

Contextual Post-processing based on the Confusion Matrix in Offline Handwritten Chinese Script Recognition

Yuan-Xiang Li^a, Chew Lim Tan^{a,*}, Xiaoqing Ding^b and Changsong Liu^b

a School of Computing, National University of Singapore, Singapore 117543

b Department of Electronic Engineering, Tsinghua University, Beijing 100084, P. R. China

Abstract The inclusion of potentially correct characters in candidate sets is key to improving accuracy in the recognition of Chinese scripts in the aspect of contextual post-processing. This paper presents two methods based on a confusion matrix to recall the correct characters. The first method uses original candidates to conjecture the most likely correct characters, and then combines the conjectured set with the original candidates to produce a new candidate set. The second method performs an approximate matching of adjoining characters in a sentence with Chinese words so as to recall the most likely correct character. Experimental results demonstrate the effectiveness of our proposed methods.

Keywords Chinese character recognition, Contextual post-processing, Statistical language model, Confusion matrix, Candidate expansion, Combination, Approximate word-matching

1. Introduction

Recognizing offline handwritten Chinese characters is a challenging pattern recognition problem [1]. Due to the large character set, complex character shapes, many confusable subsets of characters having only slightly different shapes, and great variations in writing style, it is difficult to significantly improve the accuracy of Chinese script recognition in an offline handwritten isolated Chinese character recognition system. Statistical language models (SLMs) have been successfully used for contextual post-processing to increase accuracy in the recognition of Chinese scripts [2-5]. The technology of contextual post-processing can be described as follows: Under the joint action of SLMs and candidate confidence, an efficient search strategy (such as the well-known *Viterbi* algorithm [6]) is employed to select the most likely sentence from the candidate sets provided by a character recognizer.

Since the Chinese character set is very large, the number of candidates is usually limited. When executing contextual post-processing based on word-class n-gram language models, the number of candidates is always no more than 10 [2-5]. An excessive number of candidates would increase overall processing time and decrease overall script recognition accuracy due to excessive erroneous word formations during lexicon lookup. Wong and Chan [5] only used six candidates in contextual post-processing. In fact, for well recognized scripts, the top 10 candidates may be enough to capture the correct character; however, for poorly recognized scripts, even using the top 100 candidates or more may sometimes fail to capture the correct character. Therefore, the question will naturally follow: Has the correct character (true candidate) been included in the limited candidate set at all? Obviously, if there is no correct character in the candidate set, it is impossible to correct the errors in the recognizer, no matter how precise SLMs are.

There are two approaches to dealing with this problem. One is to make use of a confidence evaluation measure to estimate not only the confidence of the first candidate in the set, but also that of subsequent candidates. Through some rules to aggregate the confidence values of top candidates

Corresponding author. Tel: + 65-6874-2900; fax: + 65-6779-4580. Email address: tancl@comp.nus.edu.sg (C.L. Tan)

until the sum exceeds a threshold, the number of candidates can then be selected [7, 8]. This approach focuses on converting the recognition distance from the k -nearest neighbor classifier into a probabilistic value. But the approach needs to know the characters' distribution in the feature space [7] or the distribution of correct data and erroneous data [8].

The other approach is to make use of the characteristics of recognition errors. As is well known, a special recognition system has its own characteristics of errors, which are based on its underlying understanding of how some characters could often be mistaken for others. This kind of recognition characteristics is represented by a confusion matrix. From the viewpoint of knowledge, a confusion matrix could be regarded as the prior knowledge of a recognizer [9]. Kernighan et al. [10] and Tong and Evans [11] employed a confusion set of Roman letters for English spelling correction. Marukawa et al. [12] used a Japanese confusion set for document retrieval that tolerates character recognition errors. Lee et al. [13] directly inserted Korean similar characters into the candidate set given by a recognizer in order to improve post-processing performance. In the Chinese language, there are many confusable subsets of characters which have only slightly different shapes among themselves. Chinese similar characters are often used in Chinese text proofreading [14]. This approach is more amenable to implementation compared to the earlier approach of utilizing confidence evaluation.

This paper focuses on the contextual post-processing of Chinese script recognition with the use of a confusion matrix of Chinese characters. Based on the confusion matrix, we propose two methods to recall potentially correct characters. One is to use the original candidates in a candidate set to conjecture the most likely correct characters, and then combine the conjectured set with the original candidates to produce a new candidate set. The other is to perform an approximate matching of adjoining characters in a sentence with Chinese words. The remainder of this paper is organized as follows. In Section 2, we introduce the framework of contextual post-processing. In Section 3, we describe a candidate expansion algorithm, discuss its performance and workings, and consider how a new candidate set is produced by combination using the algorithm. In Section 4, we describe an approximate word-matching method. Section 5 demonstrates the effectiveness of our methods by showing several post-processing experimental results and analyses. We conclude in Section 6.

2. Description of Contextual Post-processing

A typical Chinese script recognition system is shown in Fig.1. Let $X = x_1x_2 \cdots x_T$ be a sequence of Chinese character images, where x_t is the t^{th} character image in the input sequence X , and T is the length of the sequence. Let $S = s_1s_2 \cdots s_T$ be a sequence of Chinese characters given by an isolated Chinese character recognizer (ICCR), in which each output s_t may include top K candidates ($c_{t,1}c_{t,2} \cdots c_{t,K}$). Let $O = o_1o_2 \cdots o_T$ be the final Chinese sentence.

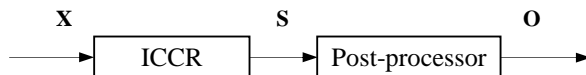


Fig.1. The basic framework of Chinese script recognition

Considering the top K candidates for each output s_t , there are K^T possible sentences. The post-processor's task is to select the most likely sentence from all of the K^T sentences. By applying the rule of maximal posterior probability, the output O in Fig.1 can be represented as:

$$O = \underset{S}{\operatorname{argmax}} p(S|X) = \underset{S}{\operatorname{argmax}} p(S) * p(X|S) \quad (1)$$

where $p(S)$ stands for a statistical language model; $p(X | S)$ is a conditional probability. Assuming that the current recognition behavior is independent of the previous decisions in ICCR, then $p(X | S)$ can be written as follows:

$$p(X | S) = \prod_{t=1}^T p(x_t | s_t) = \prod_{t=1}^T \frac{p(s_t | x_t) * p(x_t)}{p(s_t)} \quad (2)$$

In Eq. (2), $p(s_t)$ is the prior probability determined by ICCR, which can be regarded as of equal probability for all K candidates for the output s_t [1], while $p(x_t)$ is irrelevant to solving O in Eq. (1). Substituting Eq. (2) into Eq. (1), we have:

$$O = \arg \max_S p(S) * \prod_{t=1}^T p(s_t | x_t) \quad (3)$$

where $p(s_t | x_t)$ stands for the confidence of a candidate $c_{t,k}$, which can be estimated by the *Logistic Regression Model* (see Section 3.4.1) ¹.

In the Chinese language, a word consisting of one or more characters is a basic syntactically meaningful unit. However, each character in the word also has a definite meaning in itself. Thus, an n -gram Chinese language model can be based on either words or characters. If we adopt the character-based bigram, Eq. (3) can be represented as follows:

$$O = \arg \max_S [p(s_1) p(s_1 | x_1)] * [\prod_{t=2}^T p(s_t | s_{t-1}) p(s_t | x_t)] \quad (4)$$

where $p(s_t | s_{t-1})$ is the transition probability of the character-based bigram; $p(s_1)$ is the first character probability in a sentence.

If we adopt the word-based bigram in Eq. (3), we use $S = w_1 w_2 \cdots w_{T'}$ (S contains T' words)

instead of $S = s_1 s_2 \cdots s_T$. Then we have:

$$\begin{aligned} O &= \arg \max_S p(w_1) \prod_{i=2}^{T'} p(w_i | w_{i-1}) \prod_{t=1}^T p(s_t | x_t) \\ &= \arg \max_S [p(w_1) \varphi(w_1)] * [\prod_{i=2}^{T'} p(w_i | w_{i-1}) \varphi(w_i)] \end{aligned} \quad (5)$$

where $\varphi(w_i) = \prod_{t=t_i+1}^{t_i} P(s_t | x_t)$, $t_i = \sum_{j=1}^i |w_j|$, $|w_j|$ stands for the j^{th} word length;

$p(w_i | w_{i-1})$ is the transition probability of the word-based bigram; $p(w_1)$ is the first word probability in a sentence.

The optimal sentence in Eq. (4) and Eq. (5) can be searched by the *Viterbi algorithm* [6].

¹ Noting that $s_t \in \{c_{t,1}, c_{t,2}, \dots, c_{t,K}\}$, $p(s_t | x_t)$ in Eq. (3) equals $p(c_k | x)$ in Eq. (12), which will be explained later.

3. Expanded Candidate Set

Let ω_i be the correct character and ω_j be the recognition result (i.e., the first candidate) of ω_i . M is the set containing 3,755 Chinese simplified characters ($\omega_i, \omega_j \in M$). N is the number of classes in M ($N = 3755$). The confusion matrix of a recognizer can be represented as:

$$P_M = (p_{ij})_{N \times N}$$

where $p_{ij} \approx n_{ij} / n_i$, n_{ij} is the number of times that ω_i is recognized as ω_j in the training sets of isolated Chinese character recognition, $n_i = \sum_{j=1}^N n_{ij}$. Note that a character can only be recognized as a limited number of other characters, so P_M is a sparse matrix.

Because the confusion matrix P_M can be obtained from a large number of training sets in advance, we regard it as prior knowledge for a character recognition system. Based on P_M , it is possible to improve the performance of a character recognition system.

3.1. Candidate Expansion Algorithm (CEA)

It is very difficult to directly obtain the posterior probability of a candidate [9]. Broadly speaking, a decision based on maximal posterior probability is usually converted into a decision based on minimal distance in character recognition. The higher the posterior probability, the less is its correlative distance.

Let a be the correct character corresponding to a character image x . For x , a character recognizer gives top K candidates $c_1 c_2 \cdots c_K$ (called the original candidates) and their corresponding distance values $d_1 d_2 \cdots d_K$ ($d_1 \leq d_2 \leq \cdots \leq d_K$). Let $C_K = \{c_1, c_2, \cdots, c_K\}$ be the original candidate set (OCS) with top K original candidates².

Using $c_1 c_2 \cdots c_K$, we can conjecture the most likely correct character \hat{a} (called the expanded candidate). Applying the rule of maximal posterior probability, we have:

$$\begin{aligned} \hat{a} &= \operatorname{argmax}_{a \in M} p(a | c_1 c_2 \cdots c_K) \\ &= \operatorname{argmax}_{a \in M} \{p(c_1 c_2 \cdots c_K | a) * p(a) / p(c_1 c_2 \cdots c_K)\} \end{aligned} \quad (6)$$

In Eq. (6), $p(a)$ is the prior probability determined by the recognizer, which can be regarded as of equal probability [1], i.e., $p(a) = 1/N$, and $p(c_1 c_2 \cdots c_K)$ is irrelevant to solving \hat{a} . So Eq. (6) can be rewritten as follows:

$$\hat{a} = \operatorname{argmax}_{a \in M} p(c_1 c_2 \cdots c_K | a) \quad (7)$$

Theoretically, there are some relationships among $c_1 c_2 \cdots c_K$. However, from the viewpoint of engineering practice, because of the large K , it is very difficult to obtain this kind of

² For conciseness, we omit here the subscript t in $c_{t,k}$, which we have mentioned in Section 2.

relationships. So, we assume independence among $c_1 c_2 \cdots c_K$ so as to approximately solve Eq. (7).

Then, Eq. (7) can be represented as:

$$\hat{a} \approx \operatorname{argmax}_{a \in M} \prod_{k=1}^K p(c_k | a) \quad (8)$$

where $p(c_k | a)$ is an element of P_M .

Based on Eq. (8), we select the top L most likely correct characters to construct the expanded candidate set (ECS) $E_L = \{e_1, e_2, \dots, e_L\}$. Furthermore, by Eq. (8), we can also evaluate the possibility f_l for each expanded candidate e_l , which is defined as:

$$f_l = \prod_{k=1}^K p(c_k | e_l) / \sum_{l=1}^L \prod_{k=1}^K p(c_k | e_l), \quad l = 1, 2, \dots, L \quad (9)$$

where $f_l \in [0, 1]$, $f_1 \geq f_2 \geq \dots \geq f_L$, $\sum_{l=1}^L f_l = 1$.

Instead of expanding from the top candidates $c_1 c_2 \cdots c_K$, we can use each original candidate c_k to conjecture the likely correct characters. Eq. (7) can be simplified as follows:

$$\hat{a}_k = \operatorname{argmax}_{a \in M} p(c_k | a), \quad k = 1, 2, \dots, K \quad (10)$$

For each c_k , we select the top J most likely correct characters, which are usually called similar characters. For K original candidates, we can select $J * K$ likely correct characters to construct the similar candidate set (SCS).

In the following sub-sections, we shall discuss the performance and workings of CEA.

3.2. Performance of CEA

We define a performance parameter r , the recall rate, as follows:

$$r = n_{rec} / (n_{sum} - n_{ten}) \times 100\% \quad (11)$$

where n_{rec} is the number of correct characters in ECS when there is no correct character in C_{10} ($K = 10$), n_{sum} is the number of characters in the sample sets, n_{ten} is the number of correct characters in C_{10} .

In our experiment, "THOCR'97 *Synthetical and Integrated Chinese Character Recognition System*" [1] is used as the ICCR, in which a minimal distance classifier is adopted. There are 1,400 sample sets, of which 1,100 sample sets with an average recognition accuracy³ (namely the first

³ recognition accuracy = $(1.0 - \text{the number of incorrect characters} / \text{total characters}) \times 100\%$.

candidate accuracy) of 89.05% and a top 10 cumulative accuracy⁴ of 98.18% are used to obtain a confusion matrix. Every sample set consists of 3,755 offline handwritten Chinese characters. The remainder, containing 300 sample sets, is regarded as test sets with an average recognition accuracy of 87.85%, and a top 10 cumulative accuracy of 97.96%.

3.2.1. Factors Affecting Recall Rate

There are three key factors impacting r : the size of training sets (TS), the number of original candidates K in Eq. (8), and the quality of test sets. Apparently, r rises with the increase of the number of expanded candidates m . In the following figures, $r\sim m$ curves are used to illustrate the effect of these three factors.

Fig.2 shows that r rises as the size of the training sets increases. It is worth noting that 1,100 training sets are not enough to obtain the confusion matrix. If there are more training sets, r can further increase.

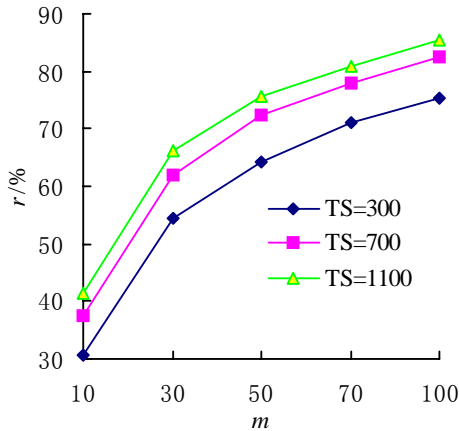


Fig.2. $r\sim m$ curves effected by the size of training samples

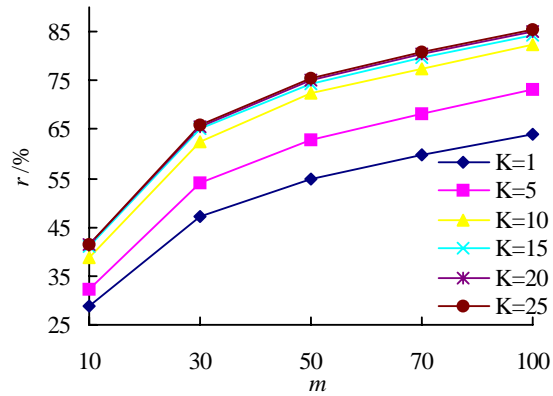


Fig.3. $r\sim m$ curves effected by the number of original candidates

Fig.3 demonstrates that r also rises with increasing K . Note that the augment gradually decreases with increasing K . K is suitably chosen to be 15.

We divide the test sets into five classes: (Class-A) best-quality samples with accuracy more than 90%; (Class-B) good-quality samples with accuracy between 80% and 90%; (Class-C) fair-quality samples with accuracy between 70% and 80%; (Class-D) bad-quality samples with accuracy between 60% and 70%; (Class-E) worst-quality samples with accuracy below 60%. Fig.4 shows that the better the quality of test sets, the higher r .

⁴ cumulative accuracy = $(1.0 \times \text{the number of correct characters in the top } k \text{ candidates} / \text{total characters}) \times 100\%$.

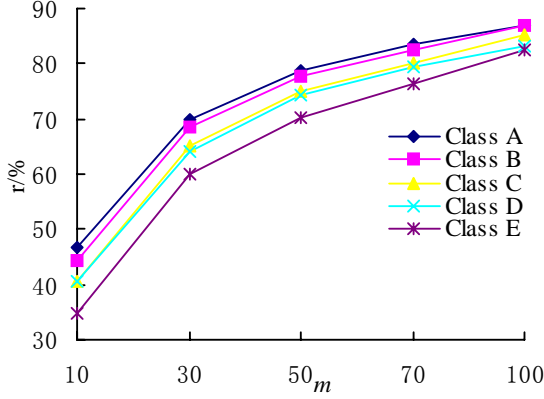


Fig.4. $r\sim m$ curves effected by the quality of test samples

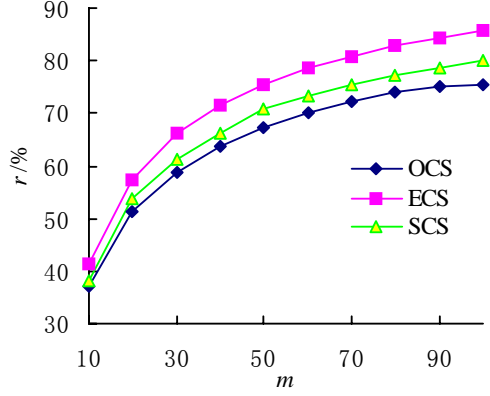


Fig.5. $r\sim m$ curves for different candidate sets

3.2.2. Comparison among ECS, SCS and OCS

We can define the recall rate for OCS in the same way as Eq. (11), except that n_{rec} is the number of correct characters beyond C_{10} in Eq. (11). Similarly, for SCS, n_{rec} is the number of correct characters within SCS when there is no correct character in C_{10} . The $r\sim m$ curves for ECS, OCS and SCS are plotted in Fig.5, from which three characteristics are observed:

- 1) The number of correct characters beyond C_{10} is limited. r with 50 more original candidates (i.e., from the 11th to the 60th) is 67.40%, while r with 100 more original candidates (i.e., from the 11th to the 110th) is only 75.15%.
- 2) In comparison with OCS, both SCS and ECS have a higher r . Among ECS, SCS and OCS, ECS has the highest r . For ECS, r with E_{50} is 75.65% (the error is reduced by 25%); furthermore, r with E_{100} is 85.57% (the error is reduced by 42%).
- 3) With the increase in the number of candidates, the gap between r in ECS and r in OCS gradually enlarges, as can also be seen in Table 1.

Intuitively, even if we simply replace the original candidates beyond C_{10} with the expanded candidates or similar characters, we can improve the cumulative accuracy of a recognition system.

3.3. The Workings of CEA

As stated before, the confusion matrix records the prior knowledge of a recognizer, which is the information about the feasible confusion characters (similar characters) in the recognition process. The workings of CEA are to “match” the original candidates with similar characters corresponding to the correct character in the confusion matrix, and find the likely correct character by way of the similar characters.

When K original candidates include similar characters of a in Eq. (8) and the confusion probability is high (namely well “matched”), CEA is able to find the correct character. The greater the similar characters of a are included in C_K , the more likely it is able to find a . However, if there is no similar character of a included in C_K (a complete “mismatch”), CEA fails.

Actually, there are so many similar characters in the Chinese language that an offline handwritten character recognizer cannot effectively discern similar characters sometimes. Therefore, generally, even if C_K does not include a , it may include the similar characters of a . So, to a large extent, CEA is able to find a .

Because of the wide variations in writing style and the limited training sets, the distribution of similar characters in the confusion matrix is so sparse that the correct character may be found in ECS, where L is large (as given in Section 3.1), as can be verified in Fig.5 and Table 1.

3.4. Combining Expanded Candidates with Original Candidates

Table 1 lists the cumulative accuracies of OCS and ECS for 300 test sets. From Table 1, we can see that although the accuracy of the first expanded candidate is rather low, the cumulative accuracy of ECS is higher than that of OCS when the number of candidates is no less than 10. Based on the analysis in Section 3.1, we believe that there is some complementary relationship between OCS and ECS, even though ECS is conjectured from OCS. The combination of OCS and ECS can be beneficial to improving cumulative accuracy [15].

Table 1 Comparison of Cumulative Accuracies among OCS, ECS and CCS (%)

Number of candidates	1	10	20	30	40	50	100
Original candidates	87.85	97.96	98.72	99.01	99.16	99.26	99.49
Expanded candidates	62.94	97.97	98.86	99.11	99.24	99.33	99.55
Combined candidates	88.21	98.20	98.93	99.22	99.45	99.54	99.69
Error reduction rate	2.96	11.76	16.4	21.21	34.52	37.88	39.22

From the viewpoint of scheme combination, the key to combining OCS and ECS is to look for a unified measurement [16] by which we can re-rank OCS and ECS in order to obtain a new candidate set. In the following, we use confidence as the unified measurement and estimate the confidence of the original candidates and the confidence of the expanded candidates respectively.

3.4.1. Confidence Measurement

For the original candidates $c_1 c_2 \cdots c_K$ corresponding to a character image x , their distance values are $d_1 d_2 \cdots d_K$. The *Logistic Regression Model* [17, 18] can directly convert the distance measurement of an original candidate c_k into its confidence value:

$$p(c_k | x) = (1 + \exp(\beta_0^k + \sum_{i=1}^y \beta_i^k d_i))^{-1}, \quad 1 \leq k \leq K \quad (12)$$

where β_i^k is the regression coefficient, which can be estimated by *Maximum Likelihood Estimation* [19] through the recognition results of some training sets. y is the order of regression model. d_i ($1 \leq i \leq K$) is normalized to the value within 0 ~ 100 in our experiment.

In our experiment, 50 sample sets with an average accuracy of 87.40% (see Section 3.2) are used to estimate the regression coefficients in Eq. (12). For the first original candidate, we have:

$$p(c_1 | x) = (1 + \exp(-0.647 + 0.439d_1 - 0.325d_2 - 0.086d_3))^{-1} \quad (13)$$

Eq. (13) means that the smaller d_1 (and the bigger d_2 or d_3) is, the more reliable is the first original candidate. For the subsequent original candidates, their confidence has the following formula: $p(c_k | x) = (1 + \exp(\beta_0^k + \beta_1^k d_1 + \beta_k^k d_k))^{-1}$ ($\beta_1^k < 0$, $\beta_k^k > 0$, $k \geq 2$), which

means that the smaller d_1 (and the bigger d_k) is, the less reliable are the subsequent original candidates.

For the expanded candidates $e_1 e_2 \cdots e_L$, their corresponding possibility values are $f_1 f_2 \cdots f_L$. Similarly, we can also use the *Logistic Regression Model* to directly convert the possibility measurement of an expanded candidate e_l into its confidence value:

$$p(e_l | x) = (1 + \exp(\gamma_0^l + \sum_{i=1}^z \gamma_i^l f_i))^{-1}, \quad 1 \leq l \leq L \quad (14)$$

where γ_i^l is the regression coefficient estimated by *Maximum Likelihood Estimation* [19] through the recognition results of some training sets. z is the order of regression model.

The expanded sample sets of the above 50 sample sets can be obtained by Eq. (8) and Eq. (9), which are used to estimate the regression coefficients in Eq. (14). For the first expanded candidate, we have:

$$p(e_1 | x) = (1 + \exp(1.587 - 3.248f_1 + 1.246f_2 + 2.184f_3))^{-1} \quad (15)$$

Eq. (15) means that the higher f_1 (and the lower f_2 or f_3) is, the more reliable is the first expanded candidate. For the subsequent expanded candidates, their confidence has the following formula: $p(e_l | x) = (1 + \exp(\gamma_0^l + \gamma_1^l f_1 + \gamma_l^l f_l))^{-1}$ ($\gamma_1^l > 0, \gamma_l^l < 0, l \geq 2$), which means that the higher f_1 (and the lower f_l) is, the less reliable are the subsequent expanded candidates.

3.4.2. Combination

Since our main purpose is not to improve first candidate accuracy but to improve the cumulative accuracy of top candidates, instead of adopting complex combination schemes [15], we directly re-rank C_K and E_L to produce Q new candidates (called the combined candidates) in light of the confidence value of the original candidates and the expanded candidates. Owing to some overlapping candidates between C_K and E_L , we choose those with the higher confidence value during ranking. Hence, we can construct the combined candidate set (CCS) $G_Q = \{g_1, g_2, \dots, g_Q\}$ with its corresponding confidence values $h_1 h_2 \cdots h_Q$ ($h_1 \geq h_2 \geq \dots \geq h_Q$). The cumulative accuracy of the combined candidate set for 300 test sets is shown in Table 1 ($K = 30, L = 100$).

From Table 1, we can see that: Although the accuracy of the first candidate in CCS increases slightly in comparison with OCS, the improvement steadily rises with increasing Q . The improvement is only between 0.20% and 0.36%, but the error is reduced greatly. With increasing Q , the error reduction rate gradually rises. For G_1 ($Q = 1$), the error is only reduced by 2.96%;

however, for G_{50} and G_{100} , the error is reduced by 37.88% and 39.22% respectively. This improvement is very beneficial to contextual post-processing.

4. Approximate Chinese Word Matching Method

As mentioned in Section 3.3, if there is no similar character of a included in C_K , CEA fails. In this section, we use the confusion matrix together with lexicon information for further search of the correct character.

Our Chinese lexicon consists of 78,986 words, which are divided into four groups according to word length. They correspond to words of one, two, three and four characters with group size equal to 6,763, 43,727, 15,239 and 13,257, which are called one-character words, two-character words, three-character words and four-character words respectively. According to the statistical results from the *People's Daily* (1993-1994) corpora of about 40 million Chinese characters, there are 20.67 million words altogether, in which one-character words, two-character words, three-character words and four-character words account for 41.54%, 51.85%, 4.70% and 1.93% respectively. The average word length is 1.67. Since the proportion of two-character words is far bigger than that of three-character words and four-character words, we distinguish two-character words from three-character words and four-character words in our method. As there are many candidates in each candidate set, we only consider the top most candidates for approximate matching with words in the lexicon.

4.1. Using Two-character Words

Let $U = u_1u_2$ be a two-character word in the lexicon, where u_1 is the head character and u_2 is the tail character. We define u_1 's capability of constructing two-character words (CCTW) as the number of words whose head character is u_1 . Similarly, we define u_2 's CCTW as the number of words whose tail character is u_2 . In the lexicon, there are 18 head characters and 18 tail characters whose CCTW are more than 100; these are illustrated in detail in the Appendix.

Let $V = v_1v_2$ be the respective top most candidates of two adjoining characters in the sentence S in Fig.1, i.e., $v_1v_2 = c_{t,1}c_{t+1,1}$ ($t = 1, 2, \dots, T-1$). Suppose the accuracy of script recognition without post-processing is not too low. Then, the possibility of error that both v_1 and v_2 are not correctly recognized is very low. For example, if the accuracy without post-processing is 80%, the possibility of both v_1 and v_2 being erroneous is only 0.04%. If $V = v_1v_2$ is a two-character word $U = u_1u_2$, then we need not process v_1 or v_2 . Otherwise, we should take the following two cases into consideration: 1) $v_1 = u_1$ and $v_2 \neq u_2$; 2) $v_2 = u_2$ and $v_1 \neq u_1$. Our aim is to select possible U through V . Applying the rule of maximal posterior probability, we have:

$$\hat{U} = \operatorname{argmax}_U p(U|V) = \operatorname{arg} \max_U p(U) * p(V|U) \quad (16)$$

where U is one of two-character words whose head character is u_1 for the first case or whose tail character is v_1 for the second case; $p(U)$ is the frequency of U .

Owing to the independence between v_1 and v_2 in the recognizer, the probability $p(V|U)$ can be expressed as follows:

$$p(V|U) = p(v_1 v_2 | u_1 u_2) = p(v_1 | u_1) * P(v_2 | u_2) \quad (17)$$

where $p(v_i | u_i)$ ($i = 1, 2$) is an element of P_M in Section 3.1, which stands for the probability of the correct character u_i being recognized as the character v_i .

As far as the first case is concerned, because $v_1 = u_1$, $p(v_1 | u_1)$ can be approximately equal to 1. Similarly, $p(v_2 | u_2) \approx 1$ in the second case. Thus, we can rewrite Eq. (16) as follows:

$$\hat{U} \approx \operatorname{argmax}_U p(U) * p(v_i | u_i) \quad i = 1 \text{ or } 2 \quad (18)$$

According to Eq. (18), we can directly insert the top two-character words with the highest likelihood into the relevant word set in the word graph [20] when doing word-based post-processing. The confidence of u_i can be replaced by the product of the confidence of candidate v_i and the confusion probability $p(v_i | u_i)$.

For instance, a Chinese character string “继续努力(continue to work hard)” is recognized as “继绿努力”. Although the CCTW of the head character “继” is 12 (the relevant tail characters contain “承”, “而”, “父”, “宏”, “母”, “任”, “位”, “续”, “轩”, “英”, “友” and “子”), only “续” is similar to “绿”. Since the confusion probability $P(\text{绿}|\text{续})$ is high, we can exclude the other 11 tail characters from joining the word set. Therefore, even if there is no correct character “续” in OCS, we can recall it by the two-character word matching method.

4.2. Using Three-character Words and Four-character Words

For three-character words and four-character words, we first calculate the edit distance, and then use the knowledge from the confusion matrix to put the most likely words into the relevant word set in the word graph [20]. The approximate four-character word matching can be described as follows:

Let $U' = u_1 u_2 u_3 u_4$ be a four-character word in the lexicon. Let $V' = v_1 v_2 v_3 v_4$ be the respective top most candidates of four adjoining characters in S in Fig.1, i.e., $v_1 v_2 v_3 v_4 = c_{t,1} c_{t+1,1} c_{t+2,1} c_{t+3,1}$ ($t = 1, 2, \dots, T-3$). We only take into account one different character between U' and V' , i.e., the edit distance is 3, their similarity is dependent on the confusion probability $p(v_i | u_i)$. The most likely four-character words \hat{U}' can be decided by the following formula:

$$\hat{U}' = \operatorname{argmax}_{U'} p(U') * p(v_i | u_i) \quad i \in \{1, 2, 3, 4\} \quad (19)$$

For instance, a Chinese character string “人工智能 (artificial intelligence)” is recognized as “人工智脱”. Since the confusion probability $P(\text{脱}|\text{能})$ is high, we can recall the correct character by four-character word matching.

As for three-character words, the process is similar to that for four-character words. Let $V' = v_1 v_2 v_3$ be the respective top most candidates of three adjoining characters in S , i.e.,

$v_1v_2v_3 = c_{t,1}c_{t+1,1}c_{t+2,1}$ ($t = 1, 2, \dots, T - 2$). The edit distance between $V' = v_1v_2v_3$ and three-character word $U' = u_1u_2u_3$ in the lexicon is 2.

5. Post-processing Experiments

We conduct our post-processing experiments on a DELL PC (Pentium-IV, CPU 2.4Ghz, 256MB RAM). ICCR is the same as mentioned in Section 3.2. SLMs are trained by the *People's Daily* (1993-1994) corpora of about 40 million Chinese characters. The *People's Daily* corpora are very comprehensive and the SLMs trained by them can be widely applied to different domains. There are 3,763 characters and 78,993 words (including seven sentence segmentation tokens) respectively in the lexicon. The sizes of the character-based bigram language model and the word-based bigram language model are 5.41MB and 11.62MB respectively.

The objects of post-processing are three scripts handwritten by 30 writers, i.e., *Script A*, *Script B* and *Script C*, whose recognition accuracies without post-processing (*Top1*) are 92.32%, 81.58% and 70.84% respectively. Each script contains about 22,000 characters, covering news, politics, and computers selected from the Internet (the contents are not in the corpus). In the experiments, in order to deal with the sparse data in the SLMs, we use Witten-Bell smoothing [21], which is given below:

$$p_{WB}(s_i | s_{i-1}) = \frac{n(s_{i-1}s_i) + N_{1+}(s_{i-1}\cdot) p(s_i)}{N_{1+}(s_{i-1}\cdot) + \sum_{s_i} n(s_{i-1}s_i)} \quad (20)$$

$$p(s_i) = (n(s_i) + \varepsilon) / \sum_j n(s_j)$$

where $N_{1+}(s_{i-1}\cdot) = |\{s_i : n(s_{i-1}s_i) > 0\}|$ is the number of novel words seen after the history s_{i-1} over the training corpora; $n(s_i)$ and $n(s_{i-1}s_i)$ represent the number of times unigram s_i and bigram $s_{i-1}s_i$ occur in the training corpora; $\varepsilon = 0.01$ is to avoid zero probability. For word-based bigrams and character-based bigrams, s_i stands for a Chinese word and a Chinese character respectively.

5.1. Post-processing based on CEA

In order to verify the performance of the proposed CEA in Section 3, character-based bigram post-processing experiments on the following six candidate sets are carried out:

Org10 - the top 10 original candidates (C_{10});

Com10 - the top 10 combined candidates (G_{10});

Org60 - the top 60 original candidates (C_{60});

Sim60 - the top 10 original candidates (C_{10}) + 5 similar characters for each original candidate;

Mix60 - the top 10 original candidates (C_{10}) + the top 50 expanded candidates (E_{50});

Com60 - the top 60 combined candidates (G_{60}).

Sim60, *Mix60* and *Com60* are produced by CEA. The confidence of candidates in the five candidate sets other than *Sim60* has already been estimated in Section 3.4. For *Sim60*, the confidence of similar characters is supposed to be the same as that of the 10th original candidate. Table 2 shows the experimental results compared to *Top1* and the cumulative accuracy of C_{10} (*Top10*).

Table 2 CEA-based Post-processing Comparison among Different Candidate Sets (%)

	<i>Top1</i>	<i>Top10</i>	<i>Org10</i>	<i>Com10</i>	<i>Org60</i>	<i>Sim60</i>	<i>Mix60</i>	<i>Com60</i>
<i>Script A</i>	92.32	99.31	98.49	98.53	98.79	98.81	98.92	99.01
<i>Script B</i>	81.58	95.73	93.34	93.61	94.80	95.03	96.16	96.33
<i>Script C</i>	70.84	87.94	84.38	85.42	90.02	90.75	92.04	92.83
Average	81.58	94.33	92.07	92.52	94.54	94.86	95.71	96.06
Error correction rate ⁵	—	—	56.95	59.39	70.36	72.10	76.71	78.61
Error reduction rate	—	—	—	5.67	31.15	35.18	45.90	50.32

From Table 2, the experimental results are characterized by the following:

- 1) CEA is fairly effective for contextual post-processing. With the same number of candidates, the recognition accuracies of *Sim60*, *Mix60* and *Com60* are better than that of *Org60*. Among *Sim60*, *Mix60* and *Com60*, *Com60* has the highest accuracy. The experimental results show that combining the original candidates and the expanded candidates can further improve recognition performance, which is in accordance with the results of the samples test in Section 3.4.
- 2) The average accuracy of *Com60* reaches 96.06%, which increases by 14.48% compared to *Top1* while the error correction rate reaches 78.61%. In comparison with *Org10* and *Org60*, *Com60* raises accuracy by 4% and 1.52% respectively, and its error reduction rates are 50.32% and 27.84% respectively.
- 3) It is noted that recognition performance improves with an increasing number of candidates. The average recognition accuracy of post-processing with 60 candidates surprisingly outperforms *Top10*. The recognition accuracies of *Com60* and *Org60* are better than that of *Com10* and *Org10* respectively. Especially, when the script is poorly recognized, increasing the number of candidates is fairly effective. For *Script C*, the recognition accuracy of *Org60* is 5.64% higher than that of *Org10*, while *Com60* improves 7.41% accuracy in comparison with *Com10*.

5.2. Post-processing based on Approximate Word-matching (AWM) Method

In order to verify the performance of the proposed AWM in Section 4, using the word-based bigram language model, the following three post-processing methods are tested:

AWM0 – word-based bigram post-processing with the top 10 original candidates (C_{10});

AWM1 – based on *AWM0*, considering two-character word-matching;

AWM2 – based on *AWM1*, considering three-character and four-character word-matching.

In word-based bigram post-processing, a word graph [20] should be constructed in advance. While considering AWM, we can directly insert possible multi-character words into the word graph. The number of top multi-character words is no more than 3. Table 3 shows the experimental results.

Table 3 AWM-based Post-processing Comparison (%)

⁵ error correction rate = $(1.0 - \frac{\text{the number of errors after post-processing}}{\text{the number of errors before post-processing}}) \times 100\%$

	<i>Top1</i>	<i>Top10</i>	<i>AWM0</i>	<i>AWM1</i>	<i>AWM2</i>
<i>Script A</i>	92.32	99.31	98.73	98.82	98.83
<i>Script B</i>	81.58	95.73	94.01	95.97	96.11
<i>Script C</i>	70.84	87.94	84.92	87.83	88.01
Average accuracy	81.58	94.33	92.55	94.21	94.32
Error correction rate	—	—	59.54	68.55	69.16

From Table 3, the experimental results are characterized by the following:

- 1) In comparison with conventional word-based bigram post-processing, two-character word matching can recall many correct characters beyond C_{10} , and thus improves the accuracy from 92.55% to 94.21%. Considering three-character word and four-character word, the accuracy only rises a little. The average error reduction rate is 23.76%.
- 2) When the accuracy in the recognition of a script without post-processing is rather low, the AWM method is very effective. The accuracies with *Script B* and *Script C* surprisingly outperform *Top10*.

5.3. Hybrid Post-processing based on Integration between Character-based Bigram and Word-based Bigram

Considering the complementary relation between Chinese words and Chinese characters, we can combine word-based bigram post-processing and character-based bigram post-processing [22]. Based on CCS, character-based bigram post-processing using *forward-backward* search [6] is first executed on a big candidate set, which not only improves recognition accuracy, but greatly boosts *Top10* as well. Then, word-based bigram post-processing using the *Viterbi* search is executed on a small candidate set (containing 10 new candidates) to further improve recognition accuracy. This kind of post-processing (called hybrid post-processing) can effectively improve script recognition accuracy while giving due attention to processing speed at the same time (see Section 5.4).

Table 4 Performance of Hybrid Post-processing (%)

	<i>Top1</i>	<i>Top10</i>	<i>Org10</i>	<i>Org60</i>	<i>AWM0</i>	<i>Word60</i>	<i>Com60_1</i>	<i>Int1</i>	<i>Int2</i>
<i>Script A</i>	92.32	99.31	98.49	98.79	98.73	99.10	98.79	99.11	99.18
<i>Script B</i>	81.58	95.73	93.34	94.80	94.01	96.35	96.56	97.14	97.29
<i>Script C</i>	70.84	87.94	84.38	90.02	84.92	92.19	92.97	94.09	94.36
Average accuracy	81.58	94.33	92.07	94.54	92.55	95.88	96.11	96.78	96.94
Error correction rate	—	—	56.95	70.36	59.54	77.63	78.88	82.52	83.39

Table 4 shows the experimental results of seven post-processing methods. In addition to *Org10*, *Org60* and *AWM0* described earlier, the other four post-processing methods are stated as follows:

Word60 – word-based bigram post-processing with the top 60 original candidates (C_{60});

Com60_1 – character-based bigram post-processing using *forward-backward* search with the top 60 combined candidates (G_{60});

Int1 – word-based bigram post-processing with 10 new candidates given by *Com60_1*;

Int2 – based on *Int1*, considering the approximate word-matching method (see *AWM2* in Section 5.2).

From Table 4, the experimental results are characterized by the following:

- 1) Word-based bigram post-processing can achieve an accuracy higher than that of character-based bigram post-processing. The accuracy of *Com60_1* is far higher than *AWM0*, and even higher than *Word60* (the computational cost of *Word60* is huge, see Section 5.4).
- 2) Although *Com60_1* has reached a high accuracy, hybrid post-processing further improves the accuracy of script recognition when word-based bigram post-processing is executed on 10 new candidates given by *Com60_1*. In comparison with *Top1*, *Int1* improves the accuracy to 96.78% from 81.58%, and achieves an error correction rate of 82.52%.
- 3) Integrating *AWM* with *Int1*, we further improve the accuracy of script recognition a little compared to *Int1*. *Int2* reaches the accuracy of 96.94, and achieves an error correction rate of 83.39% in comparison with *Top1*. Compared to the conventional *Org10* and *AWM0*, its error is reduced by 61.41% and 58.93% respectively. Especially, when the script is poorly recognized, our proposed methods are fairly effective (e.g., *Script B* and *Script C*).

5.4. Post-processing Speed

In evaluating the performance of post-processing, memory space and computational cost are also important factors. It is worth noting that character-based bigram post-processing is extremely fast and its processing time rises linearly with the number of candidates. On the other hand, word-based bigram post-processing appears very slow, and its processing time rises exponentially with an increase in the number of candidates [22]. Considering the computational cost of word-based bigram post-processing, we have, in practice, only processed the candidate set in which the first candidate’s confidence is less than 0.99.

As for CEA, additional space is needed to store the confusion matrix P_M (Section 3). Owing to the sparseness of P_M , it can be stored in 786KB using a linear link table. Compared to conventional character-based bigram post-processing, CEA-based post-processing takes more time to obtain the expanded candidate set using 15 original candidates. However, as CEA-based post-processing (such as *Com60_1*) is essentially based on the character bigram, this kind of additional time is far less than that of conventional word-based bigram post-processing with big candidate sets.

As for *AWM*, additional space (608KB) is needed to store the CCTW of all two-character words. The approximate matching of adjoining characters in a sentence with Chinese words in the lexicon is also very fast, and the processing time is almost negligible compared to conventional word-based bigram post-processing.

Since word-based bigram post-processing is executed with only 10 candidates, hybrid post-processing (such as *Int1* and *Int2*) is fairly fast compared to conventional word-based bigram post-processing with big candidate sets.

Table 5 lists the processing time of the various post-processing methods on the three scripts.

Table 5 Post-processing Time (s)

	<i>Org10</i>	<i>Org60</i>	<i>AWM0</i>	<i>Word60</i>	<i>Com60_1</i>	<i>Int1</i>	<i>Int2</i>
<i>Script A</i>	5	30	32	1056	108	148	160
<i>Script B</i>	5	30	53	3097	110	168	169
<i>Script C</i>	5	30	71	6895	111	172	183
Average processing time	5	30	52	3683	110	163	171

As shown in Table 5, *Word60* is extremely time-consuming (3,683s), while our proposed methods’ speeds are comparable to *AWM0* (conventional word-based bigram post-processing with 10 candidates). For *Int2*, the average processing time is 171s, while *Com60_1* only needs 110s. This demonstrates that hybrid post-processing can effectively improve accuracy in the recognition of scripts while being efficient in terms of processing speed at the same time. For a page of 400 handwritten Chinese characters, *Int2* only needs about 3s to complete processing.

6. Conclusion

In order to improve accuracy in the recognition of Chinese scripts, contextual post-processing is necessary. If there is no correct character in the candidate set, little improvement could be made. In this paper, we regard the confusion matrix of a special recognizer as the prior knowledge of a character recognition system. Based on the confusion matrix, we have proposed two methods to allow the correct character to be included in a fixed number of candidates: one is the candidate expansion algorithm; the other is the approximate word-matching method. Experiments show that the two methods are fairly effective in recalling the correct character within a fixed number of candidates, and therefore can improve accuracy in the recognition of Chinese scripts. Hybrid post-processing, which integrates the character-based bigram with the word-based bigram, greatly improves accuracy in the recognition of scripts (totaling about 66,000 characters in the experiments, with an improvement from 81.58% to 96.94%). In particular, our proposed methods are very effective for poorly recognized scripts while being efficient in terms of processing speed at the same time.

Our future research will focus on employing advanced language models (such as word-based trigrams, semantic-based n-grams) and advanced search strategies to further enhance accuracy in Chinese script recognition.

Acknowledgements

This research was supported in part by the Agency for Science, Technology and Research (Grant No. R252-000-123-305) in Singapore, National Science Foundation (Grant No. 69972024) and National “863” High-tech Research and Development Plan (Grant No. 863-306-ZT03-03-1) in P. R. China.

Appendix

Here, we illustrate 18 head characters and 18 tail characters in Table 6 and Table 7 respectively, whose CCTW are more than 100.

Table 6 18 Head Characters and their CCTW

Head character	不	出	大	发	分	高	公	开	上
Capability of constructing words	192	115	337	118	112	129	115	120	161
Head character	水	天	外	无	下	小	一	中	自
Capability of constructing words	138	114	101	119	117	161	147	151	110

Table 7 18 Tail Characters and their CCTW

Tail character	道	地	工	化	口	力	面	气	人
Capability of constructing words	120	131	101	115	105	119	108	132	216
Tail character	山	生	事	手	水	头	心	行	子
Capability of constructing words	125	105	119	111	160	190	150	113	485

References

- [1] Y. Chen, Research on hand-printed Chinese character recognition, Ph.D. dissertation, Tsinghua University, China, 1997.
- [2] H-J Lee, C-H Tung, A Language model based on semantically clustered words in a Chinese character recognition system, *Pattern Recognition* 30(8) (1997) 1339-1346.
- [3] C-H Chang, Simulated annealing clustering of Chinese words for contextual text recognition, *Pattern Recognition Letters* 17(1) (1996) 57-66.
- [4] C-H Tung, H-J Lee, Increasing character recognition accuracy by detection and correction of erroneously identified characters, *Pattern Recognition* 27(9) (1994) 1259-1266.
- [5] P-K Wong, C. Chan, Post-processing statistical language models for a handwritten Chinese character recognizer, *IEEE Trans. Systems, Man and Cybernetics — Part B: Cybernetics* 29(2) (1999) 286-291.
- [6] L.R. Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition, *Proc. of the IEEE* 77(2) (1989) 257-286.
- [7] C.L. Liu, M. Nakagawa, Precise candidate selection for large character set recognition by confidence evaluation, *IEEE Trans. Pattern Analysis and Machine Intelligence* 22(6) (2000) 636-642.
- [8] E. Ishidera, A. Sato, A candidate reduction method for handwritten Kanji character recognition. *Proc. 6th International Conf. on Document Analysis and Recognition*. Seattle, USA, 2001, pp. 8-13.
- [9] L. Xu, A. Krzyzak, C.Y. Suen, Methods of combining multiple classifiers and their applications to handwriting recognition, *IEEE Trans. Systems, Man and Cybernetics* 22(3) (1992) 418-435.
- [10] M.D. Kernighan, K.W. Church, W.A. Gale, A spelling correction program based on a noisy channel model. *Proc. 13th International Conf. on Computational Linguistics*. Helsinki University, Finland, 1990, pp. 205-210.
- [11] X. Tong, D.A. Evans, A statistical approach to automatic OCR error correction in context. *Proc. 4th Workshop on Very Large Corpora*. Copenhagen University, Denmark, 1996, pp. 88-100.

- [12] K. Marukawa, T. Hu, H. Fujisawa, *et al*, Document retrieval tolerating character recognition errors – evaluation and application, *Pattern Recognition* 30(8) (1997) 1361-1371.
- [13] G. Lee, J-H Lee, J. Yoo, Multi-level post-processing for Korean character recognition using morphological analysis and linguistic evaluation, *Pattern Recognition* 30(8) (1997) 1347-1360
- [14] L. Zhang, M. Zhou, C.N. Huang, *et al*, Automatic detecting/correcting errors in Chinese text by an approximate word-matching algorithm. Proc. 20th Annual Meeting of the ACL. Hong Kong, China, 2000, pp. 248-254.
- [15] A.K. Jain, R.P.W. Duin, J. Mao, Statistical pattern recognition: a review, *IEEE Trans. Pattern Analysis and Machine Intelligence* 22(1) (2000) 4-37.
- [16] X. Lin, X. Ding, M. Chen, *et al*, Adaptive confidence transform based on classifier combination for Chinese character recognition, *Pattern Recognition Letters* 19(10) (1998) 975-988.
- [17] K.T. Ho, J.J. Hull, S.N. Srihari, Decision combination in multiple classifier systems. *IEEE Trans. Pattern Analysis and Machine Intelligence* 16(1) (1994) 66-75.
- [18] Y. Li, X. Ding, Evaluation of character candidate confidence measure using logistic regression model, *Pattern Recognition and Artificial Intelligence* 15(2) (2002) 160-166 (in Chinese).
- [19] D.W. Hosmer, S. Lemeshow, *Applied logistic regression*. Wiley, New York, 1989.
- [20] H-Y Gu, C-Y Tseng, L-S Lee, Markov modeling of mandarin Chinese for decoding the phonetics sequence into Chinese characters, *Computer Speech and Language* 15(4) (1991) 363-377.
- [21] S.F. Chen, J. Goodman, An empirical study of smoothing techniques for language modeling, *Computer Speech and Language* 13(4) (1999) 359-394.
- [22] Y. Li, X. Ding, C.L. Tan, Combining character-based bigram with word-based bigram in contextual post-processing for Chinese script, *ACM Trans. Asian Language Information Processing* 1(4) (2002) 297-309.

About the Author – YUAN-XIANG LI is a research fellow with the Department of Computer Science, School of Computing, National University of Singapore. He received his B.E. degree in Communication and Electronic Engineering in 1990 from Nanjing Institute of Communication Engineering, China, and his Ph.D. degree in Signal and Information Processing in 2001 from Tsinghua University, China. His research interests include character recognition, contextual post-processing, statistical language model, Chinese information processing, image processing and data compression.

About the Author -- CHEW LIM TAN is an associate professor with the Department of Computer Science, School of Computing, National University of Singapore. He received his B.Sc. (Hons) degree in physics in 1971 from the University of Singapore, his M.Sc. degree in radiation studies in 1973 from the University of Surrey, UK, and his Ph.D. degree in computer science in 1986 from the University of Virginia, U.S.A. His research interests include document image and text processing, neural networks and genetic programming. He has more than 170 research publications in these areas. He is an associate editor of *Pattern Recognition* and has served on the program committees of International Conference on Pattern Recognition (ICPR) 2002, Graphics Recognition Workshop (GREC) 2001 and 2003, Web Document Analysis Workshop (WDA) 2001 and 2003, Document Image Analysis and Retrieval Workshop (DIAR) 2003, Document Image Analysis for Libraries Workshop (DIAL) 2004, and International Conference on Document Analysis and Recognition (ICDAR) 2005. He is a senior member of IEEE.

About the Author -- XIAOQING DING is a professor with the Department of Electronic Engineering, Tsinghua University, Beijing, China. She graduated from Tsinghua University and won the gold medal for best graduating students in 1962. She has researched and developed a series of Chinese character recognition systems, which are among the foremost internationally. She has won numerous Honors and Awards, such as the 2nd and 3rd class *China National Scientific and Technical Progress Award*, in 1999 and 1992, respectively. Her research interests include image processing, pattern recognition, character recognition, document analysis, computer vision, multimedia information processing and video surveillance.

About the Author -- CHANGSONG LIU is an associate professor with the Department of Electronic Engineering, Tsinghua University, China. He received his B.E. degree in Mechanics Engineering in 1992 and his M.E. degree in Signal and Information Processing in 1995 from Tsinghua University. He has participated in developing a series of Chinese character recognition systems, which are among the foremost internationally. His research interests include image processing, character recognition, document analysis, multimedia information processing and Chinese information processing.