2011 International Conference on Document Analysis and Recognition

Writer Identification using TF-IDF for Cursive Handwritten Word Recognition

Quang Anh BUI, Muriel VISANI, Sophea PRUM and Jean-Marc OGIER
Laboratory L3i, University of La Rochelle
La Rochelle, France
Email: qbui01@univ-lr.fr, muriel.visani@univ-lr.fr, sophea.prum@univ-lr.fr and jean-marc.ogier@univ-lr.fr

***Abstract*—In this paper, we present two text-independent writer identification methods in a closed-world context. Both methods use on-line and off-line features jointly with a classifier inspired from information retrieval methods. These methods are local, respectively based on the character and grapheme levels.**
This writer identification engine may be used to personalize our cursive word recognition engine [1] to the handwriting style of the writer, resulting in an adaptive cursive word recognizer.
Experiments assess the effectiveness of the proposed approaches in a context of writer identification as well as integrated to our cursive word recognizer to make it adaptive.

***Keywords*-Writer identification; handwriting recognition; information retrieval; on-line signature**

I. Introduction

Writer identification has been an active research topic for several decades in the image processing and pattern recognition community [2], [3]. Despite continuous effort, it remains a challenging issue with many applications in the fields of behavioral biometrics and handwriting recognition. We have previously developed an original cursive handwritten word recognition system [1]. One of the main difficulties when conceiving such a system is the large amount of variations between the visual appearance of two occurrences of a given word [4]. The final objective of the work presented in this paper is to ameliorate our cursive handwritten word recognition system by personalizing it. Many writer adaptive handwriting recognition methods have been proposed in the literature [5], [6], [7] and generally provide better results than writer-independent recognition systems. In this paper we will focus on the step of writer identification.

Our handwriting recognition system is conceived to recognize the handwriting of a limited number of known users (employees of a company) filling forms using an electronic tablet. The tablet provides on-line (dynamic) features such as the writing pressure, velocity and pen inclination. From these on-line features, we approximate the off-line signal and extract off-line features describing the visual characteristics of the handwriting such as its shape. Therefore, in this paper, we will focus on text-independent writer identification using both on-line and off-line features in a closed-world context. While many off-line writer recognition systems have been proposed in the literature [8], [4], [9], [10], [11], [12], there is much less work related to the on-line signal [13], [14], [15].

The features used for writer identification may be *global* or *local*. While global features (including the density of lines, texture, some grid-based approaches, etc...) describe the signal from a macroscopic point of view, local features are generally more robust when a small part of the input signal is affected by noise or distorsion. Depending on their applicative context, the authors may use local features [5], [9], [10], [14], [16], [15], global features [11], [17], [13], or a combination of both through fusion or sequential combination [18]. When using global features, a few text lines of handwritten material are needed in order to discriminate between a large number of writers. However, in our applicative context where we use forms, we often do not have enough material to use global features and therefore we focus on local features.

In this paper, we introduce two local approaches, respectively at the levels of the characters and graphemes, and show their effectiveness for enhancing the results of the handwriting recognition approach by personalizing it with respect to the writer or to the handwriting style of the writer.

This paper is organized as follows. In Section II, we present related work in the domain of writer identification. In Section III, we introduce our two approaches. Section IV provides experiments showing the effectiveness of these approaches, while Section V concludes this paper and gives future directions of research.

II. Related work

In this section we will focus on text-independent writer identification methods using local on-line and off-line features. The main idea of these methods is to calculate features of small pieces extracted from the documents. These features are further used to describe and compare documents using some similarity measure *sim* between these features. The unknown writer of a test document T is therefore identified as the author of the document D_i in the base of reference documents Ω (containing handwriting from all enrolled writers) with highest similarity $sim(T, D_i)$:

$$writer(T) = writer(\arg\max_{D_i \in \Omega}(sim(T, D_i)) \qquad (1)$$

1520-5363/11 $26.00 © 2011 IEEE
DOI 10.1109/ICDAR.2011.173

In [19], the authors work at the grapheme level using as a similarity measure the maximal correlation measure between off-line features of the graphemes. Experiments were carried out on a data set extracted from the PSI database [19], composed of 88 writers, with 2 documents per writer. The results reported for writer identification are around 90%. Two major drawbacks of this approach can however be pointed out. The first one is that it is especially computationally expensive due to the pattern matching technique employed. The second drawback is that all the graphemes have the same weight over all the documents, without explicit characterization of the writer style.

To overcome these drawbacks, a method using an information retrieval model was proposed in [10]. The main idea lies in the definition of a common grapheme prototype set over the entire database, where each writer style is characterized by a weight matrix over these prototypes. The prototypes $\varphi_{i\in\{1,...P\}}$ are computed by clustering off-line features extracted from the graphemes (using k-means). Each document D_j can be described by its weight vector $W_j = (w_{ij}, \ldots, w_{Pj})^T$, where the w_{ij}'s are weights assigned to each prototype φ_i as:

$$w_{ij} = TF_{\varphi_i}(D_j) IDF_{\varphi_i}(\Omega) \quad (2)$$

where $TF_{\varphi_i}(D_j)$ is the *Term Frequency* of the prototype φ_i in the document D_j and the *Inverse Document Frequency* $IDF_{\varphi_i}(\Omega)$ is computed as follows:

$$IDF_{\varphi_i}(\Omega) = \log\left(\frac{1+|\Omega|}{1+|D \in \Omega \; s.t. \; \varphi_i \in D|}\right) \quad (3)$$

The more the prototype φ_i is rare in the reference database Ω, the more the value of $IDF_{\varphi_i}(\Omega)$ is high. The similarity measure is the cosine angle of the two weight vectors W of the documents to compare.

Experiments are carried out on the same subset of the PSI database as in [19], and the writer identification rates are around 93% (82/88). Two major drawbacks of this approach are that it is based on off-line features only, which are often less informative than on-line features, and that the features are extracted from graphemes, the distribution of which may be scattered, resulting on a large number of prototypes.

Two more approaches relying on information retrieval models are introduced in [14], [15]. However, instead of using grapheme clusters as features, they use character clusters, which include less variations. Characters are extracted and recognized from documents by a recognition and segmentation system (MyScript Builder), relying on on-line features only. Each character is described by 30 representative points, from which 7 basic features (coordinates, angle, ...) are extracted. One of the main drawbacks of these approaches is that they rely on very local on-line features only which do not carry enough information. Indeed, they do not take into account the global shape of the character, which results in a higher sensitivity towards noise.

III. Proposed approaches

In this paper, we introduce two local approaches for writer identification using cursive handwriting, respectively based on characters and graphemes extracted from cursive handwriting. We consider a closed-world and text-independent context using on-line signal as well as the reconstructed off-line signal.

A. Method based on characters

The training stage of the proposed method is organized into two steps (see Figure 1).

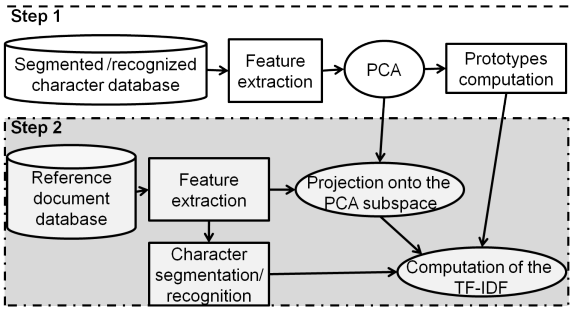

Figure 1. Method based on characters - Training stage.

During the first step, 45 features (both on-line and off-line) are extracted from a reference character database. Principal Component Analysis (PCA) is applied to the resulting feature vectors, so as to reduce the dimensionality of the problem and to disable redundant information. The prototypes $\varphi_{k\alpha}$ are computed for each letter $\alpha = 'a', 'b', \ldots 'z'$ using the k-means algorithm with an adapted initialization procedure. Based on experimental results, we use $P = 11$ prototypes for each letter. Some examples of prototypes are given in Figure 2.

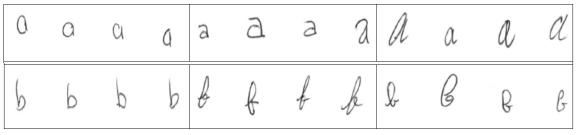

Figure 2. Some prototypes for letters a and b.

During the second step, we first segment and recognize each handwritten character x in the reference document D using its feature vector. The segmentation/recognition method used is similar to the mono-character step of the cursive word recognizer presented in [1]. Let us denote α the letter assigned to the character x. We project the feature vector of x in the PCA subspace built during the first step and obtain a reduced feature vector f_x. Then, we calculate

the similarities $p(\varphi_{k\alpha}, x)$ between the character x and each of the P the prototypes $\varphi_{k\alpha}$ of letter α, as:

$$p(\varphi_{k\alpha}, x) = \frac{exp(-\beta * dist_{(k\alpha)}(f_x))}{\sum_{i=1}^{P} exp(-\beta * dist_{(i\alpha)}(f_x))} \quad (4)$$

where $dist_{(k\alpha)}(f_x)$ is the mean Euclidean distance between f_x and the reduced feature vectors of the samples belonging to the prototype $\varphi_{k\alpha}$. Parameter $\beta > 0$ is a fixed scalar which determines the selectivity of exponential function. In our system, we use $\beta = 0.1$.

The term frequency $TF_{\varphi_{k\alpha}}(D)$ of prototype $\varphi_{k\alpha}$ in the document D is defined as follow:

$$TF_{\varphi_{k\alpha}}(D) = \frac{1}{\sum\limits_{x \in D} \delta_{letter(x)=\alpha}} \sum_{x \in D} p(\varphi_{k\alpha}, x) \delta_{letter(x)=\alpha} \quad (5)$$

where $\delta_{letter(x)=\alpha} = 1$ if x is recognized as the letter α, and 0 otherwise. The value $IDF_{\varphi_{k\alpha}}(\Omega)$ is calculated as in Equation (3), with $\varphi_{k\alpha}$ instead of φ_i. We consider that $\varphi_{k\alpha} \in D$ when at least a character x of D is such that:

$$\delta_{letter(x)=\alpha} \quad \text{and} \quad p(\varphi_{k\alpha}, x) = \underset{j=1...P}{Argmax}(p(\varphi_{j\alpha}, x))$$

Further, each document D is represented by a weight matrix

$$W = [TF_{\varphi_{k\alpha}}(D) * IDF_{\varphi_{k\alpha}}(\Omega)] \quad (6)$$

of size $26 \times P$, describing the writing style of this document.

During the recognition stage, we can use any distance (Euclidean, normalized cosine, χ^2) or dissimilarity measure between the weight matrices of the documents to compare.

The main differences with the methods in [14], [15] lie in the use of a different character segmentation/recognition method, in the use of both on-line and off-line features, of PCA and of a different similarity measure $p(\varphi_{k\alpha}, x)$.

B. Method based on graphemes

The method proposed in the previous section is very effective in general. However, when the character recognizer fails to correctly recognize a given character x, then x is assigned to an incorrect letter α and the similarities $p(\varphi_{k\alpha}, x)$ on which is based the writer recognition process are non relevant, inducing a systematic bias in the writer recognizer. In order to reduce this bias, we conceive a method based on graphemes instead of characters. This method is very similar to the method based on characters that is described in the previous section and in Figure 1, except that instead of segmenting and recognizing characters, we extract graphemes by using our segmentation method introduced in [20]. Then, the graphemes x are characterized by using the features in [20]. In order to reduce variability of the clustering input dataset as well as the weight matrices computation time, graphemes are pre-classified into 4 groups α depending on the initial writing direction θ ($\theta \in \{[0; 90[, [90; 180[, [180; 270[, [270; 360[$). Then, PCA is applied. Inside each group α, graphemes are clustered into P different cluster prototypes using a variant of the k-means algorithm. Here, based on experiments, we use $P = 20$. Then, the similarity $p(\varphi_{k\alpha}, x)$ between the grapheme x belonging to group α and the k^{th} prototype $p_{k\alpha}$ of group α is computed using equation (4) with $\beta = 0.1$ and the weight matrices are computed as in the previous method.

During the recognition stage, any two documents (and therefore two writers) are compared using any distance between the corresponding weight matrices.

IV. Experimental results

In order to assess the effectiveness of the proposed approaches, we perform a series of experiments of increasing complexity, first with the character-based approach and second with the grapheme-based approach. In both cases, the results presented in this section are obtained using χ^2 distance, as it gave the best results.

A. Results of the character-based method

First experiment: We use our own handwriting database including 32 documents written by 16 writers on our electronic tablet (2 documents per writer). Each document contains, for each of the 26 letters of the latin alphabet, 10 occurences in predefined fields. Both the reference and the test databases include 1 document per writer, and therefore $16 \times 10 \times 26 = 4160$ characters. If we use the ground-truth and skip the character segmentation/recognition step, the writer recognition rate is 100%. If we use our isolated character segmentation/recognition module described in [20], the writer recognition rate is still 100% even though the character recognition rate has dropped to 94%. Therefore we can conclude that this approach is relatively robust towards character segmentation/recognition errors.

Second experiment: We use cursive words extracted from the IRONOFF database [21] to construct both the reference and test databases. We consider words written by 10 to 300 different writers, with 30 words per writer (among which 20 words are included in the reference database, the 10 remaining words in the test database). Figure 3 (dotted lines) shows the writer recognition rates when using the ground-truth and when using our character segmentation/recognition module. The comparison of these two curves highlights the bias introduced by the automatic character segmentation/recognition errors. Indeed, the difference between the recognition rates of these two experiments in on average 12%. We can also see that the recognition rate drops faster when the character segmentation/recognition step is performed automatically, than when the ground-truth is used. This shows the limits of the robustness towards character segmentation/recognition errors (robustness illustrated by the previous experiment).

We implemented a method which is as similar to the one presented in [15] as we could implement. Instead of My

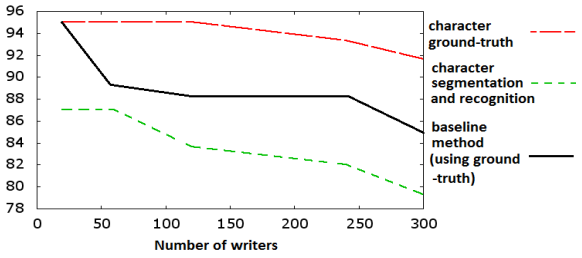

Figure 3. Recognition rates of the second experiment.

Script Builder which has high character recognition rates, this method uses the ground-truth, the same features as [15] and no PCA. We denote it "baseline method" (Figure 3, solid line). Our method is superior to this method when using the ground-truth as well, and slightly inferior when using our own character segmentation/recognition method.

B. Results of the grapheme-based method

First experiment: This experiment is performed using the grapheme-based method and the same databases as in the second experiment of the character-based method. Recognition rates when increasing the number of writers are given in Figure 4.

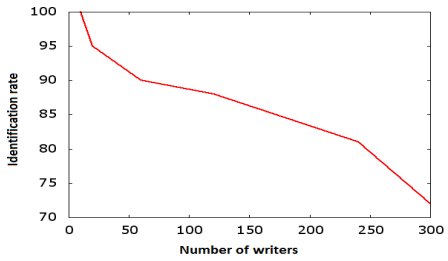

Figure 4. Recognition rates of the method based on graphemes.

By comparing Figures 3 and 4, we can see that the method based on graphemes outperforms the method based on characters, as long as the number of writers does not exceed 150. This may be explained by the fact that considering graphemes instead of characters suppresses the systematic bias introduced by the character recognizer. On the other hand, the recognition rate of the grapheme-based method drops much more abruptly when the number of writers is increased, and becomes inferior to the recognition rates of character-based method when the number of writers is superior to 150. This may be explained by the fact that, when the number of writers is increased, the amount variability of the graphemes becomes huge and our systems fails to capture it. The performances of the grapheme-based system could be enhanced by considering different features (more fitted to the grapheme description) and/or hierarchical clustering (where the number of clusters may be automatically adjusted to the problem) instead of k-means.

We can also note that the method based on graphemes is more computationally expensive than the method based on characters, because the number of graphemes in a given document is much superior to its number of characters. Using a Core i7 processor with 4GB RAM, 120 writers, 45 features, 2 documents per writer as reference and 1 document per writer as test, the character-based system takes only 10 minutes for training and testing against more than 2 hours for the grapheme-based method. It has to be noted that using some hierarchical clustering method instead of k-means could also reduce the computational complexity of the grapheme-based approach.

Second experiment: This experiment aims at showing the effectiveness of the proposed approach in an adaptive cursive word recognition framework, where the word recognizer is adapted to the handwriting style of the writer. The handwriting styles are created by clustering (using k-means) the weight matrices obtained from our method based on graphemes. The main idea of our adaptive cursive word recognizer is that we train one word recognizer per handwriting style using the method in [1], but with increased weights for the words written by writers of this handwriting style, yet not removing the other writers. While the method in [1] includes two levels: the character level and the bi-character level, here adaptation is applied only at the character level. During the recognition stage, when a word has to be recognized, the cursive word recognizer corresponding to the writer's handwriting style is used. We work in a closed-world context. Using 300 writers and 30 documents per writer (from the IRONOFF database), we compute 9 handwriting styles clusters (the number of writers per cluster varies from 7 to 49). For each writer style, we use $\frac{2}{3}$ of the characters (segmented from cursive words using the method presented in [20]) to train the adaptive word recognizer and the remaining $\frac{1}{3}$ of the characters as a test set. Based on experiments, we use weights 7 times superior for the writers of the corresponding handwriting style.

Figure 5 compares the boxplots of our adaptive word recognizer towards the non-adaptive word recognizer (among the 9 handwriting styles). This figure shows that the mean recognition rate (represented by crosses) is increased of 2%

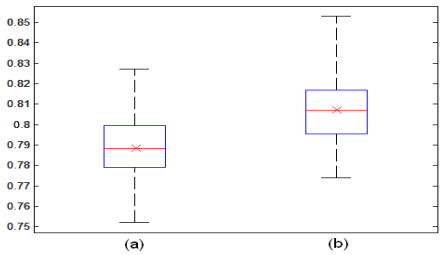

Figure 5. Boxplots of the cursive word recognizer over the 9 handwriting style clusters using (a) the generic word recognizer (without adaptation) and (b) the adaptive word recognizer.

when using the adaptive engine, even though the adaptation method is very basic as it only consists in modifying the weights of the samples. These results are very promising and show that our method may effectively enhance the recognition rates of our cursive word recognizer.

V. Conclusion

In this paper, we introduce two local writer recognition methods and show their effectiveness using cursive word databases. While the first one is based on characters, the second one is based on graphemes extracted from the documents. Each method has its own advantages and drawbacks. While the former is less computationally expensive and more effective when the number of writers is increased, the main advantage of the latter is that it is not subject to the systematic error introduced by the character segmentation/recognition module. As a future work, we plan to enhance the grapheme-based method and to reduce its computational complexity by using more adapted features and/or a hierarchical clustering method.

In this paper, we also show that a variant of the grapheme-based method may be used for adapting our cursive handwritten word recognizer to the different handwriting styles of the writers. Even though the adaptation method we use is very simple, experimental results using 300 writers are very encouraging. Another part of our future work will consist in conceiving a more effective adaptation scheme.

Acknowledgment

Grateful acknowledgement is made for financial support by the Poitou-Charentes Region (France).

References

[1] S. Prum, M. Visani, and J.-M. Ogier, "Cursive on-line word recognition using a bi-character model for large lexicon applications," in *International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 2010, pp. 194–199.

[2] R. Plamondon and G. Lorette, "Automatic signature verification and writer identification – the state of the art," *Pattern Recognition*, vol. 22, no. 2, pp. 107–131, 1989.

[3] A. Schlapbach, *Writer Identification and Verification. Dissertations in Artifical Intelligence*, 2008, vol. 311. 148 pages.

[4] S. N. Srihari, S.-H. Cha, H. Arora *et al.*, "Individuality of Handwriting," *Journal of Forensic Sciences*, vol. 47, pp. 1–17, 2002.

[5] A. Nosary, L. Heutte, T. Paquet *et al.*, "Defining writer's invariants to adapt the recognition task," in *International Conference on Document Analysis and Recognition (ICDAR)*, 1999, pp. 765–768.

[6] S. Connel and A. Jain, "Writer adaptation of online handwriting models," *IEEE Trans. on PAMI*, vol. 24, pp. 329–346, 2002.

[7] Z. Huang, K. Ding, L. Jin *et al.*, "Writer adaptive online handwriting recognition using incremental linear discriminant analysis," in *ICDAR*, 2009, pp. 91–95.

[8] H. E. S. Said, T. N. Tan, and K. D. Baker, "Personal identification based on handwriting," *Pattern Recognition*, vol. 33, pp. 149–160, 2000.

[9] M. Bulacu, L. Schomaker, and L. Vuurpijl, "Writer identification using edge-based directional features," in *ICDAR*, vol. 2, 2003, pp. 937–941.

[10] A. Bensefia, T. Paquet, and L. Heutte, "A writer identification and verification system," *Pattern Recognition Letters*, vol. 26, pp. 2080–2092, 2005.

[11] L. Schomaker and M. Bulacu, "Automatic writer identification using connected-component contours and edge-based features of uppercase western script," *IEEE Trans. on PAMI*, vol. 26, pp. 787–798, 2004.

[12] A. Schlapbach and H. Bunke, "Using HMM based recognizers for writer identification and verification," in *International Workshop on Frontiers in Handwriting Recognition (IWFHR)*, 2004, pp. 167–172.

[13] B. Li, Z. Sun, and T. Tan, "Online text-independent writer identification based on stroke's probability distribution function," in *Advances in Biometrics*, 2007, vol. 4642, pp. 201–210.

[14] S. K. Chan, Y. H. Tay, and C. Viard-Gaudin, "Online text independent writer identification using character prototypes distribution," in *International Conference on Information, Communications and Signal Processing*, 2007, pp. 1–5.

[15] G. X. Tan, C. Viard-Gaudin, and A. C. Kot, "Automatic writer identification framework for online handwritten documents using character prototypes," *Pattern Recognition*, vol. 42, no. 12, pp. 3313–3323, 2009.

[16] A. Imdad, S. Bres, V. Eglin *et al.*, "Writer identification using steered hermite features and svm," in *ICDAR*, vol. 2, 2007, pp. 839–843.

[17] R. Pareti and N. Vincent, "Global method based on pattern occurrences for writer identification," in *IWFHR*, 2006.

[18] I. Siddiqi and N. Vincent, "Text independent writer recognition using redundant writing patterns with contour-based orientation and curvature features," *Pattern Recognition*, vol. 43, pp. 3853–3865, 2010.

[19] A. Bensefia, A. Nosary, T. Paquet *et al.*, "Writer identification by writer's invariants," in *IWFHR*, 2002, pp. 274–279.

[20] S. Prum, M. Visani, and J.-M. Ogier, "On-line handwriting word recognition using a bi-character model," in *International Conference on Pattern Recognition (ICPR)*, 2010, pp. 2700–2703.

[21] C. Viard-Gaudin, P.-M. Lallican, S. Knerr *et al.*, "The ireste on/off (ironoff) dual handwriting database," in *ICDAR*, 1999, pp. 455–458.