IJDAR (2011) 14:145–157
DOI 10.1007/s10032-010-0139-z

ORIGINAL PAPER

Unconstrained handwritten document retrieval

Huaigu Cao · Venu Govindaraju · Anurag Bhardwaj

Received: 23 December 2009 / Revised: 29 July 2010 / Accepted: 25 October 2010 / Published online: 16 November 2010

Abstract With the ever-increasing growth of the World Wide Web, there is an urgent need for an efficient information retrieval system that can search and retrieve handwritten documents when presented with user queries. However, unconstrained handwriting recognition remains a challenging task with inadequate performance thus proving to be a major hurdle in providing robust search experience in handwritten documents. In this paper, we describe our recent research with focus on information retrieval from noisy text derived from imperfect handwriting recognizers. First, we describe a novel term frequency estimation technique incorporating the word segmentation information inside the retrieval framework to improve the overall system performance. Second, we outline a taxonomy of different techniques used for addressing the noisy text retrieval task. The first method uses a novel bootstrapping mechanism to refine the OCR'ed text and uses the cleaned text for retrieval. The second method uses the uncorrected or raw OCR'ed text but modifies the standard vector space model for handling noisy text issues. The third method employs robust image features to index the documents instead of using noisy OCR'ed text. We describe these techniques in detail and also discuss their performance measures using standard IR evaluation metrics.

Dr. Cao's work presented in this article was done during the fulfillment of his Ph.D. degree in the Department of Computer Science and Engineering, University at Buffalo.

H. Cao (✉)
Raytheon BBN Technologies, Cambridge, MA 02138, USA
e-mail: hcao@bbn.com

V. Govindaraju · A. Bhardwaj
Department of Computer Science and Engineering,
University at Buffalo, Amherst, NY 14260, USA
e-mail: govind@buffalo.edu

A. Bhardwaj
e-mail: rushtoanuragin@gmail.com

1 Introduction

The need for easy access to a vast repository of both historical and contemporary handwritten documents (e.g. handwritten medical records, historical manuscripts, personal notes) has led to an ever-increasing demand for an efficient information retrieval system that is able to search and retrieve handwritten documents when presented with user queries. Over the recent years, the retrieval methods have improved the quality of document search in a dramatic fashion. However, these methods are predominantly applicable to documents with ASCII text or machine-printed document images where the task of automatic word recognition is relatively easier. We are interested in recognition of unconstrained handwritten documents, which is a significantly more challenging task due to the large variations in writing styles and document image quality. The methods developed for ASCII text and machine-printed documents are rendered ineffectively due to low OCR accuracy. In this paper, we present an overview of our own research in information retrieval and extraction from noisy OCR'ed text extracted from unconstrained handwritten documents. Specifically, our focus is on handwritten medical forms (Fig. 1 [8]) that prove to be a very complex and challenging domain for any automatic recognition or retrieval system. The challenges of automatic transcription lies in three respects: (1) large variability in handwriting samples given the multiple authors even with a single document, different response format and choice of text in the case of emergency medical conditions, (2) poor image quality, and (3) a large lexicon (dictionary) of medical words that can be around 5,000 words. In this paper, we outline three separate

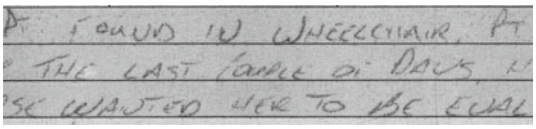

Fig. 1 The text in a handwritten medical form [8]

research directions for tacking this problem in form of OCR correction-based retrieval methods, modified vector model-based methods and keyword spotting-based methods.

The rest of the paper is organized as follows. Section 2 describes the related work in the area of information retrieval from handwritten documents. Section 4 presents OCR model-based document retrieval techniques and describes a novel term frequency estimation technique in detail. Keyword spotting-based techniques are presented in Sect. 5. Conclusions are outlined in Sect. 6.

2 Background

Several works have tried to improve the quality of information retrieval on OCR'ed text. Previous work [2,11] has shown that the IR performance is adversely affected by the noise present in the OCR output due to low recognition performance. A typical solution to this problem is to correct OCR errors using post-processing techniques. [14,19,20] propose different methods of OCR correction for improving the information retrieval performance. Mittendorf et al. [19] propose a probabilistic model for OCR errors and use it to design a term-weighting scheme for information retrieval from document images. In Ohta et al. [20], specific character transformations and character occurrence bi-grams were used to generate candidate terms for each "true" search term. Documents retrieved by each candidate term are then evaluated for inclusion into the final result set. This approach results in minor improvements in recall for moderate quality OCR documents. Jing et al. [14] build a language model that takes OCR errors into account. This model approximates an "uncorrupted" version of a particular document for efficient retrieval.

Due to the poor recognition results in handwritten documents, it is not feasible to apply probabilistic modeling of OCR output. Recently, there has been much focus on information retrieval from handwritten documents. Lee et al. [16] propose to use top-k hypothesis from the OCR instead of using just the top choice. They report that using multiple recognition choices for retrieval improves the overall recall of the system. Rath et al. [22] propose an IR model that assumes independence between each term of the query for the purpose of computing its similarity with a given document. The frequency of each term is computed using the posterior probability estimated from the word image features. Howe et al. [13] use the same IR model as [22], but they do not use word recognition probabilities. Instead, they model the ranking as a zipfian distribution where word recognition probability is inversely proportional to its rank.

As an alternative to OCR-based document indexing and retrieval techniques, keyword spotting from handwriting has gained significant research attention lately for solving the document retrieval task. Keyword Spotting is defined as an image matching task where the input query is matched against candidate word images of the document. The word-level matching scores can then be combined to generate a document level score that can be used for ranking the documents in order of query similarity. There are two major classes of keyword spotting: (i) Unconstrained Keyword Spotting: also known as "recognition-based keyword spotting" that relies on an OCR (i.e. HMM) to provide probabilistic values for each keyword in the lexicon that can later be integrated into a document-level ranking score [10,12,21,25], (ii) Isolated Keyword Spotting: also known as "recognition free keyword spotting" that is the typical image matching task for generating the document relevance score for an input query [5,18,24,26,27].

3 OCR correction-based IR

This technique can be described as a multi-pass technique to boost recognition and then perform retrieval on the refined OCR'ed text. In the first pass, an OCR correction model is employed, which improves the recognition rate of the handwritten word recognizer. It is generally believed that in context of handwritten word recognition, reducing the size of lexicon translates to an improvement in word recognition performance. In our work by Milewski et al. [17], we propose a lexicon reduction-based strategy for OCR correction in context of handwritten medical forms. It can be understood as a novel bootstrapping algorithm, where a sequence of confidently recognized characters from word recognizers are used to encode the document category that is further used to reduce the initial lexicon by considering only the category-specific terms.

First, all the forms that are manually categorized under a specific topic are used to generate a lexicon of words. These lexicons are then used to extract phrases from initial word recognition output using a cohesion-based metric. These phrases are then encoded into a phrase-category matrix on which singular valued decomposition (SVD) is performed. The resultant matrix is then used to compute a topic or category distribution for every test document. Finally, a second phase of recognition is performed that uses a reduced lexicon, corresponding to document category. In a similar work on OCR correction, Bhardwaj et al. [3] also use the document topic information to obtain a refined posterior probability of every

lexicon term. However, their approach is different from [17] since instead of reducing the initial lexicon, they create a re-weighted lexicon for word recognition. In the retrieval phase, they utilize a two-pass document scoring method for ranking each document in the collection corresponding to the input query. First, each document is ranked according to the frequency of query terms present in it. In cases, where similar number of query terms are present, a secondary phase re-scores these documents using a distance metric $d(a_i, b_j)$ that computes the distance between two matched words, a_i and b_j such that i and j, respectively, represent the word position in the document. w_{ij} represents a weight based on the frequency of occurrences of words a and b in the document.

$$d(a_i, b_j) = w_{ij} * \frac{1}{|a_i - b_j|} \tag{1}$$

3.1 Performance measures

The retrieval performance is evaluated using standard MAP(mean average precision) and R-precision metrics. MAP denotes the mean value of the average precision of all queries in the set. R-precision measures the precision of the retrieval system at a point when the total number of documents retrieved is R. Figure 2 illustrates the values of these metrics for different search experiments. 'OR' experiments refer to methods when either of the query terms are searched. 'AND' experiments refer to methods when all query terms are searched. 'RL' and 'CL' methods refer to reduced lexicon- and complete lexicon-based methods. A complete lexicon of 4,570 words is used in the experiment of [17]. Through the lexicon reduction, the complete lexicon is decomposed into smaller lexicons of 20 categories, each containing 400 to 1,200 words. The average size of the reduced lexicons is 574. As seen from the figure, reduced lexicon-based search methods outperform the complete lexicon-based searches due to better quality of the recognized text.

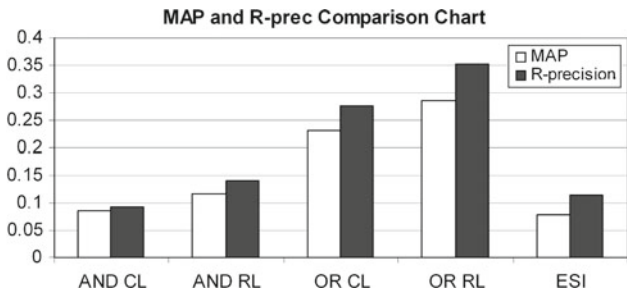

Fig. 2 Mean Average Precision and R-Precision comparison for correction based retrieval Model [17]

4 Adapted vector model-based IR

4.1 Classic vector model

In the classic Vector Model [1], the documents are represented by the vector space of terms. A term is a word from the vocabulary of all of the documents. Given the vocabulary $\{t_i\}$, $1 \le i \le N$, the term frequency $tf_{i,j}$ is defined by

$$tf_{i,j} = \frac{\text{freq}_{i,j}}{L_j}, \quad i = 1, \ldots, N \tag{2}$$

where $\text{freq}_{i,j}$ is the term count, i.e., the number of occurrences of term t_i in document d_j, and L_j is the total number of occurrences of all the terms in document d_j. The inverse document frequency (IDF) of a term is defined by

$$idf_i = \log \frac{\#\{d_j\}}{\#\{d_j | \text{freq}_{i,j} > 0\}}, \quad i = 1, \ldots, N \tag{3}$$

where $\#\{\cdot\}$ denotes the number of elements in set $\{\cdot\}$. The IDF of a term shows the importance of the term: a term that appears in most documents is less important than a term that appears in only a few documents. A query is also represented by the vector of terms. The query term frequency (QTF) of query q is defined as

$$tf_{i,q} = \begin{cases} 1, & \text{if term } t_i \text{ is in } q \\ 0, & \text{otherwise} \end{cases} \quad i = 1, \ldots, N \tag{4}$$

and the query is represented by vector $\left[tf_{1,q}, tf_{2,q}, \ldots, tf_{N,q}\right]$.

The similarity between document d_j and query q is defined as

$$\text{sim}(d_j, q) = \sum_{i=1}^{N} tf_{i,j} \cdot idf_i \cdot tf_{i,q}. \tag{5}$$

4.2 Modified vector model

The term count $\text{freq}_{i,j}$ is not immediately available from the document image and need to be estimated. Thus, we modify the definitions of TF and IDF in Eqs. (2) and (3): the modified TF is

$$tf'_{i,j} = \frac{\widehat{\text{freq}_{i,j}}}{L_j}, \tag{6}$$

and the modified IDF

$$idf'_i = \log \frac{\#\{d_j\}}{\max\left\{1, \#\{d_j | \widehat{\text{freq}_{i,j}} > 0.5\}\right\}} \tag{7}$$

where $\widehat{\text{freq}_{i,j}}$ is an estimation of $\text{freq}_{i,j}$. Note that here we require that $\widehat{\text{freq}_{i,j}} > 0.5$ which is equivalent to a rounding function of the expected value of $\text{freq}_{i,j}$, i.e., $round$

$(\widehat{\text{freq}_{i,j}}) \geq 1$. We estimate $\text{freq}_{i,j}$ using the its expected value:

$$\widehat{\text{freq}_{i,j}} = E\{\text{freq}_{i,j}\}, \tag{8}$$

The text length in Eq. (6) is estimated by

$$L_j = \sum_{i=1}^{N} \widehat{\text{freq}_{i,j}} \tag{9}$$

The similarity between document image d_j and the query q is given by

$$\text{sim}(d_j, q) = \sum_{i=1}^{N} tf'_{i,j} \cdot idf'_i \cdot tf_{i,q}. \tag{10}$$

Suppose document d_j is represented by the observed image features $\overrightarrow{o}$, and $\overrightarrow{w} = w_1 w_2 \ldots w_L$ denotes an arbitrary segmentation of $\overrightarrow{o}$, where $w_1, \ldots, w_L$ are word images. The expected value of $\text{freq}_{i,j}$ is given by

$$E\{\text{freq}_{i,j}\} = \sum_{\overrightarrow{w}} \Pr(\overrightarrow{w} \mid \overrightarrow{o}) \cdot \sum_{\overrightarrow{\tau}} \Pr(\overrightarrow{\tau} \mid \overrightarrow{w}) \cdot \#_{t_i}(\overrightarrow{\tau}) \tag{11}$$

where $\overrightarrow{\tau} = \tau_1 \ldots \tau_L$ is a sequence of terms. $\Pr(\overrightarrow{w} \mid \overrightarrow{o})$ is the probability that $\overrightarrow{w}$ is a correct segmentation. $\Pr(\overrightarrow{\tau} \mid \overrightarrow{w})$ is the word sequence recognition probability. $\#_{t_i}(\overrightarrow{\tau})$ is the number of term t_i occurring in sequence $\overrightarrow{\tau}$.

Equation (11) can be simplified in some special situations. $\overrightarrow{w}$ is unique and $\Pr(\overrightarrow{w} \mid \overrightarrow{o}) \equiv 1$, if we assume the correct segmentation $\overrightarrow{w}$ is known. Thus, Eq. (11) is equivalent to

$$E\{\text{freq}_{i,j}\} = \sum_{\overrightarrow{\tau}} \Pr(\overrightarrow{\tau} \mid \overrightarrow{w}) \cdot \#_{t_i}(\overrightarrow{\tau}) \tag{12}$$

In addition to the assumption of knowing the correct segmentation, assuming the independence of terms $\tau_1, \tau_2, \ldots, \tau_L$, i.e.,

$$\Pr(\overrightarrow{\tau} \mid \overrightarrow{w}) = \prod_{k=1}^{L} \Pr(\tau_k | w_k), \tag{13}$$

then Eq. (12) is equivalent to

$$E\{\text{freq}_{i,j}\} = \sum_{k=1}^{L} \Pr(\tau_k | w_k) \tag{14}$$

Equation (14) is used in [13,22] for retrieval. It is a solution to Eq. (11) based on the assumptions of perfect word segmentation and independence of terms. In the general case, given the probability of every single segmentation point and a language model (n-gram), we can solve Eq. (11) by dynamic programming.

4.3 Estimating term count $\text{freq}_{i,j}$

The observational sequence of a document image can be represented by a sequence of connected components sorted in the reading order. Since the following discussion focuses on a single document, we can omit the subscript j of d_j from notations like $\text{freq}_{i,j}$ without ambiguity.

Given N consecutive connected components $c_1, \ldots c_n$ and the set of terms $t_1, \ldots t_N$, we use a dynamic programming-based algorithm to compute the term count. We assume a word image is composed of at most C connected components. The term count of t_i in sequence $c_1, \ldots, c_k$ $(0 < k \leq n)$ is denoted by freq_i^k. The probability that the last word of sequence $c_1, \ldots, c_k$ is term t_i is denoted by λ_i^k. The probability that the gap after the connected component c_k is a true word gap is denoted by σ_k. When we define freq_i^k and σ_k on a sequence $c_1, \ldots, c_k$, we assume $\sigma_0 = \sigma_k = 1$. Next, we will present the formulae of estimating term counts when the language model is a bi-gram. The formulae for higher-order LM's can be derived similarly.

When $k = 0$, the sequence is empty, and thus

$$E(\text{freq}_i^0) = 0 \tag{15}$$

When $k = 1$, the only possible segmentation is that c_1 is a word image, and thus

$$E(\text{freq}_i^1) = \frac{p_i \cdot \Pr(c_1|t_i)}{\sum_{i_2=1}^{N} p_{i_2} \cdot \Pr(c_1|t_{i_2})} \tag{16}$$

When $k = 2$, the last word image can be either c_2 or $c_1 c_2$. The probability that c_2 is recognized as t_i is

$$\frac{\sum_{i_1=1}^{N} (\lambda_{i_1}^1 p_{i_1 \to i}) \cdot \Pr(c_2|t_i)}{\sum_{i_2=1}^{N} \sum_{i_1=1}^{N} (\lambda_{i_1}^1 p_{i_1 \to i_2}) \cdot \Pr(c_2|t_{i_2})}, \tag{17}$$

where $p_{i_1 \to i_2}$ represents the transition probability from term t_{i_1} to term t_{i_2} and $\Pr(c_2|t_i)$ is the probability density of observation c_2 in class t_i. The probability that $c_1 c_2$ is recognized as t_i is

$$\frac{p_i \cdot \Pr(c_1 c_2|t_i)}{\sum_{i_2=1}^{N} p_{i_2} \cdot \Pr(c_1 c_2|t_{i_2})} \tag{18}$$

Thus,

$$\begin{aligned} &E(\text{freq}_i^2) \\ &= \sigma_1 \cdot \left(\text{freq}_i^1 + \frac{\sum_{i_1=1}^{N} \left(\lambda_{i_1}^1 p_{i_1 \to i} \right) \cdot \Pr(c_2|t_i)}{\sum_{i_2=1}^{N} \sum_{i_1=1}^{N} \left(\lambda_{i_1}^1 p_{i_1 \to i_2} \right) \cdot \Pr(c_2|t_{i_2})} \right) \\ &\quad + (1 - \sigma_1) \cdot \frac{p_i \cdot \Pr(c_1 c_2|t_i)}{\sum_{i_2=1}^{N} p_{i_2} \cdot \Pr(c_1 c_2|t_{i_2})} \end{aligned} \tag{19}$$

For an arbitrary $k > 0$, we can prove that the number of terms

$$E(\text{freq}_i^k) = \sum_{c=1}^{k-1} \sigma_{k-c} \cdot \left(\prod_{k-c<q<k} (1-\sigma_q) \right) \cdot \left(\text{freq}_i^{k-c} + \frac{\sum_{i_1=1}^{N} (\lambda_{i_1}^{k-c} p_{i_1 \to i}) \cdot \Pr(c_{k-c+1} \ldots c_{k-1} c_k | t_i)}{\sum_{i_2=1}^{N} \sum_{i_1=1}^{N} (\lambda_{i_1}^{k-c} p_{i_1 \to i_2}) \cdot \Pr(c_{k-c+1} \ldots c_{k-1} c_k | t_{i_2})} \right) + \left(\prod_{0<q<k} (1-\sigma_q) \right) \cdot \frac{p_i \cdot \Pr(c_1 \ldots c_{k-1} c_k | t_i)}{\sum_{i_2=1}^{N} p_{i_2} \cdot \Pr(c_1 \ldots c_{k-1} c_k | t_{i_2})} \quad \text{if } k \le C; \tag{20}$$

and

$$E(\text{freq}_i^k) = \sum_{c=1}^{C} \sigma_{k-c} \cdot \left(\prod_{k-c<q<k} (1-\sigma_q) \right) \cdot \left(\text{freq}_i^{k-c} + \frac{\sum_{i_1=1}^{N} (\lambda_{i_1}^{k-c} p_{i_1 \to i}) \cdot \Pr(c_{k-c+1} \ldots c_{k-1} c_k | t_i)}{\sum_{i_2=1}^{N} \sum_{i_1=1}^{N} (\lambda_{i_1}^{k-c} p_{i_1 \to i_2}) \cdot \Pr(c_{k-c+1} \ldots c_{k-1} c_k | t_{i_2})} \right) \quad \text{if } k > C. \tag{21}$$

Similarly, we can prove that

$$\lambda_i^0 = \frac{1}{N};$$

$$\lambda_i^k = \sum_{c=1}^{k-1} \sigma_{k-c} \cdot \left(\prod_{k-c<q<k} (1-\sigma_q) \right) \cdot \frac{\sum_{i_1=1}^{N} (\lambda_{i_1}^{k-c} p_{i_1 \to i}) \cdot \Pr(c_{k-c+1} \ldots c_{k-1} c_k | t_i)}{\sum_{i_2=1}^{N} \sum_{i_1=1}^{N} (\lambda_{i_1}^{k-c} p_{i_1 \to i_2}) \cdot \Pr(c_{k-c+1} \ldots c_{k-1} c_k | t_{i_2})} + \left(\prod_{0<q<k} (1-\sigma_q) \right) \cdot \frac{p_i \cdot \Pr(c_1 \ldots c_{k-1} c_k | t_i)}{\sum_{i_2=1}^{N} p_{i_2} \cdot \Pr(c_1 \ldots c_{k-1} c_k | t_{i_2})} \quad \text{if } 1 \le k \le C; \tag{22}$$

and

$$\lambda_i^k = \sum_{c=1}^{C} \sigma_{k-c} \cdot \left(\prod_{k-c<q<k} (1-\sigma_q) \right) \cdot \frac{\sum_{i_1=1}^{N} (\lambda_{i_1}^{k-c} p_{i_1 \to i}) \cdot \Pr(c_{k-c+1} \ldots c_{k-1} c_k | t_i)}{\sum_{i_2=1}^{N} \sum_{i_1=1}^{N} (\lambda_{i_1}^{k-c} p_{i_1 \to i_2}) \cdot \Pr(c_{k-c+1} \ldots c_{k-1} c_k | t_{i_2})} \quad \text{if } k > C. \tag{23}$$

The term count freq_i^n $(i = 1, 2, \ldots, N)$ are obtained by calculating freq_i^k's and λ_i^k's recursively for k from 0 to n using Eqs. (15)–(22).

4.4 Estimating word segmentation probability

Word segmentation is defined as the process of segmenting a line into words. In handwritten lines, the space between words is uneven. Moreover, the same amount of space may be present between words and between characters within a word. Such cases arise due to differences in writing styles and space constraints.

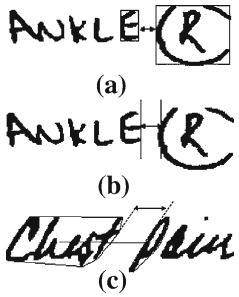

Fig. 3 Three feature representing a gap between two consecutive connected components. **a** Euclidean distance. **b** Run length distance. **c** Convex hull distance

In our word segmentation method, for every gap between any two consecutive connected components, the probability of that gap being a between-word gap is estimated. A gap between two connected components is represented by three features:

1. **Euclidean Distance.** This feature is defined as the horizontal distance between the bounding boxes of the two consecutive connected components of the line image (Fig. 3a).
2. **Minimum Run Length.** This feature represents the minimum horizontal white run length distance between the two adjacent connected components of the line image. There is a little difference between the run length and Euclidean distances. Run length is only affected by rows that are common to both the connected components. However, Euclidean distance between bounding boxes can also be affected by a particular row unique to one component (which changes the shape of the bounding box of the component).
3. **Convex Hull Distance.** The Euclidean distance between points at which this line crosses the convex hulls of two adjacent components is defined as the Convex Hull distance of the two components.

To eliminate the effect of different text sizes, we compute the average height of all the components and normalize the extracted features by dividing them by the average height of all components in the same line.

The segmentation probability of a gap g is given by the Bayes' Rule

$$\sigma_g = \Pr(g|f) = \frac{\Pr(g)p(f|g)}{\Pr(g)p(f|g) + \Pr(\bar{g})p(f|\bar{g})} \tag{24}$$

where $\Pr(g)$ and $\Pr(\bar{g})$ are the prior probabilities of between-word gaps and within-word gaps, respectively. f represents three features of g. $p(f|g)$ is the probability density of the features of between-word gaps. $p(f|\bar{g})$ is the probability density of the features of within-word gaps. Given a set

of gap features with the annotation of *"between-word"* and *"within-word"*, we can estimate $\Pr(g)$, $\Pr(\bar{g})$, $p(f|g)$ and $p(f|\bar{g})$ as follows. $\Pr(g)$ and $\Pr(\bar{g})$ are estimated from the ratio of the numbers of between-word and within-word gaps in the training set.

$$\Pr(g) = \frac{\#\{\text{between-word gaps}\}}{\#\{\text{between-word gaps}\} + \#\{\text{within-word gaps}\}} \tag{25}$$

$$\Pr(\bar{g}) = 1 - \Pr(g) \tag{26}$$

$p(f|g)$ and $p(f|\bar{g})$ are estimated non-parametrically using Parzen window technique with a Gaussian kernel function.

4.5 Estimating word recognition likelihood

We use a lexicon-driven word recognition algorithm [15] based on character segmentation and dynamic programming to find the best matching path. First, a word image is segmented into candidate character images. Then the directional features are extracted from the contours of character images and matched to every word in the lexicon by searching all possible segmentations for the minimum sum of Euclidean distances from the features of the test image and the character templates in the training set. The minimum Euclidean distance indicates the similarity between the word image and the term in the lexicon. The square of the distance is associated with a pair of a word image w, and a term t_i is denoted by $s(w, t_i)$. We can use the Bayes' rule to verify, if t_i is a genuine match of w:

$$\Pr_{t_i}(G|s(w,t_i)) = \frac{\Pr_{t_i}(G)p(s(w,t_i)|G)}{\Pr_{t_i}(G)p(s(w,t_i)|G) + \Pr_{t_i}(I)p(s(w,t_i)|I)} \tag{27}$$

where $p(s(w, t_i)|G)$ is the likelihood of the genuine matching score, $p(s(w, t_i)|I)$ is the likelihood of the imposter matching score, $\Pr_{t_i}(G)$, and $\Pr_{t_i}(I)$ are the prior probabilities of genuine and imposter matches, respectively. For simplicity, we assume those distributions are invariant to different term t_i. Thus,

$$\begin{aligned}\Pr_{t_i}(G|s(w,t_i)) &= \frac{\Pr(G)p(s(w,t_i)|G)}{\Pr(G)p(s(w,t_i)|G) + \Pr(I)p(s(w,t_i)|I)} \\ &= g(s(w,t_i))\end{aligned} \tag{28}$$

can be denoted by a function g of $s(w, t_i)$.

$\Pr(G)$, $\Pr(I)$, $p(s|G)$ and $p(s|I)$ are estimated from the scores of all of the terms. We model $p(s|G)$ and $p(s|I)$ as Gamma distributions. Actually, the matching score s is a squared sum of distances between character-level feature vectors and the centers of clusters in the training features. In other words,

$$s = \sum_{l=1}^{L} D_l^2 \tag{29}$$

where D_l is a character matching distance. If we assume all the clusters of the training feature vector space are independent normal distributions, then the squared sum of the distances can be modeled as a gamma distribution. The probability density function of the gamma distribution can be represented by

$$f_S(s;k,\theta) = s^{k-1}\frac{\mathrm{e}^{-s/\theta}}{\theta^k \Gamma(k)}, \quad s > 0 \text{ and } k, \theta > 0 \tag{30}$$

where $\Gamma(k)$ is the gamma function:

$$\Gamma(k) = \int_0^\infty x^{k-1}\mathrm{e}^{-x}\mathrm{d}x. \tag{31}$$

If k is a positive integer, then $\Gamma(k) = (k-1)!$. There is no closed-form solution for the maximum likelihood estimation of k and θ [9]. However, we can use a simple way to estimate the Gamma distribution. First, we can prove that the mean and variance of the Gamma distribution are $k \cdot \theta$ and $k \cdot \theta^2$, respectively. Then, given N genuine matching scores $s_1, s_2, \ldots s_N$, we can compute the ML estimation of mean and variance:

$$\begin{cases} \bar{\mu} = \frac{1}{N}\sum_{i=1}^{N} s_i \\ \bar{\sigma}^2 = \frac{1}{N}\sum_{i=1}^{N}(s_i - \bar{\mu})^2 \end{cases} \tag{32}$$

Let $\bar{k} \cdot \bar{\theta} = \bar{\mu}$ and $\bar{k} \cdot \bar{\theta}^2 = \bar{\sigma}^2$, then

$$\begin{cases} \bar{k} = \frac{\bar{\mu}^2}{\bar{\theta}^2} \\ \bar{\theta} = \frac{\bar{\theta}^2}{\bar{\mu}} \end{cases} \tag{33}$$

A Genuine probability/score curve estimated from 5461 genuine matching scores and 1,226,022 imposter matching scores is shown in Fig. 4.

By Bayes' rule, the likelihood

$$\Pr(t_i|w) = \frac{\Pr(t_i)\cdot\Pr(w|t_i)}{\sum_{j=1}^{N}\Pr(t_j)\Pr(w|t_j)} \tag{34}$$

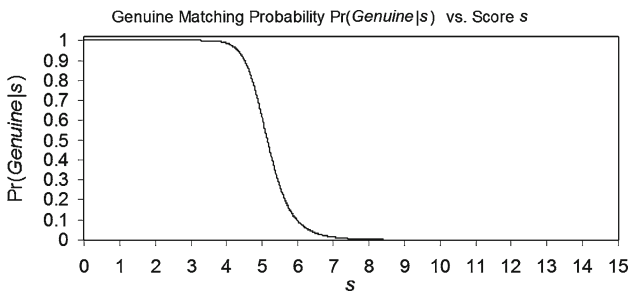

Fig. 4 Genuine matching probability/score curve estimated from training set

On the other hand, we approximately compute the posterior probabilities using

$$\Pr(t_i|w) = \frac{\Pr(t_i)g(s(w,t_i))}{\sum_{j=1}^{N}\Pr(t_j)g(s(w,t_j))}, \quad i = 1, 2, \ldots, N \tag{35}$$

From Eqs. (34) and (35),

$$\begin{aligned}\Pr(w|t_i) &= \frac{\sum_{j=1}^{N}\Pr(t_j)\Pr(w|t_j)}{\Pr(t_i)} \cdot \Pr(t_i|w) \\ &= g(s(w,t_i)) \cdot \frac{\sum_{j=1}^{N}\Pr(t_j)\Pr(w|t_j)}{\sum_{j=1}^{N}\Pr(t_j)g(s(w,t_j))} \\ &\propto g(s(w,t_i))\end{aligned} \tag{36}$$

Thus, we can replace the likelihood $\Pr(c_{k-c+1}\ldots c_k|t_i)$ with $g(s(c_{k-c+1}\ldots c_k, t_i))$ in Eqs. (15–22).

4.6 Search engine based on modified vector model

A search engine for handwritten document is built using the modified vector model and term count estimation method discussed in the previous subsections. The flowchart of the search engine (Fig. 5) shows three phases of the system: preprocessing, indexing, and document retrieval.

In the preprocessing phase, image enhancement such as noise filtering and binarization are performed, and text lines are identified by page segmentation.

Indexing includes word segmentation and recognition with the estimation of probabilities. We use these probabilities to estimate the term frequency (TF) and inverse document frequency (IDF) and store the estimated TF and IDF values for retrieval.

When searching the database for relevant documents, the user input query is converted to a query vector and the similarity of the vector model is calculated for each document. Documents are ranked in the decreasing order of similarity, and top documents are returned.

4.7 Computational issues

Only the non-zero values of the TF matrix are needed to be stored in the index, and thus, the space to store the index and time complexity of retrieval are both linear in the number of non-zero values in the TF matrix. Since the TF matrix for text retrieval is usually sparse, the size of index file and the retrieval speed are not issues. But the TF matrix is no longer sparse when indexing document images (using the proposed method). Practically, we can convert the TF matrix into a sparse one without affect performance much: we can choose a threshold $\text{THR}_{\text{sparse}}$ and turn those elements from the TF matrix that are less or equal to $\text{THR}_{\text{sparse}}$ (see circled elements in Fig. 6b). We set $\text{THR}_{\text{sparse}}$ to 0.002 in our experiments.

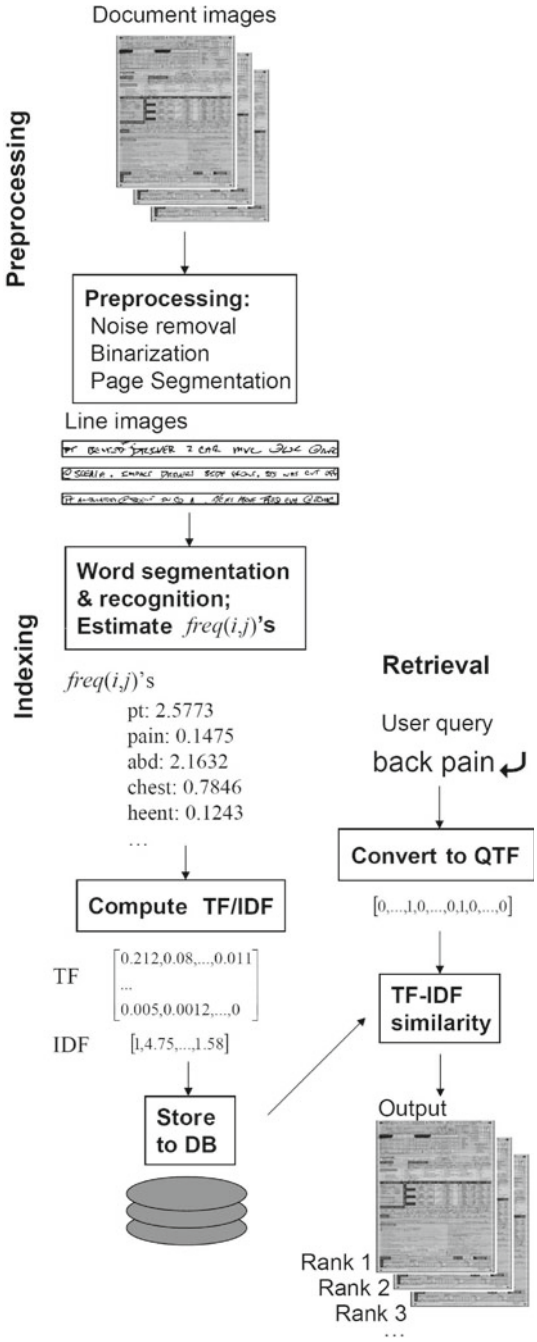

Fig. 5 Flowchart of the search engine

4.8 Test corpus

Our test corpus consists of the New York State Pre-hospital Care Reports (PCR forms). In New York State, all patients who enter the emergency medical system (EMS) are tracked through their pre-hospital care to the emergency room using the PCR. The PCR is used to gather vital patient information. Retrieval on this data set is quite challenging for several reasons: (i) handwritten responses are very loosely constrained in terms of writing style, format of response,

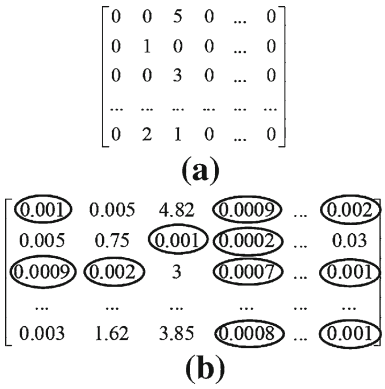

Fig. 6 TF matrices from text IR and document image IR. The TF matrix for document image IR can be approximated by a sparse matrix if we turn the circled elements that are below a threshold to 0. **a** The TF matrix from a text IR application. **b** The TF matrix from a document image IR application

and choice of text due to irrepressible emergency situations, (ii) images are scanned from noisy carbon copies and color background leads to low contrast and low signal-to-noise ratio (Fig. 7), (iii) medical lexicons of words are very large (∼5,000 entries). This leads to difficulties in the automatic transcription of forms. The word recognition rate of the forms using Word Model Recognizer (WMR) [15] is below 30%. Each PCR contains only about 100 handwritten words on average, so the content is very short and ordinary IR methods perform badly, since some of the terms are often absent from the OCR result.

4.9 Preprocessing and recognition of PCR form images

First, we detect and remove the skew of every PCR form image as follows.

1. We manually de-skew a form and take it as a template. Two regions with machine-printed text are cut from the template image as anchors.
2. We use the anchors to perform registration between the template and other test images, since we know in advance the anchors appear in all the images. The positions of two anchoring regions in any test image are located using cross-correlation.
3. The skew angle of the test image is obtained by the relative skewing between the test image and the template. We de-skew the image by rotating the test image to the opposite direction.

By aligning the test image to the template image, we can also obtain the position of each form cell containing a line of text. The template-matching-based de-skewing and page segmentation work well on the PCR form images, since they have a fixed layout and are scanned at the same resolution. Our approach is applicable to other types of forms as well.

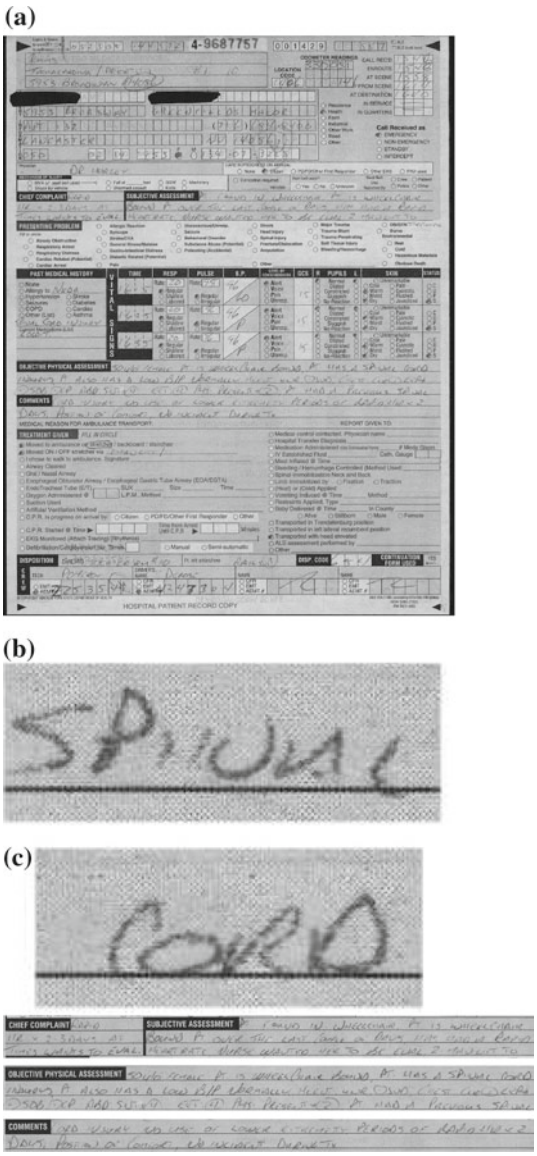

Fig. 7 A sample PCR form. **a** The entire PCR form. **b** A small local region showing blurred text and background noise. **c** Fields of interest in the PCR form

We use the MRF-based document image preprocessing algorithm [6] to binarize the form image and remove the grid lines from the image. Assuming the binarized objective image is x and the grayscale image is y, we solve the maximum a posteriori (MAP) estimation $\hat{x} = \underset{x}{\text{argmax}} \Pr(x|y)$ using the Markov Random Fields (MRF). An example of binarization and line removal result is shown in Fig. 8. The MRF-based preprocessing method improves the word recognition accuracy from 18.7% (obtained by the PCR form preprocessing algorithm in [17]) to 28.6%.

We use 1,099 between-word gaps and 5,138 within-word gaps to train the word gap classifier using the method presented in Sect. 4.4. The classifier is evaluated on a test set of 791 between-word gaps and 4,369 within-word gaps.

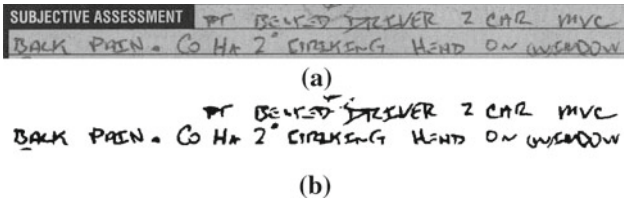

Fig. 8 An example of the binarization and line removal result. **a** The original grayscale image. **b** The bnarized image. Grid lines are removed and broken strokes are fixed

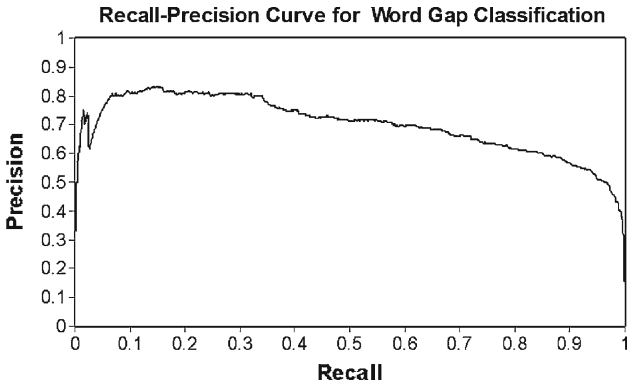

Fig. 9 The performance of word segmentation (recall-precision curve)

If we take probability p_{thr} as a threshold to determine the category of a gap, we can compute the recall and precision values obtained from the given test deck. Thus, a precision-recall curve (Fig. 9) is obtained by taking various values of threshold, p_{thr}.

The WMR handwritten word recognizer is trained using 21,054 character images collected from the handwriting on the letters provided by the US Postal Service [15]. A lexicon of 4,551 English words is generated from the ground truth of 783 PCR forms.

A bi-gram LM is trained from the above 783 forms. A word recognition rate of 28.6% is obtained on the PCR forms.

4.10 Evaluation metrics of IR test

The IR tests are evaluated in terms of mean average precision (MAP) and R-Precision [1]. The mean average precision is obtained in the following way:

1. For each query, check the returned documents starting from rank 1. Whenever a relevant document is found, record the precision of the documents from the one with rank 1 to the current one. The average value of the recorded precisions for the query is the average precision of the query.
2. The mean value of the average precisions of all the queries is the mean average precision of the test.

R-Precision of a query is the mean value of precisions computed for each query when R documents are retrieved, where R is the number of relevant documents. The mean value of the R-Precisions of all queries is the R-Precision of all of the queries. For example, suppose 100 documents are relevant to query q_1, and 30 of the top 100 retrieved documents are relevant to the query, then the R-Precision of query q_1 is $30/100 = 30\%$. Suppose the R-Precision of another query q_2 is 20%, then the R-Precision of q_1 and q_2 is $(30 + 20\%)/2 = 25\%$.

In addition to the mean average precision and R-Precision, the performance of the IR system can also be visualized using a 11-point precision. First, the 11 interpolated precisions at recalls 0, 0.1, …, 1 are calculated for each query. Then, the average precision of all of the queries at each of the 11 recalls is calculated. Finally, we get 11 precisions.

4.11 IR tests

The document images used in our IR tests are 342 PCR forms with manually transcribed ground truth and coordinates of each word. We have 28 queries and manual annotation of relevance of the 342 forms to these queries. These 342 PCR forms are different from the 791 forms used in the training of the word recognizer and LM. The queries used in our IR tests are shown in Table 1.

We compare the performances of the following 7 IR tests:

Tests 1–4: IR tests on OCR'ed text
We apply the classic vector model on OCR'ed text. First, we apply word segmentation to the 342 form images as follows. For any m ($m \leq 16$) consecutive connected components $c_q c_{q+1} \ldots c_{q+m}$, suppose $\sigma_{q-1}, \sigma_q, \ldots,$ and σ_{q+m} are gap category probabilities obtained by the gap classification algorithm presented in Sect. 4.4, then the probability of the

Table 1 28 query phrases used in our IR tests

"Head pain"	"Emesis"	"Breath difficulty short"
"Trachea"	"Lung"	"Chest pain"
"Fracture"	"Rib fracture"	"Head fracture"
"Ankle fracture"	"cancer"	"Trauma"
"Glucose"	"Diabetes"	"Foot"
"Tender"	"Hurts"	"Ambulate"
"Cardiac"	"Dizzy dizzyness dizziness"	"Cardiac monitor"
"Wrist"	"Arthritis"	"Shoulder pain"
"Syncope"	"Mri"	"Blind"
"Dementia"		

concatenation $c_q c_{q+1} \ldots c_{q+m}$ being a word image is $\sigma_{q-1} \cdot (1-\sigma_q) \cdot \ldots \cdot (1-\sigma_{q+m-1}) \cdot \sigma_{q+m}$.

We recognize all the word images with the word segmentation probability above 0.3. The OCR'ed text is composed of the top-S word recognition candidates of every word image. The parameter S = 1, 3, 7, and 15 in four separate tests. IR tests based on the classic VM are performed on the OCR'ed text of 342 form images.

Test 5: UMASS TF (vector model)
We apply the modified vector model to 342 form images for document retrieval. The term counts are estimated from handwriting recognition (HR) results using Eq. (12) by assuming perfect word segmentation and identical independent distribution (i.i.d.) of terms, i.e.,

$$E\{\text{freq}_{i,j}\} = \sum_{k=1}^{L} \Pr(\tau_k | w_k) \tag{37}$$

We use the same word segmentation method in Test 1–4.

Test 6: UMASS TF (probabilistic model)
We apply the probabilistic IR model [13,22] to 342 form images for document retrieval. In this model, the doc-query similarity is defined as

$$\text{sim}(d_j, q) = \prod_{1 \le i \le N,\ tf_{i,q}=1} tf_{i,j}, \tag{38}$$

and the term count is estimated by Eq. (37). We use the same word segmentation method in Test 1–4. The difference between [13,22] and our implementation is the way word recognition probabilities $\Pr(\tau_k | w_k)$ are estimated.

Test 7: Proposed TF (vector model)
We apply the modified vector model to 342 form images for document retrieval. The term counts are obtained using Eqs. (15)–(22).

The MAP and R-Precision values of the above IR tests are compared in Fig. 10a. A trivial average precision of 4.76% is obtained by generating random retrieval results for the 28 queries. We amend the metrics by subtracting the trivial AP from the MAP and R-Precision values. The amended metrics show the incremental improvement from the trivial result. The amended MAP and R-Precision values of the above IR tests are compared in Fig. 10b. Tests 1–4 show that the improvement of using more word recognition candidates (S = 3, 7, and 15) compared to the result of IR test on top-1 word recognition text is very slight. Even a naive estimation

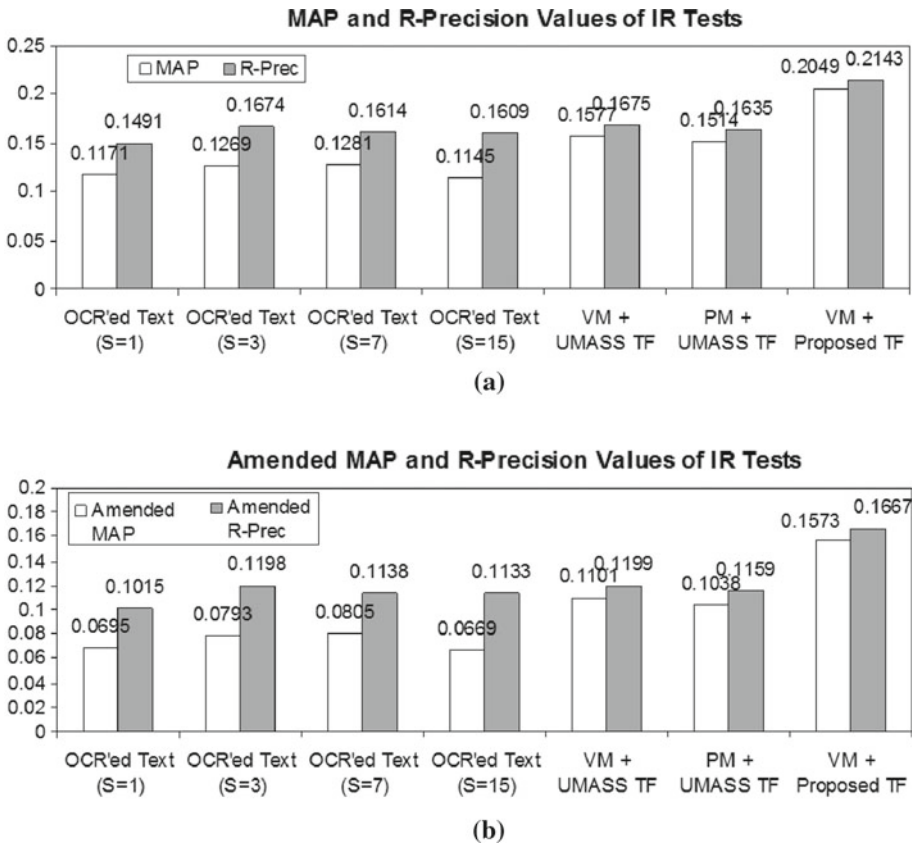

Fig. 10 The MAP and R-Precision values of 7 IR tests. **a** Original MAP's and R-Precisions. **b** Amended MAP's and R-Precisions

of the term counts (Eq. 37) improves the IR performance compared to the tests based on OCR'ed text. But the use of the word segmentation probabilities and the language model (Test 7) resulted in better IR performance than the estimation method that only uses isolated word recognition results.

The interpolated 11-point precision curves of tests 1 (OCR'ed text, $S = 1$), 5 (VM + isolated word estimation) and 7 (VM + word sequence estimation) are shown in Fig. 11a. The IR performance of building the index on the ground truth text is also shown in Fig. 11a. Tests 5 and 7 produce similar precisions at low recall (around 0), but Test 7 produces significantly higher precisions at higher recalls.

For better comparison, the above 11-point precision curves can also be amended this way: we first get two addition precisions at each recall level: trivial precision and ground-truth precision, and then normalize the recall-precision coordinates so that the trivial precision is always 0 and the ground-truth precision is always 1. The trivial precision is defined as the precision obtained by ranking all the documents randomly:

$$\text{Prec}_{\text{trivial}} = \frac{\text{average number of relevant documents per query}}{\text{number of documents}} \tag{39}$$

The ground-truth precision $\text{Prec}_{\text{truth}}$ at a recall level is the precision obtained by IR test performed on the index built on ground-truth text. The amended precision of an original precision p is defined as

$$\text{Prec}_{\text{amended}} = \frac{p - \text{Prec}_{\text{trivial}}}{p - \text{Prec}_{\text{truth}}} \times 100\% \tag{40}$$

The amended 11-point precision curves in Fig. 11b show that the proposed method obtained improvement at almost all recall levels but especially improved the precisions at high recall rates (>50%). The two existing methods perform very poorly at high recall levels by giving nearly zero precisions. But the proposed method still obtained about 10% precision at the recall level of 100%.

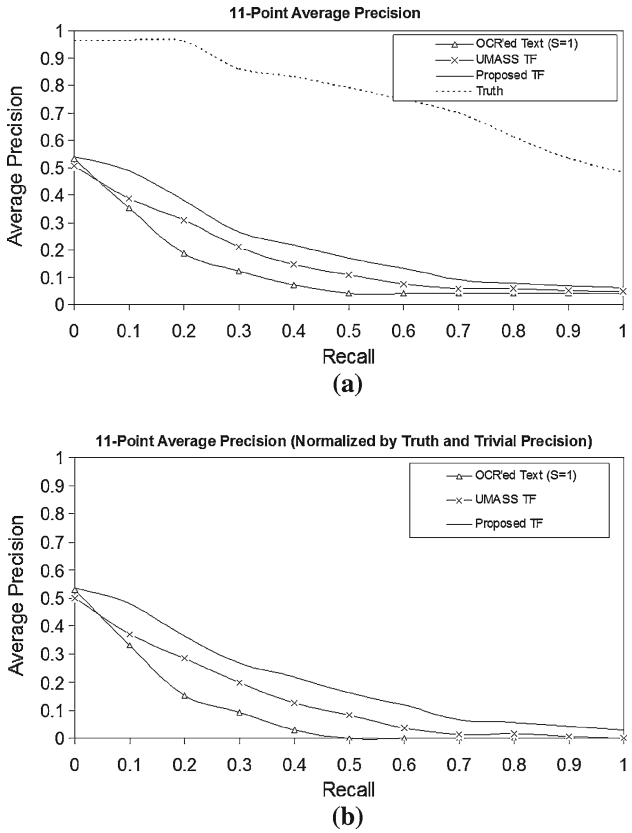

Fig. 11 The 11-point average precision curves of tests 1, 5 and 7. **a** Original recall-precision curves. **b** Amended recall-precision curves

5 Word spotting-based IR

The notion of word spotting [22] has been introduced as an alternative to OCR-based information retrieval solutions. It can be defined as an information retrieval task that finds all occurrences of a typed query word in a set of handwritten or machine-printed documents. This section presents some of our keyword spotting approaches for handwritten medical forms as well as multilingual documents.

5.1 Probabilistic word spotting model

Huaigu et al. [8] describe a keyword spotting model that provides an improved retrieval performance by combining the word recognition likelihood and word segmentation probability in a probabilistic framework. Given a series of consecutive connected components $c_1, c_2, \ldots, c_n$ and a possible word image w represented by $c_i, c_{i+1}, \ldots c_j$ $(1 \le i, j \le n)$, their model represents the similarity between w and a query word q by:

$$\begin{aligned}&\text{sim}(w, q)\\ &= \sigma_{i-1} \cdot (1 - \sigma_i) \cdot \ldots \cdot (1 - \sigma_{j-1}) \cdot \sigma_j \cdot \Pr(q|w)\end{aligned} \tag{41}$$

where σ_k $(1 \le k \le n - 1)$ is the probability of the gap between c_{k-1} and c_k being a between-word gap, $\sigma_0 = \sigma_n = 1$, and $\Pr(q|w)$ is the word recognition probability. The gaps are assumed to be independent, and therefore, the word segmentation probability can also be represented as $\sigma_{i-1} \cdot (1 - \sigma_i) \cdot \ldots \cdot (1 - \sigma_{j-1}) \cdot \sigma_j$.

Figure 12 shows the average precision curve of the proposed method that outperforms traditional keyword spotting approaches assuming perfect word segmentation.

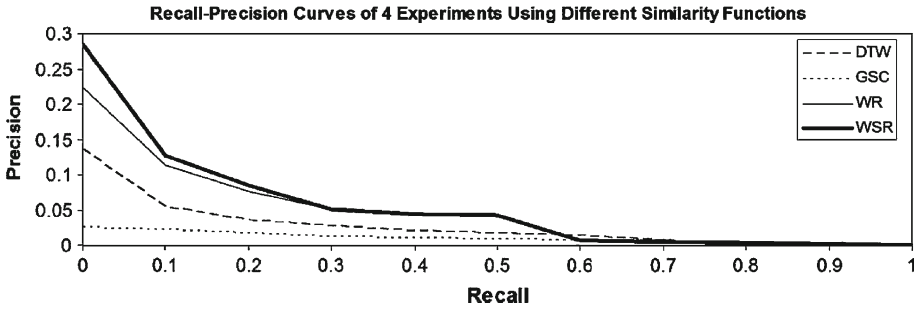

Fig. 12 11-point average precision curves of Tests 1–2 [8]

5.2 Feature-based word spotting model

Bhardwaj et al. [4] propose an image feature-based keyword spotting solution for multilingual documents. They describe an indexing process that extracts moment features from input word images and stores the feature values as indexes. The features are computed using a geometrical moment equation that is invariant under image translation and scale transformations.

$$m_{pq} = \frac{\sum_X \sum_Y (x^*)^p (y^*)^q f(x, y)}{M_{00}} \tag{42}$$

For the retrieval process, they represent all the query words and candidate words as traditional vector space model. Cosine similarity is used to compute similarity between the query images and indexed images on moment feature space. Finally, all the candidate word images are ranked in order of their similarity with the query image. Since the similarity values are computed on feature space, it's not robust to larger image variation and lower image quality. To address this issue, they use relevance feedback mechanism to re-rank all the candidate word images. This mechanism re-formulates the query feature vector by adjusting the values of the individual moment orders present in the query vector. The relevance feedback mechanism assumes a user input after the presentation of the initial results. A user enters either a 1 denoting a result to be relevant or 0 denoting a result to be irrelevant. The new query vector is computed as follows:

$$q_{new} = \gamma . q_{old} + \frac{\alpha}{|R|} . \sum_{i=1}^{i=R} d_i - \frac{\beta}{|NR|} . \sum_{j=1}^{j=NR} d_j \tag{43}$$

where α , β, and γ are term re-weighting constants. R denotes a relevant result set, and NR denotes a non-relevant result set.

Table 2 describes their results on 3 different scripts before and after applying relevance feedback (RF).

Table 2 Average precision rate for word spotting in all 3 Scripts [4]

Script	Before RF	After RF
English	66.30	69.20
Hindi	71.18	74.34
Sanskrit	87.88	92.33

6 Conclusion

Information retrieval from handwritten documents is a challenging task primarily due to lower word recognition rates in the case of unconstrained handwritten documents when compared to machine-printed document images. Traditional information retrieval techniques therefore fail to perform efficiently in case of noisy OCR'ed text. In this paper, we presented some of our existing methods that deal with retrieval from noisy OCR'ed text. We discussed three approaches that address this issue in different ways. First approach refines the OCR output and then performs retrieval over the cleaned text. The second approach uses the uncorrected OCR'ed text, but modifies the traditional retrieval model to account for OCR errors. The third approach uses image-processing techniques to compute similarity between query and word images and retrieves them accordingly. Each of the discussed approaches have their own merits or pitfalls and have been applied to different applications. Our future work will focus on exploring techniques leading to higher IR performance on handwritten documents including combining the benefit of all OCR correction methods with modified retrieval, integrating the stemming technique into the language model used in OCR, as well as broader applications such as the detection of out-of-vocabulary (OOV) items—name identities and so on—in noisy text.

References

1. Baeza-Yates, R.A., Ribeiro-Neto, B.A.: Modern Information Retrieval. ACM Press/Addison-Wesley (1999)
2. Beitzel, S.M., Jensen, E.C., Grossman D.A.: A survey of retrieval strategies for OCR text collections. In: Proceedings of the Symposium on Document Image Understanding Technologies. Greenbelt, Maryland, April 2003
3. Bhardwaj, A., Farooq, F., Cao, H., Govindaraju, V.: Topic based language models for OCR correction. In: Proceedings of the second workshop on Analytics for noisy unstructured text data, pp. 107–112, Singapore (2008)

4. Bhardwaj, A., Jose, D., Govindaraju, V.: Script independent keyword spotting for multilingual documents. In: Cross Lingual Information Access Workshop (2008)
5. Cao, H., Govindaraju, V.: Template-free word spotting in low-quality manuscripts. International Conference on Advances in Pattern Recognition (2007)
6. Cao, H., Govindaraju, V.: Handwritten carbon form preprocessing based on Markov random field. In: Proceedings of the 2007 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'07) (2007)
7. Cao, H., Govindaraju, V.: Vector model based indexing and retrieval of handwritten medical forms. In: Proceedings of Ninth Internation Conference on Document Analysis and Recognition (ICDAR) **1**, 88–92 (2007)
8. Cao, H., Bhardwaj, A., Govindaraju, V.: A probabilistic method for keyword retrieval in handwritten document images. In: J. Pattern Recognit. **42**(12), Elsevier Press (2009)
9. Choi, S.C., Wette, R.: Maximum likelihood estimation of the parameters of the gamma distribution and their bias. Technometrics **11**(4), 683–690 (1969)
10. Choisy, C.: Dynamic handwritten keyword spotting based on the NSHP-HMM. In: International Conference on Document Analysis and Recognition, pp. 242–246. ICDAR (2007)
11. Croft, W.B., Harding, S.M., Taghva, K., Borsack, J.: An evaluation of information retrieval accuracy with simulated OCR output. In: Proceedings of the Symposium on Document Analysis and Information Retrieval (1994)
12. Frinken, V., Fischer, A., Bunke, H.: Combining neural networks to improve performance of handwritten keyword spotting. Mult. Classif. Syst. **5997**, 215–224 (2010)
13. Howe, N.R., Rath, T.M., Manmatha, R.: Boosted decision trees for word recognition in handwritten document retrieval. In: Proceedings of the SIGIR, pp. 377–383 (2005)
14. Jing, H.: Using hidden Markov modeling to decompose human-written summaries. Comput. Linguis. **28**(4), 527–543 (2002)
15. Kim, G., Govindaraju, V.: A lexicon driven approach to handwritten word recognition for real-time applications. IEEE Trans. Pattern Anal. Mach. Intell. **19**(4), 366–379 (1997)
16. Lee, D.-R., Kim, W.-Y., Oh, I.-S.: Hangul document image retrieval system using rank-based recognition. In: Proceedings of the International Conference on Document Analysis and Recognition **2**, 615–619 (2005)
17. Milewski, R., Govindaraju, V., Bhardwaj, A.: Automatic recognition of handwritten medical forms for search engines. Int. J Doc. Anal. Recognit. **11**(4), 203–218 (2009)
18. Manmatha R., Han C., Riseman, E.M.: Word spotting: a new approach to indexing handwriting. Computer Vision and Pattern Recognit, p. 631. CVPR (1996)
19. Mittendorf, E., Schauble, P., Sheridan, P.: Applying probabilistic term weighting to OCR text in the case of a large alphabetic library catalogue. In: Research and Development in Information Retrieval, pp. 328–335 (1995)
20. Ohta, M., Takasu, A., Adachi, J.: Retrieval methods for English text with misrecognized OCR characters. In: Proceedings of the International Conference on Document Analysis and Recognition (1997)
21. Perronnin, F., Rodriguez-Serrano, J.A.: Fisher kernels for handwritten word-spotting. International Conference on Document Analysis and Recognition, pp. 106–110. ICDAR (2009)
22. Rath, T.M., Manmatha, R., Lavrenko, V.: A search engine for historical manuscript images. In: Proceedings of the 27th annual international ACM SIGIR conference on research and development in information retrieval (2004)
23. Reynolds, D.A., Quatieri, T.F., Dunn, R.B.: Speaker verification using adapted gaussian mixture models. Digit. Signal Process. **10**(1–3), 19–41 (2000)
24. Rodriguez-Serrano, J.A., Perronnin, F.: Local gradient histogram features for word spotting in unconstrained handwritten documents. In: International Conference on Frontiers in Handwriting Recognition. ICFHR (2008)
25. Rodriguez-Serrano, J.A., Perronnin, F.: Handwritten word-spotting using hidden Markov models and universal vocabularies. Pattern Recognit. **42**(9), 2106–2116 (2009)
26. Terasawa, K., Nagasaki, T., Kawashima, T.: Automatic keyword extraction from historical document images. Doc. Anal. Syst. VII **3872**, 413–424 (2006)
27. van der Zant, T., Schomaker, L., Haak, K.: Handwritten word-spotting using biologically inspired features. IEEE Trans. Pattern Anal. Mach. Intell. **30**(11), 1945–1957 (2008)