Keyword Searching for Arabic Handwritten Documents

Raid Saabni[1,2] *Jihad El-Sana*[1,3]

[1]Department of Computer Science
Ben-Gurion University of the Negev
Beer-Sheva , Israel 84105
saabni,el-sana@cs.bgu.ac.il

[2]Triangle R&D Center
Kfar Qarea, Israel 30075

[3]Negev Research Center
Hura, Israel 85730

Abstract

In this paper we present a system for searching keywords in Arabic handwritten and historical documents using two algorithms, Dynamic Time Warping (DTW) and Hidden Markov Models (HMM). The HMM based system provides satisfying results when it is possible to provide adequate training samples (which is not always possible in historical documents). The DTW algorithm with a slight modification provides better results even with a small set of training samples. The observation sequences for the matching algorithms are generated by extracting a set of geometric features that already shown to obtain good recognition rates for on-line Arabic handwriting. We have adopted the segmentation-free approach, i.e., continuous word-parts are used as the basic alphabet, instead of the usual alphabet letters. The contours of the complete word-parts are used to represent the shapes of the compared word-parts. Additional strokes, such as dots and detached short segments, which are very common in Arabic scripts, are used via a rule-based system to improve the search algorithm and determine the final comparison decision. The search for a keyword is performed by the search for its word-parts, including the additional strokes, in the right order. The results for our modified DTW algorithm are very encouraging, even when using a small set of samples for training.

Keywords: Keyword Searching, Word Spotting, Handwriting Recognition, Dynamic Time Warping, Hidden Markov Models

1 Introduction

The advances in digital scanning and electronic storage have driven the digitization of historical documents for preservation and analysis of cultural heritage. This process enables important knowledge to be accessible to the wide public, while protecting historical documents from aging and deterioration by frequent handling. These documents are usually stored as a collection of images, which complicates searching through them for a specific word or phrase. To utilize the digital availability of these documents, it is essential to develop an indexing and searching mechanism. Currently, indexing is built manually and the search is performed on the scanned pages one by one. Since this procedure is expensive and very time consuming, an automation process is desirable. One may consider using off-line handwriting recognition to convert these document images into text files. However, The research on off-line handwritten script recognition has been limited to domains with small vocabularies, such as automatic mail sorting and check processing. Historical documents add another level of complexity resulting from lower quality sources due to various aging and degradation factors, such as faded ink, stained paper, dirt, and yellowing.

This paper deals with searching for a keyword in historical documents written in Arabic script, which is more complex due to cursiveness and similarity among letters. The results for off-line Arabic script recognizers are still very limited due to the lack of research in this field, compared to the Latin scripts. More than 40 million documents survived the last ten centuries and fortunately are preserved in different libraries around the word. For the processing of these documents, it is essential to develop efficient searching, indexing, and archiving for Arabic documents.

Keyword searching is designed to give users the ability to search for specific words in a given collection of document images automatically, without converting them into their ASCII equivalences. Spotting words aims to to cluster similar words within documents into different classes in order to generate indexes for efficient search.

In this work we have developed a keyword searching system for historical documents in Arabic. The features we use are extracted from the segment's angles and length of the word-part's simplified contour. We have experimented with two probabilistic classifiers – HMM and DTW, using the same set of extracted features. In addition, we have slightly modified the DTW algorithm to include different costs for substitution, insertion, and deletion of segments from the compared sequences. The same preprocessing techniques and similar feature sets were used for the two classifiers. The HMM based system requires many training samples of the keywords, which are generated manually from the processed documents. The DTW based system uses a simplified representation of the component's contour to constructed the feature vector, which is used to compare the word-parts.

In the rest of this paper we will first review closely related work, and then present our approach followed by experimental results. Finally, we draw some conclusions and suggest directions for future work.

2 Related Work

Keyword-searching algorithms provide the ability to automatically search through a collection of document images for a pictorial representation of a given word without converting them into their ASCII equivalences. Spotting words is a special kind of indexing on document images by clustering the results of a keyword-searching algorithm into different classes.

Word-matching algorithms roughly fall into two categories [7]: Pixel-based Matching and Feature-based Matching. Pixel-based matching approaches measure the similarity on the pixel domain between the two images using various metrics, such as Euclidean Distance Map (EDM), XOR difference, or the Sum of Square Differences (SSD) [9]. In Feature-based Matching, two images are compared using representative features, extracted from the images. Similarity measurements, such as Dynamic Time Warping and point correspondence are defined on the feature domain.

The Dynamic Time Warping(DTW) technique had been used and tested in many systems using various sets of features and shown to have better results than the competing techniques [7]. Manmatha *et al.* [7] examined several matching techniques and showed that DTW, in general, provides better results. Using a set of 2000 word images, they have reported an average match rate of 70%, which motivated them to develope algorithms to accelerate the computation of the DTW. Rath and Manmatha[12] preprocessed segmented word images to create sets of one-dimensional features, which were compared using DTW. Experimental results using different datasets from the George Washington collection have yield matching rates that range between 51.81% and 73.71%. They also analyzed a range of features suitable for matching words using DTW [11].

Rothfeder *et al.* [13] presented an algorithm which recovers correspondences of points-of-interest in two word images. These correspondences are used to measure the similarity between word images. They reported a correct matching rate of 62.57% and 15.49% using set of 2372 images of reasonable quality and a set of 3262 images of poor quality, respectively. Srihari *et al.* [15] presented a system for spotting words in scanned document images for three scripts, Devanagari, Arabic, and Latin. The system retrieved the candidate words from the documents and ranked them based on global word shape features. They reported better results for printed text compared to handwritten and showed that combining prototype selection and word matching yield better results for handwritten document. They obtained a correct match of 60% for handwritten English and 90% for printed Sanskrit documents.

Shrihari *et al.* [16] used global word shape features to measure the similarity between the spotted words and a set of prototypes from known writers. They reported results for manually segmented documents, using five writers to provide prototypes and another five for testing. They obtained 55% correct matching rate and commented that the match rate increases as more writers are used for training. In [14] they presented a design for a search engine for handwritten documents. They indexed documents using global image features, such as stroke width, slant, word gaps, as well as local features that describe the shapes of characters and words. Image indexing is done automatically using page analysis, page segmentation, line separation, word segmentation, and recognition of characters and words. Rath *et al.* [10] and [18] extract discrete feature vectors that describe word images, which are used to train the probabilistic classifier. They reported 89% correct matching rate for 4-word queries on a subset of George Washington's manuscripts.

A segmentation-free approach was adopted by Lavrenko *et al.* [6]. They used the upper word and projection profile features to spot word images without segmenting into individual characters. They showed that this approach is feasible even for noisy documents. Their experimental results show a recognition accuracy of 65%. Another segmentation-free approach for keyword search in historical documents was proposed by Gatos *et al.* [4]. Their system combines image preprocessing, synthetic data creation, word spotting, and user feedback technologies.

Manmatha and Rothfeder[8] describe a novel scale space algorithm for automatic segmentation of handwritten documents into words. They clean margins, segment lines and use anisotropic Laplacian at several scales to segment lines into words. They reported 17% incorrect matching on 100 handwritten documents from the George Washington corpus of handwritten document images.

An algorithm for robust machine recognition of keywords embedded in a poorly printed document was presented by Kuo and Agazzi [5]. For each keyword, two statistical models are generated – one represents the actual keyword and the other represents all irrelevant words. They adopted dynamic programming to enable elastic matching using the two models. They created a synthetic database that includes about 26000 words and reported 99% recognition rate for words that share the same font size and 96% for those that do not. Chen *et al.* [1] developed a font-independent system, which is based on HMM to spot user-specified keywords in a scanned image. The system extracted potential keywords from the image using a morphology-based preprocessor and then used external shape and internal structure of the words to produce feature vectors. Duong *et al.* [2] presented an approach that extracts regions of interest from gray scale images. The extracted regions are classified into textual and non-textual using geometric and texture features. Farooq *et al.* [3] present preprocessing techniques for Arabic handwritten document to overcome the ineffectiveness of conventional preprocessing for such documents. They described techniques for slant normalization, slope correction, and line and word separation for handwritten Arabic documents.

3 Our Approach

In this paper we present a novel keyword searching algorithm for handwritten Arabic Documents include historical Arabic manuscripts with reasonable quality. Our algorithm is based on geometric features, which can be used for any feature-based matching technique, such as DTW and HMM. We assume the input documents can be segmented into words and word-parts, which boundary contours are well defined. Next we will overview the various stages of this algorithm.

3.1 Component Labeling

Prior to component labeling procedure we horizontally align the base line of the rows of the input page. Such alignment is archived by first computing the page's vertical density histogram and then analyzing it's standard deviation to determine the optimum points. We then segment the page into lines and calculate the lower and upper base lines, which are used to extract the various components.

Since we are tracing the contour of each component independently, segmenting the line into words is not necessary. The extracted components are classified into *main* and *secondary* based on their size and location with respect to the base line. We use *main component* to denote the continuous body a word-part and *secondary component* to refer to an additional stroke. Each *secondary component* is associated with a *main component*. A main component with its secondary components represent an Arabic word-part, which will be denoted *Meta Component* (see Figure 1).

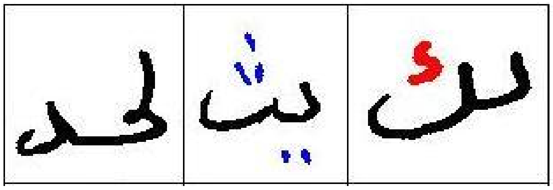

Figure 1. Meta Components with different numbers of additional strokes.

3.2 Simplification

The pixels on a component's contour form a 2D polygon. However, such a representation includes more than required vertices, which often complicate processing and handling these contours. Therefore, we simplify the contour polygon to work with a small number of representative vertices. In each iteration of the simplification process we remove the vertex with the smallest distance from the line passing through its two adjacent neighbors. The process terminates when an error threshold or a satisfying number of vertices is reached.

Since we are using two major classification approaches that rely on inherently different classification measures, we generate two simplified versions for each contour polygon. For the HMM classifier there is a need to control the number of fed vertices (points). Therefore, the simplified polygon is refined by adding k vertices from the original polygon, which are distributed nearly uniformly between each two consecutive vertices. The point sequence $P = [p_1, p_2, \cdots, p_n]$ includes all the vertices on the refined polygon. The size of P is determined based on the characters of the keyword and a predefined table that provides an estimation for the number of points required to describe each character.

The DTW requires nearly equal-length edges of the contour polygon, i.e., similar distances between consecutive vertices. Since the geodesic distance between the vertices of the simplified model could dramatically vary, we use the short edges and a predefined tolerance value to subdivide the long edges to satisfy the requirements of the DTW.

3.3 Feature Extraction

In machine learning, feature vectors are used to generate observation sequences. In this work we have adopted

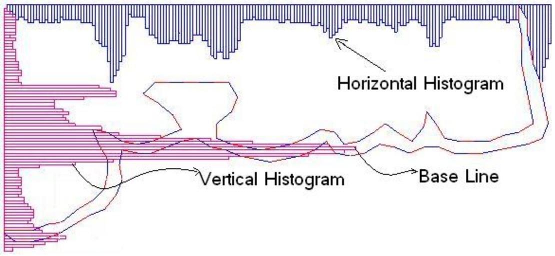

Figure 2. Horizontal and Vertical Density histograms on top of a simplified word-part.

a set of features which provide good recognition rates for the on-line Arabic handwriting [17]. In addition, we have developed several features that capture the special structure of the Arabic script. These features capture local, semi-global, and global behaviors.

- The angle α_i, which is the angle between the two vector (p_{i-1}, p_i) and (p_i, p_{i+1}).
- The length of the vector (p_i, p_{i+1}).
- The angle β_i, which is the angle between the vector (p_i, p_{i+1}) and (p_j, p_{j+1}), where p_j and p_{j+1} are consecutive vertices in the simplified polygon and the p_i vertex was inserted between them by the refining process.
- Loops Number: the number of loops found in the component.

In this work we have used different subsets of the mentioned features for the HMM-based and the DTW-based classifiers. Our HMM classifier have used the features α and β. The contour length and number of loops were ignored since they do not behave consistently in handwriting style. In the DTW classifier we have used α and $Length$ features.

4 Matching

Matching algorithms form the core of any search algorithm. Keyword search relies on a matching technique to determine the similarity between word images. In general, these matching techniques could be categorized into: pixel-based and feature-based approaches. The pixel based approaches compare pixels or blocks from the two images. The feature-based techniques extract a set of features from the two images and compare them. In this paper we use a feature-based technique as it provides flexible comparison, which is essential to handling varying handwriting styles.

In this research we avoid segmenting words into letters and consider the continuous word-part as the basic alphabet of the Arabic language. As a result, the search for a given keyword is performed by the search for its word-parts in the right order. For that reason, the basic matching procedure compares word-parts, i.e., computes the match between two word-parts. We have embedded our feature's set into two statistical matching schemes – Hidden Markov Models and Dynamic Time Warping.

4.1 Hidden Markov Models

The Hidden Markov Model (HMM) is a finite set of states, each is associated with a (generally multidimensional) probability distribution. Transitions among the states are governed by a set of probabilities called transition probabilities.

In this research we have manually extracted different occurrences of the word-parts of the keyword from the searched document. The basic postulation assumes that the extracted occurrences capture the different shapes of each word-parts according to document's writing style. The extracted word parts are used to produce feature vectors (as explained in Section 3.3), which are used to train the HMM system. The number of states is determined by the letters in each word-part of the keyword according to a predefined table. The search for a keyword is performed by searching for its word-parts, which are later combined into words (the keywords). For each processed word-part an observation sequence is generated and fed to the trained HMM system to determine its proximity to each of the keyword's word-parts. This approach is suitable for large documents authored by the same writer, as is the case for many large historical Arabic manuscripts.

4.2 Dynamic Time Warping

Dynamic Time Warping is an algorithm for measuring similarity between two sequences which may vary in time or speed. It suites matching sequences with missing information or with non-linear warping. For 1D sequences DTW run at polynomial time complexity and is usually computed by dynamic programming using Equation 1.

$$\begin{aligned} D(i,j) =& min\{D(i,j-1), D(i-1,j), D(i,j)\} \\ &+ cost \end{aligned} \tag{1}$$

In this research we have slightly modified the classic DTW to include different costs for insertion, deletion, and substation. The DTW is computed by taking the minimum of the three options including the cost of each operation, as shown in Equation 2. Typical DTW considers the same cost for the three operations – deletion, insertion, and substitution. Such typical configuration tends to reduce the distance between sequences that have relatively similar small subsequences, with respect to the entire sequence.

We assign different cost functions for deletion, insertion, and substitution based on the introduced change. In general handwriting, including Arabic, the difference between two point sequence that represent two different words is very small, i.e., inserting/deleting just a few elements can change the sequence to represent a different word-part, see Figure 3.

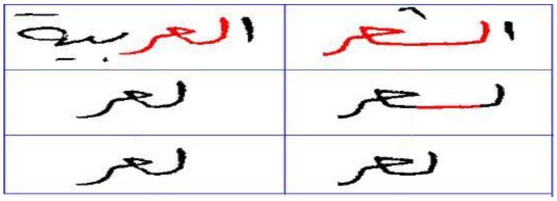

Figure 3. The deletion of similar segments can lead to a different word-part, which illustrates the need for different costs for insertion, deletion, and substitution.

For a these given two sequences s_x and s_y, we define the $cost_{ins}$, $cost_{del}$, and $cost_{sub}$ as the cost of inserting a new element into the sequence s_x, the deletion of an element from the sequence s_x, and the substituting of the element x_i in s_x by the element y_i in s_y, respectively.

$$\begin{aligned} MD(i,j) = min\{ & MD(i-1,j) + cost_{del} \\ & MD(i,j-1) + cost_{ins} \\ & MD(i-1,j-1) + cost_{sub}\} \end{aligned} \quad (2)$$

We have developed four searching schemes using the TDW algorithm. These schemes differ in the way we generate the keyword from its textual description and the searched document.

In the first scheme we manually extract the keyword's word-parts from the document. Then we search for the extracted word-parts in the input document. Note that extracting the word-parts of the keyword does not necessary require locating the keyword itself. Extracting multiple shapes of the same keyword have yielded better results. Since word spotting generates the index for all the words in a document, this scheme is appropriate for word spotting without the need to extract word-parts manually.

In the second scheme a human operator generates several versions of the keyword by extracting letter shapes, from the document, and assembling them into word-parts. Then the generated keyword shapes are used to search the document images.

The third scheme automatically generates multiple shapes of the keyword using predefined fonts and handwritten templates. The generated shapes are used to search the documents for the keyword. The best matches among the located keywords are used to extend the keyword shapes for future search (within the same session). The process accumulatively proceeds until locating all the appearance of the keyword.

In the fourth scheme a human operator mimics the document handwriting by following the shapes of letters in the input document, using a digitizing device, such as Tablet-PC. Generating multiple samples for the keyword has improved the matching results.

It is important to note that there is no need for a prototype database in the first, second, and forth schemes. In the third scheme we maintain a dictionary that includes the predefined handwritten templates of word-parts, using variety of common handwriting styles.

The match between the shapes of two word-parts is estimated by computing the feature vectors, mentioned in section 3.3, for each word-part. Let wp_k denote a keyword's word-part and wp_d denote a word-part in the searched document. The function $\mathcal{F}(wp)$ calculates the feature vector for the word-part, wp. The cost for the substitution, deletion, and insertion operations are defined in Equations 3, 4, and 5, respectively, where $0 \leq i \leq n$ and $0 \leq j \leq m$.

$$cost_{sub}(x_i, y_j) = |\mathcal{F}(wp_k)(i) - \mathcal{F}(wp_d)(j))| \quad (3)$$

$$cost_{del}(x_i) = (\mathcal{F}(wp_k)(i+1) - \mathcal{F}(wp_k)(i-1))^2 \quad (4)$$

$$cost_{ins}(y_j) = (\mathcal{F}(wp_d)(j+1) - \mathcal{F}\Im(wp_d)(j-1))^2 \quad (5)$$

4.3 Pruning

Since matching algorithms are usually very expensive, a pruning step is necessary to avoid comparing word images that are very different from the keyword image.

The compared word-parts are normalized according to the average height of the document's word-parts. In this work we perform pruning by using the contour properties and the density histograms. The ratio between the width and the contour's length of the compared word-parts are computed. A pre-computed ratio is used to prune word-parts with large ratios. The horizontal and vertical density histograms are computed for the two compared word-parts. We then calculate the sum of the square differences between the two horizontal and vertical density histograms, separately. An experimentally determined threshold is used to eliminate the irrelevant word-parts (see Figure 4).

4.4 Rule-based system

The system treats each word-part as a meta component – one main component and associated secondary components. Recall that the secondary components represent additional strokes associated with the word-parts, represented by the main component. The shape of an additional stroke could be a dot, detached vertical segment, or

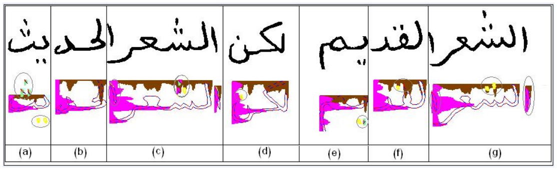

Figure 4. The columns (c) and (g) show the similarity of the density histograms of the same word-parts .

small curve (usually similar to " s " or " ˜ "). Additional strokes can appear above or below word-parts. The additional dots are associated in groups that include one, two, or three dots. Our rule-based system utilizes the number, shape, and position of the additional strokes to prune irrelevant words and verify the match between two word-parts. In addition, the rule-based system determines whether a located set of word-parts could be combined into one keyword or not. Recall that our four searching schemes (see Section 4.2) use accumulative search process, i.e., quality results from an iteration are used as the sample set for the next one. The rule-based system also determines the samples for the next iteration, by using the match probabilities of the results to determine their quality.

5 Experimental results

We have performed several experiments to test our system. We have used a dataset of 40 pages written in Arabic and included more than 8000 words and more than 15000 word-parts. These pages are classified into three groups: printed, handwritten, and historical documents, each include five documents. The printed documents are in different fonts and the handwritten documents were written by different writers. Each one of the printed and handwritten documents includes two pages and each one of the historical documents is composed of four pages.

A two phases process has been used to complete the search task. In the first phase the system recognizes the main bodies of the word-parts and the additional strokes, separately. The second phase combines the recognized word-parts and the additional strokes into keywords using the rule-based part of the system.

We have run experiments using the four searching schemes. In order to highlight the insufficient training problem, we have used two training sets of different sizes – small (S) and large (L). It is important to notice that the results we are presenting consider word-parts, since the focus is on the matching algorithm and the geometric features. In addition, the four search schemes use accumulative search.

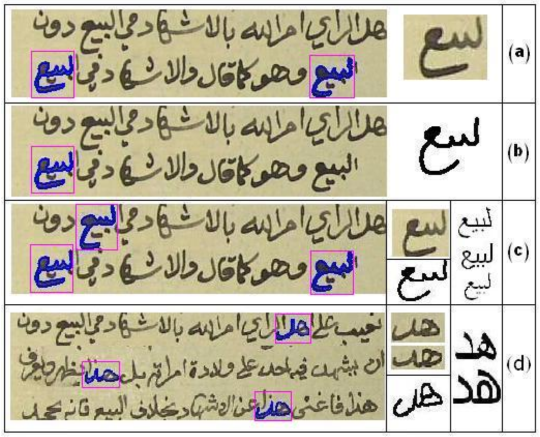

Figure 5. The results from the first (a) and the four(b) search schemes; and the final results, using the accumulative process, are shown in (c) and (d)

Table 1. Results of DTW and HMM classifiers. The improvement achieved by our modification are depicted using the numbers inside the parentheses

—		DTW Results		HMM Results	
Data	Sc	Small	Large	Small	Large
Printed	1	86(+2)%	88(+6)%	76%	86%
	2	86(+2)%	87(+7)%	72%	83%
	3	88(+0)%	89(+1)%	60%	71%
	4	88(+0)%	90(+2)%	62%	85%
HWR	1	82(+6)%	86(+6)%	70%	76%
	2	78(+8)%	86(+6)%	60%	74%
	3	79(+6)%	84(+7)%	60%	71%
	4	76(+6)%	84(+6)%	41%	56%
History	1	80+2%	81(+5)%	61%	66%
	2	80(+0)%	81(+5)%	46%	63%
	3	76(+6)%	79(+8)%	39%	52%
	4	74(+6)%	76(+9)%	34%	44%

Table 1 shows samples of our results. The $5th$ and $6th$ columns show that the HMM based system is highly dependent on size of the training set. It also shows that the more variation we use to train the system, the better the recognition rates we achieve. The $3rd$ and $4th$ columns show that in general the DTW provides better results and it is less sensitive to the size of the training set. The DTW classifier performance slightly deteriorates when using a small training set. It is also able to find close variations of a given word-part better than HMM. The results are excellent for the Handwritten (HWR) and the Printed document and very good for the Historical documents.

Since it is not always possible to provide enough samples to train a probabilistic classifier, our experimental results show that it is better to use DTW rather than HMM for keyword searching in Arabic historical documents.

The *CEDARABIC* system [16] is a well known system for Arabic word spotting. Our system differs from

CEDARABIC in several ways. They spot the entire word as one component, while our system searches for word-parts without additional strokes. As a result, our system deals with a much smaller dictionary that includes only the word-parts without the additional strokes. The *CEDARABIC* system accepts the spotted words in English characters, which are used to guide the search for the appropriate Arabic prototype. In contrast, our system accepts the search words directly in Arabic (handwritten, and printed), which are used to automatically generate the prototypes for searching. Our system relies on local features to preform the DTW-based search, while The *CEDARABIC* system uses correlation similarity measure based on global word shape features.

6 Conclusion

We have presented keyword searching algorithms for Arabic documents. Our experimental results show that the used geometrical features can capture the real behavior of the written script for matching purposes. The non-linearity of the DTW method provides very good results and seems to be adequate for keyword searching in Arabic handwritten documents. Since it is not always possible to provide enough samples to train an HMM system, it does not seem to be the right choice for keyword searching, when very few training samples are available.

The scope of future work includes replacing the component's contour by a representative skeleton which preserve the small features of the Arabic script such as tooth and closed loops.

References

[1] F. Chen, L. Wilcox and D. Bloomberg, ”Word spotting in scanned images using hidden Markov models”, *Acoustics, Speech, and Signal Processing, 1993. ICASSP-93., 1993 IEEE International Conference on*, 27-30 April 1993, volume 5, pp 1–4vol.5.

[2] J. Duong, M. Côte, H. Emptoz and C. Y. Suen, ”Extraction of text areas in printed document images”, pp 157–165, 2001.

[3] F. Farooq, V. Govindaraju and M. Perrone, ”Pre-processing Methods for Handwritten Arabic Documents”, pp 267–271, 2005.

[4] B. Gatos, T. Konidaris, K. Ntzios, I. Pratikakis and S. Perantonis, ”A segmentation-free approach for keyword search in historical typewritten documents”, *Document Analysis and Recognition, 2005. Proceedings. Eighth International Conference on*, 29 Aug.-1 Sept. 2005, pp 54–58Vol.1.

[5] S. S. Kuo and O. E. Agazzi, ”Keyword Spotting in Poorly Printed Documents using Pseudo 2-D Hidden Markov Models”, *IEEE Trans. Pattern Anal. Mach. Intell.*, 16(8):842–848, 1994.

[6] V. Lavrenko, T. Rath and R. Manmatha, ”Holistic Word Recognition for Handwritten Historical Documents”, January 2004, pp 278–287. PARC, Institute of Electrical & Electronics Engineering.

[7] R. Manmatha and T. Rath, ”Indexing Handwritten Historical Documents - Recent Progress”, 2003.

[8] R. Manmatha and J. Rothfeder, ”A Scale Space Approach for Automatically Segmenting Words from Degraded Handwritten Documents”, *TPAMI*, 27(8):1212–1225, 2004.

[9] T. Rath, S. Kane, A. Lehman, E. Partridge and R. Manmatha, ”Indexing for a Digital Library of George Washingtons Manuscripts: A Study of Word Matching Techniques”, 2002.

[10] T. Rath, V. Lavrenko and R. Manmatha, ”Retrieving Historical Manuscripts using Shape”, (328), 2003.

[11] T. Rath and R. Manmatha, ”Features for word spotting in historical manuscripts”, *Document Analysis and Recognition, 2003. Proceedings. Seventh International Conference on*, 3-6 Aug. 2003, pp 218–222vol.1.

[12] T. Rath and R. Manmatha, ”Word image matching using dynamic time warping”, *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, 18-20 June 2003, volume 2, pp II–521–II–527vol.2.

[13] J. L. Rothfeder, S. Feng and T. M. Rath, ”Using Corner Feature Correspondences to Rank Word Images by Similarity”, 2003.

[14] C. H. S. N. Srihari and H. Srinivasan, ”A Search Engine for Handwritten Documents”, *Document Recognition and Retrieval XII, San Jose, CA,Society of Photo Instrumentation Engineers (SPIE)*, pp pp. 66–75, January 2005.

[15] C. H. S. N. Srihari, H. Srinivasan and S. Shetty, ”Spotting Words in Latin, Devanagari and Arabic Scripts”, *Vivek: Indian Journal of Artificial Intelligence,*, 16(3):2–9, 2003.

[16] P. B. S. N. Srihari, H. Srinivasan and C. Bhole, ”Handwritten Arabic Word Spotting using the CEDARABIC Document Analysis System”, *Proc. Symposium on Document Image Understanding (SDIUT 05), College Park, MD*, November 2005.

[17] R. Saabni, F. Biadsy, J. ElSana and N. Habash., ”Segmentation-Free Online Arabic Handwriting Recognition”, Technical report, Ben Gurion University of the Negev., Beer Sheva , Israel., 2008.

[18] V. L. Toni Rath and R. Manmatha, ”A Statistical Approach to Retrieving Historical Manuscript Images”, 2003.