Journal of Electronic Imaging

JElectronicImaging.org

Mental model for handwritten keyword spotting

Youcef Brik
Djemel Ziou

Youcef Brik, Djemel Ziou, "Mental model for handwritten keyword spotting," *J. Electron. Imaging* **27**(5), 053027 (2018), doi: 10.1117/1.JEI.27.5.053027.

Journal of Electronic Imaging 27(5), 053027 (Sep/Oct 2018)

Mental model for handwritten keyword spotting

Youcef Brik[a,b,c,*] **and Djemel Ziou**[b]
[a]Université des Sciences et de la Technologie Houari Boumediene, Faculté d'Electronique et d'Iinformatique, Bab Ezzouar, Algeria
[b]Université de Sherbrooke, MOIVRE Laboratory, Sherbrooke, Canada
[c]Université Mohamed Boudiaf de M'sila, Msila, Algeria

Abstract. Most of existing approaches in keyword spotting are system-oriented, which did not take into consideration the user's needs. However, a user may want to find words, sentences, or texts that match his target image in his mind. The challenge here is how to formulate one's mental image to reach what he is looking for. The key idea is to design and build a model that properly adapts the human reasoning in information searching through an interactive process. We propose a mental model for handwritten keyword spotting based on relevance feedback, feature weighting, and optimization. This model meets simultaneously the user's needs, the system behavior, and the user–system relationship. In an appropriate feature space, the query is progressively built from user-supplied keywords, old queries, and spotted images. This dynamic process not only converges toward the desired word images, but also helps the hesitant user to clarify progressively what he is looking for. The proposed model was showcased via a user-friendly interface, which we tested including real users on three well-known handwritten datasets; Institute for Communications, Braunschweig University, Germany/École Nationale d'Ingénieurs de Tunis, Tunisia, Institut für informatik und Angewandte Mathematik, and George Washington. The experimental results show that the proposed method provides promising scores with a reasonable number of refinements. [DOI: 10.1117/1.JEI.27.5.053027]

Keywords: information retrieval; keyword spotting; mental model; handwritten documents; relevance feedback; feature weighting; optimization.

Paper 180406 received May 23, 2018; accepted for publication Sep. 13, 2018; published online Oct. 4, 2018.

1 Introduction

A large number of handwritten documents stored in libraries, which have important values, should be accessible online to every interested user. These documents include books, historical manuscripts, scribes, architectural plans, and religious and civil leaflets. For protecting them from various deterioration factors and making them amenable to browsing, a subset of these documents is usually digitized forming collections of electronic images. However, it is necessary to build adequate systems for automatic indexing and online access of their content.[1] Among the strategies implemented to build the useful system, the handwritten keyword spotting (HKWS) has been proposed for efficient browsing or searching a specific content.[2,3] A recent survey of this domain can be found in Ref. 4.

HKWS is part of content-based information retrieval (CBIR) and it has become a very active research area since 1996.[1,3,5] The main idea of HKWS is based on the retrieval of all relevant images containing similar content as the input query. Basically, the images in a collection are ranked according to their similarity to the query images.

This large number of digitized documents continues to increase rapidly, which renders it necessary to develop more efficient systems for word or sentence searching.[6,7] Thus it leads us to design a powerful HKWS model capable of serving all types of user's requirements taking into account the interaction between the user and the system. First, from the user perspective, an HKWS system must be precise and fast in its responses to a query. In addition, it must offer mechanisms to facilitate the formulation of the query, to display images similar to the query and to refine the retrieved images. Furthermore, it must provide access to similar images or refine the query according to what the user is looking for. Second, from the designer point-of-view, the user requirements guide the choice of strategies to implement them for characterizing the images well and indexing them using interaction mechanisms.[8]

Most existing approaches in HKWS are system oriented, which did not take into consideration the user's needs.[9,10] However, from the user side, the research paradigm is slightly different. A user may have a mental image about what he is looking for, but the images provided by HKSW system will not satisfy him.[7] He, then, performs several refinements until satisfaction using images provided by the HKWS system. During this interaction, the user builds a reasoning leading to clarify and meet his needs to find the relevant images. For example, at any time, the query can be influenced by the previous queries as well as the previous ranking outputs. Hence, the search in documents is governed by a dynamic process called "mental model." The mental model can be defined as a dynamic process built and adapted by the means of the user/system interaction for retrieving documents that better match the explicit or implicit user needs.[7]

The challenge we face, in this work, is the implementation of this mental model as a dynamic process in HKWS task. The aims of this mental model are to: (1) construct a dynamic reasoning about the system's input/output, (2) anticipate the behavior of both user/system for helping the user reach what he is looking for, (3) elucidate why the system reacts as it does, and (4) emulate the actual reasoning by constructing an analytic form of the query using system outputs and

*Address all correspondence to: Youcef Brik, E-mail: youcef.brik@usherbrooke.ca

1017-9909/2018/$25.00

previous queries. In this way, our model can be adapted according to the user's reasoning.

This paper is organized as follows: Section 2 discusses the related work in the areas of word spotting and mental model frameworks. Section 3 explains the proposed model and its formulation. Section 4 introduces the datasets used in this experiment, clarifies the experimental protocol, and shows the obtained scores. Finally, the conclusions drawn from this work are in Sec. 5.

2 Related Works

HKWS is a very active domain in information retrieval area.[4,11] It consists of retrieving all relevant images containing keyword images similar to the ones specified by the query. Overall, HKWS has been considered under two distinct paradigms: query-by-string[11–14] and query-by-example.[3,5,9,15–17] The first paradigm usually consists of introducing the query image in a textual representation to retrieve the similar images. Conversely, the latter consists of finding candidate images, which are similar to a given image, i.e., the system ranks all candidate images in descending order of resemblance to the query.

The query-by-example (QbE) techniques are the most popular in HKWS systems because they do not require a large labeled data and the query can be constructed from one or a few images.[4,10] Nevertheless, the performances of QbE techniques suffer from two major shortcomings. First, the query is not always available or it can take a lot of forms with particular intraclass variability.[18] Second, the user may not have a precise idea about what he is looking for, which causes a bad selection of queries. In order to overcome these shortcomings, we need to build a dynamic model that produce more accurate output images according to the user's mental target. Hence, mental models present useful mechanisms for meeting the user's requirements and understanding the user/system interaction using psychovisual aspects and sources of variability in the decision making.[7,19]

The mental model, which is originally from psychology,[20] has been described as organized knowledge frameworks constructed in his mind. This latter uses them to anticipate behaviors, to reason about system, and to underlie an explanation.[21,22] Also, Jacobson[23] described mental models as held images of thinking and acting. Translated to information retrieval, the mental model can be seen as a dynamic process built and adapted by the means of the user/system interaction for retrieving documents that better match the user needs.[7] In CBIR context, there are few studies that have adopted mental model-based systems. Cox et al.[24] have proposed a mental matching framework, called "PicHunter," which aims to retrieve a mental image that the user is looking for without introducing an explicit query. PicHunter predicts the target the users want, using a general Bayesian framework. Fang and Geman[19] have proposed a relevance feedback system to find mental face image from a large dataset. The user, at each iteration, can select the closest face picture, which matches his target, among several displayed faces. The authors of this work used a Bayesian framework to select which pictures to display and to model the user's response. Another statistical framework proposed in Ref. 25 based also on relevance feedback. The objective of this work is to find the mental picture in an unstructured image dataset with no semantic annotation. The search process was initiated from random images. At each round, the user could select one image among a set of displayed images, which is closer to his mental target to refine the search outputs. The number of iterations required to display relevant images that satisfy the user was considered to measure the model performance. More recently, Bdiri et al.[7] have extended the work of Ref. 25 by including the nonselection and multiselection options within a single mental retrieval process. The user can interact with the system to express his mental target using visual features. Generalized inverted Dirichlet mixture was used to model data and the search process gave encouraging results by reducing the number of iterations required to reach the mental target. Furthermore, several relevance feedback mechanisms have been implemented in CBIR context. The interested reader can find a more complete survey about them in Refs. 26–28.

However, we notice very little consideration concerning the mental models in HKWS systems. We found only four references that studied HKWS framework involving the user in the spotting loop either by the user's feedback or by the relevance feedback techniques. The first system for HKWS that uses user's feedback has been proposed by Konidaris et al.[16] The authors combined synthetic data and the user feedback to perform HKWS in ancient printed documents. To initiate the spotting process, the authors created synthetic images (i.e., the synthesis of the keyword images from their ASCII equivalent). The user could then select the correct keywords from the initial list. In this system, the critical role of the user feedback is the transition from synthetic to real data. Another HKWS system that used relevance feedback was proposed by Zagoris et al.[29] The proposed framework allowed the user to define a number of retrieved images as correct or wrong, then exploiting them with the first query to provide new training samples to the support vector machines (SVMs). Then new retrieval images are computed using the decision function of the trained SVMs. Recently, Wei et al.[30] proposed an HKWS system for historical Mongolian documents using data fusion method and pseudorelevance feedback. Finally, Zagoris et al.[31] have proposed a similar work to Ref. 30. The authors coupled HKWS with a relevance feedback mechanism to significantly reduce the cost of training data construction by enhancing the manual transcription procedure.

3 Proposed Model

As mentioned in the previous section, our goal is to define a spotting scenario where the user can select a few input images and their respective degrees of relevance at each iteration. This allows us, first, to get an idea about what the user initially thinks and how he reacts after each feedback, then to refine the ranked images by revising the query using the user's interaction. Our proposed model takes into account the evolution of the query in time depending on what the user thinks through his feedback by combining the initial words, the selected words from displayed results, and the previous query in a new query. Second, the proposed model aims to iteratively update the weight of feature components according to its relevance to the modified query. Finally, these two strategies (i.e., modified query and feature reweighting) are integrated in our system to make the

similarity measures more efficient and to perform retrieval in a reasonable number of iterations.

Now, let us explain, in more details, the model we propose. Initially, the user can choose word images that will participate in the query and give them their appropriate weight according to their resemblance to his preferences. Then the query is compared to all the candidate words using image matching in order to find all similar words in the data. At this level, if the user is satisfied with the obtained results, we can say that the "final objective" of the spotting is reached. Conversely, if the spotting results are not satisfied, we then move from a "temporary objective" to another, in order to reach the "final objective." During this moving process, the query can be iteratively refined by specifying more relevant images by the user and the feature weighting can be implicitly updated to improve the similarity matching.

3.1 *Formulation of the Proposed Model*

Before giving details on our proposed model formulation, we will begin by explaining the image model and the adopted similarity measures. Consider that we have a dataset consisting of N_{total} images. At every time (henceforth $t = 1,2,3,\ldots$ denotes time), each image will be represented by a feature vector $\overrightarrow{x_n}^{(t)} = [x_{n1}^{(t)},\ldots,x_{ni}^{(t)},\ldots,x_{nI}^{(t)}]$, where $x_{ni}^{(t)}$ is the i'th feature vector of the n'th image at time t. Also, the query will be represented by $\vec{q}^{(t)} = [q_1^{(t)},\ldots,q_i^{(t)},\ldots,q_I^{(t)}]$. However, these components of the feature do not have the same relevance with the query, which means that they do not capture the user's requests well. Feature selection can remedy this problem either by eliminating uninformative and redundant components or by giving them appropriate weights according to their relevance with the query. Consequently, we wish to iteratively enhance each component of the feature generated from images by maximizing those, which have much relevance with the query. At the same time, we reduce the importance of those that do not have much relevance with the query. This process can be done by transforming the original feature space into a new one that better corresponds with the the user's requirements. To realize this, we introduce an optimal vector weight $\vec{u}^{(t)}$ in the similarity measure between all candidate images and a query. Our similarity measure is then given as follows:

$$\text{Dist}[\overrightarrow{x_n}^{(t)},\vec{q}^{(t)}] = \sum_{i=1}^{I} d[u_i^{\xi^{(t)}} x_{ni}^{(t)}, u_i^{\xi^{(t)}} q_i^{(t)}], \tag{1}$$

where $u_i^{(t)}$ is a global scalar weight assigned to the i'th feature at time t and ξ is a parameter for attribute the weight $u_i^{(t)}$.

Several similarity measures can be used including Euclidean, Canberra, Kullback–Leibler, and Bhattacharyya.[4,32] In our case, we choose the Euclidean distance to perform one-to-one alignments between the components of the feature of the query and those of the candidate images. Equation (1) becomes:

$$\text{Dist}[\overrightarrow{x_n}^{(t)},\vec{q}^{(t)}] = \sum_{i=1}^{I} u_i^{\xi^{(t)}} [x_{ni}^{(t)} - q_i^{(t)}]^2. \tag{2}$$

After defining our similarity measure, now we want to reach two objectives that are: the estimation of the query $\vec{q}^{(t)}$ and the feature weights $\vec{u}^{(t)}$. At time t, the optimal query must fulfill the following requirements: (1) by assuming that there is a continuity in user mental target, when moving from $\vec{q}^{(t-1)}$ to $\vec{q}^{(t)}$, these two queries must share relevant features and they inevitably differ in some others. In other words, the query $\vec{q}^{(t)}$ is formulated using $\vec{q}^{(t-1)}$ such that a new information is constructed, which moves the spotting toward the relevant images and away from those irrelevant. (2) Till satisfaction, the user can iteratively refine the output word images by selecting some relevant images among the retrieved ones to formulate a new query. At this level, $\vec{q}^{(t)}$ will noticeably share relevant features with these word images. Thus the feature weights need to be updated by transforming the original features into a new feature space that better models the user's needs.

To translate the aforementioned objectives into a mathematical formulation, we consider that the user, at any time t, expresses the query $\vec{q}^{(t)}$ by a set $S^{(t)}$ of $N^{(t)}$ images and their relevance $P_n^{1^{(t)}}$ for $n = 1,\ldots,N^{(t)}$, as well as a set $R^{(t)}$ of $M^{(t)}$ images and their relevance $P_m^{2^{(t)}}$ for $m = 1,\ldots,M^{(t)}$. $S^{(t)}$ are the initial images that participate in the query and $R^{(t)}$ are the selected images among those retrieved in $t-1$. The relevance is the degree of resemblance to the sought word images given by the user in such a way that images with high degrees should have small distance from $\vec{q}^{(t)}$. The computation of the optimal parameters $\vec{u}^{(t)}$ and $\vec{q}^{(t)}$ is the trade-off between the similarity measures defined in Eq. (2) and the requirements mentioned above. Several combinations of them are possible. In our case, we formulate the objective function as follows:

$$\begin{aligned} J(u,q) = &\sum_{i=1}^{I} u_i^{\xi^{(t)}} \sum_{n=1}^{N} P_n^{1^{(t)}} [q_i^{(t)} - S_{ni}^{(t)}]^2 \\ &+ \sum_{i=1}^{I} u_i^{\xi^{(t)}} \sum_{m=1}^{M} P_m^{2^{(t)}} [q_i^{(t)} - R_{mi}^{(t)}]^2 \\ &+ \sum_{i=1}^{I} u_i^{\xi^{(t)}} [q_i^{(t)} - q_i^{(t-1)}]^2, \end{aligned} \tag{3}$$

where $S_{ni}^{(t)}$ is the i'th feature vector of the n'th introduced word image S_n at iteration t. $R_{mi}^{(t)}$ denotes the i'th feature vector of the m'th retrieved image R_m that participates in creating the new query keyword at time t, and $q_i^{(t-1)}$ is the i'th feature vector of the query at $t-1$, i.e., the previous iteration.

Subject to the constraint:

$$\sum_{i=1}^{I} u_i^{(t)} = 1, \qquad 0 \le u_i^{(t)} \le 1. \tag{4}$$

Here, it is worth noting that the characterization space of the word images has a finite number of features. The weighting scheme of the features is based on the assumption that, at any time, each word image shares with the query at least one relevant features and it inevitably differs in others. If all features are not relevant, then the weight of at least one features will be high. In this case, the spotting output will be noise. However, because the features are real variables, then the probability that this situation happen is too low. Based on this, the proposed model assigns to each feature a weight to express its relevance to the corresponding feature in the query. This can be a number between 0 and 1. The relevance

value of any feature component must be expressed in the relativity to the others. Hence, all features share between them the all. To address this fact, the relevance values of all features must necessarily verify the constraint given in Eq. (4), where the all in our case is 1. Then the optimization problem can be solved using the Lagrange multipliers to reduce the constrained problem in Eqs. (3) and (4) to an unconstrained one. Let λ be the multiplier and L be the Lagrangian:

$$L = 3\sum_{i=1}^{I} u_i^{\xi(t)} q_i^{(t)^2} - 2\sum_{i=1}^{I} u_i^{\xi(t)} q_i^{(t)} \Phi_i^{(t)} + \sum_{i=1}^{I} u_i^{\xi(t)} \Psi_i^{(t)} - \lambda\left[1 - \sum_{i=1}^{I} u_i^{(t)}\right], \tag{5}$$

where

$$\Phi_i^{(t)} = \sum_{n=1}^{N} P_n^{1(t)} S_{ni}^{(t)} + \sum_{m=1}^{M} P_m^{2(t)} R_{mi}^{(t)} + q_i^{(t-1)}, \tag{6}$$

and

$$\Psi_i^{(t)} = \sum_{n=1}^{N} P_n^{1(t)} S_{ni}^{(t)^2} + \sum_{m=1}^{M} P_m^{2(t)} R_{mi}^{(t)^2} + q_i^{(t-1)^2}. \tag{7}$$

The minimization of L is decoupled by solving $q_i^{(t)}$ first and then $u_i^{(t)}$, so that the gradient in both variable sets must be equal to zero.

3.2 *Optimal Solution of $q_i^{(t)}$*

To find the optimal solution for $q_i^{(t)}$ at each iteration, we should calculate the partial derivative of L with respect to $q_i^{(t)}$. Straightforward mathematical manipulations allow to write:

$$\frac{\partial L}{\partial q_i^{(t)}} = 6u_i^{\xi(t)} q_i^{(t)} - 2u_i^{\xi(t)} \Phi_i^{(t)}. \tag{8}$$

By setting Eq. (8) to zero, we obtain the final solution to $q_i^{(t)}$:

$$q_i^{(t)} = \frac{1}{3}\Phi_i^{(t)}. \tag{9}$$

We can see that, at every iteration, the optimal query keyword $q^{(t)}$ depends on the average of the initial word images $S^{(t)}$, the selected images among those retrieved by the previous query $R^{(t)}$, and the previous query $q^{(t-1)}$. The $S^{(t)}$ and $R^{(t)}$ sets are multiplied by their degrees of relevance introduced by the user $P^{1(t)}$ and $P^{2(t)}$, respectively. In this way, the optimal query shares the relevance of $S^{(t)}$, $R^{(t)}$, and $q^{(t-1)}$ so that it should be moved toward what the user wants to and away from what he does not want to.

3.3 *Optimal Solution of $u_i^{(t)}$*

To find the optimal solution for $u_i^{(t)}$, we should take the partial derivative of L with respect to $u_i^{(t)}$. After some mathematical manipulations, we obtain:

$$\frac{\partial L}{\partial u_i^{(t)}} = 3\xi u_i^{\xi-1(t)} q_i^{(t)^2} - 2\xi u_i^{\xi-1(t)} q_i^{(t)} \Phi_i^{(t)} + \xi u_i^{\xi-1(t)} \Psi_i^{(t)} - \lambda. \tag{10}$$

After substituting Eq. (9) into Eq. (10) and setting it to zero, we get

$$u_i^{(t)} = \left\{\frac{\lambda}{\xi\left[\Psi_i^{(t)} - \frac{1}{3}\Phi_i^{(t)^2}\right]}\right\}^{\frac{1}{\xi-1}}. \tag{11}$$

Since $\sum_{j=1}^{I} u_j^{(t)} = 1$, we obtain

$$\sum_{j=1}^{I}\left\{\frac{\lambda}{\xi\left[\Psi_j^{(t)} - \frac{1}{3}\Phi_j^{(t)^2}\right]}\right\}^{\frac{1}{\xi-1}} = 1, \tag{12}$$

which leads to

$$\lambda = \left[\left(\sum_{j=1}^{I}\left\{\frac{1}{\xi\left[\Psi_j^{(t)} - \frac{1}{3}\Phi_j^{(t)^2}\right]}\right\}^{\frac{1}{\xi-1}}\right)^{\xi-1}\right]^{-1}. \tag{13}$$

Substituting Eq. (13) into Eq. (11), we obtain the proper weight $u_i^{(t)}$ assigned to the i'th feature component as follows:

$$u_i^{(t)} = \left\{\sum_{j=1}^{I}\left[\frac{\Psi_i^{(t)} - \frac{1}{3}\Phi_i^{(t)^2}}{\Psi_j^{(t)} - \frac{1}{3}\Phi_j^{(t)^2}}\right]^{\frac{1}{\xi-1}}\right\}^{-1}. \tag{14}$$

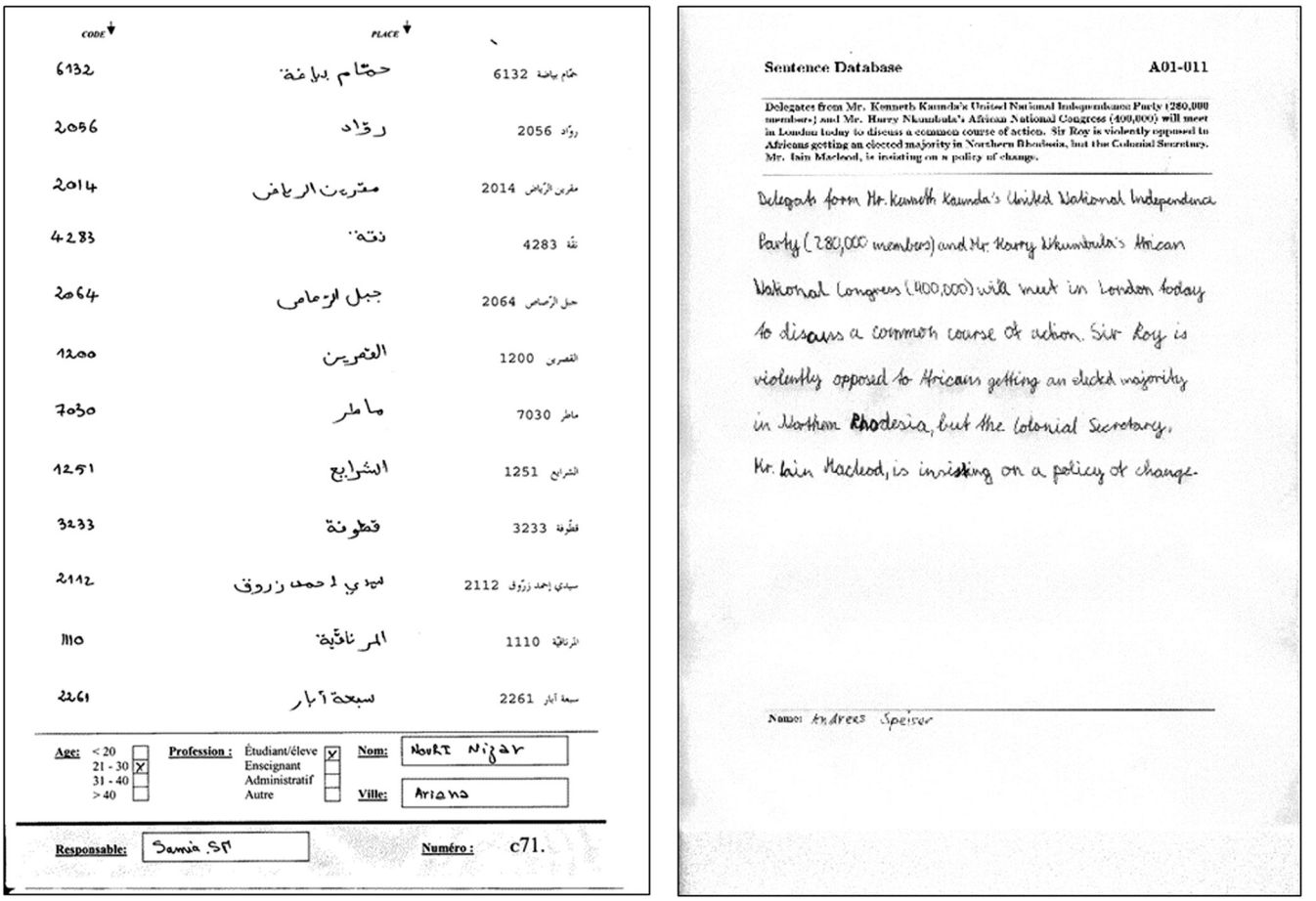
By considering Eqs. (6) and (7), we can see that the numerator of Eq. (14), i.e., $[\Psi_i^{(t)} - \frac{1}{3}\Phi_i^{(t)^2}]$, describes the intradispersion of the i'th feature and the denominator, i.e., $[\Psi_j^{(t)} - \frac{1}{3}\Phi_j^{(t)^2}]$, describes the inter dispersion of all features, which is constant for all $j \in I$. Consequently and for a given ξ, if the intradispersion of the i'th feature is relatively important to the inter dispersion, this feature should receive a low weight. With the same reasoning, if the intradispersion of the i'th feature is relatively low to the inter dispersion, this component of the feature should receive a high weight. The aforesaid behavior of $u_i^{(t)}$ fulfills our objective for selecting relevant features against others by giving them much importance.

After computing the optimal parameters $q_i^{(t)}$ and $u_i^{(t)}$, it is worth noting that by ignoring the time (i.e., the dynamic process) in the spotting system, our model becomes similar to the work that was proposed in Refs. 6 and 33. The feature weights in those two models were updated without considering the user's thoughts before.

4 Experiments and Results

4.1 *Dataset*

The experiments of this work are carried out on three different real datasets to assess the performance of the proposed model. The first one is the well-known Institute for Communications, Braunschweig University, Germany/École Nationale d'Ingénieurs de Tunis, Tunisia (IFN/ENIT) collection.[34] This dataset characterizes Arabic script in the form of handwritten Tunisian town names written by 411 different writers, making a total of more than 26,000 binary word images divided into five subsets (from *a* to *e*). IFN/ENIT has been used as a benchmark for several handwritten recognition systems and recently word spotting frameworks. An example of the IFN/ENIT forms is



(a) (b)

Fig. 1 Sample form images for (a) IFN/ENIT dataset and (b) IAM dataset.

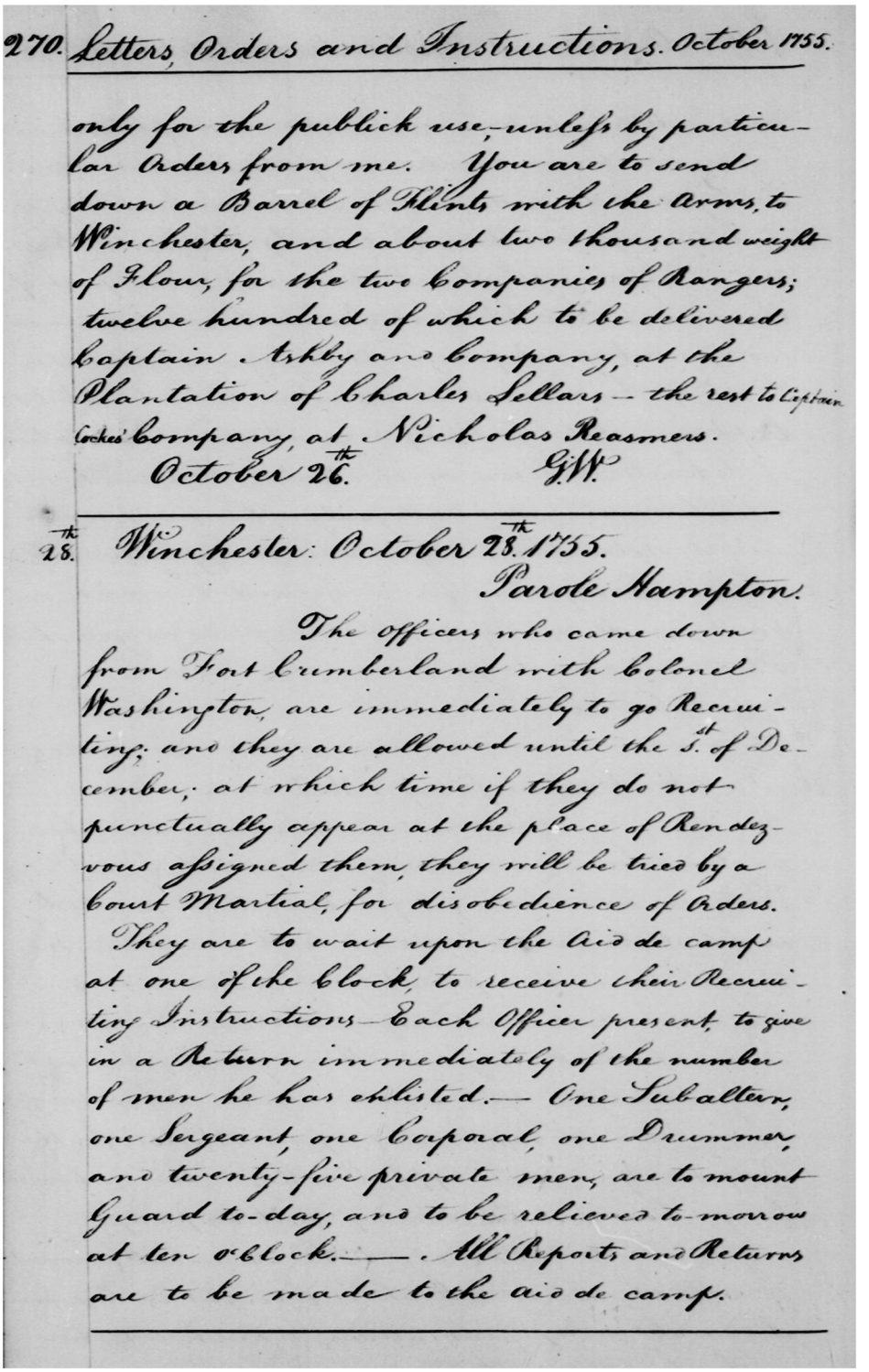
presented in Fig. 1(a). The second dataset is the Institut für informatik und Angewandte Mathematik (IAM) handwritten dataset.[35] It is probably the most widely used dataset for English script in both handwritten recognition and word spotting. IAM handwritten dataset comprises 1539 pages with more than 115,000 word images written by 675 different writers. Figure 1(b) shows an example of IAM forms. In addition, an official partition has been considered to evaluate handwritten frameworks such as word recognition,[36] writer identification,[37] gender classification,[38] and word spotting.[4] It contains three different sets (training, validation, and testing).

Finally and to verify that the proposed model is also applicable to the historical script, we perform the same experiments on the well-known George Washington (GW) dataset.[3] It consists of 20 pages of letters from GW and his assistants dating back to 1755, which contain more than 4900 word images. An example of the GW correspondences is shown in Fig. 2.

4.2 *Feature Generation*

In order to properly characterize the word images, we use the Curvelet transform to generate our features. The Curvelet transform is an advanced generalization of the Wavelet transform to represent an image at different scale levels and various directions.[39] Unlike wavelet transform, curvelet transform offers an enhanced directional capacity to characterize edges and singularities not only across curves but also along them.[40] For generating the coefficients of curvelet transform, we use the fast discrete curvelet transform (FDCT) via wedge wrapping[41] with two different strategies, global and local. The first one consists of applying this transform on the whole word images and then considering only the finest and the coarsest levels. We find that the obtained curvelet coefficients are large. Hence, for an efficient word characterization, it is necessary to reduce the feature dimensionality. In our work, we apply the local phase quantization (LPQ)[42] on the gray level finest and coarsest images, where the quantized coefficients of LPQ will be integers between 0 and 255. The histogram of this coefficients (a vector of 256 features) is estimated from each level (finest and coarsest) describing the number of occurrence of each integer value for all pixel positions. We recall (RC) here that the sliding rectangular neighborhood used in the implementation of LPQ at each pixel position is of size 3×3. A more detailed description for the implementation of LPQ can be found in Ref. 43.

While to store the local writing features, the local method consists of dividing the word images of each dataset into $\eta_x \times \eta_y$ cells of the same size, where η_x is the number of rows and η_y the number of columns. Then we apply the FDCT on each cell and we consider all the subbands of


270. Letters, Orders and Instructions. October 1755.

only for the publick use, unless by particular Orders from me. You are to send down a Barrel of Flints with the Arms, to Winchester, and about two thousand weight of Flour, for the two Companies of Rangers; twelve hundred of which to be delivered Captain Ashby and Company, at the Plantation of Charles Sellars — the rest to Captain Cocke's Company, at Nicholas Reasmers.

October 26th GW.

28th Winchester: October 28th 1755.

Parole Hampton.

The officers who came down from Fort Cumberland with Colonel Washington, are immediately to go Recruiting; and they are allowed until the 5th of December; at which time if they do not punctually appear at the place of Rendezvous assigned them, they will be tried by a Court Martial, for disobedience of Orders.

They are to wait upon the Aid de camp at one of the clock, to receive their Recruiting Instructions — Each Officer present, to give in a Return immediately of the number of men he has enlisted. — One Subaltern, one Sergeant, one Corporal, one Drummer, and twenty-five private men, are to mount Guard to-day, and to be relieved to-morrow at ten o'clock. — All Reports and Returns are to be made to the Aid de camp.

Fig. 2 Sample image from GW letters.

the transform except the finest and the coarsest. To reduce the dimensionality of the obtained coefficients, the standard deviation (Std) and the entropy (Ent) of each subband in each cell are computed to generate the local features. Experimentally, we found that the best regular grid size applied on each word image is ($\eta_x = 1$, $\eta_y = 14$) for the IFN/ENIT dataset, ($\eta_x = 1$, $\eta_y = 8$) for the IAM dataset, and ($\eta_x = 1$, $\eta_y = 10$) for the GW dataset. Each cell is then decomposed into four scale levels using the FDCT leading with 50 ($1 + 16 + 32 + 1$) subbands. By ignoring the finest and the coarsest levels and discarding the half of the subbands at scales 2 and 3 due to the symmetry property of the FDCT (i.e., the curvelet coefficients produced at angle θ are the same as those produced at angle $\theta + \pi$), each cell is, therefore, represented by 48 features.

Consequently, the final feature vector of each word image is constructed by concatenating each feature from the global and local methods. The final length of the feature vectors is

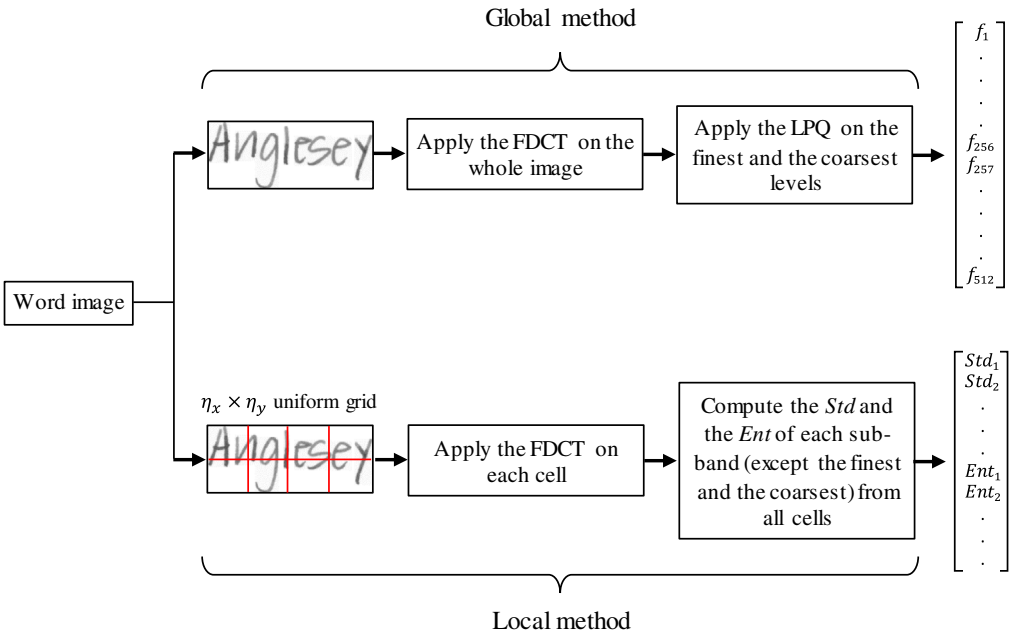


Fig. 3 Feature generation pipeline.

$48 \times 14 + 512 = 1184$, $48 \times 8 + 512 = 896$, and $48 \times 10 + 512 = 992$ for the IFN/ENIT, IAM, and GW word images, respectively. Figure 3 shows the overall pipeline of the proposed feature generation (in 4 scales and 16 orientations).

4.3 *Performance Measures*

Usually, the performances of the spotting methods are summarized by two performance measures namely RC and precision (PR).[4] However, the RC has been considered less meaningful in image retrieval since it is often low.[6,44] Furthermore, since the PR is calculated for the whole retrieved images, it is unaffected by the respective rankings of relevant images in the retrieved list.[45]

To address the aforesaid limitations, the mean average precision (MAP) has been considered as the main metric used to evaluate CBIR systems as well as word spotting. This metric has been proposed by the National Institute of Standards and Technology and the Text Retrieval Conference Community. The MAP is calculated as the mean of the average precision (AveP) obtained by each submitted query to the system. This measure takes values in a rang of (0 to 1), where 0 means no correctly retrieved images and 1 means that all images are correctly retrieved. Before giving the AveP equation, we first consider another measure called the PR at top k retrieved word images ($P@k$). This measure defines how successfully the system produces relevant results to the first k positions of the ranking list.[31] $P@k$ is defined as follows:

$$P@k = \frac{|\{\text{relevant word images}\} \cap \{k \text{ retrieved word images}\}|}{|\{k \text{ retrieved word images}\}|}. \tag{15}$$

While the AveP is given by

$$\text{AveP} = \frac{\sum_{k=1}^{n} [P@k \times \text{rel}(k)]}{|\{\text{relevant word images}\}|}, \tag{16}$$

where n designs the number of retrieved word images and the $\text{rel}(k)$ is defined as

$$\text{rel}(k) = \begin{cases} 1, & \text{if word at rank } k \text{ is relevant} \\ 0, & \text{if word at rank } k \text{ is not relevant} \end{cases}. \tag{17}$$

Finally, the MAP for a set of query words Q is the mean of the AveP scores for each query word:

$$\text{MAP} = \frac{1}{|Q|} \sum_{q=1}^{|Q|} \text{AveP}(q). \tag{18}$$

4.4 *Baseline Systems*

We remind here that there were no methods before that used a mental model in searching the handwritten/historical documents. For this reason, we consider two scenarios: the first one is based on the famous Rocchio equation[46] and the second consists of using the dynamic time warping (DTW) algorithm[3] in the matching process. We remind also that neither the Rocchio equation nor DTW algorithm is a concept that simulates the human reasoning in the image retrieval. Therefore, we adapt the first algorithm to the human reasoning in the spotting procedure and take DTW as a reference.

4.4.1 *Scenario 1*

Rocchio's algorithm is a well-known retrieval system that implements the user's feedback. It consists of a query-point movement equation to assign much importance to images that are close to the user's preferences. The mathematical definition of Rocchio's equation is given:

$$q_m = \alpha q_0 + \beta \frac{1}{|D_r|} \sum_{d_j \in D_r}^{I} d_j - \gamma \frac{1}{|D_{nr}|} \sum_{d_j \in D_{nr}}^{I} d_j, \tag{19}$$

where q_m and q_0 are the updated (i.e., modified) and the initial query, D_r and D_{nr} are the relevant and the nonrelevant images, and α, β, and γ are the weights assigned to q_0, D_r, and D_{nr}, respectively. We adapt this algorithm to the human concept and reasoning in the spotting procedure by setting, and its parameters are set by trial and errors. Furthermore, the Euclidean distance [i.e., Eq. (20)] is used to evaluate the Rocchio algorithm:

$$D[\overrightarrow{x_n}^{(t)}, \vec{q}^{(t)}] = \sum_{i=1}^{I} [x_{ni}^{(t)} - q_i^{(t)}]^2. \tag{20}$$

4.4.2 *Scenario 2*

We adopt DTW in the spotting process. In fact, DTW is the most popular matching algorithm that computes efficiently the similarity between two vector sequences, and it can also better adapt to the different variations of written style and word deformation.[47] Below, we briefly review the DTW algorithm and explain how it works. Let us consider the two vector sequences $\overrightarrow{x_n}$ and $\vec{q}$ defined in Eq. (1). DTW aligns these two sequences so that their difference is minimized. Unlike the traditional Minkowski distances, DTW breaks the drawback of one-to-one alignment and absorbs the temporal deformation between the two sequences. To this end, we first build the matrix D, where each element $D(i,j)$ is computed by the following recurrence:

$$D(i,j) = d_e(x_{ni}, q_j) + \min\{D(i-1,j), D(i,j-1), D(i-1,j-1)\}, \tag{21}$$

where $(0 < i \leq I, 0 < j \leq I)$.

Finally, the optimal warping path that minimizes the distance between the two vector sequences $\overrightarrow{x_n}$ and $\vec{q}$ is defined as

$$\mathrm{sim}_{\mathrm{DTW}}(\overrightarrow{x_n}, \vec{q}) = \frac{D(I,I)}{L}, \tag{22}$$

where $I \leq L \leq 2I-1$ is the length of the optimal warping path. The DTW algorithm is performed by matching the query with all candidate images, where the query q_x is selected from the training set using the following equation:

$$q_x = \arg \min_{x_n \in N_x} \sum_{g=1}^{N_x} \mathrm{sim}_{\mathrm{DTW}}(\overrightarrow{x_n}, \overrightarrow{x_g}), \tag{23}$$

where N_x is the number of training samples for a given keyword x.

4.5 *Experimental Protocol*

A good search system based on mental model is usually one that satisfies the user in a reasonable number of refinements. The performance is then evaluated according to the number of iterations. Also, we experimentally found that with more than two successive iterations with a change <1% on MAP, the system does not bring any noticeable enhancement in the spotting results. We remind that our model does not require neither training phase nor large amount of labeled data, only a very limited word images needed to start the system (generally from one to ten samples for each query). To perform our experiments, we use the following protocol:

Our model with IFN/ENIT dataset. We have taken the same protocol used in Ref. 10, where a set of 50 most popular keywords selected from set a to be spotted in set b (6714 word images). The 10 initial word images for each query are randomly selected from set a to evaluate our model. Figure 4 shows the 10 selected samples for the village name with ZIP code 6060.

Our model with IAM dataset. Concerning IAM handwritten dataset, we use the testing set of the official partition available for writer-independent text line recognition where 1861 lines are automatically segmented at more than 8800 isolated word images, whereas the initial word images for each query keyword are randomly selected from the training set, but excluding the stop words such as a, an, the, and from. Figure 5 shows a selection of samples of the query "government" from IAM handwritten dataset.

Our model with GW dataset. We follow the same protocol that was adopted by Refs. 12 and 36, where the GW dataset was divided into two sets containing 2/3 and 1/3 of the words. The 10 initial word images for each query are

Fig. 4 The 10 initial word samples for the ZIP code 6060.

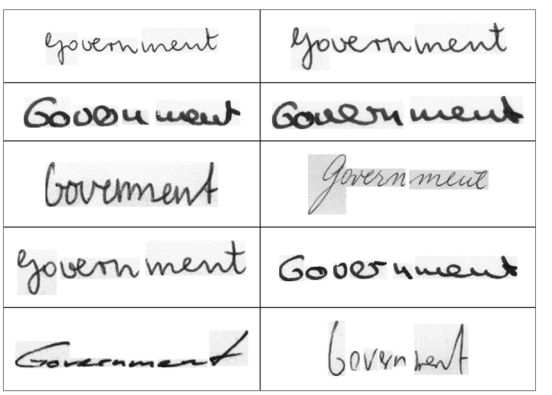


Fig. 5 The 10 initial samples of the word "government" from IAM dataset.

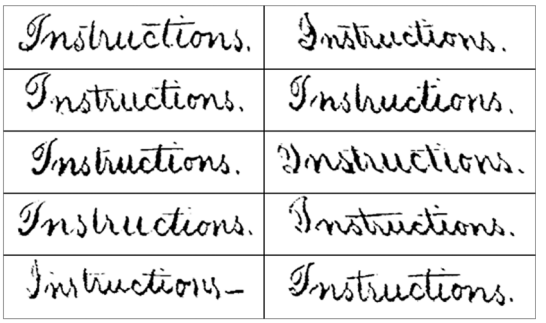


Fig. 6 The 10 initial samples of the word "instructions" from GW dataset.

Table 1 Number of word images used in the spotting evaluation.

	IFN/ENIT	IAM	GW
# Query words	50	30	25
# Initial word images	From 1 to 10 instances for each query		
# Testing word images	6714	8807	1632

randomly selected from the first set, and the second set is used for testing. Figure 6 shows selected samples of the query "instructions" from the GW dataset. Table 1 summarizes the statistics about the total number of word images used in the evaluation task.

4.6 *Results*

Let us remind that the effectiveness of our method is evaluated on IFN/ENIT, IAM, and GW datasets, respectively. Three people have participated in all of the experiments through a user-friendly interface. The MAP and $P@k$ are computed to assess the performance of the proposed method. Therefore, we carry out three experiments, each of which evaluates a given aspect of the proposed method. The first experiment aims to study the effect of the parameter ξ in Eq. (14) on the overall performance of our system and to show how the feature weight u_i behaves. The second experiment is designed to measure the variation of the performance (i.e., the accelerating convergence in the system) as a function of the number of initial word images participated in the query. The last experiment reveals the combination of our model with DTW and its impact on the overall performance.

4.6.1 *First experiment*

In the first experiment, we perform a set of tests on the weight u_i in Eq. (9) with different ξ values ranging from -5 to 5. We note that the case of $\xi = 0$ corresponds to the nonselection of features (i.e., $u_i^{\xi} = u_i^0 = 1$) and the case of $\xi = 1$ is a hard selection of features (i.e., assigning 1 to the most relevant feature and 0 to the rest). In order to formulate the query in this experiment, we initially take $|S| = |R| = 5$ samples.

Figure 7 shows the MAP obtained with different values of the parameter ξ on the IFN/ENIT dataset. We reached the highest value of MAP at $\xi = 3$ with 69.7%. The negative

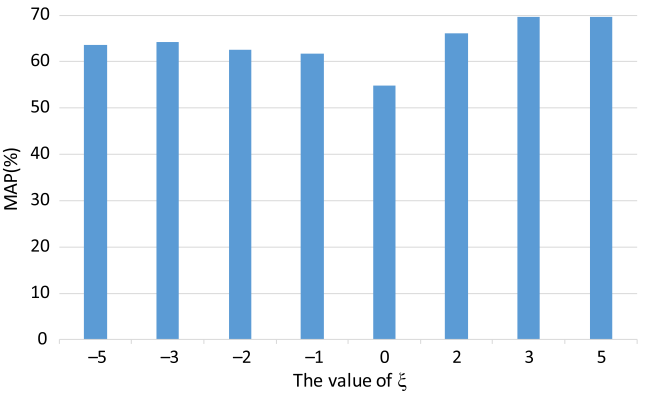


Fig. 7 The spotting scores produced with various ξ values on IFN/ENIT dataset.

Table 2 Word spotting comparison with the baseline systems on IFN/ENIT dataset.

Method	Performance		
	$P@5$	$P@10$	MAP
Rocchio baseline	30.45	47.89	38.24
DTW baseline	56.30	58.02	49.76
Our model without feature weighting ($\xi = 0$)	63.40	67.52	54.83
Our model with feature weighting ($\xi = 3$)	82.31	78.13	69.70

values of ξ, however, are lower, because the small values of u_i are favored over large ones when ξ is negative. This means that when ξ is negative the irrelevant features are favored. Also, the scores of our baseline systems (Rocchio and DTW algorithms) are shown in Table 2. We clearly observe how our model with appropriate value of ξ outperforms the baseline systems as well as our model without feature weighting. This is done by increasing the importance of feature components that help in spotting relevant word images and reducing the weight of those which do not. In addition, DTW baseline outperforms Rocchio baseline results. This superiority is justified, on the one hand, by the power of DTW in finding the optimal alignment between the query and the retrieved words. On the other hand, Rocchio parameters can be set only by using trial and error.

Qualitative results on IFN/ENIT dataset. Using the first five samples shown in the right side of Fig. 4 as initial images to formulate the query, we show qualitative results of the $P@5$ in Fig. 8. We observe how some samples are incorrectly retrieved in the case of Rocchio, DTW, and our model without feature weighting. Otherwise, the first five retrieved words using our model with $\xi = 3$ are properly correct. This can be justified by the role of vector weights in changing the original feature space into a new one that corresponds the user's requirements better. Also the corresponding feature weights to the same query with different values of ξ are shown in Fig. 9. Note that the final feature weights represented by the curves in Figs. 9(a)–9(d), which were

Fig. 8 $P@5$ qualitative results on word spotting on the IFN/ENIT dataset for: (a) Rocchio baseline, (b) DTW baseline, (c) our model without feature weighting ($\xi = 0$), and (d) our model with feature weighting ($\xi = 3$). The relevant samples to the query are outlined in green and the irrelevant ones are in red.

produced with negative values of ξ, could not perfectly distinguish the relevant features among the all features because their weights are almost in the same range. For positive values of ξ [Figs. 9(e)–9(g)], our model promotes the relevant features with high weights. For $\xi = 3$, the relevant features are in the intervals (440, 560) and (890, 1060), which correspond the user's preferences better.

Figure 10 shows the scores obtained in terms of MAP with different values of ξ on the IAM dataset. We reached the best MAP at $\xi = 3$ with 51.67%. However, the MAP

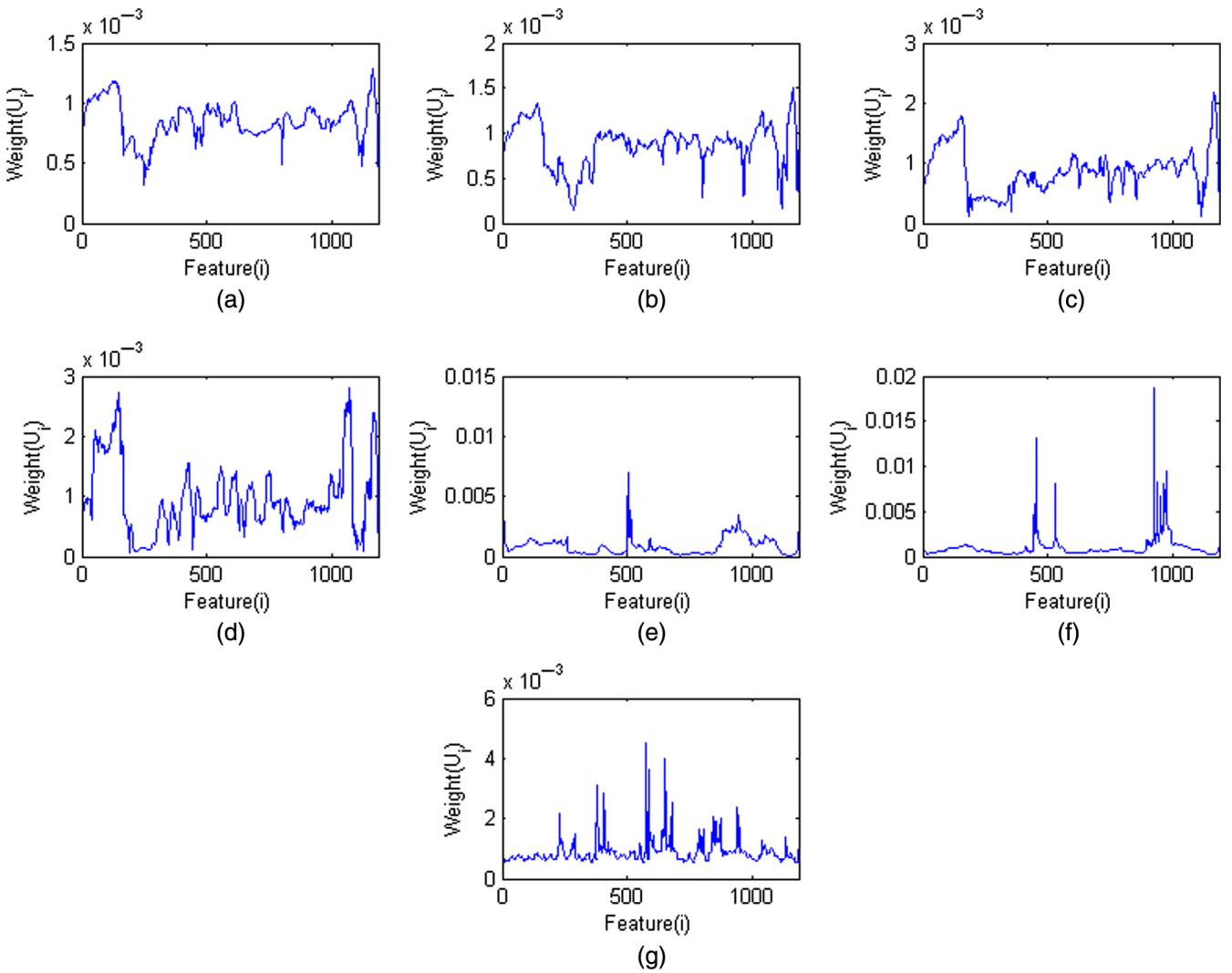


Fig. 9 Final feature weights for: (a) $\xi = -5$, (b) $\xi = -3$, (c) $\xi = -2$, (d) $\xi = -1$, (e) $\xi = 2$, (f) $\xi = 3$, and (g) $\xi = 5$.

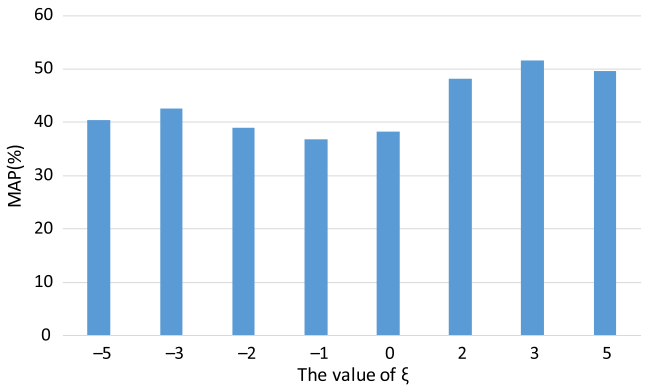


Fig. 10 The spotting scores produced with various ξ values on IAM dataset.

decreases to 49.72% with $\xi = 5$. In addition, the performance of our model with negative values of ξ is <43% at its best case. Furthermore, the results of our baseline systems (Rocchio and DTW algorithms) and those of our model (with and without feature weighting) are shown in Table 3. We observe that our model with an appropriate value of ξ outperforms both the baseline systems and our model without feature weighting. This is because our model captures the user's requirements by transforming the initial feature space into a new one that responds better to what he is looking for.

Table 3 Word spotting comparison with the baseline systems on IAM dataset.

Method	Performance		
	P@5	*P*@10	MAP
Rocchio baseline	49.12	47.67	30.85
DTW baseline	53.10	54.97	34.66
Our model without feature weighting ($\xi = 0$)	55.22	56.07	38.28
Our model with feature weighting ($\xi = 3$)	87.24	70.45	51.67

Qualitative results on IAM dataset. Figure 11 shows the qualitative results of the *P*@5 using the first five samples shown in the right side of Fig. 5 as initial images to formulate the query. Unlike our model with feature weighting ($\xi = 3$), which properly retrieved relevant words, the baseline systems (Rocchio and DTW) and our model without weighting produced some incorrect word images. Among retrieval errors include words with similar shapes ("government" versus "German"). Also the feature weights to the same query with different values of ξ are shown in Fig. 12. The final weights represented by the curve in Fig. 12(f), which is produced with $\xi = 3$, distinguished better the relevant features among the all features, especially in the intervals (140, 290), (540, 590), and (640, 660). For $\xi = 2$, the feature weighting [the curve in Fig. 12(e)] seems to be a hard selection (some few components have significant weights and the others are almost zero). However, the relevant features lost their significant weights when $\xi = 5$. For negative values of ξ [Figs. 12(a)–12(d)], our model cannot better promote the relevant features and the weights assigned to feature components are roughly close to each other.

The MAP obtained with different values of ξ on the GW dataset is depicted in Fig. 13. We also show the results of our baseline systems in Table 4. We notice that our model with $\xi = 3$ outperforms the other systems with 68.17% in terms of MAP. In addition, the performance of our model with negative values of ξ is <61%.

Qualitative results on GW dataset. Figure 14 shows the *P*@5 using the first five samples shown in the right side of Fig. 6 as initial images to formulate the query. We find that our model with and without feature weighting ($\xi = 3$) has properly retrieved the top five relevant words, the baseline systems have produced some incorrect word images. Also the corresponding final feature weights to the same query with different values of ξ are shown in Fig. 15. It can be clearly seen that the weights represented by the curve in Fig. 15(f), which is produced with $\xi = 3$, distinguished better the relevant features, especially in the intervals (345, 420), (630, 710), and (965, 992). For negative values

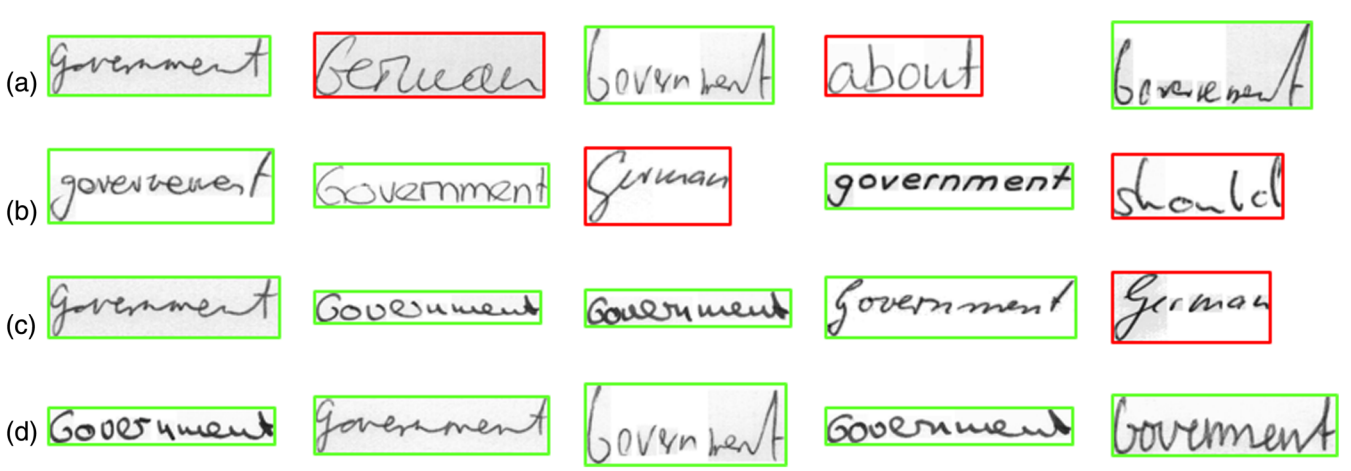


Fig. 11 *P*@5 qualitative results on word spotting on the IAM dataset for: (a) Rocchio baseline, (b) DTW baseline, (c) our model without feature weighting ($\xi = 0$), and (d) our model with feature weighting ($\xi = 3$). The relevant samples to the query are outlined in green and the irrelevant ones are in red.

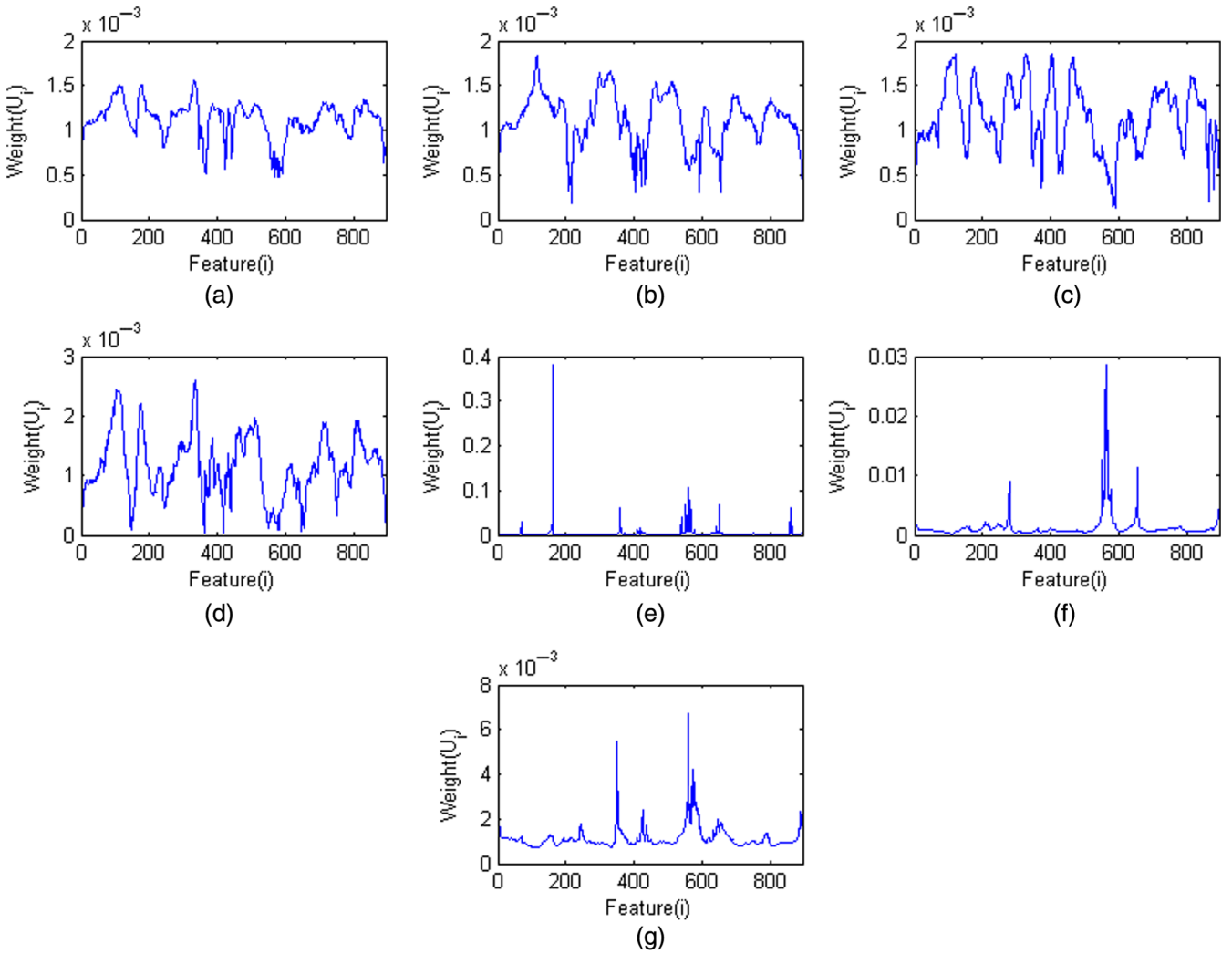


Fig. 12 Final feature weights for: (a) $\xi = -5$, (b) $\xi = -3$, (c) $\xi = -2$, (d) $\xi = -1$, (e) $\xi = 2$, (f) $\xi = 3$, and (g) $\xi = 5$.

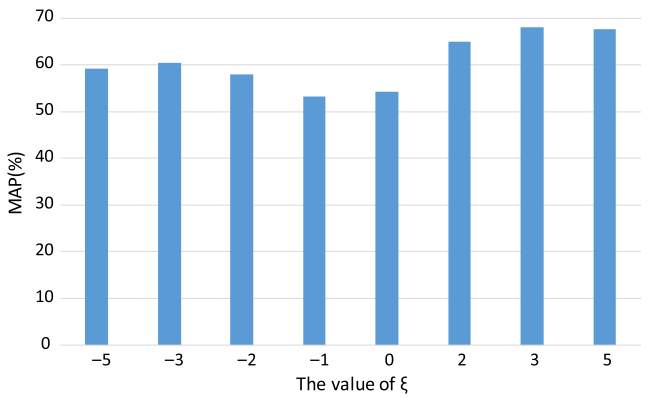


Fig. 13 The spotting scores produced with various ξ values on GW dataset.

Table 4 Word spotting comparison with the baseline systems on GW dataset.

Method	Performance		
	P@5	*P*@10	MAP
Rocchio baseline	44.81	47.56	38.05
DTW baseline	67.18	52.03	44.27
Our model without feature weighting ($\xi = 0$)	60.20	63.12	54.33
Our model with feature weighting ($\xi = 3$)	83.05	80.2	68.17

of ξ [Figs. 15(a)–15(d)], our model has failed to distinguish the most relevant features.

4.6.2 *Second experiment*

Here we study the variation in the performance of our method as a function of the number of word images participated in the query and also the system convergence (i.e., the number of refinements iterations required). In fact, the user can first select the initial word images from set $S^{(0)}$ to formulate $\vec{q}^{(0)}$ and then select more word images from the displayed result [i.e., the set $R^{(1)}$] as well as the set $S^{(1)}$ to formulate $\vec{q}^{(1)}$, and so forth. Based on that, the performance of our system as well as the convergence of our

Fig. 14 *P*@5 qualitative results on word spotting on the GW dataset for: (a) Rocchio baseline, (b) DTW baseline, (c) our model without feature weighting ($\xi = 0$), and (d) our model with feature weighting ($\xi = 3$). The relevant samples to the query are outlined in green and the irrelevant ones are in red.

algorithm depend mainly on the number of word images selected from both sets S and R. Therefore, we perform our method with different numbers of initial and displayed word images that range from one to ten samples. We keep $|S| = |R|$ at each iteration so that the selection process will not be tiring and tedious. Figure 16 shows the MAP values obtained on IFN/ENIT, IAM, and GW datasets when varying the number of word images that participated in the query.

We clearly notice that the MAP proportionally increases with the number of word images that participated in the

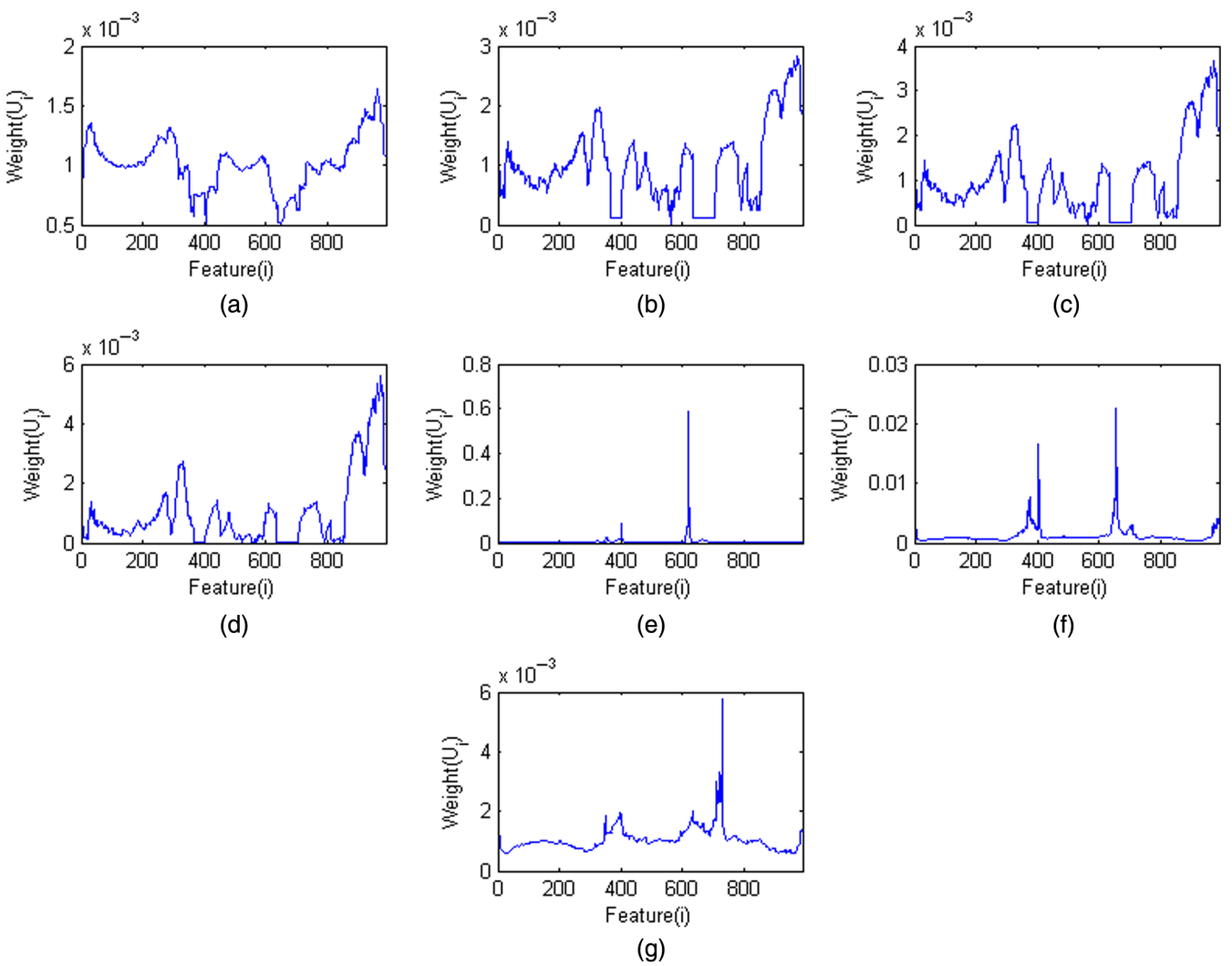


Fig. 15 Final feature weights for: (a) $\xi = -5$, (b) $\xi = -3$, (c) $\xi = -2$, (d) $\xi = -1$, (e) $\xi = 2$, (f) $\xi = 3$, and (g) $\xi = 5$.

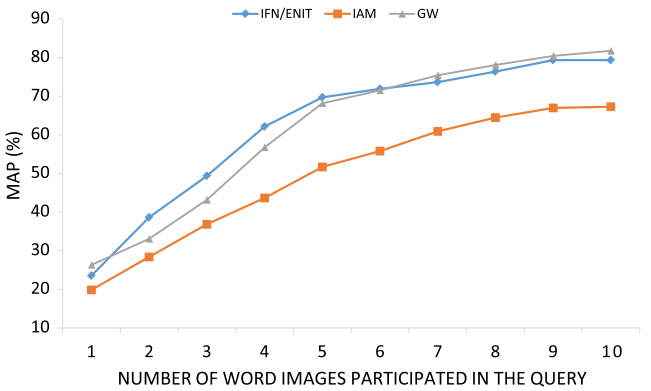


Fig. 16 The influence of the number of word images that participated in the query in IFN/ENIT dataset.

query. The best score obtained on IFN/ENIT is when nine word images participated in the query with 79.34% (the difference in terms of score between nine and ten samples is less than 1%). For IAM and GW datasets, the MAP proportionally increases with the number of word images participated in the query. The best score obtained is 67.27% and 81.74%, respectively.

Now, we study the number of refinements required to reach a high performance as a function of the number of samples participating in the query. To do this, we compute the average number of refinements, denoted ANREF, for all queries. As shown in Fig. 17 for the three datasets, our method with one sample cannot give good scores. With any noticeable improvement, the user can stop the spotting process after three or four refinements.

From the IFN/ENIT and IAM curves, we find that when using two and three samples to build the query, the average of refinements increases from 11.70 to 12.25 and from 10.4 to 12.47 iterations for IFN/ENIT and IAM, respectively. This improvement means that the spotting process needs more samples to build the query and to find the optimal weights that meet the user requirements. From the GW curve, the average number of iterations when varying the number of samples from two to four increases from 9.5 to 11.02 but the system provides an enhancement in MAP with 23.65%. Thereafter, the use of more than four samples appreciably reduces the number of refinements, where we find the average numbers of iterations over the experiments that give the best results (i.e., when using ten samples to build the query) are 6.85, 8.12, and 6.2 for IFN/ENIT, IAM, and GW, respectively. We also note that the spotting on GW dataset needs less refinements than on IFN/ENIT and IAM due to its homogeneous writing style (GW is considered as single-writer dataset). Consequently, this experiment clearly shows the convergence speed of our model as a function of the initial word images that participated in the query.

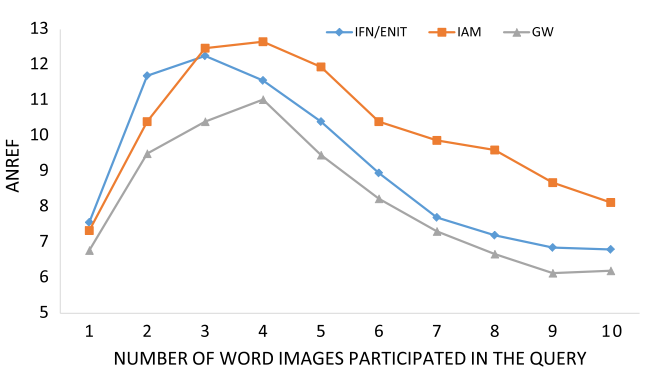


Fig. 17 The MAP as function of the number of word images that participated in the query.

4.6.3 *Third experiment*

In this experiment, we combine the advantages of our model with the benefits of DTW in the decision rule to improve the overall performance of our system. In fact, DTW is the most common similarity measure between two sequences adopted in the literature of HKWS. It consists of finding the optimal warping path or elastic alignment between the two feature vectors and then minimizing the vector-to-vector cumulative distances along the optimal warping path. Subsequently, we integrate the DTW in our similarity measure defined previously in Eq. (1), which becomes:

$$\mathrm{Dist}[\overrightarrow{x_n}^{(t)}, \vec{q}^{(t)}] = \mathrm{sim}_{\mathrm{DTW}}[\vec{u}^{(t)}\overrightarrow{x_n}^{(t)}, \vec{u}^{(t)}\vec{q}^{(t)}]. \tag{24}$$

We note that we do not have to reformulate the mathematical modeling to find the optimal weight $\vec{u}$ and the query $\vec{q}$. This is because of the DTW recurrent equation, which is based on dynamic programming where the analytic computation of the objective function derivative [Eq. (3)] will be then so complicated. We denote our model with Euclidean similarity by OMEucl and our model with DTW similarity by OMDTW.

The final results obtained on the three datasets using 10 selected samples from both sets S and R are shown in Table 5. We can clearly see that OMDTW outperforms OMEucl. This success is justified by, on the one hand, the mathematical formulation of the query and the feature weighting that provides a powerful tool to identify and anticipate the user's needs for helping him to reach his mental target with high performance. On the other hand, the ability of DTW minimizes the difference between the query and the similar candidate images using the vector-to-vector cumulative distance to find the best warping path.

4.7 *Comparison*

To the best of our knowledge, we propose the first mental model for word image spotting. In order to give an idea on where our mental model ranks performance-wise, we compare with works that used the same experimental protocol, the same performance measures, and the same datasets. It should be noted that the methods with which we compare

Table 5 Spotting results obtained with OMEucl and OMDTW on IFN/ENI, IAM, and GW datasets by introducing 10 samples from both sets S and R at each refinement.

Method	IFN/ENIT		IAM		GW	
	MAP	ANREF	MAP	ANREF	MAP	ANREF
OMEucl	79.34	6.85	67.27	8.12	81.74	6.2
OMDTW	88.28	6.05	74.64	7.04	87.53	4.83

Table 6 Comparison of spotting system performances on the three datasets.

Reference	Dataset	Method	Training setup	MAP(%)
Rodriguez-Serrano and Perronnin et al.[10]	IFN/ENIT	Learning method based on SCHMM	10 samples of each query are randomly selected from set *a*	41.60
Proposed method	IFN/ENIT	Mental model	Same as in Ref. 10	**88.28**
Almazan et al.[36]	IAM	Learning method based on GHMM and SVMs	The training set of Official partition for "writer-independent text line recognition"	55.73
Sharma and Sankar[48]	IAM	Learning method based on GMM and SVMs	40% of the whole IAM dataset	46.53
Sudholt et al.[49]	IAM	Learning method based on CNN	Same as in Ref. 36	72.51
Proposed method	IAM	Mental model	10 samples of each query are randomly selected from the same training set used in Ref. 36	**74.64**
Almazan et al.[36]	GW	Learning method based on GHMM and SVMs	75% of the whole GW dataset	**92.90**
Sudholt et al.[49]	GW	Learning method based on CNN	Same as in Ref. 36	92.64
Proposed method	GW	Mental model	10 samples of each query are randomly selected from the same training set used in Ref. 36	87.53

are not mental models. The scores of our work compared to the state-of-the-art methods can be seen in Table 6.

For IFN/ENIT and IAM datasets, Table 6 noticeably shows that our method outperforms the state-of-the-art approaches, which are based on powerful learning techniques. For GW, the methods proposed in Ref. 36, which is based on semicontinuous hidden Markov model and SVMs, and in Ref. 49, which is based on convolutional neural networks (CNN), outperform our method by 5.37% and 5.11% in terms of MAP, respectively. However, we find that this increase in spotting score can be traded against a significant increase of the number of training samples, which makes the learning phase very expensive in data and time. On the contrary, our mental model starts the spotting with only 10 samples for each query and provides a competitive results after less than seven refinements.

This comparison proves that our mental model can significantly give promising performances similar or better than the state-of-the-art on spotting task. The advantages of our method can be summarized as follows: (1) it does not require large amount of labeled data. (2) Our method is not based on complicate learning algorithms. (3) It allows the user at any time to change the spotting trend according to what he is looking for. This can be reached by selecting more relevant word images from the displayed results. (4) It helps the hesitant user to clarify iteratively his idea about what he wants to. (5) Our method can be easily generalized or reused for new application or new data whatever the user's type.

5 Conclusion

The main objective of this paper is to build a mental model for keyword spotting that both integrates the user and the system requirements. This model helps a user to retrieve his mental target based on a dynamic process. In fact, it is proved that the integration of the user's needs is very important to reach a high performance in the spotting task, which can be identified through his selections. However, the use of the previous query in the construction of the new query can give the spotting much PR reached in less time. The main advantage of our proposed method is that it allows the user to construct reasoning about what he is looking for. This is done by selecting a few word images to start the spotting, then exploiting the displayed word images with the previous query to provide new training samples to the next refinement, and so forth. The proposed method also enhances iteratively each feature's component by maximizing as much importance as those that are relevant to the query. At the same time, it reduces the importance of those that do not. This process is assured by transforming the original feature space into a new one that better corresponds the user's requirements. The effectiveness of the proposed method has been proved through different handwritten collections: Arabic, modern English, and historical English documents. Our mental model is generic and can be spread on any information retrieval system.

References

1. R. Manmatha, C. Han, and E. M. Riseman, "Word spotting: a new approach to indexing handwriting," in *Proc. of IEEE Computer Society Conf. on Computer Vision and Pattern Recognition CVPR'96*, pp. 631–637, IEEE (1996).
2. T. M. Rath and R. Manmatha, "Word image matching using dynamic time warping," in *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, Vol. 2, pp. II–II, IEEE (2003).
3. T. M. Rath and R. Manmatha, "Word spotting for historical documents," *Int. J. Doc. Anal. Recognit.* **9**(2–4), 139–152 (2007).
4. A. P. Giotis et al., "A survey of document image word spotting techniques," *Pattern Recognit.* **68**, 310–332 (2017).
5. A. L. Kesidis et al., "A word spotting framework for historical machine-printed documents," *Int. J. Doc. Anal. Recognit.* **14**(2), 131–144 (2011).
6. M. L. Kherfi, D. Ziou, and A. Bernardi, "Combining positive and negative examples in relevance feedback for content-based image retrieval," *J. Visual Commun. Image Representat.* **14**(4), 428–457 (2003).

7. T. Bdiri, N. Bouguila, and D. Ziou, "A statistical framework for mental targets search using mixture models," in *Artificial Intelligence Applications in Information and Communication Technologies*, pp. 99–118, Springer (2015).
8. R. Datta et al., "Image retrieval: ideas, influences, and trends of the new age," *ACM Comput. Surv.* **40**(2), 1–60 (2008).
9. A. L. Kesidis and B. Gatos, "Efficient cut-off threshold estimation for word spotting applications," in *Int. Conf. on Document Analysis and Recognition (ICDAR)*, pp. 279–283, IEEE (2011).
10. J. A. Rodríguez-Serrano and F. Perronnin, "A model-based sequence similarity with application to handwritten word spotting," *IEEE Trans. Pattern Anal. Mach. Intell.* **34**(11), 2108–2120 (2012).
11. M. Khayyat, L. Lam, and C. Y. Suen, "Learning-based word spotting system for arabic handwritten documents," *Pattern Recognit.* **47**(3), 1021–1030 (2014).
12. V. Frinken et al., "A novel word spotting method based on recurrent neural networks," *IEEE Trans. Pattern Anal. Mach. Intell.* **34**(2), 211–224 (2012).
13. A. H. Toselli and E. Vidal, "Word-graph based handwriting key-word spotting: impact of word-graph size on performance," in *11th IAPR Int. Workshop on Document Analysis Systems (DAS)*, pp. 176–180, IEEE (2014).
14. C. Choisy, "Dynamic handwritten keyword spotting based on the NSHP-HMM," in *Ninth International Conf. on Document Analysis and Recognition*, Vol. 1, pp. 242–246, IEEE (2007).
15. J. Puigcerver, A. H. Toselli, and E. Vidal, "ICDAR 2015 competition on keyword spotting for handwritten documents," in *13th Int. Conf. on Document Analysis and Recognition (ICDAR)*, pp. 1176–1180, IEEE (2015).
16. T. Konidaris et al., "Keyword-guided word spotting in historical printed documents using synthetic data and user feedback," *Int. J. Doc. Anal. Recognit.* **9**(2–4), 167–177 (2007).
17. A. Fornés et al., "A keyword spotting approach using blurred shape model-based descriptors," in *Proc. of the 2011 Workshop on Historical Document Imaging and Processing*, pp. 83–90, ACM (2011).
18. E. Vidal, A. H. Toselli, and J. Puigcerver, "High performance query-by-example keyword spotting using query-by-string techniques," in *13th Int. Conf. on Document Analysis and Recognition (ICDAR)*, pp. 741–745, IEEE (2015).
19. Y. Fang and D. Geman, "Experiments in mental face retrieval," *Lect. Notes Comput. Sci.* **3546**, 637–646 (2005).
20. K. J. W. Craik, *The Nature of Explanation*, Vol. **445**, Cambridge University Press, Cambridge (1967).
21. D. Gentner and A. Stevens, *Mental Models*, pp.7–14, Psychology Press, New York (1983).
22. P. Bayman and R. E. Mayer, "Instructional manipulation of users' mental models for electronic calculators," *Int. J. Man-Mach. Stud.* **20**(2), 189–199 (1984).
23. R. D. Jacobson, *Leading for a Change*, Routledge (2012).
24. I. J. Cox et al., "The bayesian image retrieval system, pichunter: theory, implementation, and psychophysical experiments," *IEEE Trans. Image Process.* **9**(1), 20–37 (2000).
25. M. Ferecatu and D. Geman, "A statistical framework for image category search from a mental picture," *IEEE Trans. Pattern Anal. Mach. Intell.* **31**(6), 1087–1101 (2009).
26. M. L. Kherfi, D. Ziou, and A. Bernardi, "Image retrieval from the World Wide Web: issues, techniques, and systems," *ACM Comput. Surv.* **36**, 35–67 (2004).
27. S. Boutemedjet and D. Ziou, "Long-term relevance feedback and feature selection for adaptive content based image suggestion," *Pattern Recognit.* **43**(12), 3925–3937 (2010).
28. X.-Y. Wang et al., "A new SVM-based relevance feedback image retrieval using probabilistic feature and weighted kernel function," *J. Visual Commun. Image Represent.* **38**, 256–275 (2016).
29. K. Zagoris, K. Ergina, and N. Papamarkos, "Image retrieval systems based on compact shape descriptor and relevance feedback information," *J. Visual Commun. Image Represent.* **22**(5), 378–390 (2011).
30. H. Wei, G. Gao, and X. Su, "A multiple instances approach to improving keyword spotting on historical Mongolian document images," in *13th Int. Conf. on Document Analysis and Recognition (ICDAR)*, pp. 121–125, IEEE (2015).
31. K. Zagoris, I. Pratikakis, and B. Gatos, "A framework for efficient transcription of historical documents using keyword spotting," in *Proc. of the 3rd Int. Workshop on Historical Document Imaging and Processing*, pp. 9–14, ACM (2015).
32. M. Hatzigiorgaki and A. N. Skodras, "Compressed domain image retrieval: a comparative study of similarity metrics," *Proc. SPIE* **5150**, 439–448 (2003).
33. Y. Rui and T. Huang, "Optimizing learning in image retrieval," in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, Vol. **1**, pp. 236–243, IEEE (2000).
34. M. Pechwitz et al., "IFN/ENIT-database of handwritten Arabic words," in *Proc. of Francophone Int. Conf. on writing and Document CIFED*, Vol. 2, pp. 127–136, Citeseer (2002).
35. U.-V. Marti and H. Bunke, "The IAM-database: an English sentence database for offline handwriting recognition," *Int. J. Doc. Anal. Recognit.* **5**(1), 39–46 (2002).
36. J. Almazán et al., "Word spotting and recognition with embedded attributes," *IEEE Trans. Pattern Anal. Mach. Intell.* **36**(12), 2552–2566 (2014).
37. G. J. Tan, G. Sulong, and M. S. M. Rahim, "Writer identification: a comparative study across three world major languages," *Forensic Sci. Int.* **279**, 41–52 (2017).
38. N. Bouadjenek, H. Nemmour, and Y. Chibani, "Fuzzy integrals for combining multiple svm and histogram features for writer's gender prediction," *IET Biometrics* **6**(6), 429–437 (2017).
39. D. L. Donoho and M. R. Duncan, "Digital curvelet transform: strategy, implementation, and experiments," *Proc. SPIE* **4056**, 12–30 (2000).
40. E. J. Candès and D. L. Donoho, "New tight frames of curvelets and optimal representations of objects with piecewise c2 singularities," *Commun. Pure Appl. Math.* **57**(2), 219–266 (2004).
41. E. Candès et al., "Fast discrete curvelet transforms," *Multiscale Model. Simul.* **5**(3), 861–899 (2006).
42. V. Ojansivu and J. Heikkilä, "Blur insensitive texture classification using local phase quantization," *Lect. Notes Comput. Sci.* **5099**, 236–243 (2008).
43. Y. Hannad, I. Siddiqi, and M. E. Y. El Kettani, "Writer identification using texture descriptors of handwritten fragments," *Expert Syst. Appl.* **47**, 14–22 (2016).
44. Y. Rui, T. S. Huang, and S.-F. Chang, "Image retrieval: current techniques, promising directions, and open issues," *J. Visual Commun. Image Represent.* **10**(1), 39–62 (1999).
45. H. Min and Y. Shuangyuan, "Overview of content-based image retrieval with high-level semantics," in *3rd Int. Conf. on Advanced Computer Theory and Engineering (ICACTE)*, Vol. 6, IEEE (2010).
46. J. J. Rocchio, "Relevance feedback in information retrieval," in *SMART Retrieval System, Experiments in Automatic Document Processing*, pp. 313–323 (1971).
47. A. Papandreou, B. Gatos, and K. Zagoris, "An adaptive zoning technique for word spotting using dynamic time warping," in *12th IAPR Workshop on Document Analysis Systems (DAS)*, pp. 387–392, IEEE (2016).
48. A. Sharma et al., "Adapting off-the-shelf CNNs for word spotting & recognition," in *13th Int. Conf. on Document Analysis and Recognition (ICDAR)*, pp. 986–990, IEEE (2015).
49. S. Sudholt and G. A. Fink, "Phocnet: a deep convolutional neural network for word spotting in handwritten documents," in *15th Int. Conf. on Frontiers in Handwriting Recognition (ICFHR)*, pp. 277–282, IEEE (2016).

Youcef Brik received his BEng degree in electronics from the University of M'sila, Algeria, in 2007, and his magister degree in signal and image processing from the Faculty of Electronic and Computer Science, University of Sciences and Technology Houari Boumediene, Algiers, Algeria, in 2010. He is a PhD student at the same faculty. His research interests include document image analysis, information retrieval, machine learning, and pattern recognition.

Djemel Ziou received his BEng degree in computer science from the University of Annaba, Algeria, in 1984, and his PhD in computer science from the Institut National Polytechnique de Lorraine (INPL), Lorraine, France, in 1991. From 1987 to 1993, he did teaching and research at several universities in France. Presently, he is a full professor in the Department of Computer Science, Université de Sherbrooke, Canada. He has served on numerous conference committees as member or chair. He heads the laboratory MOIVRE and the consortium CoRIMedia, which he founded. His research interests include image processing, information retrieval, computer vision, and pattern recognition.