2009 10th International Conference on Document Analysis and Recognition

Integrating Language Model in Handwritten Chinese Text Recognition

Qiu-Feng Wang, Fei Yin, Cheng-Lin Liu
National Laboratory of Pattern Recognition (NLPR),
Institute of Automation, Chinese Academy of Sciences
95 Zhongguancun East Road, Beijing 100190, P.R. China
{wangqf, fyin, liucl}@nlpr.ia.ac.cn

Abstract

This paper describes a system for handwritten Chinese text recognition integrating language model. On a text line image, the system generates character segmentation and word segmentation candidates, and the candidate paths are evaluated by character recognition scores and language model. The optimal path, giving segmentation and recognition result, is found using a pruned dynamic programming search method. We evaluate various language models, including the character-based n-gram, word-based n-gram, and hybrid n-gram models. Experimental results on the HIT-HW database show that the language models improve the recognition performance remarkably.

1. Introduction

The recognition of handwritten Chinese generic text is a challenging task due to the diversity of writing styles, unconstrained language domain, large character set, character shape variation, and the character segmentation problem caused by variable character size, confusing inter-radical and inter-character gaps, character touching and overlapping, etc. A general approach to overcome the ambiguity of character segmentation is to integrate segmentation and recognition in a combinatorial optimization framework, where character segmentation candidates are generated and evaluated by character recognition scores and linguistic context [1].

The application of word-based language models to Chinese text involves a word segmentation tasks since there is no extra space between words. For handwritten text recognition, we face two candidate segmentation issues: character segmentation and word segmentation. Some previous works applied word-based language models only to post-processing without considering character segmentation [2][3]. On the other hand, some works use character-based n-gram models combined with character segmentation, without considering word segmentation [4]. Character-based n-grams have also been widely used in Japanese text recognition (e.g., [5]). The work of [4] also use a word lexicon to match candidate character segmentation and recognition results but the lexicon is only for a constrained domain like address reading. To our best of knowledge, the integration of language model with character segmentation and word segmentation in Chinese text recognition has not been reported.

In this paper, we propose a handwritten Chinese text recognition scheme integrating character-based and word-based language models. The scheme evaluates candidate character segmentation and recognition, word segmentation and language model in a unified probabilistic framework, and the optimal path giving segmentation and recognition result is found using pruned dynamic programming (DP) search. We evaluate various character-based and word-based language models and demonstrate the large benefits of language models in improving the performance of handwritten Chinese text recognition.

2. System Overview

Fig. 1 shows the block diagram of our system for handwritten Chinese text recognition. First, the input text line image is over-segmented into primitive segments using the method of [6], such that each primitive is a character or a part of a character (Fig. 2(a)). Consecutive segments are combined to generate candidate character patterns, forming a segmentation candidate lattice (Fig. 2 (b)). Each candidate pattern is classified to assign candidate character classes, forming a character candidate lattice (Fig. 2(c)). In classification, some candidate classes are abandoned because the difference of their scores from the top rank class is too large. Each sequence of candidate characters is matched with a lexicon to segment into candidate words, forming a word candidate lattice (Fig.2 (d)). Last, each character sequence or word sequence is evaluated by character recognition scores and language model, and the optimal sequence (path) is searched to give the segmentation and recognition result.

978-0-7695-3725-2/09 $25.00 © 2009 IEEE
DOI 10.1109/ICDAR.2009.96

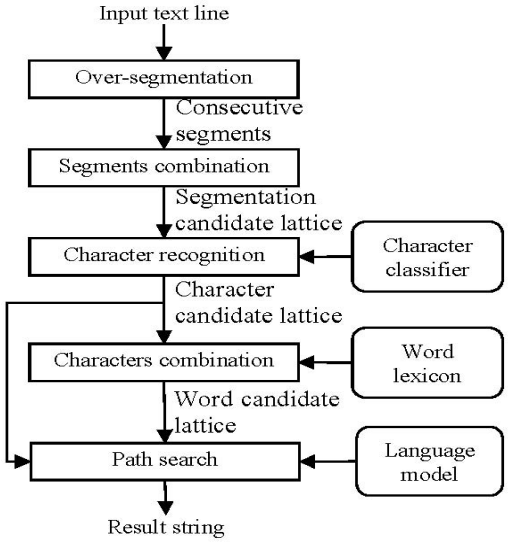


Figure 1. System diagram for handwritten Chinese text recognition.

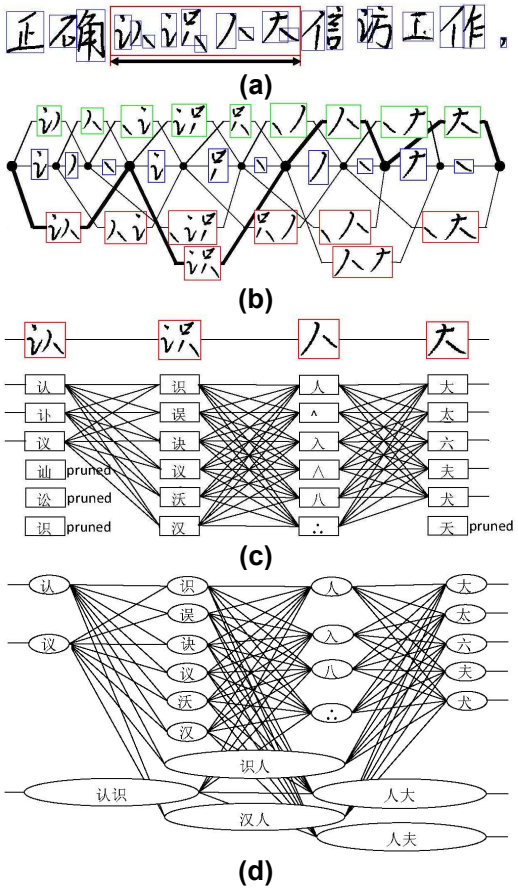


Figure 2. (a) Over-segmentation of a text line; (b) Segmentation candidate lattice of a part; (c) Character candidate lattice of a segmentation; (d) Word candidate lattice for (c).

3. Path Evaluation and Search

Based on candidate character segmentation, character classes assignment and candidate word segmentation, the task of string recognition is to find the optimal path (combination) of segmentation-recognition. A path corresponds to a sequence of candidate character patterns $X = \mathbf{x}_1 \cdots \mathbf{x}_n$ paired with character classes $C = c_1 \cdots c_n$. The path is evaluated by a likelihood score:

$$f(X,C) = \lambda_1 \log P(C) + \lambda_2 \sum_{i=1}^{n} k_i \log P(x_i \mid c_i), \quad (1)$$

and the path of maximum score over all combinations (X,C) gives the segmentation-recognition result. The function in Eq. (1) is modified from $\log P(C)P(X \mid C)$, but to overcome the bias of $P(X|C)$ to small number of segmented characters, we weight the likelihood of each character pattern with its number k_i of constituent segments [7] (similar to the variable length HMM of [8]). The empirical weights λ_1, λ_2 are used to balance the effects of language model and character recognition score.

The character recognition score is given by a character classifier (MQDF), which inputs character shape features and output scores proportional to the log-likelihood $\log P(x_i \mid c_i)$. The language model $P(C)$ is to be elaborated in Section 4.

The summation nature of path score in Eq. (1) guarantees that the optimal path can be found by dynamic programming (DP) search. The search proceeds in frame-synchronous fashion: at each primitive segment s_t, examine all the candidate patterns $\mathbf{x}_i$ ending at s_t and the candidate classes c_i assigned to $\mathbf{x}_i$, and for each class c_i, examine the words w_j ending at c_i. Denote the preceding candidate pattern of $\mathbf{x}_i$ as $\mathbf{x}_{i-1}$ ending at segment s_{t-k} and assigned classes c_{i-1}, and the preceding word of w_j as w_{j-1} ending at character c_{i-1}. If using bi-gram language model, for each triplet (s_t,c_i,w_j), an optimal partial path (with maximum partial score over k, c_{i-1} and w_{j-1}) is retained. If using tri-gram language model, the partial path should be maximized over k, c_{i-1}, w_{j-1}, as well as the further preceding characters/words c_{i-2} and w_{j-2}. Consider that at each t, the cardinality of combinations (c_i,w_j) is very large, to accelerate search, we only retain a limited number N_R of partial paths with maximum scores over (c_i,w_j) at each t. We call this beam search method as pruned DP. N_R was set as 10 in our experiments.

To accelerate the recognition using word-based n-gram models, we also use a hybrid search method split into two stages. In the first stage, we use pruned DP with charBi model to generate a reduced character

candidate lattice, composed of the candidate character patterns and classes in the N_R optimal partial paths ending at each segment found by pruned DP. In the second stage, pruned DP with word-based n-gram model is used on the reduced character candidate lattice.

4. Statistical Language Models

A statistical language model (SLM) estimates the probability of a sequence of text units (characters, words, phrases, and so on). We take the Chinese characters and words using n-gram models due to their simplicity and efficiency [9].

For a sequence of characters $C = c_1 c_2 \cdots c_n$, character-based bigram (charBi) and trigram models are formulated by

$$p(C) = p(c_1) \prod_{i=2}^{n} p(c_i \mid c_{i-1}), \quad (2)$$

$$p(C) = p(c_1) p(c_2 \mid c_1) \prod_{i=3}^{n} p(c_i \mid c_{i-2} c_{i-1}). \quad (3)$$

For a sequence of words $C = w_1 w_2 \cdots w_L$ (each word comprises one character or multiple characters), the word-based bigram model (wordBi) is

$$p(C) = p(w_1) \prod_{i=2}^{L} p(w_i \mid w_{i-1}). \quad (4)$$

Due to the very large size of word lexicon, we cluster words to a smaller number of word classes. Among several methods of clustering word classes, we take the IBM clustering, which interpolates the word class-based bigram with a normal word-based bigram [9]:

$$p_{ibm}(w_i \mid w_{i-1}) = \lambda * p_w(w_i \mid w_{i-1}) + (1-\lambda) * p_c(w_i \mid w_{i-1}), \quad (5)$$

where $p_w(w_i \mid w_{i-1})$ is a word bigram used in Eq. (4), $p_c(w_i \mid w_{i-1})$ is a word class-based bigram defined as

$$p_c(w_i \mid w_{i-1}) = p(w_i \mid W_i) * p(W_i \mid W_{i-1}), \quad (6)$$

where W_i is the class of word w_i. We can see that if the word class number is the same as the number of words, the model of (6) degenerates as a normal word-based bigram; on the other hand, if the class number is one, the model of (6) becomes a normal word-based unigram. Replacing $p(w_i \mid w_{i-1})$ with $p_{ibm}(w_i \mid w_{i-1})$ in Eq. (4), we obtain the IBM clustering bigram model (ibmBi).

In addition, smoothing and pruning are two critical techniques to the performance of language model because of the data sparseness and large number of parameters. We adopt the very commonly used Katz smoothing [9] and entropy-based pruning [10]. In pruning, the n-grams that raise the perplexity (due to pruning them) less than a threshold (empirically selected) are removed. After removal, the backoff weights are recomputed.

The language models are summarized in Table 1.

Table 1. Language models used.

charBi	character-based bigram model, Eq. (2)
charTri	character-based trigram model, Eq. (3)
wordBi	word-based bigram model, Eq. (4)
ibmBi	word classes interpolated with wordBi, Eq. (5)

5. Experiments

Our experiments were implemented on a personal computer (Intel Core2 CPU 1.86 GHz). We evaluated the performance on a small subset of the HIT-MW database. This subset has 383 text lines (8,424 characters in total), and was used as the test set in [11].

5.1. Implementation

We evaluate the performance of character string recognition term of two measures: segmentation correct rate (SCR) and recognition correct rate (RCR). The SCR is the percentage of characters with both left and right boundaries matched with the ground-truth boundaries (difference of x coordinate within a threshold of 3 pixels), and the RCR is the percentage of characters both segmented correctly and assigned correct classes.

For assigning classes to candidate character patterns, we use a modified quadratic discriminant function (MQDF) classifier [12] on contour direction histogram features extracted by continuous NCFE (normalization-cooperated feature extraction) [13]. The output values of MQDF are proportional to the log-likelihood of classes under Gaussian density assumption. The classifier parameters were trained on a sample set of 2,534,678 isolated character images of 7,356 classes (7,185 Chinese characters, 10 digits, 52 English letters and 109 frequently used symbols).

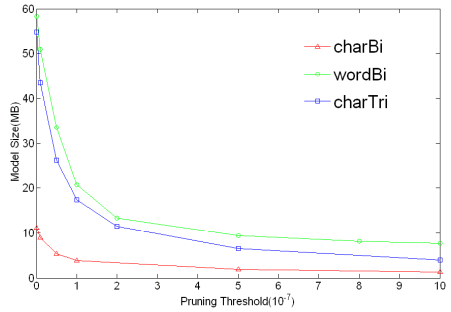
On a candidate character pattern, the classifier outputs a number of candidate classes with high scores. To achieve a good tradeoff between the number of candidate classes and the accumulated accuracy of multiple paths, we select dynamic number of classes for each candidate pattern: order the classes in descending order of scores and prune the classes when the difference of their scores from the top rank class exceeds a threshold. The threshold was empirically to achieve a high performance of string recognition. As result, by pruning classes from 30 candidate classes

(accumulated accuracy 87.64%) output by the classifier, we retained 25 classes (accumulated accuracy 86.91%) on average.

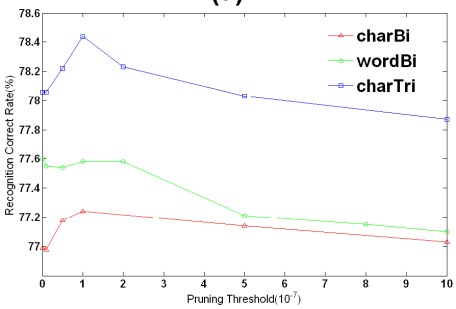
Our language models were trained on a Chinese corpus from the CLDC (Chinese Linguistic Data Consortium), which contains about 50,000,000 characters in 7,356 classes and 32,000,000 words in 289,598 types. We used the bigram word clustering algorithm of [14] to obtain word classes. The algorithm is an exchange algorithm similar to ISODATA with the maximum log-likelihood criterion. The SRILM (SRI Language Model) toolkit [15] was used to give the n-gram models.

5.2. Evaluating Language Models

We first investigated the effect of pruning threshold (PT) in character and word n-grams reduction on the storage size and recognition performance. Fig. 3 plots the model sizes of three language models and recognition correct rate (RCR) with variable PTs. We can see that an appropriate threshold (10^{-7} in our cases) can yield good tradeoff between model size and RCR.



(a)



(b)

Figure 3. (a) Model size with variable PTs and (b) RCR with variable PTs

Table 2. Effects of number of word classes in ibmBi

#Class	100	500	1000	1500	2000	3000	5000
Size(MB)	3.79	5.38	9.06	13.3	17.3	23.2	30.2
RCR(%)	77.80	77.91	78.04	77.93	77.92	77.98	77.98

We did not prune the ibmBi because the size of word classes in this model is not very large. Instead, we evaluated the effect of the number of word classes, as shown in Table 2, where the size denotes the size of ibmBi while for saving time, the RCR was obtained using hybrid search. We can see that a class number 1,000 of ibmBi leads to a high recognition rate.

Table 3 shows the results of recognition and segmentation with n-gram models and without language model, and the time of processing 383 text lines and the memory sizes of n-gram models. We can see that compared to recognition without language model, the character-based and word-based n-gram models improve the segmentation-recognition performance remarkably, say, the RCR was improved from 57.6% to over 77%. The word-based n-gram models outperform the character bigram but the character trigram even outperforms the word-based bigram and hybrid models. Recognition with word-based n-gram models is also extremely time consuming.

Table 3. Performance with and without n-gram models.

	w/o	charBi	charTri	wordBi	ibmBi
RCR (%)	57.60	77.18	**78.44**	77.58	77.85
SCR (%)	74.42	85.21	86.12	85.78	85.75
Time (s)	118	142	214	57920	58993
Size (MB)	---	5.4	17.4	20.7	29.8

Table 4. Performance of hybrid search.

	RCR (%)	SCR (%)	Time (s)	Size (MB)
char-wordBi	**77.84**	85.81	195	26.1
char-ibmBi	**78.04**	85.83	215	35.2

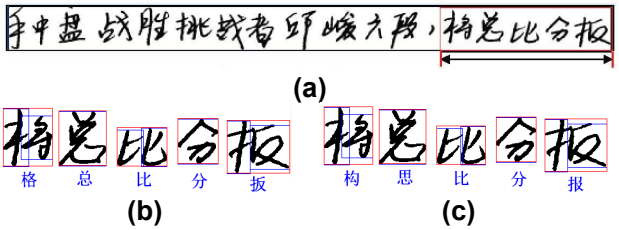
Table 4 shows the performance of hybrid search for acceleration: using character bigram to generate a reduced candidate character lattice and then using word-based model for final recognition. The hybrid search methods for two word-based models wordBi and ibmBi are denoted as “char-wordBi” and “char-ibmBi”, respectively. Obviously, the hybrid search method accelerates the recognition with word-based n-gram models significantly: the recognition time is now comparable to that of character trigram. Meanwhile, the recognition rates of word-based n-gram models were improved by hybrid search. This is because character candidates reduction by pruned DP using character bigram also reduces many noisy word candidates. Actually, the accumulated accuracy before pruned DP was 86.91%, which was only slightly decreased to 86.47% after pruned DP.

It is noteworthy that the character trigram outperforms the word-based bigram models, although

words are more meaningful semantic unit than characters. We investigated into some text lines found two reasons below.

(1) The text line is not a whole sentence, so, its first or last character may not compose a valid word. For example, in the text line of Fig. 4(a), the last character "扳" is only a part of the word "扳平" following the word "比分". So, the wordBi score *P*("扳"|"比分") is smaller than *P*("报"|"比分"). This problem is not with the charTri.

(2) If a character in a word is misrecognized, the word-based language model influences the adjacent characters more than the charTri because of the word lexicon. In Fig.4(c), the character " 将 " was misrecognized as "构", which resulted in the character pattern "总" misrecognize as "思", because "构思" is a common word in the lexicon, while "构总"is not.



(a) (b) (c)

Figure 4. (a) A text line; (b) Result with charTri; (c) Result with wordBi.

6. Conclusions

We presented an integrated segmentation-recognition system for offline handwritten Chinese text with character-based and word-based statistical language models (SLMs). The result of character-segmentation is given by pruned DP or hybrid search. We evaluated four SLMs: charBi, charTri, wordBi and ibmBi. Experimental results show that the charTri performs best in terms of character recognition and segmentation rates, and hybrid search significantly accelerates recognition with word-based language models. In the future, we will seek for strategies for better utilizing the semantics of word-based language models and incorporating segmentation scores as well.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (NSFC) under grants no.60775004 and no.60825301. The authors thank Bo Xu for creating the character classifier.

References

[1] H. Murase, Online recognition of free-format Japanese handwritings, *Proc. 9th ICPR*, 1988, Vol.2, pp.1143-1147.

[2] Y.X. Li, C.L. Tan, X.Q. Ding, A hybrid post-processing system for offline handwritten Chinese Script recognition, *Pattern Analysis and Applications*, 8: 272-286, 2005.

[3] R.F. Xu, D.S. Yeung, D.M. Shi, A hybrid post-processing system for offline handwritten Chinese character recognition based on a statistical language model, *Int. J. Pattern Recognition and Artificial Intelligence*, 19(3): 415-428, 2005.

[4] Y. Jiang, X.Q. Ding, Q. Fu, Z. Ren, Context driven Chinese string segmentation and recognition, *Structural, Syntactic, and Statistical Pattern Recognition: Joint IAPR Int. Workshops*, LNCS Vol.4109, pp127-135.

[5] M. Nakagawa, B. Zhu, M. Onuma, A model of on-line handwritten Japanese text recognition free from line direction and writing format constraints, *IEICE Trans. Information and Systems*, E88-D(8): 1815-1822, 2005.

[6] C.-L. Liu, M. Koga, H. Fujisawa, Lexicon-driven segmentation and recognition of handwritten character strings for Japanese address reading, *IEEE Trans. Pattern Analysis and Machine Intelligence*, 24(11): 1425-1437, 2002.

[7] J.L. Yu, X.D. Zhou, C.L. Liu, Search strategies in online handwritten character string recognition, *Proc. Chinese Conference on Pattern Recognition (CCPR 2007)*, 2007, pp.299-305.

[8] M.Y. Chen, A. Kundu, and S.N. Srihari, "Variable duration hidden Markov model and morphological segmentation for handwritten word recognition", *IEEE Trans. Image Processing*, 4(12): 1675-1688, 1995.

[9] J.T. Goodman, A bit of progress in language modeling: extended version, Technical Report MSR-TR-2001-72, Microsoft Research, 2001.

[10] A. Stolcke, Entropy-based pruning of backoff language models, *Proc. DARPA Broadcast News Transcription and Understanding Workshop*, 1998, pp. 270-274.

[11] T.H. Su, T.W. Zhang, D.J. Guan, H.J. Huang, Off-line recognition of realistic Chinese handwriting using segmentation-free strategy, *Pattern Recognition*, 42(1): 167-182, 2008.

[12] F. Kimura, K. Takashina, S. Tsuruoka, Y. Miyake, Modified quadratic discriminant functions and the application to Chinese character recognition, *IEEE Trans. Pattern Analysis and Machine Intelligence*, 9(1): 149-153, 1987.

[13] C.-L. Liu, K. Nakashima, H. Sako, H. Fujisawa, Handwritten digit recognition: investigation of normalization and feature extraction technique, *Pattern Recognition*, 37: 265-279, 2004.

[14] S. Martin, J. Liermann, H. Ney, Algorithms for bigram and trigram word clustering, *Speech Communication*, 24(1): 19-37, 1998.

[15] A. Stolcke, SRILM - an extensible language modeling toolkit, *Proc. Int. Conf. on Statistical Language Processing*, Denver, Colorado, 2002.