# A probabilistic method for keyword retrieval in handwritten document images

## ABSTRACT

Keyword retrieval in handwritten document images is a challenging task because handwriting recognition does not perform adequately to produce the transcriptions, especially when using large lexicons. Existing methods build indices using OCR distances or image features for the purpose of retrieval. These alternative methods are complimentary to the traditional approaches that build indices on OCR'ed text. In this paper, we describe an improvement to the existing keyword retrieval (word spotting) methods by modeling imperfect word segmentation as probabilities and integrating these probabilities into the word spotting algorithm. The scores returned by the word recognizer are also converted into probabilities and integrated into the probabilistic word spotting model.

## 1. Introduction

Keyword retrieval in handwritten document images is a high- level application that relies on document analysis and recognition techniques. There are two common approaches to keyword retrieval from handwritten documents. In the first approach [1–8], image-to- image matching is used. During retrieval, each keyword is converted into a word image. This is done by annotating a small set of word images or collecting the user's handwriting on-line. When a user pro- vides a query word, the similarity between the query and any word image in the database is computed. All of the word images are re- turned in the decreasing order of the similarities between them and the query. The similarity between two word images is measured as a distance between the two features vectors computed from the word images. In [1,3], the similarity between the feature vectors of two word images is computed by dynamic time warping (DTW) matching of profile features using various definitions of matching distances [1,9,10,3,11] in the feature space. The GSC-matching method [2,12] is based on bitwise matching of the corresponding GSC features of two word images. Thus, word spotting is a useful alternative when a full-fledged handwriting recognition system is not available.

However, word spotting requires on-line matching which is time-consuming. Trade-off between accuracy and speed has to be made in order to scale to large databases. Thus, in order to befast matching-based indexing approaches are limited in feature selection and the complexity of matching and training methods. This also limits their scope to applications dealing with a single writer or small lexicons. In contrast, OCR score-based indexing approaches [13–15] do not face the speed problem. In these methods, the indices are built from OCR scores such as posterior probabilities or feature vector observational likelihoods (probability density) obtained from distances re- turned by word recognizer. These methods [13–15] perform hand- writing recognition followed by an indexing step to keep track of the transcription and other useful information (positions and recognition scores of word images). The similarity between the keyword and another word image is computed using the recognition scores, which are usually the likelihood of the feature space, probabilities, or some other distance-based measurements. One question is whether to adopt a word lexicon. The index for fast retrieval can be built on the results of word level recognition in lexicon-driven mode [14,15]. In this mode, any word that is not in the lexicon cannot be retrieved. Ref. [13] performs recognition at the character level and searches for words in a series of character recognition scores. However, this approach is once again difficult and time-consuming which does not scale to larger data sets. We have taken a word-lexicon-driven method and get affected by the out-of-vocabulary (OOV) problem.

We have improved the OCR score-based indexing method by integrating word segmentation probabilities into the retrieval similarity metric. Word spotting methods this far has assumed perfect word segmentation: word images are given by word segmentation algorithm, and the ranks of word images are obtained by sorting the word recognition scores. However it is unrealistic to expect perfect word segmentation in unconstrained handwriting given the variation in the gap sizes between words. The performance ofword spotting can be improved by modeling the word segmentation probabilities. In this paper, we describe a probabilistic model of word spotting that integrates word segmentation probabilities and word recognition probabilities. The word segmentation probabilities are obtained by modeling the conditional distribution of multivariate distance features of word gaps. The word recognition results are also represented by a probabilistic model. The modeling of the word recognition probabilities is obtained from the distances returned by the word recognizer (Fig. 1).

## 2. Background in handwritten keyword retrieval

### *2.1. Image-to-image matching—word spotting*

Word spotting was initially proposed as an alternative approach for indexing and retrieving handwritten documents, that is one could search handwritten document images without using a handwriting recognizer. In order to search for a keyword, the user needs to write a copy of the keyword (a word template) and provide the word image as the query. One could also obtain the word templates by labeling a training set. The system executes the query by computing the distance between the query template and each word image in the document images.

*DTW-based keyword spotting*:

In the DTW-based method [1,3,11], the following preprocessing steps are commonly used.

1. Word segmentation is performed and the background of every word image is cleaned by removing irrelevant connected components from other words that reach into the word's bounding box.

2. Inter-word variations such as skew and slant angle are detected and eliminated.

3. The bounding box of any word image is cropped so that it tightly encloses the word.

4. The baseline of word images is normalized to a fixed position by padding extra rows to the images.

A normalized word image is represented by a multivariate time series composed of features from each column of the word image. These features include projection profile, upper/lower word profile, and number of background-to-foreground transitions.

1. Projection profile. The projection profile of a word image is com- posed of the sum of foreground pixels in each column.

2. Upper/lower profiles. The upper profile of a word image is made of the distances from the upper boundary to the nearest foreground pixels in each column.

3. Background-to-foreground transitions. The number of back- ground pixels whose right neighboring pixels are foregroundpixels is taken as the number of background-to-foreground transitions of the column.

The DTW-based method has been tested on GeorgeWashington's manuscripts (CIIR, University of Massachusetts [1,11]). The performance of keyword spotting was evaluated using the mean average precision (MAP) measure [16]:

1. For each query, check the returned word images starting from rank 1. Whenever a relevant word image is found, keep track of the precision of the word images from the one with rank 1 to the current one. The average value of the recorded precisions for the query is taken as the average precision (AP) of the query.

2. The mean value of the AP of all of the queries is the MAP of the test.

### *2.2. Keyword retrieval using word recognizers*:

Word spotting methods are useful when one does not have a handwriting recognizer. On the other hand, the word matching, which is essential to word spotting, can be thought of as a prototype of word recognizer, although its performance is considerably poorer than that of a well-trained word recognizer. But handwriting recognition remains very challenging task due to the wide variations in the handwriting. Thus matching against a single template is not a robust approach.

The advantage of word recognizer-based word retrieval over simple word matching was observed in our prior work [13] by comparing the performance of DTW-based word spotting method with the recognition-based keyword retrieval method.

## 3. Keyword retrieval, an important component of the search engine for off-line handwriting

### *3.1. A search engine for off-line handwriting*

A handwritten document retrieval system is presented in our prior work [19]. The goal of document retrieval is to search for “documents” that are relevant to the user query, as opposed to key- word retrieval that aims at searching for keywords. In document retrieval, we use standard indexing techniques such as TF-IDF to build indices from the documents. The major challenge in retrieving handwritten documents is the difficulty of computing the term frequency (TF) due to recognition errors. Our approach is to maintain an N-best list of the handwriting segmentation and recognition hypotheses, and estimate the TF using each result of the N-bestlist. The final TF is defined as a weighted sum of all the above TFs where the weights are the probabilities of validity of the segmentation and recognition hypothesis.

When we search for documents relevant to our query, usually we also want to get the positions of the query words and highlight them in documents, because we may only want to read upon the context around the query words. Text retrieval systems usually keep track of the positions of all the term in the indexing file. In our application, since the word segmentation is not perfect, we can only obtain hypotheses of word images. In addition to the positions, we also need to keep track of the similarities between word images and terms. The similarities can be defined and computed with very little effort given that the indexing of document retrieval has been done.

### *3.2. Word spotting using segmentation probabilities*

#### *3.2.1. Word spotting model*

Givenaseriesofconsecutiveconnectedcomponents and wordimage, the similarity between wordimageandaqueryword is defined.

#### *3.2.2. 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 un- even. Moreover, the space of the same size may be present between words, and between characters within a word. Such cases arise due to differences in writing styles, and the limited blank space left for writing. In our word segmentation method, the word segmentation probabilities are estimated from distance-based features. The gap betweenany two consecutive connected components is represented by three distance 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. 4(a)).
2. *Minimum run length*:This feature represents the minimum horizontal white run length distance between the two adjacent connected components of the line image.
3. *Convex hull distance*:We compute the convex hulls of two consecutive connected components and draw a line connecting the mass centers of the two convex hulls. The Euclidean distance

between points at which this line crosses the two convex hulls is defined as the convex hull distance of the two adjacent components.

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

### *3.2.3. Estimating word recognition probability*

In our system, the matching distance between a word image and a word is obtained by the word recognition algorithm of [21]. In this word recognition method, for any word image, all possible locations of the ligatures connecting two characters are identified by heuristic analysis of the concavity and convexity of the contour image. Then the word image can be divided into several pieces.By assuming that a character consists of at most four consecutive pieces, we can create a series of hypotheses of character images. Various features including the directions along the image contour are computed from each hypothesis of character image.

## 4. Experimentalresults

### *4.1. Preprocessing*

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 of pre-printed headlines are cropped from the template as anchors.

2. The positions of two anchoring regions in any test image are found by 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 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 de-skewing and page segmentation method using template-matching works 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. We use the Markov random fields (MRF)-based document image preprocessing algorithm [24] to binaries the form image and remove the grid lines from the image.

### *4.2. Evaluation metrics*

The performance of word spotting is evaluated using the precisions at 11 recall levels. We also use single value measures such as the MAP [16] to evaluate the word spotting performance.

## 5. Conclusion

In this paper we present a novel keyword retrieval method for the handwritten document images. Unlike the existing approaches using the image-to-image matching-based approaches, we use the word recognition distances to improve the word matching accuracy. We estimate the probabilities of word boundary segmentation using the distances between connected components and combine the segmentation and recognition distances to create a probabilistic word matching similarity.We show the improvement obtained by our approach by comparing the image-to-image matching approaches [11,12] with ours. Although the recognition-based approach shows the advantage over the image-to-image matching methods, we may notice that our method does not always have the highest MAP in every query. This suggests the future works can be done to improve the overall performance by combining multiple systems using different image features and similarity measurements. System combination may also effectively fix the intrinsic drawbacks of every single system. For example, we can use the recognition-based method to index the common words for higher performance, and use the image-to-image matching method to search for those OOV keywords.