Recognition-free Retrieval of Old Arabic Document Images

Toufik Sari and Abderrahmane Kefali

Laboratoire de Gestion Electronique de Documents (LabGED),
University Badji Mokhtar, Annaba, Algeria
sari@labged.net, kefali@labged.net

Abstract. Searching of old document images is a relevant issue today. In this paper, we tackle the problem of old Arabic document images retrieval which form a good part of our heritage and possess an inestimable scientific and cultural richness. We propose an approach for indexing and searching degraded document images without recognizing the textual patterns in order to avoid the high cost and the difficult effort of the optical character recognition (OCR). Our basic idea consists in casting the problem of document images retrieval from the field of document analysis to the field of information retrieval. Thus, we can combine symbolic notation and semic representation and exploit techniques from the two fields, in particular, the techniques of suffix trees and approximate string matching. Each document of the collection is assigned an ASCII file of word codes. Words are represented by their topological features, namely, ascenders, descenders, etc. So, instead of searching in the image, we look for word codes in the corresponding file code. The tests performed on two types of documents, Arabic historical documents and Algerian postal envelopes, have showed good performance of the proposed approach.

Keywords. Document retrieval, Arabic handwriting recognition, approximate string matching, document analysis.

Recuperación de documentos árabes antiguos a partir de imágenes sin usar reconocimiento de caracteres

Resumen. La búsqueda en imágenes de documentos antiguos es en la actualidad un tema relevante. En este artículo abordamos el problema de recuperación de documentos árabes antiguos a partir de imágenes sin usar el reconocimiento de caracteres (OCR). Dichos documentos forman una buena parte de nuestra herencia y poseen una riqueza científica y cultural invaluable. Nosotros proponemos un enfoque para indexar y buscar imágenes degradadas de documentos sin recurrir al reconocimiento de patrones textuales para así evitar el esfuerzo considerable y el alto costo que conlleva el OCR. La idea básica consiste en migrar el problema de la recuperación de estos documentos, desde el campo del análisis de documentos hacia el campo de la recuperación de información. Así, podemos combinar la notación simbólica y la representación sémica y explotar las técnicas que provienen de ambos campos de investigación, particularmente, las técnicas de árboles de sufijos y búsqueda aproximada de cadenas. A cada documento de la colección se le asigna un archivo en ASCII con códigos de palabras. Las palabras son representadas por sus características topológicas; ej. ascendientes, descendientes, etc. De esta forma, en vez de buscar en la imagen, nosotros buscamos en los códigos de palabra dentro del archivo de códigos correspondiente. Las pruebas se realizan en dos tipos de documentos: documentos históricos árabes y sobres postales argelinos. El enfoque propuesto muestra un buen rendimiento.

Palabras clave. Recuperación de documentos, reconocimiento de manuscrito árabe, búsqueda aproximada de cadenas, análisis de documento.

1 Introduction

Since the invention of the writing about the year 3200 BC, knowledge has been transmitted by writing. Currently, important electronic document collections exist in the libraries, museums and other institutions of pedagogic or sociopolitic nature. The historical documents of old civilizations and the public archives are typical examples of such riches which represent the heritage, the history and the dignity of the nations. Indeed, the condition of old documents becomes gradually worse with the time or when they do not get spoiled in a natural manner, they suffer from being handled for consultation too often. In order to overcome these problems, digitalization was instigated. Digitalization allows preserving the original documents, sharing them with other people, but it does not facilitate their access for consultation and retrieval. In fact, the access to these collections requires efficient strategies of indexing and retrieval. The majority of indexes are created manually. While this approach is possible for a small number of documents, its cost and effort become very high for large collections. Automatic approaches are thus desirable.

To reach the contents of the documents, two automatic approaches are possible. The first approach known as *analytical* consists in the development of algorithms and methods allowing us to recognize the documents contents and thus get a textual transcription and annotations. The second approach known as *holistic* allows the user to search and navigate in the collection without a complete identification of the content.

Actually, optical character recognition (OCR) can be applied only over Arabic printed documents or handwritten documents with a limited lexicon. When documents are degraded, with complex structures and covering large domains, the OCR becomes ineffective [3, 11, 17] (see Fig. 1).

Fig. 1. Examples of complexes document images

In the case of the old Arabic documents, the difficulty increases because of the topological features of Arabic writing which further complicate the task of automatic recognizers.

We should point out here that the ultimate goal of document images retrieval systems is not to recognize the textual or graphical patterns in the documents, but to find the original materials corresponding to the user´s need [15].

While a lot of effort has been put in OCR-free processing and retrieval of document images, no work dealt with Arabic handwritten documents even within the TREC framework.

In order to deal with the defectiveness of pattern recognition techniques, it is necessary to understand the complex structures and the heterogeneous content of the documents and to keep in mind the goal of IR, so we propose a recognition-free approach for retrieving handwritten Arabic document images from huge collections in order to avoid the high cost and the difficult effort of the OCR. The search strategy proposed is based on the combination of the techniques of *suffixes trees* [27] and *approximate string matching* [16]. We do not perform any recognition of words or characters. Another important innovation in our current work is that end-users can search document images by textual queries, which we consider as a noisy copy of the text already present in documents.

The rest of this paper is organized in the following way. Section 2 gives some previous works concerning document images retrieval; in Section 3, we present our approach illustrated by preliminary investigations, and the experiments are described in Section 4. Conclusions are given in Section 5.

2 Previous Work

The search of words in the Latin documents has attracted considerable attention recently. Many works was carried out on word searching by *Word spotting* and by character recognition in the document images. But in spite of this significant number of works, the results obtained until now are not sufficient to treat important volumes of data [8].

As we can see in the literature, the work of Spitz et *al.* seems to be one of the closest to our investigation. A. Spitz in [25] proposed to code the characters of the printed texts according to their forms. For each word of the document, the method extracts features of the characters based on the connected component count of each character, and on their position in relation to two base lines. The characters are then coded according to their features. For example, letters with ascenders may be mapped to the code *A*. Those with descendents may be mapped to the code *g*. The sequence of obtained codes form a *word shape token* (WST). Thus, the word “Goods” can be coded by the WST AxxAx. Query words are mapped in the same way to WSTs. Indexing and retrieval of documents can now be done as usual, but using WSTs instead of words. Note that this method only determines the general shape of a character rather than trying to identify individual characters. The hope here is that shape information can be obtained more accurately and at a lower cost than full OCR.

Later, Smeaton and Spitz [24] showed that this technique is useful only if the images are of bad quality implying a failure of the OCR. It was noted that WST-based retrieval performed worse than the conventional word-based retrieval even on a poor quality OCR output.

Chen *et al.* [6] proposed an approach based on the information of the word forms instead of the character forms. Firstly, they identify upper and lower contours of each word using mathematical morphology. From these contours, they extract the form information based on the pixels location. Then, the Viterbi decoding of the coded word is used to match the word image with the query.

If keyword spotting is a non-trivial problem for images of printed text, it is even more difficult for handwritten text. One of the first works on handwritten document retrieval is that of Manmatha *et al.* [13] in which a semi-automatic approach to indexing handwriting documents was proposed. In this work, the similar images of words are grouped into equivalence classes. The most frequently occurring classes are eliminated since they represent function words and the majority of the remaining equivalence classes are manually coded in ASCII and used as index.

Several works were also carried out on the historical document retrieval. For example, [5] proposed an approach for searching in the cards of military incorporation of the 19th century. The idea of this approach is to index automatically the old forms by an ordered chain of graphemes associated to the case of a handwritten patronym, and also to transform the alphabetical query of the user into a chain of graphemes. The comparison between the query and the document indexes is made by using the traditional edit-distance in order to take account of possible errors [18] in searching among manuscripts written in the Telugu language. These manuscripts were characterized through representations by wavelets of the words. The representation by wavelets provides information on the contents of the image to various scales. It exploits the characteristics inherent in the characters of Telugu. The application of this representation by wavelets for Latin characters does not give good results.

Rath and Manmatha [20, 21] presented a holistic approach to word search in historical handwritten documents. This approach consists of grouping word images in similar groups by using word images matching. Then it seeks the set of the most representative groups and labels them. Each labeled group is used as index. In [21], the authors propose to represent word images by four features of profile which are then matched by using various methods. [20] used the correspondences between the angular points to classify word images in historical manuscripts. The Harris detector of angular points is employed in words images. Correspondences between these points are established by comparing local windows and by using the sum of the squares of the differences. The similarity measurement between the words is given by the Euclidean distance between the points put in correspondence.

Motivated by the work of Rath and Manmatha, Adamek *et al.* [1] proposed to compare word contours instead of whole words for the holistic recognition in historical manuscripts. After binarization of a given document, the method runs a multi-scales segmentation in order to detect word contours. The search of a word in a document is implemented then by comparing word contours.

Another recent work of indexing and searching of old documents is that of Ramel *et al*. [19]. In this work, the authors proposed an approach to document search in the collection of the Higher Study Center of the Rebirth (from the 14th to the 17th century) without prior recognition of the document model. They made a study of the structural characteristics of these documents. Then, they applied a hybrid analysis which benefits from the advantages of both methods of analysis, i.e. the ascending and downward ones. Although this approach gives a higher recognition rate, it is slow and sensitive to skews.

In [8], the authors are interested in searching words in the images of old printed documents. The authors propose to work with characters and not with words, and to represent each character by a set of features. In this research, the query is treated in a way similar to the total document and the features of the query characters are matched with the features of the word characters already stored in index files.

Bai *et al*. [2] proposed an approach based on the coding of words according to their forms. For each word, they extract a set of seven features and each word is coded thereafter by a sequence of codes. For searching, a given query is coded in the same manner as a sequence of codes and the query code is matched with the code of each word in each document using dynamic programming. The features used are character ascenders, descenders, deep eastward and westward concavity, holes, i-dot connectors and horizontal-line intersection. The tests were performed on Latin documents.

The current work extends the work of Sari *et al*. [23] in three ways. First, most of the errors were due to the deficiency of word feature extraction. This in turn is caused by skewed and degraded images. To overcome such problems, we implemented and tested many thresholding and skew correction techniques. The second problem is that no indexation was used. Every time a user enters a query, the system re-accesses the whole collection. So, we added a small index table to keep more recent queries. In addition, we employed suffix tries techniques to overcome the problem of search time and storage spaces, the two main drawbacks of information retrieval systems in general and image retrieval in particular.

3 Proposed Approach

The objective is to develop an information retrieval system for old Arabic handwritten documents without recognizing the contents in order to avoid the high cost and the difficult effort of the OCR.

Our basic idea consists in transposing the problem of document image indexation and retrieval from the field of document analysis to the field of information retrieval. Thus, we can combine symbolic notation and semic representation and exploit the techniques from the two fields.

Each document of our collection is represented by a signature made of the set of relevant information which we can extract from the document image. This signature contains the characteristics of the text and it is coded in ASCII which will allow us to easily employ sophisticated information retrieval techniques: approximate string matching and suffixes trees.

We proceeded in two phases, one phase of processing and analysis of document images and another phase of retrieval in which we exploit the results of the previous phase in answering the user´s query.

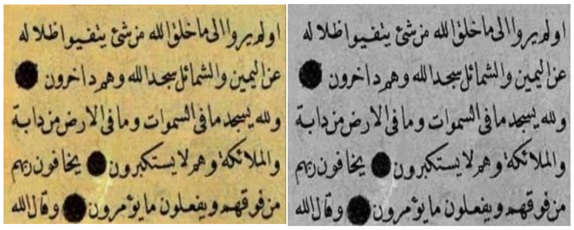
3.1 Phase 1: Document Analysis+

The goal of this phase is to analyze the images included in our collection in order to extract their characteristics. It starts with the acquisition of the documents, preprocessing, line and sub-word segmentation and finally the extraction of sub-word features. The features will be coded and saved in ASCII files. The retrieval will be performed later on these files and not on the images.

a. Preprocessing

Images of Arabic historical documents are very noisy. This is due to a degenerated quality of the original media and technical/numerical reasons related to computerizing. The preprocessing of these documents is thus necessary. It allows to reduce the noise and to keep only the significant information in order to prepare the ground for the following stages. In our system, the preprocessing includes gray level transformation, binarization, smoothing, and skew correction.

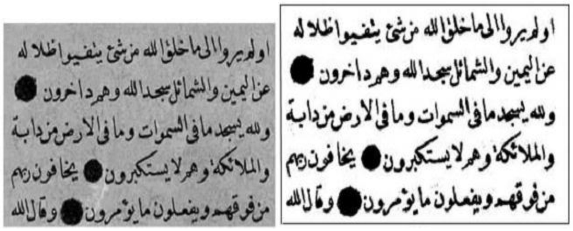
Gray level transformation: the images of our collection are mostly in color. These images should be transformed into gray levels (256 gray levels). This transformation can be carried out simply by assigning each pixel of the image the average of its values of red, green, and blue color (Fig. 2).



(a) Colored image (b) Gray image

Fig. 2. Image color transformation

Binarization: it is an irreversible processing which allows transforming an image with gray levels or color into a black and white image according to a threshold (Fig. 3). Its goal is to decrease the quantity of information present in the image, and to keep only relevant ones. Several techniques were proposed in the literature for the binarization of document images in gray levels. In [7], the authors performed a comparative study of 12 thresholding methods most frequently referred to in literature by applying them to old document images. This study showed that the method used by Khurshid *et al.* [9] was the best.



(a) Gray level image (b) Binarized image

Fig. 3. Binarization of a gray level image

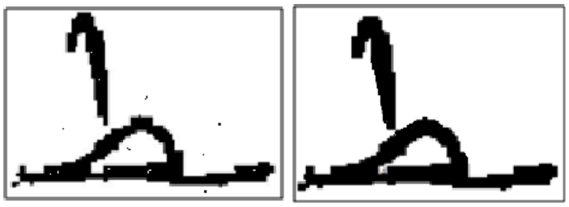
In this method, the threshold of binarization is calculated for each pixel of the image by using the following formula:

$$\boldsymbol{T = m + k\sqrt{\frac{(\sum p_i^2 - m^2)}{NP}}} \qquad (1)$$

where k varies between -0.1 and -0.2, m is the average gray level on a window centered over the current pixel, p_i is the gray level of the pixel i and *NP* is the total number of pixels.

In our experiments, we chose k=-0.2 and a window size=19*19.

Smoothing: in order to eliminate the irregularities which can emerge after binarization, we apply smoothing according to the algorithm described in [12] which reduces the noise of a binary image by eliminating the insulated pixels on the one hand and by filling the empty holes on the other hand (Fig. 4).



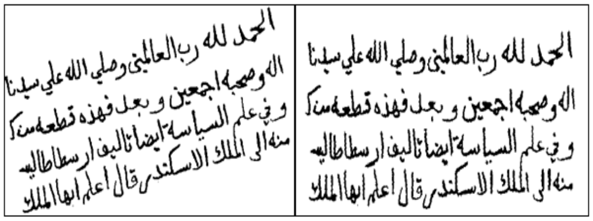
(a) Binarized image (b) Smoothed image

Fig. 4. Smoothing of an image of the letter *Tad*

Skew correction: the old documents in our collection are sometimes slanted. The presence of skews influences considerably the stages of segmentation and feature extraction and thus the final result of the system. To deal with this problem, we propose to correct the skew of text lines by the partial projection technique. The method proceeds in five steps as follows:

1. Division of the image in columns of fixed size (100 pixels in our experiments).
2. Calculation of the histogram of horizontal projections for each column.
3. Extraction of baselines corresponding to the peaks of histograms calculated before.
4. Calculation of the skew angle θ of the document as the average of all the angles formed by 2 baselines of 2 successive columns.
5. Rotation of the image by the angle θ.

The final result of this stage is shown in Fig. 5.



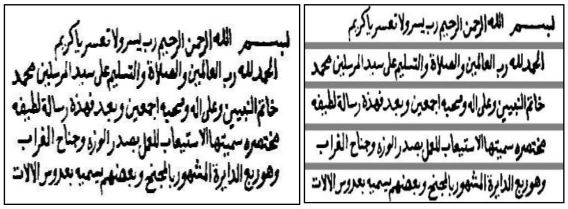
(a) Skewed image (b) Aligned image

Fig. 5. Skew correction

b. Line Segmentation

In this phase, we extract lines from text by applying a method based on the technique of horizontal projections. It consists in separating the lines using the density measure of white lines in horizontal projections. The valleys of the histogram correspond to the separation zones between successive lines (Fig. 6). The method proceeds in five steps as follows:

1. Calculation of the horizontal projections histogram of the image.
2. Extraction of the local minima corresponding to the separation zones between lines.
3. Filtering of the minima by eliminating the minima having a width lower than the longest minimum/2 and by removing one of two minima which are very close to each other.
4. Resolution of conflicts by assigning the black pixels existing in the separating zones to the nearest text line by proximity analysis.



(a) Non segmented image (b) Line detection

Fig. 6. Line detection and segmentation

c. Sub-word Decomposition

As it was said previously, we chose to work with sub-words and not with the characters, because the latter cannot be easily separated. This difficulty was clearly defined by Sayre in 1973 and it can be expressed in brief by the following dilemma: to recognize letters, we should segment words, and to segment words, we should recognize letters, also see [22].

Sub-words extraction consists in detecting various connected components (CC) in an image, i.e., gathering the neighboring pixels in one element called a connected component. At this level, we consider the diacritics as CC (Fig. 7).

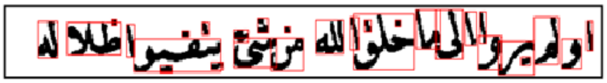


Fig. 7. Connected components extraction from a text line

The following algorithm performs line segmentation in CC:

Input: line images
Output: connected components

1. Find a non-visited black pixel.
2. Find all its neighbors: if one of the neighbors is a black pixel we gather it with the first one and reiterate recursively the operation for all the neighbors.
3. Stop when all the black pixels are visited.
4. Determine the four extreme points (high, low, left, right) to frame this CC. The four extreme points define the bounding box.
5. Return to stage 1.

d. Feature Extraction and Coding

This phase should guarantee maximum reliability, because the subsequent processing will not handle the image any more but rather the results provided by this module. Its goal is to identify the most important properties for form discrimination.

From each sub-word, we chose to extract four topological features, i.e., ascenders, descenders, loops (or holes) and diacritic points described in the following paragraphs.

Baseline detection: the baseline in the Arabic text carries significant information of text orientation and diacritic positions. The most used method for baselines detection is the horizontal projections of the text line. The baseline corresponds to the line whose projection contains the greatest number of black pixels (the red line in Fig. 8).

Median zone detection: this step consists in representing the body of the words or characters. To obtain high and low median zone, baselines were located according to the principal baseline; the space between these two lines is the median zone (the blue lines in Fig. 8).

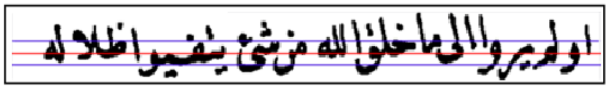


Fig. 8. The baseline and the median zones of a text line

Contour following: the contour following is commonly used for extracting topological features of characters. The Freeman string is the most used method for the description of contours in images. The contour of a sub-word is then a sequence of points where each is coded by its direction according to the Freeman code. After localizing the baseline, the median zone of a line and the contours, the following features can now be extracted:

Diacritics: they are simple or multiples points, which vary from one to three points which can be written above or below the primary part of characters. A connected component is considered as a diacritic point if its size is lower than a certain threshold.

Descenders: they are the most used primitives in handwriting recognition. They are detected by a stroke stretching below the median zone, i.e., located in the lower zone.

Ascenders: being opposite to descenders, ascenders are detected by writing above the median zone; therefore, we check the presence of ascenders in the higher zone.

Loops or holes: in Arabic writing, loops are generally located in the median zone; they are inner contours.

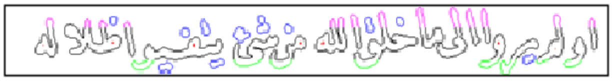


Fig. 9. Feature extraction from a text line

Each feature is indicated by different color in Fig. 9.

Then, the document image will be coded according to its sub-words features by a code corresponding to each feature (ascenders: *h*, descenders: *j*, loops: *b*, high diacritic points: *p*, and low diacritic points: *q*). We added another character "#" for inter sub-words space. For example, the text in Fig. 9 is coded as

```
h#j#hbj#qj#bj#h#h#h#h#phjp#h#hhb#bpj#pjp#qpb
pqj#h#pbhh#hb
```

Since the treatments are done on each text line, we obtain a sequence of codes for every line. These codes will be stored in a file associated with the document.

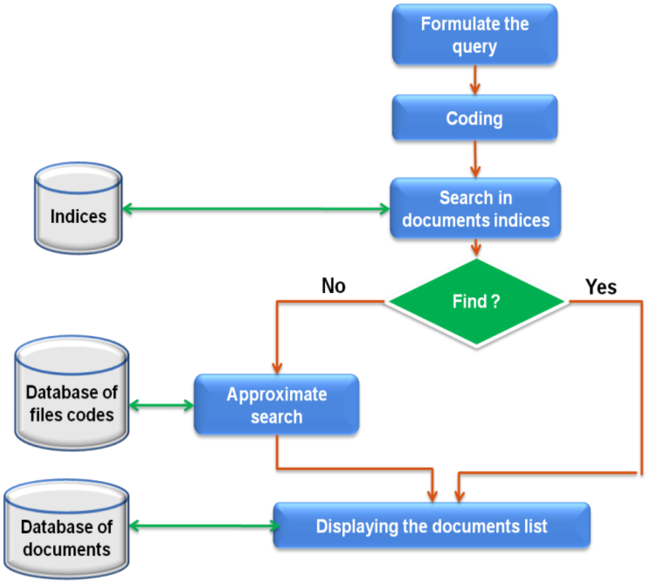


Fig. 10. Overall architecture of the retrieval phase

e. Indexing

Indexing is a technique widely used in retrieval systems. Its goal is to extract and represent the documents so that they can be found quickly by the user. In our system we employ a semi-automatic indexation. We run several retrieval sessions with different queries and for each document we keep the list of words that really exist in the document.

The search process for a query will first be done in the index table and then in file codes.

3.2. Phase 2: Document Retrieval

In this phase, we exploit the results obtained in the first phase, in particular, the file codes and the index table to answer a user query. The overall architecture of this stage is shown in Fig. 1.

a. Formulating and Coding the Query

To search images containing a particular word, the user formulates a query in natural language (Arabic in our case) (Fig. 11).

This query should be also coded as it was done for sub-words in the first stage, i.e., it will be coded according to topological features of the letters forming the query.

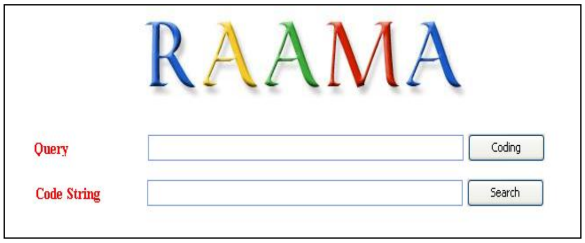


Fig. 11. The user interface

The query is coded by a precise code corresponding to each letter of the query, and this procedure outputs the code sequence. Search will be carried out using this code string and not the query itself. We developed the correspondence table of Arabic letter codes (Table 1).

b. Search in the Index Table

As the first stage of search, we carry out a search in the index table. Note that this type of search is an exact search because the documents are indexed manually. In this stage, the query will be compared with the indexes of all documents included in the collection. A document is considered to be relevant for the user if at least one of its indexes contains the query. For comparing the query with the indexes, we use the suffix tree technique. Suffix trees are very effective and fast techniques for searching a pattern *P*, of length *m*, in a text of length *n* within time *O(m)*, which allow to reduce the search time considerably. For constructing suffix trees, we use the algorithm of McCreight [14].

Table 1. Feature codes of Arabic letters

Character	Code	Designation
ا - ـا - كـ - لـ	h	Ascender
إ	hq	Ascender +Down Diacritic
أ - ـأ	ph	Up Diacritic +Ascender
ل	hj	Ascender +Descender
ط	bh	Loop +Ascender
ظ	bph	Loop +Up Diacritic +Ascender
لا	hbh	Ascender +Loop+ Ascender
ك	hp	Ascender +Up Diacritic
ي	jq	Descender +Down Diacritic
غـ - خـ - ذ- ت- ئـ - نـ	p	Up Diacritic
غ	jp	Descender +Up Diacritic
شـ - ث	pp	Up Daicritic +Up Diacritic
ن- ز- خ - ئ	jp	Descender +Up Diacritic
ش	ppj	Up Diacritic + Up Diacritic + Descender
ض	bpj	Loop + Up Diacritic + Descender
ضـ - فـ- ة- ـغـ - قـ	bp	Loop +Up Diacritic
ق	pbj	Up Diacritic + Loop + Descender
يـ - جـ - ب	q	Down Diacritic
ح-ع- س- ر- ى	j	Descender
ج	jq	Descender + Down Diacritic
ـعـ - مـ - صـ- ـه - ه	b	Loop
هـ	bb	Loop +Loop
ع - و- ص- م	bj	Loop +Descender
ـلا	hh	Ascender +Ascender
ـة	pb	Up Diacritic +Loop
ـغ	pbj	Up Diacritic + Loop + Descender
ؤ	bjp	Loop + Descender + Up Diacritic
لأ	hbhp	Ascender + Loop + Ascender + Up Diacritic
لإ	hbqh	Ascender+Loop+Down Diacritic+ Ascender

Given *P* of the query, search in the tree is very easy; we start from the tree root and follow the branch whose label begin with the same symbol as *P*. Following this branch, we arrive at a new node and run the procedure again for the remaining part of *P*.

c. Approximate Search in the Code Files

If no document is found in the first stage of research, it does not mean that the query is not present in any document. It is possible that the query does not appear in the index table but exists in codes files. To provide an adequate treatment, we propose to search in the codes files. Since the extracted sub-word codes are imprecise and in certain cases are incomplete, we propose to compute an approximate string matching algorithm in order to take account of these possible errors.

In fact, the addition of the approximation in the search allows detecting of approximate patterns but also increases the number of false occurrences. In our personal opinion, to have false occurrences is better than to ignore relevant ones.

The approximate search for a word *P* in a text *T* consists of finding all occurrences in *T* close to the word *P*. A decision concerning closeness of words requires the introduction of a threshold *K* which determines the allowed error count.

In this stage, we use two distance measures between sequences: the famous edit distance using Ukkonen's algorithm [26], which consists in calculating only a part of the distance table, and the Jaro-Winkler distance [28].

The general edit distance is the distance that makes it possible to transform a string X into a string Y using three kinds of basic operations: substitution of a letter of X by a letter of Y, deletion of a letter of X, or the insertion of a letter of Y. A cost is assigned to each of these edit operations for each letter of the alphabet:

- Sub (a, b) is the cost of the substitution of the letter a by the letter b.
- Del (a) is the cost of the deletion of the letter a.
- Ins (b) is the cost of the insertion of the letter b.

The general problem consists in finding a sequence of such basic operations to transform X into Y which minimize the total cost of the operations used. The total cost is equal to the sum of the costs of each of the basic operations. This cost is the distance between words.

The main difference of Jaro-Winkler distance from the general edit distance is that computation of the distance depends on the string length. In other words, two pairs of strings with different lengths will have different distances even for the same edit operations and the same number of admitted errors.

A document is considered to be relevant if the corresponding code file contains an occurrence of the query having at most *K* errors. The final result of this stage will be a combination of the results obtained by the two distance measures.

d. Displaying of Results

After the search, the results are displayed ordered by their relevance for the user. The relevance for the user is expressed by the minimal distance with the query.

4 Experimentations and Results

In order to evaluate our system performance, and due to the absence of a benchmark database of old Arabic documents available to be used in validation, we built a database of 150 images. The images in this database were collected from the Web, and thus they do not have equal characteristics. They are of very different nature, coming from several sources, covering several fields, present different types of degradations and structure complexity. Most of them deal with religion and sociology and others talk about mathematics, natural science and medicine including herbal medicine. Since we are interested essentially in documents with textual content rather than graphical illustrations, the first two kinds were preferred.

In our tests, we used another database composed of 398 images of Algerian postal envelopes in order to evaluate the performance of our system on another type of documents for which the application was not designed or projected. The envelopes in this database are written by various writers and they are digitized with the same device and parameters.

We interrogated these two databases by a hundred of textual queries in Arabic, their length varied from 4 to 20 characters, for which we have drawn up the list of relevant documents manually. Table 2 shows some examples of queries together with the number of corresponding relevant documents.

The evaluation of the obtained results is made in terms of recall, precision and response time. For each query, we computed the number of documents found and the number of relevant documents found.

Table 2. Some examples of queries and the number of relevant documents

Query	Number of relevant documents	Query	Number of relevant documents
الله	86	الأعمال	10
الرحمن	36	التراب	9
كتاب	20	مبعثرة	1
الملك	12	اللغات	7
ارسطا	5	الإنسان	21
طاليس	10	سيدي عمار	18
سوق أهراس			

Firstly, we evaluated the first stage of search which is the search in the index table. In this experiment, we tested several algorithms of exact search, especially the naive (brute force), Boyer Moore [4] and Knuth-Morris-Pratt [10] algorithms in order to compare them with the suffix tree technique employed in this stage. The results obtained for the two databases are summarized by Tables 3 and 4.

Table 3. Average recall, precision and search time obtained on the database of old Arabic documents

Criterion Method	Recall	Precision	Search time (ms)
Brute Force	0.147	1	190.180
Boyer Moore	0.147	1	211.415
Knuth Morris Pratt	0.147	1	190.390
Suffix trees	0.147	1	153.225

Table 4. Average recall, precision and search time obtained on the database of envelopes

Criterion Method	Recall	Precision	Search time (ms)
Brute Force	0.194	1	220.250
Boyer Moore	0.194	1	311.625
Knuth Morris Pratt	0.194	1	220.375
Suffix trees	0.194	1	182.125

Tables 3 and 4 show the perfect average precision (100%) of search results for various algorithms. This is explained by the fact that all returned documents contain true occurrences of the query, because the indexing of the documents was performed manually. On the other hand, the average recall obtained on the two databases is not sufficient (almost 15% for the first database and about 20% for the second one). Indeed, several relevant documents were ignored because the query is not present in the index table. This is possibly due to the fact that index tables could not contain all the possible words. With regard to the response time, the experiments have showed that the technique of suffix trees is faster than the other algorithms, which confirms our choice to use this technique at this stage.

We also tested the performances of the second stage which uses approximate string matching in code files.

Table 5. Average recall, precision and search time obtained using the edit distance on the first database

Value of K	Recall	Precision	Search time (ms)
0	0.297857143	0.9824	339
1	0.529166667	0.72723214	338
2	0.839456202	0.68648825	341.25
3	1	0.623125	353.75
4	1	0.39253	346.89
5	1	0.112501	339.20

Table 6. Average recall, precision and search time obtained using the edit distance on the second database

Value of K	Recall	Precision	Search time (ms)
0	0.243694	0.978571	405.36
1	0.383612	0.768045	411.89
2	0.672036	0.671100	413.23
3	0.871713	0.567266	402.00
4	0.935992	0.547570	389.79
5	0.985658	0.531372	392.75
6	1	0.416013	412.12
7	1	0.329079	405.03

Table 7. Average recall, precision and search time obtained using the Jaro-Winkler distance on the first database

Value of K	Recall	Precision	Search time (ms)
0.00	0,297857143	0,9824	572
0.01	0,312541283	0,9375	613
0.02	0,37202381	0,83333333	537
0.03	0,791666667	0,64096791	572
0.04	0,833333333	0,53647186	589
0.05	0,9375015	0,50343615	602
0.06	1	0,4701043	590
0.07	1	0,3885063	598

As approximate string matching allows returning occurrences close to the query by *K* errors. For each of the two distance measures, we tested several values of *K* in order to choose the best. Concerning information retrieval, we should find a better compromise between recall and precision. To do this, we should set a “good value” of *K*. This problem can be solved empirically.

Tables 5 and 6 recapitulate the results obtained by using the edit distance over the two databases and for several values of *K*.

The results obtained on the two databases using the Jaro-Winkler distance and for several values of *K* are summarized in Tables 7 and 8.

The obtained results show that the search recall and precision depend on the value of the threshold *K*, and vary in an opposite way for the two distances measures. Indeed, the best precision is obtained with the value 0 of *K*. On the other hand, the recall is worse for this value. This is justified by the fact that *K*=0, which means that the system returns only the exact occurrences, which increases the search precision until reaching 100% in most queries. But as our code files are not precise, and in some cases incomplete, a search with *K*=0 ignores several relevant documents containing occurrences close to the query thus decreasing recall.

Table 8. Average recall, precision and search time obtained using the Jaro-Winkler distance on the second database

Value of K	Recall	Precision	Search time (ms)
0.00	0.243694	0.978571	622.50
0.01	0.272959	0.898571	678.23
0.02	0.376226	0.877551	596.00
0.03	0.552354	0.836904	599.32
0.04	0.651388	0.772164	612.00
0.05	0.749364	0.711803	603.47
0.06	0.879509	0.660912	622.00
0.07	0.927804	0.601744	665.00
0.08	0.961240	0.585478	644.96
0.09	1	0.463174	638.71
0.10	1	0.367143	603.18

When the threshold *K* (the allowed error) increases, the number of returned documents increases, and the search recall increases also until reaching 100%.

For the database of the old documents, the maximum recall is obtained setting $k=3$ for the edit distance, and $k=0.06$ for the Jaro-Winkler distance. For the database of envelopes, the maximum recall is obtained with $k=6$ for the edit distance, and *k=0.09* for the Jaro-Winkler distance. These values give the worst precision.

As it was said, recall and precision vary in the opposite way; the best value of *K* is thus which presents the best compromise between recall and precision. The various tests carried out, show that the best compromise between the recall and precision is not constant but it depends on the length of the query codes. Table 9 gives some examples.

In order to take the above fact into account, we propose to adapt the value of *K* according to the length of the queries codes (the attribution of the values of *K* is made by experiment).

Table 9. Best values of *k* for some queries

Query	Query code	Length	Best k value	
			Edit distance	Jaro-Winkler distance
الملك	h#hbhhp	7	1	0.03
ارسطا طاليس	h#j#bhh#bhh#hqj	15	3	0.05
كتاب	hph#q	5	1	0.02
صلى الله	bhj#h#hhb	9	2	0.04

After having evaluated the two distance measures and chosen the best values of the thresholds, we combine the two distance measures in order to obtain best results. Finally, the overall documents returned are those retrieved using the edit distance and those retrieved using the Jaro-Winkler distance, and of course the intersection result is put in the first place.

Considering optimal threshold values for various word lengths, our system reaches an accuracy of 72% while obtaining a recall of 89% for the database of the old documents, and 82% of recall and 66% of precision for the database of the envelopes.

In our opinion, the preliminary test results are very good for such a challenging field of Arabic handwriting processing and document image retrieval. Our confidence goes toward Arabic sub-word processing which we consider a very promising, fast and reliable research direction rather than word processing, which is the main spotlighted track of some well-known research groups.

5 Conclusion

In this paper, we presented and discussed an Arabic historical document retrieval system without resorting to recognition of the contents in order to avoid the high cost and the difficult effort of the OCR. The old Arabic documents are characterized by a bad quality due to several factors. These factors together with characteristics of the Arabic writing complicate the processing of these documents at several levels. The basic idea of our approach consists in transposing the problem of document image indexing and retrieval from the field of document analysis to the field of information retrieval. The proposed system runs in two phases: a phase of processing and analysis which is performed off line, in which each document of our collection is represented by a signature made of the set of relevant information that we can extract from the document image. This description (signature) contains the morphological features of the writing (loops, ascenders, descenders, up and down diacritics) and it is coded in ASCII. When searching for a new query, the system proceeds in two stages: the first stage of search in the index table which is organized in a suffix tree in order to reduce the search time. In the second stage, the system runs an approximate string matching in the code files of the documents not returned in the first stage by combining two distance measures (Levenshtein edit distance and Jaro-Winkler distance). Several tests were performed in order to evaluate the performance of our system employing several exact and approximate search algorithms with varying parameters. For the approximate search, the adaptation of the allowed error *K* according to the query length shows to be an effective solution. This approach was tested on two databases: one containing 150 images of old Arabic documents and the other made up of 398 images of Algerian postal envelopes. The obtained results in terms of recall and precision are very promising if we consider the problems of the old documents and Arabic writing processing. Thus, the results show the feasibility and robustness of the employed approach as well as its generality so that it can be employed on other types of documents for which the application was not designed.

References

1. **Adamek, T., O'Connor, N.E. & Smeaton, A.F. (2007).** Word matching using single closed contours for indexing handwritten historical documents, *IJDAR*, 9, 153–161.
2. **Bai, S., Li, L. & Tam, C.L. (2009).** Keyword Spotting in Document Images through Word Shape Coding. *10th International Conference on Document Analysis and Recognition ICDAR*, Barcelona, Spain.
3. **Baird, H.S. (2004).** Difficult and Urgent Open Problems in Document Image Analysis for Libraries. *Third International Workshop on Document Image Analysis for Libraries DIAL.*
4. **Boyer, R.S & Moore, J.S. (1977).** A fast string searching algorithm. *Communications of the ACM*, 20(10), 762–772.
5. **Camillerapp, J., Pasquer, L. & Coüasnon, B. (2004).** Indexation automatique de formulaires anciens par reconnaissance du patronyme manuscrit. *Reconnaissance des Formes et Intelligence Artificielle RFIA*, Toulouse, France, 1493–1502.
6. **Chen, F. & Bloomberg, D. (1998).** Summarization of imaged documents without OCR. *Computer Vision and Image understanding*, 70(3).
7. **Kefali, A., Sari, T. & Sellami, M, (2009).** Implémentation de plusieurs techniques de seuillage d'images de documents arabes anciens, *5th International Symposium Images Multimédias Applications Graphiques et Environnements IMAGE*, Biskra, Algeria, 123–134.
8. **Khurshid, K., Faure, C. & Vincent, N. (2008).** Recherche de mots dans des images de documents par appariement de caractères. *10th Colloque International Francophone sur l'Écrit ET le Document CIFED.*
9. **Khurshid, K., Siddiqi, A., Faure, C. & Vincent, N. (2009).** Comparison of Niblack inspired Binarization methods for ancient documents. *16th Document Recognition and Retrieval Conference DRR*, USA.
10. **Knuth, D.E., Morris, J.H. & Pratt, V.R. (1974).** *Fast pattern matching in strings*. TR CS-74-440, Stanford University, ford, California.
11. **Leedham, G., Varma, S., Patankar A. & Govindaraju, V. (2002).** Separating Text and Background in Degraded Documents Images – A Comparison of Global Thresholding Techniques *for Multi-Stage Thresholding, Proc. Eighth IWFHR*, Niagara-on-the-Lake, 244–249.
12. **Mahmoud, A.S. (1994).** Arabic Character Recognition Using Fourier Descriptors and Character Contour Encoding. *Pattern Recognition*, 27(6), 815–824.
13. **Manmatha, R., Han, C. & Risemen, E. (1996).** Word spotting: a new approach to indexing handwriting. *IEEE Conference on Computer Vision and Pattern Recognition CVPR* 96, 631–637, 1996.
14. **McCreight, E.M. (1976).** A Space-Economical Suffix Tree Construction Algorithm. *Journal ACM*, 23(2), 262–272.
15. **Mitra, M. & Chaudhuri, B.B. (2000).** Information Retrieval from Documents: A Survey. *Information Retrieval*, Kluwer Academic Publishers, 2, 141–163.
16. **Navarro, G. (2001).** A guided tour to approximate string matching. *ACM Computing Surveys*, 33(1), 31-88.
17. **Plamondon, R. & Srihari, S.N. (2000).** On-line and off-line handwriting recognition: A comprehensive survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1), 63–84.
18. **Pujari, A.K., Naidu, C.D. & Jinaga, B.C. (2002).** An adaptive character recogniser for Telugu scripts using multiresolution analysis and associative memory. *3rd Indian Conference on Computer Vision, Graphics and Image Processing ICVGIP*, Ahmadabad, India.
19. **Ramel, J.Y. (2007).** User driven page layout analysis of historical printed books. *International Journal of Document Analysis and Recognition IJDAR*, 05-21.
20. **Rath, T.M. & Manmatha, R. (2003).** Features for Word Spotting in Historical Manuscripts. *Seventh International Conference on Document Analysis and Recognition ICDAR.*
21. **Rath, T.M. & Manmatha, R. (2007).** Word Spotting for historical documents. *International Journal of Document Analysis and Recognition,* 9, pp. 139–152.
22. **Sari, T. & Sellami, M. (2007).** State of the art of Off-line Arabic Handwriting Segmentation, *International Journal of Computer Processing of Oriental Languages.*
23. **Sari, T. & Kefali, A. (2008).** A search engine for Arabic documents. Proc. *10th Colloque International Francophone sur l'Écrit et le Document CIFED*, Rouen, France, 97–102.
24. **Smeaton, A.F. & Spitz, A. (1997).** Using character shape coding for information retrieval, *Fourth*

International Conference on Document Analysis and Recognition 97, IEEE Computer Society Press, 974–978.

25. **Spitz, A. (1995).** Using character shape codes for word spotting in document images. Dori, D. & Bruckstein A. (Eds.), Shape*, Structure and Pattern Recognition, World Scientific* 95, Singapore, 382–389.
26. **Ukkonen, E. (1985).** Finding approximate patterns in strings. *Journal of Algorithms*, 6, 132–137.
27. **Weiner, P. (1973).** Linear pattern matching algorithm. 14th *IEEE Symposium on Switching and Automata Theory* 1973, 1–11.
28. **Winkler, W.E. (1999).** *The state of record linkage and current research problems*. Technical report, Statistics of Income Division, Internal Revenue Service Publication R99/04.

Toufik Sari received his MSc. and Ph.D. degrees in Computer Science from Badji Mokhtar – Annaba University in 2000 and 2007, respectively. He is currently an associate professor of the Department of Computer Science and Information Engineering at Badji Mokhtar – Annaba University, Algeria. He is currently involved in many international projects (PURAQ, AIDA-EuroMed) concerned with Ancient document processing. His research interests include handwriting analysis and recognition, document retrieval and neural networks.

Abderrahmane Kefali was born in Drean-El Tarf, Algria. He obtained his MSc degree in Computer Science from Badji-Mokhtar University, Annaba, Algeria, in 2010. He is enrolled in a PhD in information technology on Arabic handwriting analysis and recognition. He is currently involved in many international projects (PURAQ, AIDA-EuroMed). His principal interests include document analysis and retrieval, information extraction, image mining and Arabic handwriting processing.

Article received on 11/15/2010; accepted 05/06/2011.