2013 12th International Conference on Document Analysis and Recognition

Keyword Spotting in Online Chinese Handwritten Documents with Candidate Scoring Based on Semi-CRF Model

Heng Zhang[1], Xiang-Dong Zhou[2], Cheng-Lin Liu[1]
[1]National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences
95 Zhongguancun East Road, Beijing 100190, P.R. China
{hzhang07, liucl}@nlpr.ia.ac.cn
[2] Beijing Key Lab of Human-Computer Interaction, Institute of Software, Chinese Academy of Sciences
No.4 Zhongguancun South 4th Street, Beijing, China
xiangdong@iscas.ac.cn

***Abstract*—For text-query-based keyword spotting from handwritten Chinese documents, the index is usually organized as a candidate lattice to overcome the ambiguity of character segmentation. Each edge in the lattice denotes a candidate character associated with a candidate class. Character similarity (between character and class) scores are calculated on each edge, and the similarity between a query word and handwriting is obtained by combining these edge scores. In this paper, we propose a document indexing method using semi-Markov conditional random fields (semi-CRFs), which provide a principled framework for fusing the information of different contexts. For fast retrieval and to save storage space, the lattice is first purged by a forward-backward pruning approach. On the reduced lattice, we estimate the character similarity scores based on the semi-CRF model. Experimental results on a large handwriting database CASIA-OLHWDB justify the effectiveness of the proposed method.**

Keywords*—Online Chinese handwritten documents; keyword spotting; semi-Markov conditional random fields.

I. Introduction

Keyword spotting [1] is to detect a word (or a phrase) in a document, which is usually accomplished by computing a similarity measure between the query word and a segmented candidate in the document. For fast retrieval of documents from large database, it is necessary to build and store an index beforehand, on which the spotting algorithm is run and gives the spotting results for a query word. Thus, a typical keyword spotting system should include two parts: index generation and keyword detection (Fig. 1). This paper is concerned with keyword spotting from online handwritten Chinese documents using text (keyboard) query. In addition to the ambiguity of character segmentation (there is no extra space between characters or words) and the shape variation, Chinese handwritten documents also suffer from the large alphabet (over 5,000 characters and 200,000 words are daily used).

To overcome the difficulties of character segmentation, the index [2–4] is generally organized as a candidate lattice which contains alternative segmentation-recognition hypotheses of text lines (the document has already been segmented as text lines before building the index). For each text line in the document, it is first over-segmented into a sequence of components according to the overlapping between strokes (Fig. 2(a)),

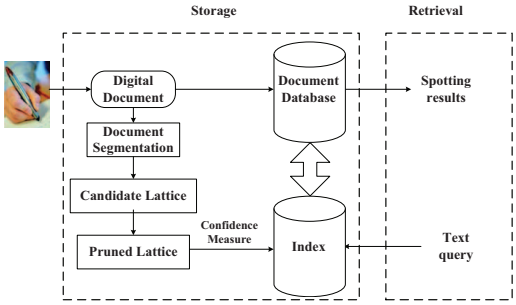

Fig. 1. Block diagram of the keyword spotting system.

with the hope that each component is a character or part of a character. Subject to constraints of character width, consecutive components are combined to generate candidate character patterns, which constitute the segmentation candidate lattice (Fig. 2(b) and Fig. 2(c)). On assigning each candidate pattern a number of candidate classes using a character classifier, the segmentation-recognition candidate lattice (referred to as lattice for brevity) is constructed. Each path in the lattice corresponds to a segmentation-recognition hypothesis of the text line. The lattice also confines the number of possible labels of each candidate character, for the state set actually includes all the categories modeled by the character classifier. However, for fast retrieval and less storage space, a more compact lattice should be generated from the initial dense lattice, on which the character similarity scores (the scores of character-label pairs in the lattice, which are also referred to as edge scores in this paper) are calculated. By concatenating the reduced lattice of each text line, the index of the document is achieved.

In text search, the query word is matched with sequences of candidate characters (paths in the candidate lattice) with every component as the start [2]. The word similarity is obtained by combining the character similarity scores (edge scores). When the word similarity is greater than a threshold, a word instance is located in the document. So, we can see that the estimation of character similarity score, which measures the similarity between a candidate character and a candidate class, is of great importance. For the retrieval of keywords from large databases of multi-writer or writer independent documents, to alleviate the effects of character shape variation, the edge scores are

1520-5363/13 $26.00 © 2013 IEEE
DOI 10.1109/ICDAR.2013.118

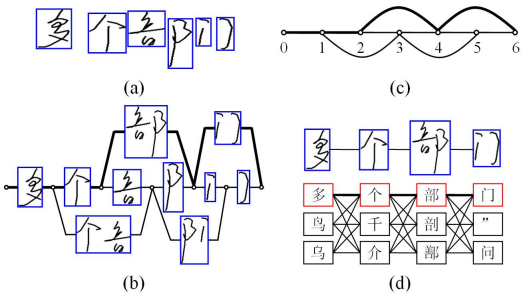

Fig. 2. Generation of segmentation-recognition candidate lattice. (a) Component sequence; (b) Candidate characters; (c) Segmentation candidate lattice, where each node denotes a candidate segmentation point, each edge corresponds to a candidate character, and the bold lines indicate the desired segmentation; (d) Candidate classes of the desired segmentation path.

usually given by a character classifier [2–4]. Ideally, the score of a true character should be higher than any imposters. To improve the discriminability, besides the character classifier outputs, more information, such as the geometric and linguistic contexts, should be incorporated when calculating the edge scores.

In this paper, we propose an indexing method using semi-Markov conditional random fields (semi-CRFs) [7], which are probabilistic graphical models defined on the candidate lattice and provide a principled framework for fusing the information of different contexts. With this model, we reduce the lattice complexity by a forward-backward pruning method which avoids break high-possibility paths, and the edge scores are derived from the marginal probabilities.

II. Related Work

With regard to the word similarity scoring techniques, keyword spotting methods can be categorized into two groups: word shape matching [1] and model-based scoring [2]. The word shape matching technique is based on a distance measure between the query (input image or an image synthesized from the text query) and all candidate word images. Without model training, it is vulnerable to word shape variation and suffers from low matching accuracy. In contrast, the model-based method can be used for retrieving multi-writer or writer independent documents in large database, such as the handwritten Chinese documents used in our experiments, by storing similarity scores computed by a trained character/word model in an index file.

For model-based scoring of handwritten Chinese documents, context information are usually integrated when calculating the character similarity scores. In [3], besides the character classifier outputs, four geometric models are taken into consideration, with the combining weights trained by optimizing a one-vs-all discrimination objective so as to maximize the similarity of true words and minimize the similarity of imposters. Similar method can be found in [4] for text retrieval in online handwritten Japanese documents. Our former work [5] estimates the character confidence based on a N-best recognition list, by combining character classification scores, linguistic and geometric contexts. Compared to the N-best list, lattice can incorporate more competing hypotheses and the estimation of edge scores will be more accurate. In contrast to other scoring methods, semi-CRFs can directly output a probabilistic measure for each edge. N-best list based and word graph based confidence estimation can also be found in speech recognition [6].

III. Building Index Using Semi-CRFs

In this section, we will introduce the semi-CRF model, and how to use this model to prune the candidate lattice as well as how to estimate the edge (character-label pair) scores on the reduced lattice. The compact lattices together with the edge scores (for text lines in a document) are concatenated and stored as index for retrieval.

A. Semi-CRFs on Candidate Lattice

In [7], the semi-CRF model is defined on the lattice (cf. Fig. 2) to directly estimate the a posteriori probability $P(S,Y|X)$ of a hypothesized segmentation-recognition path (S,Y) given the string X:

$$P(S,Y|X) = \frac{1}{Z(X)} \prod_{c \in S} \Psi_c(X, Y_c) \tag{1}$$

where S denotes a segmentation of X (character sequence) and Y denotes a label sequence of S. $\Psi_c(X, Y_c)$ is the potential function on maximal clique c (consecutive characters in the lattice):

$$\Psi_c(X, Y_c) = \exp\left\{\sum_{k=1}^{K} \lambda_k f_k(X_c, Y_c)\right\}. \tag{2}$$

$f_k(X_c, Y_c)$ is the k-th feature function defined on clique c, which models character recognition, geometric or linguistic context. $\Lambda = \{\lambda_k | k = 1, \ldots, K\}$ are the weighting parameters to be learned. $Z(X)$ is the partition function defined as the summation over all the paths in the lattice:

$$Z(X) = \sum_{(S',Y')} \prod_{c \in S'} \Psi_c(X, Y'_c). \tag{3}$$

Given N training samples: $\{(X^i, S^i, Y^i) | i = 1, \ldots, N\}$ (strings with segmentation points and character classes labeled), following the standard MAP estimation, the weighting parameters Λ can be learned by minimizing the negative log-likelihood loss with L_2-norm regularization:

$$L_{NLL}(\Lambda) = -\sum_{i=1}^{N} \log P(S^i, Y^i | X^i; \Lambda) + \frac{C}{2} \|\Lambda\|^2 \tag{4}$$

where C is a positive constant balancing the loss term against the regularization term.

B. Calculation of Edge Scores

To calculate the edge scores, we should first estimate the marginal probabilities on each lattice edge (the probability that an edge is on the desired segmentation-recognition path) using the forward and backward algorithms, and the edge scores are just the logarithm of the marginal probabilities.

Let $m \geq 2$ be the maximal clique size. Assume that the candidate segmentation points are indexed from 0 to T in the lattice (cf. Fig. 2). We denote a sub-segmentation path (character sequence) by $\mathbf{t}_{a:b}$, with $t_a, t_{a+1}, \ldots, t_b$ being $b-a+1$

ordered candidate segmentation points, and the labeling of $\mathbf{t}_{a:b}$ is denoted by $\mathbf{y}_{a+1:b}$, with $y_{a+1}, y_{a+2}, \ldots, y_b$ being $b-a$ labels for each character of $\mathbf{t}_{a:b}$. Let $(S_{t_a}^{t_b}, Y_{t_a}^{t_b})$ be an arbitrary sub-path (character sequence and labeling) from candidate segmentation point t_a to t_b. The forward variables $\{\alpha_{\mathbf{t}_{1:n}}(\mathbf{y}_{2:n})\}$ are calculated on each $(\mathbf{t}_{1:n}, \mathbf{y}_{2:n})$ in the lattice, where t_n=1, ..., T. When t_1=0, 2≤n≤m, otherwise n=m, that is, the size (character number) of $\mathbf{t}_{1:n}$ can be smaller than $m-1$ when it starts from the first segmentation point. $\alpha_{\mathbf{t}_{1:n}}(\mathbf{y}_{2:n})$ is a summation of potential products over all the partial paths (character sequence and labeling) starting from segmentation point 0 and ending at $(\mathbf{t}_{1:n}, \mathbf{y}_{2:n})$. The forward variables and the partition function $Z(X)$ can be calculated by the forward algorithm (sum-product):

(1) Initialization

$$\alpha_{\mathbf{t}_{1:n}}(\mathbf{y}_{2:n}) = \sum_{j=2}^{n} \Psi_{\mathbf{t}_{1:j}}(X, \mathbf{y}_{2:j}), \quad \text{for } t_1 = 0,\ 2 \le n \le m, \tag{5}$$

(2) Recursion

$$\begin{aligned}\alpha_{\mathbf{t}_{1:n}}(\mathbf{y}_{2:n}) &= \sum_{\substack{(S_0^{t_n}, Y_0^{t_n}):\\ (\mathbf{t}_{1:n}, \mathbf{y}_{2:n}) \in (S_0^{t_n}, Y_0^{t_n})}} \prod_{c \in S_0^{t_n}} \Psi_c(X, Y_c) \\ &= \sum_{t_0, y_1} \Psi_{\mathbf{t}_{0:n}}(X, \mathbf{y}_{1:n}) \alpha_{\mathbf{t}_{0:n-1}}(\mathbf{y}_{1:n-1}), \\ &\qquad \text{for } t_1 \neq 0, n = m,\end{aligned} \tag{6}$$

(3) Termination

$$Z(X) = \sum_{\mathbf{t}_{1:n}, \mathbf{y}_{2:n}} \alpha_{\mathbf{t}_{1:n}}(\mathbf{y}_{2:n}), \text{ for } t_n = T. \tag{7}$$

In the above algorithm, $\Psi_{\mathbf{t}_{0:n}}(X, \mathbf{y}_{1:n})$ denotes the potential on maximal clique-labeling pair $(\mathbf{t}_{0:n}, \mathbf{y}_{1:n})$. $\{\alpha_{\mathbf{t}_{1:n}}(\mathbf{y}_{2:n})\}$ are calculated sequentially from t_n=1 to t_n=T.

Similarly, we can deduce the backward variables $\{\beta_{\mathbf{t}_{0:n-1}}(\mathbf{y}_{1:n-1})\}$ defined on each $(\mathbf{t}_{0:n-1}, \mathbf{y}_{1:n-1})$ in the lattice, where t_0=0, ..., $T-1$. When t_{n-1}=T, 2≤n≤m, otherwise n=m. $\beta_{\mathbf{t}_{0:n-1}}(\mathbf{y}_{1:n-1})$ is a summation of potential products over all the partial paths starting from $(\mathbf{t}_{0:n-1}, \mathbf{y}_{1:n-1})$ and ending at segmentation point T.

(1) Initialization

$$\beta_{\mathbf{t}_{0:n-1}}(\mathbf{y}_{1:n-1}) = 1, \text{ for } t_{n-1} = T,\ 2 \le n \le m, \tag{8}$$

(2) Recursion

$$\begin{aligned}\beta_{\mathbf{t}_{0:n-1}}(\mathbf{y}_{1:n-1}) &= \sum_{\substack{(S_{t_0}^{T}, Y_{t_0}^{T}):\\ (\mathbf{t}_{0:n-1}, \mathbf{y}_{1:n-1}) \in (S_{t_0}^{T}, Y_{t_0}^{T})}} \prod_{c \in S_{t_{n-1}}^{T}} \Psi_c(X, Y_c) \\ &= \sum_{t_n, y_n} \Psi_{\mathbf{t}_{0:n}}(X, \mathbf{y}_{1:n}) \beta_{\mathbf{t}_{1:n}}(\mathbf{y}_{2:n}), \\ &\qquad \text{for } t_{n-1} \neq T, n = m,\end{aligned} \tag{9}$$

$\{\beta_{\mathbf{t}_{0:n-1}}(\mathbf{y}_{1:n-1})\}$ is calculated in reverse order from t_0=$T-1$ to t_0=0.

From the forward and backward variables, we can calculate the marginal probability on an edge $(\mathbf{t}_{0:1}, y_1)$:

$$\begin{aligned}P(\mathbf{t}_{0:1}, y_1|X) &= \sum_{(S,Y):(\mathbf{t}_{0:1}, y_1) \in (S,Y)} P(S, Y|X) \\ &= \frac{1}{Z(X)} \sum_{\substack{\mathbf{t}_{2-m:1},\\ \mathbf{y}_{3-m:1}}} \sum_{\substack{\mathbf{t}_{0:m-1},\\ \mathbf{y}_{1:m-1}}} \alpha_{\mathbf{t}_{2-m:1}}(\mathbf{y}_{3-m:1}) \\ &\quad \times \beta_{\mathbf{t}_{0:m-1}}(\mathbf{y}_{1:m-1}) \prod_{c \in \mathbf{t}_{1:m-1}} \Psi_c(X, Y_c), \\ &\qquad \text{for } m \ge 2.\end{aligned} \tag{10}$$

where $\{\alpha_{\mathbf{t}_{2-m:1}}(\mathbf{y}_{3-m:1})\}$ denotes all the forward variables ending at $(\mathbf{t}_{0:1}, y_1)$, and $\{\beta_{\mathbf{t}_{0:m-1}}(\mathbf{y}_{1:m-1})\}$ denotes all the backward variables starting from $(\mathbf{t}_{0:1}, y_1)$.

C. Lattice Pruning

To build a compact index, the edge scores are computed on a pruned lattice. The difficulty of lattice pruning lies in that the edges are not independent, i.e., to remove an edge will break the paths through it. To avoid breaking the high-possibility paths, we consider a forward-backward lattice pruning method.

Replacing the summation by maximization in Eq. (6) and Eq. (9) will yield another type of forward and backward variables ($\{\hat{\alpha}_{\mathbf{t}_{1:n}}(\mathbf{y}_{2:n})|t_n$=1, ..., $T\}$ and $\{\hat{\beta}_{\mathbf{t}_{0:n-1}}(\mathbf{y}_{1:n-1})|t_0$=0, ..., $T-1\}$, respectively), with which we can calculate the posterior probability of the best path traversing an edge by replacing the summation with maximization in Eq. (10). In this paper, the initial dense lattice will be purged by a first-order semi-CRF (m=2), thus

$$\begin{aligned}&P(S^*, Y^*, \mathbf{t}_{0:1} \in S^*, y_1 \in Y^*|X) \\ &= \max_{\substack{(S,Y):\\ (\mathbf{t}_{0:1}, y_1) \in (S,Y)}} P(S, Y|X) = \frac{1}{Z(X)} \hat{\alpha}_{\mathbf{t}_{0:1}}(y_1) \hat{\beta}_{\mathbf{t}_{0:1}}(y_1).\end{aligned} \tag{11}$$

In Eq. (11), $Z(X)$ is a constant with trained parameters. The value $\hat{\alpha}_{\mathbf{t}_{0:1}}(y_1)\hat{\beta}_{\mathbf{t}_{0:1}}(y_1)$ will be used as a score of $(\mathbf{t}_{0:1}, y_1)$ for pruning the lattice. Note that $\hat{\alpha}_{\mathbf{t}_{0:1}}(y_1)\hat{\beta}_{\mathbf{t}_{0:1}}(y_1)$ is also the score of the best path traversing $(\mathbf{t}_{0:1}, y_1)$.

Considering that a high-score path is unlikely to go through a low-score edge, to remove a low-score edge will unlikely break a high-score path. Denote the best path score in the dense lattice by Q_{max}, an edge is reserved if the following condition is held:

$$\log Q_{max} - \log(\hat{\alpha}_{\mathbf{t}_{0:1}}(y_1)\hat{\beta}_{\mathbf{t}_{0:1}}(y_1)) \le \gamma_p, \tag{12}$$

where γ_p>0 is the pruning threshold. With this method, the paths with higher scores are retained, while those with lower scores are discarded.

IV. Keyword Detection

In keyword spotting, the similarity between the query word $\mathbf{y}_{1:n}$=$y_1, y_2, \ldots, y_n$ and a sequence of n candidate characters $\mathbf{t}_{0:n}$, with $t_0, t_1, \ldots, t_n$ being n+1 ordered candidate segmentation points, is obtained by combining the similarity of candidate characters [2]:

$$SIM(\mathbf{y}_{1:n}, \mathbf{t}_{0:n}) = \frac{1}{n} \sum_{i=1}^{n} sim(y_i, \mathbf{t}_{i-1:i}), \tag{13}$$

where $sim(y_i, \mathbf{t}_{i-1:i})$ denotes the similarity between candidate character $\mathbf{t}_{i-1:i}$ and class y_i, and is defined as the logarithm of marginal probability on edge $(\mathbf{t}_{i-1:i}, y_i)$ (cf. Section III-B).

We use a character-synchronous dynamic search algorithm for efficient search of query word from the index [2]. The query word is matched with sequences of candidate characters (paths in the candidate lattice) with every component as the start. When the word similarity score is greater than a certain threshold, a word instance is located in the document. For overlapping instances after search, only the one with highest word score is retained.

V. Experiments

We evaluated the performance of the proposed keyword spotting method on a database of online Chinese handwriting: CASIA-OLHWDB [8]. This database is divided into six datasets, three for isolated characters DB1.0-1.2 (called DB1 in brief) and three for handwritten texts DB2.0-2.2 (DB2 in brief). There are 3,912,017 isolated character samples and 5,092 handwritten pages (52,220 text lines) in total. Both the isolated data and handwritten text data have been divided into standard training set (816 writers) and test set (204 writers). The training set contains 3,129,496 isolated character samples of 7,356 classes and 4,072 pages of handwritten texts (41,710 text lines, including 1,082,220 characters of 2,650 classes). The presented system was evaluated on the test set of DB2, which contains 10,510 text lines from 1,020 text pages, including 269,674 characters of 2,631 classes.

A. Experimental Setting

The features functions employed in our experiments include character recognition score, geometric and linguistic contexts [7]. The character recognition scores are given by the modified quadratic discriminant function (MQDF) [10]. The scores for class-dependent and class-independent geometries are given by the quadratic discriminant functions (QDFs) and the linear SVMs, respectively. All the classifier outputs are transformed to confidence [7]. Unless otherwise stated, the marginal probabilities are estimated by a second-order semi-CRF (the maximal clique size m is 3), and the default lattice pruning threshold γ_p is 10.

For evaluating the retrieval performance on the test dataset, we use the high-frequency words in the lexicon of the Sogou labs [11] as query words. The top 60,000 frequently used words, including 39,057 two-character words, 9,975 three-character words and 9,451 four-character words, were tested in our experiments. The keyword spotting performance is measured using three metrics: recall (R, percentage of correctly detected words among the true words), precision (P, percentage of correct words among the detected ones) and F-measure (the harmonic mean of P and R) [2].

Our experiments were implemented on a PC with Intel(R) Core(TM)2 Duo CPU E8400, 3.00 GHz processor and 2GB RAM, and the algorithms were programmed using C++.

B. Effects of Lattice Pruning

In this paper, the initial dense lattice will be purged by a first-order semi-CRF (the maximal clique size m is 2) (cf. Section III-C). Table I lists the effects of different lattice pruning thresholds (γ_p), where the precision (P) and recall (R) rates correspond to the maximum F-measure (F). The total recall (TR) rate is defined as the number of true instances in the lattice divided by the number of instances in the transcript, which is a upper bound of the recall rate. Lattice edge density (LED) is defined as the total number of edges divided by the total number of characters in the transcript, which is used to measure the complexity of the index. From Table I we can see that, by decreasing γ_p, lattice pruning can effectively reduce LED and consequently the size of index, while P, R and F decrease just slightly when γ_p is not too small. The default threshold γ_p=10 performs sufficiently well in respect of P, R and F. Enlarging the threshold, though increases the total recall rate, does not improve three metrics.

TABLE I. Effects of lattice pruning.

γ_p	1	5	10	15	20	30
P (%)	97.46	97.91	98.05	98.07	98.07	98.07
R (%)	92.91	93.32	93.30	93.28	93.28	93.28
F (%)	95.13	95.56	95.61	95.62	95.62	95.62
TR (%)	93.65	95.05	95.49	95.64	95.73	95.80
LED	1.05	1.45	2.94	6.07	10.86	23.83
Index-Size (Mb)	5.96	6.49	8.51	12.73	19.19	36.68

C. Effects of Contexts

Fig. 3 compares the precision-recall curves of different contexts, which are modeled by feature functions in semi-CRF (cf. Section III-A). In Fig. 3, "c", "g" and "l" denote the feature functions for character recognition, geometric context and linguistic context, respectively. The details of feature functions can be found in [7]. From Fig. 3 we can see that when using the character recognition model ("c") only, the performance is remarkably improved by combining geometric models ("c+g"). The incorporation of language model ("c+l") is much more effective than the geometric model. The best result is given by combining both the two types of contexts ("c+g+l").

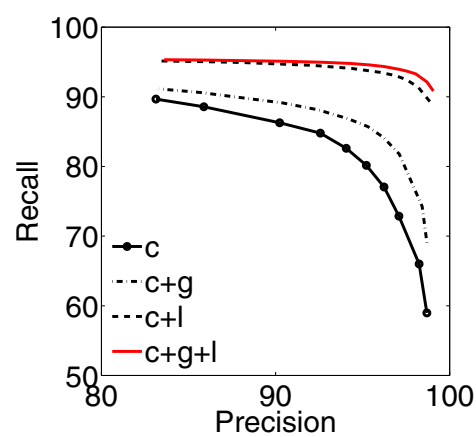

Fig. 3. Precision-Recall curves of spotting results of different contexts.

D. Comparison with Transcription-Based Search

Because of the limited accuracy in handwriting recognition, most previous works on keyword spotting avoid using a text recognition system to transcribe the handwriting and search on the output text. However, some experimental studies (e.g., the

one in [12]) show that transcription-based spotting can perform competitively. Here, on handwritten Chinese documents, we compared the proposed method with transcription-based search using a semi-CRF based text line recognition method [7]. Since text line recognition gives unique text output, transcription-based word search gives a unique point of precision-recall rates (Fig. 4). In contrast, the proposed method provides flexible options of tradeoff between the two metrics. By properly sacrificing the precision rate, much higher recall rate can be achieved than the transcription-based method.

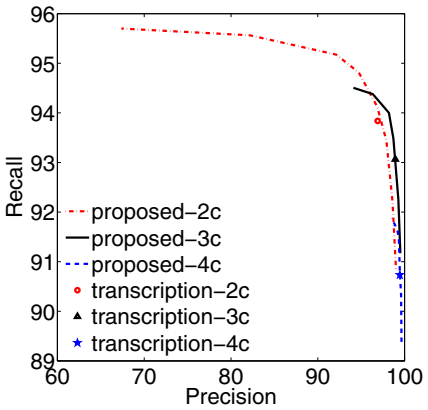

Fig. 4. Precision-Recall curves of keyword spotting by proposed method and transcription-based word search (-2C, -3C and -4C means words of two, three and four characters, respectively).

E. Comparison with N-Best-Based Scoring

Our former work [5] estimates the character confidence based on a N-best recognition list, searched for using the beam search algorithm. The paths are evaluated by combining character classification outputs, geometric context and linguistic context (bigram language model) [9]. The scores of N-best paths are converted to posterior probabilities using soft-max, and the probabilities of character classes are computed from the path probabilities. In contrast, the edge probabilities in the proposed method are directly estimated on the reduced lattice. Fig. 5 compares the precision-recall curves of the proposed method and the N-best-based method. For fairly comparison, bigram language model is used for both the two methods, and N is set to 50 as in [5]. From Fig. 5 we can see that the proposed method outperforms the N-best-based approach, which means that the probabilities estimated on the lattice are more accurate than those estimated on the N-best list.

VI. Conclusion

In this paper, we present a method for indexing online handwritten Chinese documents based on semi-CRFs, which provide a principled framework for fusing the information of character recognition, geometric and linguistic contexts. Using a forward-backward lattice pruning method, we achieve a compact index, in which the character similarity scores are estimated by the semi-CRF model. Experimental results on CASIA-OLHWDB database demonstrate that the proposed method outperforms transcription-based search and N-best based scoring.

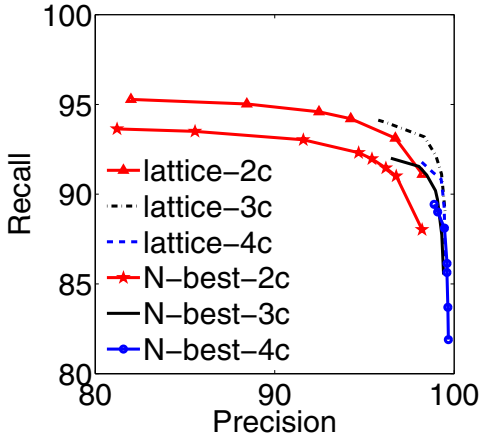

Fig. 5. Precision-Recall curves of keyword spotting by the previous and proposed method

Acknowledgment

This work is supported by the National Natural Science Foundation of China (NSFC) under grants no. 60933010 and no. 61273269.

References

[1] R. Manmatha, C. Han and E. M. Riseman, Word spotting: a new approach to indexing handwriting, in: Proc. CVPR, 1996, pp. 631-637.
[2] H. Zhang, D. H. Wang and C. L. Liu, Keyword spotting from online Chinese handwritten documents using one-vs-all trained character classifier, in: Proc. ICFHR, 2010, pp. 271-276.
[3] L. Huang, F. Yin, Q. H. Chen and C. L. Liu, Keyword spotting in offline Chinese handwritten documents using a statistical model, in: Proc. ICDAR, 2011, pp. 78-82.
[4] C. Cheng, B. Zhu and M. Nakagawa, Digital ink search based on character-recognition candidates compared with feature-matching-based approach, IEICE Trans. on Inf. & Syst. 96(3) (2013), in press.
[5] H. Zhang, D. H. Wang and C. L. Liu, A Confidence-Based Method for Keyword Spotting in Online Chinese Handwritten Documents, in: Proc. ICPR, 2012, pp. 525-528.
[6] F. Wessel, R. Schlter, K. Macherey and H. Ney, Confidence measures for large vocabulary continuous speech recognition, IEEE Trans.Speech Audio Process. 9(3) 2001 288-298.
[7] X.D. Zhou, D.H. Wang, F. Tian, C.L. Liu, M. Nakagawa, Handwritten Chinese/Japanese text recognition using semi-Markov conditional random fields, *IEEE Trans. Pattern Anal. Mach. Intell.*, doi:10.1109/TPAMI.2013.49, in press.
[8] C. L. Liu, F. Yin, D. H. Wang and Q. F. Wang, CASIA online and offline Chinese handwriting databases, in: Proc.ICDAR, 2011, pp. 37-41.
[9] D.-H. Wang, C.-L. Liu, X.-D. Zhou, An approach for real-time recognition of online Chinese handwritten sentences, Pattern Recognition 45 (10) (2012) 3661-3675.
[10] F. Kimura, K. Takashina, S. Tsuruoka, Y. Miyake, Modified quadratic discriminant functions and the application to Chinese character recognition, IEEE Trans. Pattern Anal. Mach. Intell. 9 (1) (1987) 149-153.
[11] SogouLab: http://www.sogou.com/labs/resources.html.
[12] V. Frinken, A. Fischer, R. Manmatha, H. Bunke, A novel word spotting method based on recurrent neural networks, IEEE Trans. Pattern Anal. Mach. Intell. 34 (2) (2012) 211-224.