SSR	SAR
HMMS1	0.5128	0.3390	0.4615	0.4091
HMMS2	0.6410	0.2404	0.7949	0.2153

Obviously, two algorithms have nearly the same performance and it proves that merging similar characters is a powerful improvement again. What is more, in results of current algorithm, SAR in HMMS1 is more than previous one but SAR in HMMS2 is less than previous one. This implies similar characters classes need more states to be modeled than single character classes.

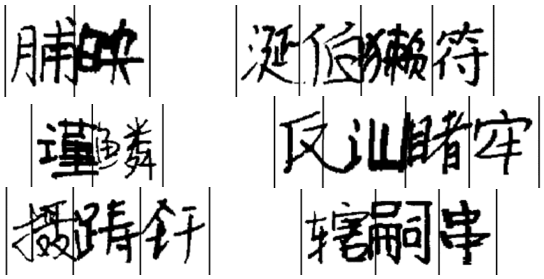


Figure 6. Some examples of decoded test sample in HMMS1

VI. FUTRUE WORK

As an extra attempt, we decode some long text lines with current segmentation algorithm in HMMS1 and surprisingly it does quite a good job shown in Fig. 7. But there are still many detailed investigations to do in the situation of long text lines.

In Fig. 7, there are some segmenting positions not detected successfully and a few segmenting errors existing as well. Because the height of characters in a long text line is always waving, this causes less accordance observed through sliding window between test sample and training sample. That is, our method is sensitive to actual observed aspect ratio of characters and a kind of adaptive height normalization is necessary to improve algorithm performance. Furthermore, some suitable after treatments are also extremely important to adjust final segmenting positions for the purpose of defeating wide fluctuation from probabilistic method.

VII. CONCLUSION

We design a segmentation algorithm using HMM for short merged off-line handwritten Chinese character strings. Through investigating parameters of HMM, advancing two improvements and introducing merging similar characters strategy, the algorithm performance is enhanced obviously and more than 75% segmenting positions can be detected successfully only with 1% amount of HMMs than before. In addition, our algorithm also shows an enormous segmentation potential in the situation of long text lines.

ACKNOWLEDGMENT

This work was supported by the National Basic Research Program of China (973 program) under Grant No. 2007CB311004 and the National Natural Science Foundation of China under Grant No. 60872086.

REFERENCES

[1] C. Hong, G. Loudon, Y.Wu, et al, "Segmentation and Recognition of Continuous Handwriting Chinese Text," International Journal of Pattern Recognition and Artificial Intelligence, vol. 12(2), pp. 223-232, 1998.

[2] C.L. Liu, M. Koga, H. Fujisawa, "Lexicon-driven Segmentation and Recognition of Handwritten Character Strings for Japanese Address Reading," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24(11), pp. 1425-1437, 2002.

[3] Q. Fu. "A Study on Unconstrained Offline Handwritten Chinese Character and Character String Recognition," PhD Thesis, Tsinghua University, 2008.

[4] T.H. Su, "Off-line Recognition of Chinese Handwriting: From Isolated Character to Realistic Text," PhD Thesis, Harbin Institute of Technology, 2008.

[5] T.H. Su, "Improving HMM-Based Chinese Handwriting Recognition Using Delta Features and Synthesized String Samples," 12th International Conference on Frontiers of Handwriting Recognition, 2010.

[6] Gernot A. Fink, "Markov Models for Pattern Recognition: From Theory to Applications," New York: Springer, 2008.

[7] B. Feng. "Hidden Markov Models Based Chinese Handwriting Recognition," PhD Thesis, Tsinghua University, 2004.

[8] B. Zhang, "Handwritten Chinese Similar Characters Recognition Based On AdaBoost," proceedings of 26th Chinese Control Conference, pp. 576-579, 2007.

[9] Z.Q. Lin, J. Guo, "An Algorithm for the Recognition of Similar Chinese Characters," Journal of Chinese Information Processing, vol. 16(5), pp. 44-48, 2002.

[10] X. Chen, "Discriminatively Train Classifiers Embedding on Synthetic String Samples for Chinese Handwritten String Recognition," Master Thesis, Harbin Institute of Technology, 2010.

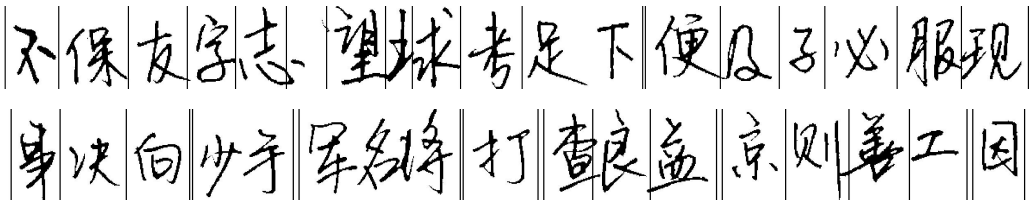


Figure 7. Some attempts of long text lines with current system