Pattern Anal Applic (2005) 8: 272–286
DOI 10.1007/s10044-005-0009-3

THEORETICAL ADVANCES

Yuan-Xiang Li · Chew Lim Tan · Xiaoqing Ding

A hybrid post-processing system for offline handwritten Chinese script recognition

Received: 4 September 2004 / Accepted: 31 August 2005 / Published online: 21 October 2005

Abstract In the recognition of offline handwritten Chinese scripts, contextual post-processing plays a vital role in improving accuracy. In this paper, we systematically analyze the key factors that have an impact on the performance of contextual post-processing: statistical language models (LMs), candidate confidence, candidate set size, and search strategy. We then present a hybrid post-processing system, which integrates various kinds of information available. Next, we investigate seven LMs, four estimation methods of candidate confidence and different size of candidate set, and illustrate their influence on the performance of contextual post-processing in detail. Experimental results justify that the performance of the LMs are affected by training corpora size, smoothing method, and model pruning, and that lower perplexity correlates with a high accuracy. Comparing different estimation methods of candidate confidence shows that, it is vital to the contextual post-processing. We also show that allowing the correct characters to be captured in a limited number of candidates is extremely important for obtaining good post-processing performance. By adopting the hybrid post-processing, we can obtain high accuracy while paying attention to post-processing speed and memory space at the same time. It is shown that the average recognition accuracy of three Chinese scripts (about 66,000 characters in total) can reach 97.65%, which means 87% error correction rate in comparison with the 81.58% average accuracy before post-processing. In the end, we give some proposals for choosing a proper post-processing method for real script recognition tasks.

Keywords Chinese character recognition · Contextual post-processing · Statistical language model · Perplexity · Candidate confidence · Candidate set size

Y.-X. Li
Institute of Meteorology,
PLA University of Science and Technology,
Nanjing 211101, People's Republic of China

C. L. Tan (✉)
School of Computing, National University of Singapore,
Singapore 117543, Singapore
E-mail: tancl@comp.nus.edu.sg
Tel.: +65-6874-2900
Fax: +65-6779-4580

X. Ding
Department of Electronic Engineering, Tsinghua University,
Beijing 100084, People's Republic of China

1 Originality and contributions

Previous works of contextual post-processing for Chinese script recognition have mostly employed class-based language models (LMs) and only ten candidates. This paper systematically presents the key factors that have an impact on the performance of contextual post-processing: statistical LMs, candidate confidence, and candidate set size. Except for class-based LMs, we also investigate the conventional character-based and word-based LMs, and hybrid LMs by combining word-based LMs and class-based LMs. We compare four estimation methods of candidate confidence and indicate that candidate confidence is vital to the contextual post-processing. We discuss the influence of candidate set size on the post-processing time and accuracy, and propose an empirical method of estimating the suitable number of candidates for each script. We build a hybrid post-processing system, which makes full use of the various sources of information available. This kind of hybrid post-processing can effectively improve script recognition accuracy while giving due attention to both processing speed and memory space at the same time. Finally, we give proposals for choosing a suitable post-processing method according to the requirement of a practical recognition system.

To the best of our knowledge, this is the first work that systematically investigates the factors that have an impact on the performance of contextual post-processing for offline handwritten Chinese script recognition. This work is also readily applicable to online handwritten Chinese script recognition.

2 Introduction

Recognizing offline handwritten Chinese characters is still a challenging pattern recognition problem [1, 2]. Mainly because of the large character set, complex character shapes, many confusable subsets of characters with only slightly different shapes, and great variations of writing style in both shape and thickness of strokes [3, 4], it is therefore difficult to significantly improve the accuracy of Chinese script recognition in an offline handwritten isolated Chinese character recognition system.

Statistical language models (LMs) have been widely employed for the contextual post-processing to improve the accuracy in recognizing the Chinese scripts [5–8]. In some earlier literatures, owing to the limitation of the corpus size and the required memory, class-based LMs were often used. Tung and Lee [5] used POS (parts-of-speech) bigram LMs, Chang [6] used bigram LMs based on words clustered by simulated annealing method, Lee and Tung [7] used semantically clustered word-based bigram LMs, like Wong and Chan [8] who also used word-class bigram LMs. Class-based LMs have proved effective for training on small datasets and for fast LM adaptation. For large training datasets, word-based LMs are still superior in capturing the collocational relations between words [9]. With the rapid advancement of computer technology, it is now feasible to obtain large-scale corpora and to execute a large LM with many parameters.

In the Chinese language, conventional n-gram LMs can be based either on Chinese words or on Chinese characters. In our previous works [10, 11], we employed conventional n-gram Chinese LMs including character-based bigram, character-based trigram, and word-based bigram in the post-processing of script recognition. On the other hand, class-based LMs have frequently been shown to improve the performance of speech recognition systems when combined with conventional word-based LMs even when a large amount of training data is available [12]. So, in this paper, in addition to the traditional class-based LMs and conventional n-gram LMs, we will also employ hybrid bigram LMs in the post-processing, which combine both word and class-based bigrams.

Due to the large number of characters in Chinese character recognition, the number of candidates is usually limited. When executing the post-processing using class-based LMs [5–8], the number of candidates was always not more than 10. In practice, for well-recognized scripts, the top ten 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. Apparently, if there is no correct character included in the candidate sets, it is impossible to correct the errors in the recognizer no matter how precise the LMs are. On the other hand, if one recognition result (the first candidate) is very reliable, we can take it as the correct character; otherwise, we should measure its reliability. This problem is called candidate confidence estimation [13]. In this paper, we will also investigate the influence of candidate confidence and candidate set size on the performance of post-processing.

For script recognition, high accuracy is certainly the most important aspect to be pursued. However, the other two aspects, namely memory requirement and computational complexity are also important in the real recognition tasks. While [11], considering the complementary action between the Chinese characters and words [14], the character-based bigram post-processing and the word-based bigram post-processing were combined to improve the script recognition accuracy while giving attention to processing speed at the same time.

The aim of this paper is to integrate the various kinds of information available to construct a proper post-processing system in real Chinese script recognition tasks. The rest of this paper is organized as follows: in Sect. 3, we first analyze the factors affecting the performance of contextual post-processing for Chinese script recognition and then present a hybrid post-processing system that can make use of the various kinds of information available. Statistical LMs are briefly introduced in Sect. 4. The problem of candidate confidence estimation is introduced in Sect. 5 and Sect. 6 discusses the selection of suitable candidate set size, and the modification of candidate set. Sect. 7 shows the experimental results in detail and Sect. 8 gives the proposals in choosing a suitable post-processing method according to the requirement of a practical recognition system. Finally, the conclusion is given in Sect. 9.

3 Hybrid post-processing system

3.1 Problem formulation

A typical Chinese script recognition system is depicted in Fig. 1. The input to the system $X = x_1\ x_2\ \ldots\ x_T$ is a sequence of handwritten Chinese character images, where x_t is the tth character image, and T is the length of X. An example of X is illustrated in Fig. 2, where its corresponding correct sentence is "语言模型千差 (LMs differ in thousands of ways)". Let $S = s_1\ s_2\ \ldots\ s_T$ be a sequence of Chinese characters given by an isolated Chinese character recognizer (ICCR), in which each output s_t may include the top K candidates $c_1 c_2\ \ldots\ c_k$ with the corresponding distance measurement values $D = d_1\ d_2\ \ldots\ d_K$. The output of the system $O = o_1\ o_2\ \ldots\ o_T$ is the final Chinese sentence.

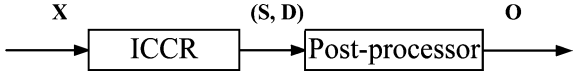


Fig. 1 The basic framework of Chinese scrip recognition

Fig. 2 An example for a sequence of Chinese character images

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 possible sentence from all the K^T sentences. By applying the rule of maximal posterior probability, the output O in Fig. 1 can be formulated as [10]:

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

where $p(S)$ denotes a statistical LM; $p(s_t|x_t)$ denotes the posterior probability of s_t given x_t, which can be computed through candidate confidence [13].

3.2 Analysis of post-processing factors

According to (1), the technology of contextual post-processing can be expressed as follows: under the joint action of $p(S)$ and $p(s_t|x_t)$, a path searching strategy [15–18] is employed to select the most possible sentence from the candidate sets given by the ICCR. In other words, there are four key factors that have an impact on the performance of contextual post-processing, namely LMs, candidate confidence, candidate set size, and search strategy.

From the viewpoint of system, we should systematically take into account these four factors. First, precisely estimating both $p(S)$ and $p(s_t|x_t)$ is the key to obtain the optimal O. If we cannot appropriately estimate $p(S)$ and $p(s_t|x_t)$, the accuracy of script recognition may even decrease somewhat after post-processing. Second, if there is no correct character in a limited number of candidates, the errors cannot be corrected in the ICCR, no matter how precise both $p(S)$ and $p(s_t|x_t)$ are. So, we should try to capture the correct character in a limited candidate set. Third, we must employ an efficient search strategy to obtain O from as many possible sentences produced from all the candidate sets.

$p(S)$ can be estimated from linguistic corpora, which is regarded as a kind of prior knowledge. $p(s_t|x_t)$ is computed through an estimation of candidate confidence, which can be learned from the training samples.

Since the Chinese character set is very large, if the number of candidates is not constrained, the search space will be huge, and so searching O would be computationally expensive. In order to capture the correct character in a limited candidate set, a confusion matrix of Chinese characters can be employed [10, 19].

As for the path searching, there are many algorithms in speech recognition [15–17], such as dynamic programming method, Viterbi search, forward–backward search, A* algorithm, beam search, etc. Most of the search techniques in handwriting recognition are inherited from speech recognition [18].

3.3 Architecture of hybrid post-processing system

In this subsection, in order to obtain the overall optimal post-processing performance for script recognition, we present a hybrid post-processing system that integrates the various kinds of information available, as shown in Fig. 3.

In contrast to Fig. 1, the post-processor consists of eight modules in Fig. 3. For script recognition tasks, except that *Statistical LMs* and *Confidence Evaluation* which are integral modules, the other six modules may be optional. *Character-based Post-processing* is a post-processing module using conventional character-based LMs. *Word-based Post-processing* is also a post-processing module, which can employ traditional class-based LMs, or conventional word-based LMs, or hybrid LMs combining word-based LMs and class-based LMs. *Word Graph Construction* is a prerequisite module to *Word-based Post-processing*, which constructs a word graph [15] to carry out word-based post-processing.

Script Quality Evaluation is an optional module, which can roughly estimate the accuracy of script recognition before post-processing and further decide the suitable number of candidates in post-processing. Candidate set expansion (CSE) is also an optional module, which uses the ICCR characteristics of errors (represented by a confusion matrix) and the original candidates produced in ICCR to conjecture the most likely correct character. *Approximate Word Match* is another

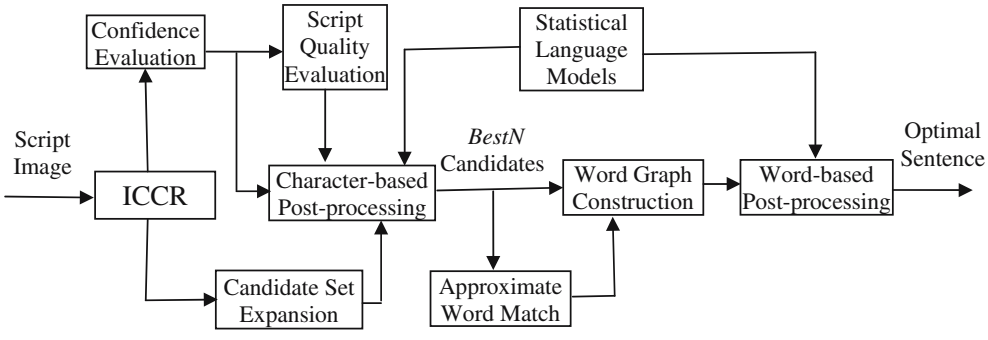


Fig. 3 The schematic diagram of hybrid post-processing system

optional module, which performs an approximate matching of adjoining characters in a sentence with Chinese words so as to recall the most likely correct character. The above three optional modules aim to solve the problem of candidate set size.

Character-based Post-processing and *Word-based post-processing* can be employed alternatively or jointly in script recognition tasks. Employing either of them means a common post-processing, while a hybrid post-processing employs both of them.

By integrating all the above modules, the work flow of recognizing a script is stated as follows: after recognizing the script image by ICCR, two types of information are produced: one is candidate information and the other is candidate measurement information. Using the measurement information, we can estimate the confidence of a candidate and thus estimate the script quality. Using the candidate information, we can modify the original candidate set to allow the correct character to be captured in a limited number of candidates. Under the joint action of candidate confidence and character-based LMs, a *forward–backward* search [16] is first employed to produce new candidates (called *BestN* candidates, *BestN* usually equals 10) from the modified candidate set. Then, based on the *BestN* candidates, we can further modify the candidate set using AWM, and construct a word graph to carry out word-based post-processing by the *Viterbi* search [15].

By synthetically employing various kinds of information available, this hybrid post-processing system can obtain high script recognition accuracy while giving due attention to the processing speed and memory space at the same time.

In this paper, we will not compare the various search strategies, but only use the Viterbi search and forward-backward search. In the following sections, we will discuss statistical language model, candidate confidence, and candidate set size, respectively.

4 Statistical language model

The most widely used LMs, by far, are the n-gram models. In the Chinese language, a word is a basic syntax-meaningful unit. Although a word consists of one or more Chinese characters, each character in the word also has the definite meaning in itself. Thus, conventional n-gram Chinese LMs can be based on either characters or words.

4.1 Description of Chinese LMs

Based on Chinese characters, for $n=2$, 3, we have the character-based bigram model (*charBi*) and the character-based trigram model (*charTri*) expressed, respectively as follows:

$$p(S) = p(s_1)\prod_{t=2}^{T} p(s_t|s_{t-1}) \tag{2}$$

$$p(S) = p(s_1)p(s_2|s_1)\prod_{t=3}^{T} p(s_t|s_{t-2}s_{t-1}) \tag{3}$$

Considering Chinese words, we use $S = w_1 w_2, \ldots, w_{T'}$ (S contains T' words, $T' \leq T$) instead of $S = s_1\, s_2 \ldots s_T$. For example, in Fig. 2, $S =$ 语言 模型 千差万别 where $T=8$ and $T'=3$. Based on Chinese words, for $n=2$, we have the word-based bigram model (*wordBi*) expressed as follows:

$$p(S) = p(w_1)\prod_{t=2}^{T'} p(w_t|w_{t-1}) \tag{4}$$

Considering Chinese word classes, we partition the vocabulary of size W into a fixed number G of word classes by mapping function $G{:}w \rightarrow g(w)$, in which each word w of the vocabulary belongs exactly to one class $g(w)$. For a class-based bigram model (*classBi*), we then have:

$$p_c(w_t|w_{t-1}) = p(g(w_t)|g(w_{t-1})) \times p(w_t|g(w_t)) \tag{5}$$

where $p_c(w_t|w_{t-1})$ can be used to replace $p(w_t|w_{t-1})$ in (4).

For obtaining word classes, the exchange algorithm using the criterion of perplexity improvement was employed [12]. In this paper, we test 500 and 2,000 word classes, from which we obtain the class-based bigram models called *class500* and *class2k*, respectively.

While class-based LMs generalize better to unseen word sequences, word-based LMs in general have better performance when enough training corpora are available. It is desirable to retain the advantages of each of these models by combining their word predictions [20]. So, we can construct a hybrid bigram model (*hybridBi*) that combines word-based bigrams with class-based bigrams by linear interpolation expressed as follows:

$$p_h(w_t|w_{t-1}) = \lambda \times p(w_t|w_{t-1}) + (1-\lambda) \times p_c(w_t|w_{t-1}) \tag{6}$$

The optimal value of λ can be estimated by optimizing over the held-out data. Interpolating *wordBi* with *class500* and *class2k*, we obtain *hybrid500* and *hybrid2k*, respectively.

4.2 Perplexity

The most common metric for evaluating a LM is the probability that the model assigns to characters or words in a test corpus, or the perplexity [21]. The perplexity (PP) can be defined as follows:

$$PP = p(M)^{-1/L} \tag{7}$$

where M is a sequence of the test corpus with length L (the total number of characters); $p(M)$ is the probability of M, which can be computed using various LMs trained from corpora. Intuitively, PP can be interpreted as the average number of possible successors of a Chinese word

or character. In applications, lower PP normally leads to better performance [21].

For a test corpus, the PP of a given LM is affected by the size of training corpus, the smoothing method for unseen n-grams, and count cutoffs. For statistical LMs, smoothing technology for sparse data is a central issue. Chen and Goodman [22] investigated the most widely used smoothing methods for addressing the English sparse data issues. For large training corpora, count cutoffs (pruning) are often used to restrict the size of the n-gram model constructed. With model pruning, all n-grams with fewer than a given number of occurrences in the training corpora are ignored.

In Sect. 7.2, we will display the PPs of various LMs and give the comparative experiments and results in detail.

5 Candidate confidence

In this section, we discuss the estimation of the posterior probability $p(s_t|x_t)$ of a candidate c_k in (1). Without loss of generalization, we replace $p(s_t|x_t)$ with $p(c_k|x)$[1] in the following statements.

It is very difficult to directly obtain the posterior probability of a candidate [13, 23]. The minimal distance classifier is a common selection in ICCR for its simplicity, where the decision based on the maximal posterior probability is converted into the decision based on the minimal distance. The higher the posterior probability, the lesser is its correlative distance.

Confidence measurement is an important issue in character recognition, which is a quantitative estimation of the potential correctness of recognition candidates. For a given candidate, the confidence value ideally equals the posterior probability. Several approaches have been proposed to convert the distance value of c_k into its confidence value.

Xu et al. [23] used an empirical distance formula (EDF1) to compute $p(c_k|x)$, as expressed in (8).

$$p(c_k|x) = \frac{(1/d_k)}{\sum_{i=1}^{K} 1/d_i},\ k = 1, 2, \ldots,\ K \tag{8}$$

Lee and Chen [24] used another empirical distance formula (EDF2) to compute $p(c_k|x)$, as expressed in (9).

$$p(c_k|x) = \frac{\text{score}_k}{\sum_{i=1}^{K} \text{score}_i},\ score_k = \frac{1}{d_k - d_1 + 1},\quad k = 1, 2, \ldots, K \tag{9}$$

Lin et al. [25] proposed the adaptive confidence transform (ACT) method to estimate $p(c_k|x)$. The ACT method first constructs a so-called generalized confidence from $d_1\ d_2\ \ldots\ d_K$, and then maps a generalized confidence into probability through a transform. This transform can be trained using typical samples.

Li and Ding [26] proposed the logistic regression model (LRM) to directly convert the distance measurement of a candidate c_k into its confidence value. The LRM method defines $d_1\ d_2\ \ldots\ d_K$ as independent variables and the correctness of c_k as a dependent variable (Y). If c_k is the correct character, $Y = 1$; otherwise, $Y = 0$. The mean value of Y can be regarded as $p(c_k|x)$, which is expressed as:

$$p(c_k|x) = \left(1 + \exp\left(\beta_0^k + \sum_{i=1}^{z} \beta_i^k d_i\right)\right)^{-1},\quad 1 \leqslant k \leqslant K \tag{10}$$

where $\beta^k{}_i$ is the regression coefficient, which can be estimated by *maximum likelihood estimation* [27, 28] through the recognition results of some training samples. z is the order of regression model.

In Sect. 7.3, we will show the influence of different estimation methods of candidate confidence on script recognition accuracy.

6 Candidate set size

The size of candidate set is vital to both the processing speed and the improvement of recognition accuracy in the contextual post-processing.

6.1 Analysis of candidate set size

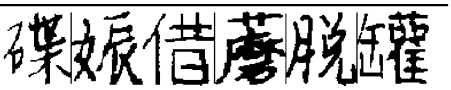
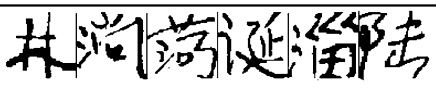
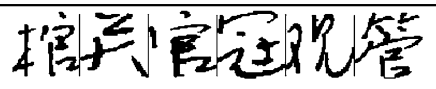
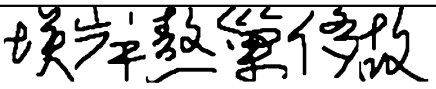
Owing to the large character set, the number of candidates K in a candidate set is usually limited. If K is too small to capture the true candidate in a limited candidate set, we cannot select the correct character through post-processing, no matter how precise both the LMs and candidate confidence are. On the other hand, if K is very large in order to capture the true candidate, the contextual post-processing with the excessive number of candidates would be very time-consuming even if the correct character can be selected [11]. In order to obtain an ideal performance, how many candidates are suitable in the contextual post-processing?

In our experiment, 300 test sample sets are divided into five classes (see Sect. 7.1): (Class-A) best-quality samples with an accuracy of more than 90%; (Class-B) good-quality samples with an accuracy between 80 and 90%; (Class-C) fair-quality samples with an accuracy between 70 and 80%; (Class-D) bad-quality samples with an accuracy between 60 and 70%; (Class-E) worst-quality samples with an accuracy below 60%. Some samples for the five classes are illustrated in Table 1. With the above five classes of samples, Table 2 shows the cumulative recognition accuracy[2] (CRA) with increasing K.

[1] x denotes a character image and c_k is the kth recognition candidate of x.

[2] Cumulative recognition accuracy = (1.0×the number of correct characters in the top K candidates / total characters)×100%

Table 1 Some samples with various writing styles

ClassA	
ClassB	
ClassC	
ClassD	
ClassE	

From Table 2, we can see that:

1. For Class-A samples, the first candidate's accuracy (FCA) is high and CRA rises slowly with increasing K. The CRA discrepancy between $K=10$ and $K=1$ is less than 5%, whereas the CRA discrepancy between $K=50$ and $K=20$ is only 0.09%. $K=20$ may be enough in this case.
2. For Class-C samples, CRA rises fast with increasing K, where the discrepancy between $K=10$ and $K=1$ is about 20%, whereas the discrepancy between $K=90$ and $K=20$ is near 2%. CRA with $K=100$ is only 0.06% higher than that with $K=90$. $K=90$ may be enough in this case.
3. For Class-E samples, FCA is very low and CRA rises rapidly with increasing K. The discrepancy between $K=10$ and $K=1$ surprisingly reaches 30%, and the discrepancy between $K=100$ and $K=20$ also reaches 6%. CRA with $K=100$ is still 0.23% higher than that of $K=90$. In this case,$K=100$ is not enough yet.

Intuitively, the number of candidates K should be small if a script is well recognized (FCA is high); otherwise, K should be large. Fig. 4 illustrates an empirical relationship curve between K and FCA, where higher FCA leads to smaller K and lower FCA leads to larger K. According to this empirical curve, we can select a suitable number of candidates through estimating FCA for a script, as shown in Sect. 7.3.

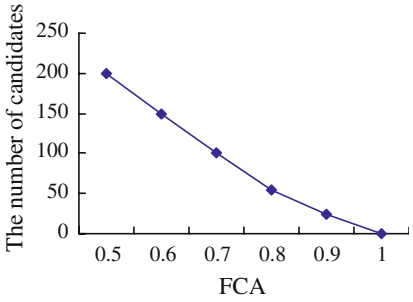


Fig. 4 An empirical curve between the number of candidates and FCA

6.2 Candidate set modification

From Table 2, we know that for the poorly recognized samples, even using 100 candidates may sometimes fail to capture the correct characters. On the other hand, post-processing with a large number of candidates could be very time-consuming. The inclusion of potentially correct characters in a limited number of candidates is very important for improving the contextual post-processing performance of script recognition in both accuracy and speed.

For a special ICCR, it has its own characteristics of errors which are based on its underlying understanding of how some of the 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 character recognition system.

In our previous work [10], based on the confusion matrix, two methods were proposed to recall the potentially correct characters. The one called the CSE method is to use the original candidates in a candidate set to conjecture the most likely correct characters, and then combine them with the original candidates to produce a new candidate set. The other one called the approximate word match (AWM) method performs an approximate matching of adjoining characters in a sentence with Chinese words so as to recall the most likely correct character.

In Sect. 7.4, we will display the influence of candidate set size on both script recognition accuracy and post-processing speed. The influence of CSE and AWM on hybrid post-processing will be shown in Sect. 7.5.

Table 2 CRA comparison for the five classes of sample sets (%)

K	1	10	20	30	40	50	60	70	80	90	100
Class-A	94.68	99.59	99.76	99.81	99.83	99.85	99.86	99.87	99.87	99.88	99.88
Class-B	86.13	98.09	98.86	99.13	99.26	99.35	99.41	99.46	99.49	99.52	99.54
Class-C	76.4	95.09	96.9	97.62	97.98	98.23	98.41	98.56	98.68	98.77	98.83
Class-D	66.74	90.6	93.76	95.16	95.91	96.48	96.85	97.13	97.28	97.48	97.64
Class-E	57.86	85.63	90.37	92.39	93.6	94.42	95.01	95.38	95.8	96.13	96.36

7 Experimental results

This section illustrates the influence of various factors on the performance of contextual post-processing in detail, through a series of experiments. We have conducted these experiments on a DELL PC (Pentium-IV, CPU 2.4 GHz, 256 Mb RAM).

7.1 Experimental design

The experiments aims to address the issues discussed in sections 4 to 6 and to finally test our hybrid models. Thus there will be four experiments in stages to be discussed in detail in Sects. 7.2–7.5: (1) experiments on LMs, (2) experiments on candidate confidence, (3) experiments on candidate set size and (4) experiments on hybrid models.

The corpora used here come from the *People's Daily* (1993–1996). *People's Daily* corpora are very comprehensive and LMs trained by them can be widely applied to different domains. In the Chinese lexicon, there are 3,763 character types and 79,029 word types, respectively. There are four training corpora, named as *set1* to *set4*. *set1*, *set2* and *set3* consist of 1993 newspapers (19.4 million characters, 12.5 million words), 1993–1994 newspapers (39.4 million characters, 25.5 million words) and 1993–1995 newspapers (63.7 million characters, 41.4 million words). *set4* consists of *set3* and 1996 newspapers excluding November and December, which contain 83.8 million characters (54.4 million words). The texts of November 1996 are used as held-out data. The test corpus is made of the texts containing 2.2 million characters (1.4 million words) from December 1996. In the following experiments except for Sect. 7.2.1, *set4* is referred to as the training corpus. In our experiment, seven LMs described in Sect. 4.1 are trained, namely *charBi*, *charTri*, *wordBi*, *class500*, *class2k*, *hybrid500* and *hybrid2k*.

"*THOCR'97 Synthetical and Integrated Chinese Character Recognition System*" [29] 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 accuracy[3] (RA) of 89.05% are regarded as training sets. The remainder, containing 300 sample sets, is regarded as test sets with an average RA of 87.85%. Every sample set consists of 3,755 offline handwritten Chinese simplified characters.

The three scripts used in the post-processing experiment were handwritten by 30 writers, i.e., *Script-A*, *Script-B* and *Script-C*, whose RAs without post-processing (Top1) are 92.32, 81.58, and 70.84%, respectively. Their CRAs of the top ten candidates (Top10) are 99.31, 95.73 and 87.97%, respectively. Each script consists of about 22,000 characters, involving news, politics, and computer selected from the Internet (the contents are not in *set4*).

[3] recognition accuracy = (1.0 − the number of incorrect characters / total characters)×100%

7.2 Experiments on language models

As stated in Sect. 4, the performance of a given LM is affected by the size of training corpus, the smoothing method, and the pruning threshold. With the above seven LMs, experiments in this subsection investigate the influence of these three LM factors on both the PPs for the test corpus and the RA for *Script-B*. While doing the following contextual post-processing, ten original candidates and candidate confidence estimated by LRM (see Sect. 7.3) are employed.

7.2.1 *The size of training corpus*

With different corpus size, we test RAs and PPs using the Jelinek-Mercer smoothing method [22], as shown in Figs. 5 and 6, respectively.

From Fig. 5, we can see that:

1. Obviously, *charTri* has the highest RA while *charBi* has the lowest RA. *wordBi* has a higher RA than *charBi*.
2. The RA of *classBi* is higher than that of *charBi*, but a little lower than that of *wordBi*. Obviously, *class2k* has a higher RA than *class500*.
3. Both *hybrid500* and *hybrid2k* have higher RAs than *wordBi*. It is worth noting that *hybrid500* almost has the same RA as *hybrid2k*. This result indicates that more word classes are hardly beneficial for constructing *hybridBi*. Small classes may be enough to construct *hybridBi*.
4. With an increasing size of the training corpus, the RAs of all LMs increase. Note that the RAs of both *classBi* and *hybridBi* increase slowly while the RAs of conventional n-gram LMs increase fast. For small training corpora, *hybridBi* is beneficial to improve RA. For example, with *set1*, its RA is even a little higher than that of the RA of *charTri*.

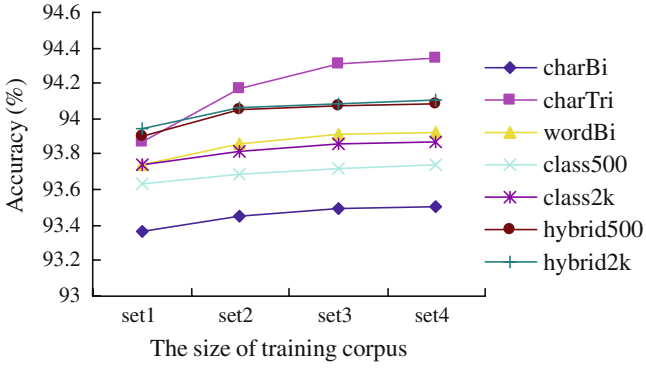


Fig. 5 The RA affected by the size of training corpus

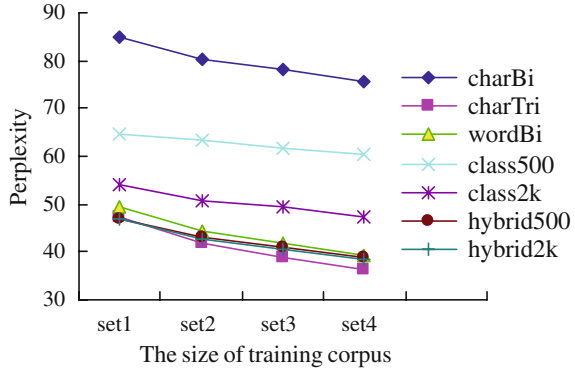


Fig. 6 Perplexity affected by the size of training corpus

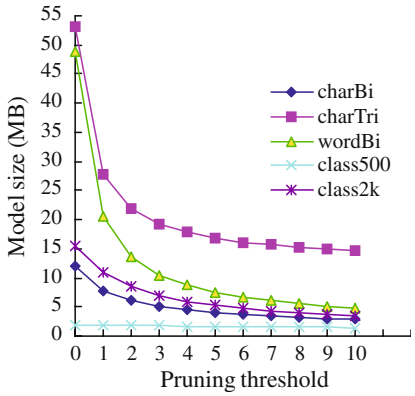


Fig. 7 The model size affected by model pruning

From Figs. 5 and 6, we can see that lower PP correlates with a higher accuracy. Obviously, *charBi* has the highest PP while *charTri* has the lowest PP. *wordBi* has a little higher PP than *charTri*.

7.2.2 Smoothing method

Comparing the following four smoothing methods: Jelinek-Mercer (J-M), Witten-Bell (W-B), Katz, and Kneser-Ney (K-N) smoothing (see details in [22]), we test RAs and PPs for the seven LMs, as shown in Tables 3 and 4, respectively.

From Table 3, we can see that different smoothing methods could impact RAs to some extent; however, the discrepancy for a given LM is trivial. For implementing simplicity, the J-M smoothing method is a good method for the contextual post-processing and is adopted in the following post-processing.

From Tables 3 and 4, we can also see that lower PP correlates with a higher accuracy.

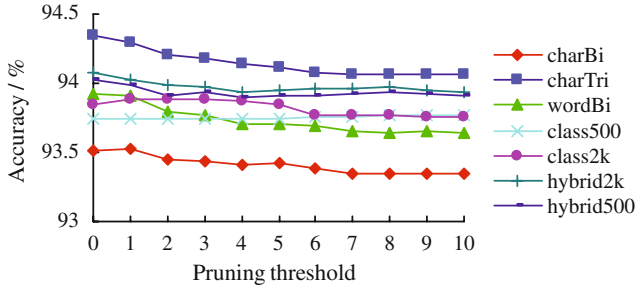


Fig. 8 The RA affected by model pruning

7.2.3 Pruning language model

Figure 7 demonstrates that the memory requirement varies with pruning threshold (PT) for *charBi*, *charTri*, *wordBi*, *class500* and *class2k*. Without count cutoffs, the sizes of these five LMs are 12, 53, 49, 2, and 16 Mb, respectively. Since *hybridBi* consists of *wordBi* and

Table 3 RA affected by different smoothing methods (%)

	J-M	W-B	Katz	K-N
CharBi	93.51	93.56	93.58	93.58
CharTri	94.34	94.27	94.38	94.30
WordBi	93.92	93.96	93.97	94.04
class500	93.74	93.77	93.74	93.77
class2k	93.85	93.80	93.82	93.77
hybrid500	94.08	94.12	94.14	94.14
Hybrid2k	94.11	94.05	94.02	93.99

Table 4 Perplexity affected by different smoothing methods

	J-M	W-B	Katz	K-N
charBi	75.7	75.2	74.9	74.9
charTri	36.2	34.9	34.6	35.6
wordBi	39.2	38.4	37.9	37.5
class500	60.3	58.8	58.8	58.8
class2k	47.5	46.4	46.3	46.4
hybrid500	38.7	37.9	37.5	37.2
hybrid2k	38.5	37.7	37.3	37.0

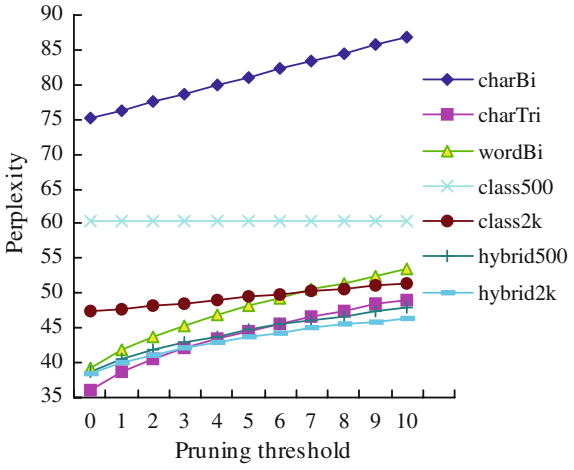


Fig. 9 Perplexity affected by model pruning

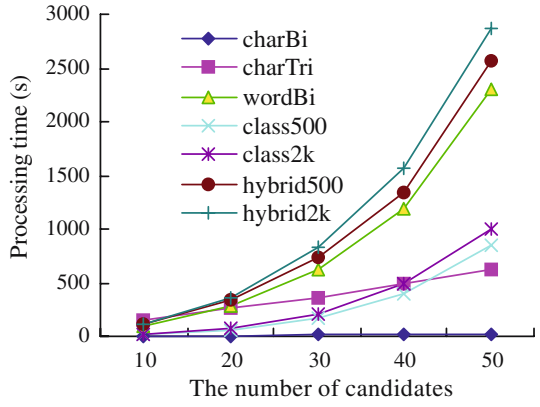


Fig. 10 Post-processing time as a function of the number of candidates

classBi, its size is certainly larger than *wordBi*. *hybrid500* and *hybrid2k* need 51and 65 Mb, respectively.

With increasing PT, except for *class500*, the other LMs' sizes decrease exponentially. Especially, pruning the n-grams with one occurrence can greatly decrease the size of a model. For example, the memory space is only 28 Mb for *charTri* and 20 Mb for *wordBi* in this case. The effects of count cutoffs on both RAs and PPs are shown in Fig. 8 and Fig. 9, respectively.

From Fig. 8, we can see that:

1. With increasing PT, RA decreases for *charTri* and *wordBi* ; for *charBi*, its RA decreases very slowly as compared to *charTri* and *wordBi*.
2. In comparison with *hybridBi* and *class2k*, conventional n-gram LMs (i.e., *charBi*, *charTri*, and *wordBi*) are rather sensitive to PT. Especially, *class500* is fairly robust to PT.
3. For *class500*, its RA almost remains unchangeable when increasing PT. For *class2k*, its RA with pruning fewer counts even outperforms that without pruning, its RA only decreases when PT > 5.
4. For *hybrid500* and *hybrid2k*, their RAs almost equal and decrease very slowly with increasing PT, although their model sizes reduce greatly.

From Fig. 8 and Fig. 9, lower PP also correlates with a higher accuracy. In speech recognition, it is well-known that lower PP correlates with better performance. Through investigating the influence of these three LM factors on PPs and RAs, we see that there is a similar relationship: lower PP correlates with a higher accuracy in script recognition.

7.3 Experiments by candidate confidence

In this subsection, we compare the four estimation methods of candidate confidence introduced in Sect. 5, namely EDF1, EDF2, ACT and LRM.

For ACT method, 1,100 training sample sets are used to obtain a confidence look-up table. In practice, candidate confidence can be directly obtained from the look-up table.

For LRM method, 50 training sample sets with an average RA of 87.40% are used to estimate the regression coefficients in (10). The distance value $d_i (1 \leqslant i \leqslant K)$ is normalized to the value within 0–100. For the first original candidate, we have

$$p(c_1|x) = (1 + \exp(-0.647 + 0.439d_1 - 0.325d_2 - 0.086d_3))^{-1} \quad (11)$$

For the second candidate, we have:

$$p(c_2|x) = (1 + \exp(-0.221 - 0.377d_1 + 0.392d_2))^{-1} \quad (12)$$

For the subsequent candidates ($k > 2$), their confidence can be computed by (13), which depends on d_1 and the corresponding distance value d_k.

$$p(c_k|x) = (1 + \exp(0.595 - 0.334d_1 + 0.344d_k))^{-1}, k > 2 \quad (13)$$

The posterior probability estimated by ACT and LRM should be more accurate than that designed artificially in EDF1 and EDF2.

Table 5 The influence of four confidence estimation methods on RA (%)

	Top1	Equal confidence	EDF1	EDF2	ACT	LRM
Script-A	92.32	89.75	93.19	98.09	98.49	98.51
Script-B	81.58	82.26	84.63	92.66	93.44	93.51
Script-C	70.84	75.04	76.94	83.68	84.38	84.44
Average	81.58	82.35	84.92	91.48	92.10	92.15

7.3.1 Candidate confidence affecting accuracy

For the three scripts, using *charBi* and ten candidates, we test the influence of four estimation methods of candidate confidence on their RAs, as shown in Table 5.

From Table 5, we can see that candidate confidence is very important for the contextual post-processing. Although EDF1 can reflect the reliability of candidates, its performance is rather bad. Considering the distance discrepancy between the first candidate and the related candidate, EDF2 has better performance than EDF1. ACT can be trained by a large number of recognition results, so it has fairly better performance than EDF2. From a statistical point of view, LRM can also be trained by a large number of recognition results, hence it has the best post-processing performance among the above four confidence estimation methods.

On the other hand, in order to further indicate the importance of candidate confidence, we assume that all candidates are of equal confidence value ($p(c_k|x)$ is equal for $k = 1,2, ..., 10$), in other words, the influence of candidate confidence on the post-processing is omitted and the post-processing thoroughly depends on LMs. In this case (see the third column in Table 5), the RA of *Script-A* does not increase but decrease, while the improvement of poorly recognized scripts is very limited.

7.3.2 Script quality evaluation

In Chinese script recognition, we should first estimate FCA of a script so that we can select a suitable K to execute the post-processing. Lin [30] proved that the mean value of all the first candidates' confidence in a sample set is equal to the expectation value of character recognition accuracy. Thus, we can estimate the accuracy of script recognition before post-processing, that is to say, the quality of recognition results in ICCR can be evaluated (called script quality evaluation). Apparently, K should vary with each script. In practice, after estimating the FCA of a script using the first candidate's confidence, we can assign a suitable K to the post-processing for the script according to Fig. 4.

Using LRM and ACT, the estimation results of FCA on the five classes of samples from 300 test sets are shown in Table 6.

Table 6 illustrates that the estimation results of both LRM and ACT have fairly good preciseness. Compared to ACT, LRM has better preciseness. For not poorly recognized sample sets, the estimation error of LRM is below 2% (such as Class-A, Class-B, and Class-C). But for badly recognized sample sets, the estimation error is large, such as Class-D and Class-E. The reason is that the overall quality of training samples is good (FCA is more than 87%), whereas there are not enough worse-quality training samples. In practice, we only need to roughly estimate FCA of a script, and then decide the appropriate value of K.

In script quality evaluation, the FCAs are estimated by LRM as 93.38, 83.56, and 72.45% for *Script-A*, *Script-B*, *Script-C*, respectively. Their suitable K could be roughly estimated as 20, 50, 100 for each script according to Fig. 4.

In the next subsection, we will still see that the excessive number of candidates would not only increase the overall processing time but also decrease the overall recognition accuracy of scripts due to excessive erroneous word formations in the lexicon lookup. Therefore, selecting a suitable K is very important for the post-processing of script recognition.

7.4 Experiments on candidate set size

In this subsection, we will display the influence of candidate set size on both post-processing speed and script recognition accuracy.

7.4.1 Effect of candidate set size on post-processing speed

The contextual post-processing speed mainly depends on three factors: the complexity of looking up LM parameters, the complexity of searching optimal sentence, and the complexity of constructing a word graph. Apparently, the parameters of *charBi* are far fewer than those of *charTri* and *wordBi* (see Fig. 7), and the searching space of *charBi* post-processing is also far smaller than that of *charTri* and *wordBi* post-processing. On the other hand, neither *charBi* post-processing nor *charTri* post-processing requires the construction of word graph. For *classBi*, although its parameters are extremely few, its post-processing needs the construction of word graph like *wordBi* post-processing. Intuitively, *hybridBi* post-processing is more complex than *both wordBi* post-processing and *classBi* post-processing.

We adopt the seven LMs without pruning to obtain the relationship curve between the post-processing time and the number of candidates K for *Script-B*, as shown in Fig. 10. Noting that the complexity of constructing a word graph rapidly rises with increasing K [15], we have,

Table 6 FCA estimation using candidate confidence

	ClassA	ClassB	ClassC	ClassD	ClassE
The number of sets	145	100	40	10	5
True (%)	94.68	86.13	76.65	67.69	58.31
ACT (%)	93.00	85.75	79.53	73.97	70.10
LRM (%)	93.46	86.02	78.32	71.58	67.08

in practice, only processed the candidate set in which the first candidate's confidence is less than 0.99.

As can be seen from Fig. 10, *charBi* post-processing is extremely fast and its processing time is almost negligible as compared to others. Since *charTri* post-processing does not require the construction of word graph, its post-processing is faster and its processing time rises linearly with increasing K, while *wordBi* post-processing appears very slow and its processing time rises exponentially with increasing K. For *classBi*, its processing time also rises exponentially with increasing K, although its post-processing is rather fast with small K. Obviously, *hybridBi* post-processing is a little slower than *wordBi* post-processing.

It is noticeable that both *wordBi* post-processing and *hybridBi* post-processing are very slow when K is large. In Fig. 10, for $K = 50$, *wordBi* post-processing, *hybrid500* post-processing and *hybrid2k* post-processing take 38, 43, and 48 min, respectively; while *charTri* post-processing, *class500* post-processing and *class2k* post-processing take 10, 14, and 17 min, respectively. However, for $K = 50$, *charBi* post-processing only needs 23 s.

7.4.2 *Effect of candidate set size on accuracy*

With the seven LMs, we test the relationship between the recognition accuracy and K for *Script-A* *Script-B*, and *Script-C*, as illustrated in Fig. 11, 12, and 13, respectively. We also display the CRA with varying K for each script in its corresponding figure.

From Fig. 11, 12, and 13 experimental results are characterized by the following:

1. Obviously, the post-processing with suitable K candidates can obtain fairly higher RA than the traditional 10 candidates. For *Script-B*, its RA approaches its Top10; for *Script-C*, its RA even surprisingly outperforms its CRA of the top 20 candidates.
2. We should select a suitable K for each script. For *Script-A*, $K = 20$ is optimal for the post-processing (the accuracy reaches 99.26%), while $K > 20$ does not improve its accuracy, instead decreases it. Similarly, $K = 50$ is optimal for *Script-B*, its accuracy reaches 96.69%. However, from Fig. 13 it seems that even $K = 100$ is not enough for the post-processing of *Script-C*, since we can steadily improve the accuracy with increasing K. The accuracy of *Script-C* can reach 94.05% with 100 candidates.
3. Obviously, the recognition accuracy of a script with post-processing is confined by its CRA. If there is no correct character in the limited candidate set, we cannot select the correct character through the post-processing. We find an interesting phenomenon that the optimal K depends on the variation trend of CRA. This phenomenon can be explained as follows: when CRA barely improves with increasing K, additional candidates can rarely include the correct character, but also produce the excessive erroneous word (or co-occurrence pair) formations resulting in a decrease in recognition accuracy.

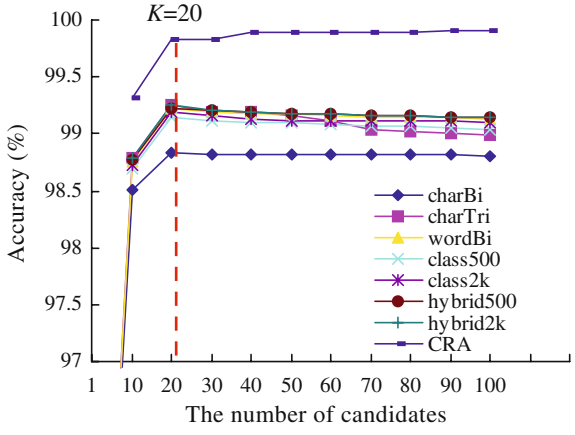


Fig. 11 The RA of *Script-A* varying with the number of candidates

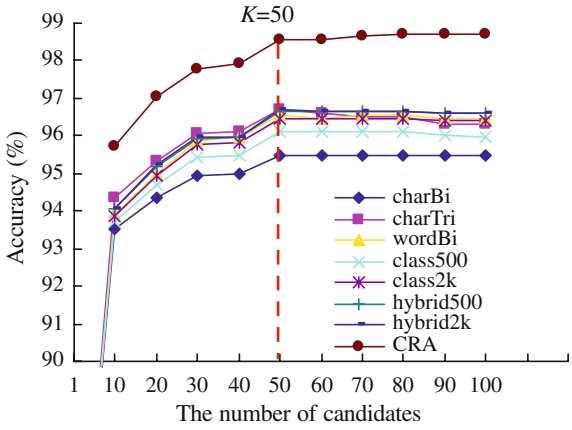


Fig. 12 The RA of *Script-B* varying with the number of candidates

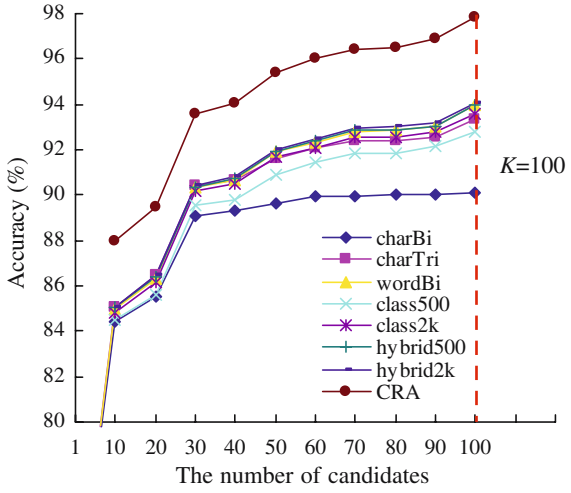


Fig. 13 The RA of *Script-C* varying with the number of candidates

Table 7 Baseline: common post-processing performance

	Script-A (%)	*Script-B* (%)	*Script-C* (%)	Average (%)	ECR (%)	APT (s)
Top1	92.32	81.58	70.84	81.58	–	–
charBi_K	98.88	95.61	90.37	94.95	72.60	31
charTri_K	99.26	96.69	93.35	96.43	80.62	708
wordBi_K	99.22	96.54	93.94	96.57	81.38	13,217
class500_K	99.15	96.13	92.77	96.02	78.39	10,028
class2k_K	99.16	96.44	93.60	96.40	80.46	10,429
hybrid500_K	99.24	96.69	93.95	96.63	81.70	14,009
hybrid2k_K	99.25	96.71	94.05	96.67	81.92	14,475

7.5 Experiments on hybrid post-processing

From Sect. 7.4.2, we know that the post-processing with suitable K candidates can effectively improve the accuracy of script recognition; however, post-processing is rather slow with increasing K, especially for *wordBi* and *hybridBi* post-processing. On the other hand, allowing the correct character to be captured in a limited number of candidates is vital to post-processing.

As described in Sect. 3.3, *Script Quality Evaluation* is employed to roughly estimate the suitable number of candidates (see Sect. 7.3.2) in post-processing. The other two optional modules (Candidate Set Expansion and *Approximate Word Matching*) are employed to capture the potential correct characters, in which the confusion matrix is obtained from 1,100 training sets (see Sect. 7.1). Combining *Character-based Post-processing* and *Word-based post-processing* can obtain high script recognition accuracy while giving due attention to processing speed at the same time. Considering the processing speed, we use *charBi* post-processing here.

This subsection shows the performance of the hybrid post-processing in details. In contrast to Fig. 11, 12 and 13, Table 7 shows the post-processing results with the optimal number of candidates K, which is regarded as the baseline of common post-processing performance for the seven LMs. The error correction rate[4] (ECR) and the average processing time (APT) are also listed in Table 7. Comparing *charBi_K* and *hybrid2k_K*, although the latter obtains higher accuracy, the former is far faster than the latter.

In the hybrid post-processing system, after the *charBi* post-processing using *forward-backward* search is first executed on the optimal number of candidates, the word-based post-processing using the *Viterbi* search is executed on a small candidate set with 10 new candidates. Here, we consider five types of the word-based post-processing, which employ *wordBi*, *class500*, *class2k*, *hybrid500*, and *hybrid2k*, respectively. Thus, using these five LMs, we have five corresponding types of hybrid post-processing.

We investigate the influence of CSE and AWM on the performance of hybrid post-processing, as shown in Tables 8, 9, 10, 11, where H, E and A denote hybrid post-processing, CSE and AWM, respectively. The performance of the hybrid post-processing without CSE and AWM is given in Table 8. Considering the CSE method in the *charBi* post-processing, where combined candidate sets are employed to replace original candidate sets, the performance of hybrid post-processing is shown in Table 9. Considering the AWM method in the word-based post-processing, where additionally approximate words are inserted into the word set in a word graph, the performance of hybrid post-processing is show in Table 10. Table 11 shows the performance of hybrid post-processing with both CSE and AWM.

In Table 8, since the word-based post-processing is executed with only ten candidates, the hybrid post-processing speed is fairly fast compared to the conventional word-based post-processing with the suitable number of candidates in Table 7. Meanwhile, the hybrid post-processing in Table 8 can obtain the comparable RAs as the word-based post-processing in Table 7. For example, *hybrid2k_H* with 96.64% is 100 times faster than *hybrid2k_K* with 96.67%.

Compared to Table 8, using CSE can further improve RA in Table 9. Although there is only a little improvement for *Script-A*, the improvement is fairly obtained for *Script-B* and *Script-C*. In comparison with Table 8, using AWM can also improve the accuracies of *Script-B* and *Script-C* in Table 10. However, for *Script-A*, its accuracy instead of increasing, decreases a little. The reason may be explained as follows: since the accuracy of *Script-A* in Table 9 is very high (not less than 99.15%), the AWM method may produce the excessive erroneous word formations so that the accuracy decreases. Compared to Table 9, using AWM can further improve RAs of *Script-B* and *Script-C* in Table 11. For *Script-A*, its RA decreases a little, similar to that in Table 10.

From Tables 9, 10 and 11, one can see that both CEA and AWM are very beneficial in improving the accuracy of not well-recognized scripts. For the well-recognized scripts, since their CRA of the top 10 or 20 candidates is very high, neither CEA nor AWM is needed in the hybrid post-processing.

With our proposed hybrid post-processing system, we can obtain high script recognition accuracy while

[4] error correction rate = (1.0 − the number of errors after post-processing / the number of errors before post-processing)×100%

Table 8 Hybrid post-processing performance without CSE and AWM

	Script-A (%)	*Script-B* (%)	*Script-C* (%)	Average (%)	ECR (%)	APT (s)
Top1	92.32	81.58	70.84	81.58	–	–
wordBi_H	99.23	96.68	93.77	96.56	81.32	89
class500_H	99.15	96.44	93.16	96.25	79.64	34
class2k_H	99.18	96.56	93.59	96.44	80.69	41
hybrid500_H	99.25	96.75	93.82	96.61	81.58	93
hybrid2k_H	99.26	96.75	93.92	96.64	81.78	144

Table 9 Hybrid post-processing performance with CSE

	Script-A (%)	*Script-B* (%)	*Script-C* (%)	Average (%)	ECR (%)	APT (s)
Top1	92.32	81.58	70.84	81.58	–	–
wordBi_HE	99.24	97.49	94.83	97.19	84.73	174
class500_HE	99.17	97.27	94.29	96.91	83.22	123
class2k_HE	99.19	97.44	94.64	97.09	84.20	135s
hybrid500_HE	99.26	97.57	94.93	97.25	85.09	189
hybrid2k_HE	99.27	97.59	94.96	97.27	85.20	230

Table 10 Hybrid post-processing performance with AWM

	Script-A (%)	*Script-B* (%)	*Script-C* (%)	Average (%)	ECR (%)	APT (s)
Top1	92.32	81.58	70.84	81.58	–	–
wordBi_HA	99.19	97.52	94.49	97.07	84.08	133
class500_HA	99.15	97.31	93.88	96.90	83.15	48
class2k_HA	99.16	97.44	94.26	96.95	83.46	60
hybrid500_HA	99.23	97.59	94.63	97.15	84.53	144
hybrid2k_HA	99.23	97.63	94.64	97.17	84.62	190

Table 11 Hybrid post-processing performance with both CSE and AWM

	Script-A (%)	*Script-B* (%)	*Script-C* (%)	Average (%)	ECR (%)	APT (s)
Top1	92.32	81.58	70.84	81.58	–	–
wordBi_HEA	99.21	97.97	95.38	97.52	86.54	198
class500_HEA	99.17	97.63	94.78	97.19	84.76	132
class2k_HEA	99.16	97.82	95.18	97.39	85.81	152
hybrid500_HEA	99.25	98.03	95.64	97.64	87.19	207
hybrid2k_HEA	99.24	98.06	95.65	97.65	87.24	248

Table 12 Performance comparison of several post-processing schemes

Post-processing method	LM size (Mb)		Accuracy (%)		Processing time (s)	
	No pruning	Pruning	No pruning	Pruning	No pruning	Pruning
charBi_K	12	3	94.95	94.94	31	30
charTri_K	53	22	96.43	96.26	708	620
wordBi_K	49	14	96.57	96.47	13,217	12,407
class500_K	**2**	-	96.02	–	10,028	–
wordBi_H	61	17	96.56	96.12	89	63
hybrid500_H	63	19	96.61	96.42	93	67
hybrid500_HEA	63	20	97.64	97.49	207	179

giving due attention to processing speed at the same time. Compared to the traditional post-processing with only ten original candidates, this hybrid post-processing system can greatly improve the accuracy of script recognition from 92.66 to 97.65%, which means 87% ECR in comparison with traditional 60% ECR.

8 Proposals for real recognition tasks

In evaluating the performance of post-processing, besides recognition accuracy, memory space and computational cost are also important factors.

According to the experimental results in Sect. 7.5, we can make an appropriate decision in choosing a suitable contextual post-processing method for constructing a practical post-processor when a script is recognized. Which post-processing method to be employed really depends on the available memory and computational resources as well as the requirement of response time in real recognition tasks.

We summarize several typical post-processing schemes in terms of post-processing method, model size, recognition accuracy, and processing time in Table 12. The hybrid post-processing needs 768 kb to store the confusion matrix for CSE and 608 kb to store the capability of constructing two-character words for AWM.

As shown in Table 12, both *wordBi_K* and *class500_K* are extremely time-consuming, while the hybrid post-processing schemes' speeds are acceptable. For *hybrid500_HEA*, the average recognition accuracy reaches 97.64%, while the processing time is only 207 s. This demonstrates that hybrid post-processing can effectively improve the accuracy of script recognition while giving attention to the processing speed at the same time. For a page of 400 handwritten Chinese characters, *hybrid500_HEA* only needs 3–4s to process it.

It is noticeable that model pruning can greatly reduce the size of a LM, while the model's capability of improving accuracy only decreases a little. For *charBi_K*, there is almost no change of RA when the model size reduces from 12 Mb to 3 Mb. For *hybrid500_HEA*, there is only a little decrease of RA when the model size reduces from 63 to 20 Mb.

According to the requirement of a real recognition system, one can select a suitable post-processing method. It is quite clear that if an application has to run on a platform with only very limited memory, then *class500_K* is the choice to build a practical post-processor. If processing speed is strictly required in some applications, *charBi_K* is a practicable method. If high recognition accuracy is the main concern of the application, *hybrid500_HEA* can be used. On the other hand, if the confusion matrix of a recognizer is inaccessible, *hybrid500_H* is a good choice to obtain high accuracy.

9 Conclusion

In this paper, we analyze the key factors that have an impact on the performance of contextual post-processing: statistical LMs, candidate confidence, and candidate set size. We show the perplexities of several LMs and their influence on the recognition accuracy, and confirm that lower perplexity correlates with a higher accuracy. We compare several estimation methods of candidate confidence, and indicate that candidate confidence is vital to the contextual post-processing. We discuss the influence of candidate set size on post-processing time and accuracy, and point out that a suitable number of candidates should be selected for each script.

We build a hybrid post-processing system integrating the above factors with script quality evaluation, CSE and AWM, by which the average recognition accuracy of three Chinese scripts (about 66,000 characters in total) can reach 97.65%, that means 87.24% error correction rate in comparison with the 81.58% average accuracy before post-processing. We also give a proposal in choosing a suitable post-processing method according to the requirement of a practical recognition system.

This kind of hybrid post-processing can effectively improve the accuracy of script recognition while giving due attention to both processing speed and memory space at the same time. The proposed hybrid post-processing system can be readily applicable to online handwritten Chinese script recognition.

In order to obtain good post-processing performance (high accuracy and rapid processing), improving the effectiveness of candidate sets is extremely important. For poorly recognized scripts, we will still strive to allow the correct character to be captured in a limited number of candidates.

Yuan-Xiang Li is an Associate Professor in the Department of Atmospheric Sounding and Information Engineering, Institute of Meteorology, PLA University of Science and Technology. From 2002 to 2004, he was a research fellow in 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 image processing, pattern recognition, character recognition, Chinese information processing, information retrieval, data mining, and data compression. He is a member of the IAPR.

Chew Lim Tan is an Associate Professor in the Department of Computer Science, School of Computing, National University of Singapore. He received the B.Sc. (Hons) degree in physics in 1971 from the University of Singapore, the M.Sc. degree in radiation studies in 1973 from the University of Surrey, UK, and the Ph.D. degree in computer science in 1986 from the University of Virginia, USA. His research interests include document image and text processing, neural networks and genetic programming. He has published more than 230 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 and 2006, Graphics Recognition Workshop (GREC) 2001 and 2003, Web Document Analysis Workshop (WDA) 2001, 2003,

and 2005, Document Image Analysis and Retrieval Workshop (DIAR) 2003, Document Image Analysis for Libraries Workshop (DIAL) 2004, International Conference on Image Processing (ICIP) 2004, IEEE Workshop on Applications of Computer Vision (WACV) 2005, Camera-Based Document Analysis and Recognition Workshop (CBDAR) 2005, International Conference on Document Analysis and Recognition (ICDAR) 2005, and Pacific Rim International Conference on Artificial Intelligence (PRICAI) 2006. He is the current President of the Pattern Recognition and Machine Intelligence Association (PREMIA) in Singapore. He is a member of the Governing Board of the International Association of Pattern Recognition. He is also a senior member of IEEE.

Xiaoqing Ding is a Professor in the Department of Electronic Engineering, Tsinghua University, China. She graduated from Tsinghua University and won the gold medal for excellent student in 1962. She has published more than 290 papers and is the co-author of four books. Her research interests include image processing, pattern recognition, character recognition, document image analysis, biometric authentication, computer vision, and video surveillance. She has researched and developed a series of Chinese character recognition systems, and multi-inguinal document recognition systems including Japanese, Korean, Arabic, Tibetan, Uyghur, Kazakh, Kirghiz characters etc., which are among the foremost internationally. She has also developed multi-modal biometric verification and identification algorithms including Face, Hand-writer, Signature and Iris, etc. She has won numerous Honors and Awards, such as the 2nd class *China National Scientific and Technical Progress Award* in 1999 and 2004, and the Awards of Best Overall Performing Face Verification Algorithm in the 2004 Face Authentication Test on ICPR2004.

Acknowledgements This work is partly supported by the Agency for Science, Technology and Research (Grant No. R252-000-123-305) in Singapore. The authors would like to thank the anonymous reviewers for their valuable suggestions.

References

1. Suen CY, Mori S, Kim SH, et al (2003) Analysis and recognition of Asian scripts—the state of the art. In: Proceedings of 7th international conference on document analysis and recognition, Edinburgh, UK, pp 866–878
2. Xiong Y, Huo Q, Chan C (2001) A discrete contextual stochastic model for the offline recognition of handwritten Chinese characters. IEEE Trans Pattern Anal Mach Intell 23(7):774–782
3. Zhang J, Ding X, Liu C (2000) Multi-scale feature extraction and nested-subset classifier design for high accuracy handwritten character recognition. In: Proc 15th international conference on pattern recognition, Barcelona, Spain 2:581–584
4. Tang YY, Tu LT, Liu J, et al (1998) Offline recognition of Chinese handwriting by multifeature and multilevel classification. IEEE Trans Pattern Anal Mach Intell 20(5):556–561
5. Tung CH, Lee HJ (1994) Increasing character recognition accuracy by detection and correction of erroneously identified characters. Pattern Recogn 27(9):1259–1266
6. Chang CH (1996) Simulated annealing clustering of Chinese words for contextual text recognition. Pattern Recogn Lett 17(1):57–66
7. Lee HJ, Tung CH (1997) A Language model based on semantically clustered words in a Chinese character recognition system. Pattern Recogn 30(8):1339–1346
8. Wong PK, Chan C (1999) Post-processing statistical language models for a handwritten Chinese character recognizer. IEEE Trans Syst Man Cybern Part B Cybern 29(2):286–291
9. Samuelsson C, Reichl W (1999) A class-based language model for large-vocabulary speech recognition extracted from part-of speech statistics. In: Proceedings of international conference on acoustics, speech and signal processing, Phoenix, USA 1:537–540
10. Li YX, Tan CL, Ding X, et al (2004) Contextual post-processing based on the confusion matrix in offline handwritten Chinese script recognition. Pattern Recogn 37(9):1901–1912
11. Li Y, Ding X, Tan CL (2002) Combining character-based bigram with word-based bigram in contextual post-processing for Chinese script. ACM Trans Asian Lang Inform Process 1(4):297–309
12. Martin S, Liermann J, Ney H (1998) Algorithms for bigram and trigram word clustering. Speech Commun 24:9–37
13. Liu CL, Nakagawa M (2000) Precise candidate selection for large character set recognition by confidence evaluation. IEEE Trans Pattern Anal Mach Intell 22(6):636–642
14. Wu LD (1997) Large-scale Chinese text processing. Fudan University Press, China
15. Gu HY, Tseng CY, Lee LS (1991) Markov modeling of mandarin Chinese for decoding the phonetics sequence into Chinese characters. Comput Speech Lang 15(4):363–377
16. Rabiner LR. (1989) A tutorial on hidden Markov models and selected applications in speech recognition. Proc IEEE 77(2):257–286
17. Ney H, Ortammns S (1999) Dynamic programming search for continuous speech recognition. IEEE Signal Process Mag 16(5):64–83
18. Koerich AL, Sabourin R, Suen CY (2003) Large vocabulary off-line handwriting recognition: a survey. Patt Anal Appl 6(1):97–121
19. Xu R, Yeung D, Shu W (2002) A hybrid post-processing system for handwriting Chinese character recognition. Int J Patt Recogn Artif Intell 16(6):657–679
20. Perraud F, Viard-Gaudin C, Morin E et al (2003) N-gram and n-class models for online handwriting recognition. In: Proceedings of 7th international conference on document analysis and recognition. Edinburgh, UK, pp 1053–1057
21. Jurafsky D, Martin JH (2000) Speech and language processing: an introduction to natural language processing, computational linguistics and speech recognition. Prentice Hall, New Jersey
22. Chen SF, Goodman J (1999) An empirical study of smoothing techniques for language modeling. Comput Speech Lang 13(4):359–394
23. Xu L, Krzyzak A, Suen CY (1992) Methods of combining multiple classifiers and their applications to handwriting recognition. IEEE Trans Syst Man Cybern 22(3):418–435
24. Lee YS, Chen HH (1996) Analysis of error count distributions for improving the post-processing performance of OCCR. Commun COLIPS 6(2):81–86
25. Lin X, Ding X, Chen M, et al (1998) Adaptive confidence transform based on classifier combination for Chinese character recognition. Pattern Recogn Lett 19(10):975–988
26. Li Y, Ding X (2002) Evaluation of character candidate confidence measure using logistic regression model (in Chinese). Pattern Recogn Artif Intell 15(2):160–166
27. Webb A (2002) Statistical pattern recognition. Wiley, England
28. Hosmer DW, Lemeshow S (1989) Applied logistic regression. Wiley, New York
29. Chen Y (1997) Research on hand-printed Chinese character recognition. PhD Thesis, Tsinghua University
30. Lin X (1999) Theory and application of confidence analysis and multiple classifier combination in character recognition. PhD Thesis, Tsinghua University