

HANDWRITTEN TEXT RECOGNITION USING A MULTIPLE-AGENT ARCHITECTURE TO ADAPT THE RECOGNITION TASK

L. HEUTTE, T. PAQUET, A. NOSARY AND C. HERNOUX

Laboratoire PSI, Université de Rouen, F-76821 Mont-Saint-Aignan Cedex, France.

E-mail: Laurent.Heutte@univ-rouen.fr

This communication investigates the automatic reading of unconstrained omni-writer handwritten texts. It shows how to endow the reading system with learning faculties necessary to adapt the recognition to each writer's handwriting. In the first part of this communication, we explain how the recognition system can be adapted to a current handwriting by exploiting the graphical context defined by the writer's invariants. This adaptation is guaranteed by activating interaction links over the whole text between the recognition procedures of word entities and those of letter entities. In the second part, we justify the need of an open multiple-agent architecture to support the implementation of such a principle of adaptation. The proposed platform allows to plug expert treatments dedicated to handwriting analysis. We show that this platform helps to implement specific collaboration or cooperation schemes between agents which bring out new trends in the automatic reading of handwritten texts.

1 Introduction

Like a human reader, an automatic reading system should be able to meet two different requirements. It should have omni-writer capabilities in order to recognise any handwriting. It should also have mono-writer capabilities in order to take into account the potential fantasy of each writer. Therefore, learning machines to read any hand-written text requires of course sophisticated and highly adapted algorithms of pattern recognition but requires also to manage all together the various interpretation levels (i.e. from graphical level up to lexical and syntactical levels).

The human expertise in managing these different interpretation levels relies on some abilities of learning the current handwriting. The current recognition systems do not have these learning abilities and consider the recognition to be a pure omni-writer problem. They try to recognize handwritten words or letters one independently from the others in a sequential manner [20][22]. Two main approaches are used to perform the recognition of handwritten cursive words. The first, called analytical, is a data-driven bottom-up approach in which letters are recognized before a lexical analysis is performed [16][21]. To counteract the problem of letter segmentation before (without) recognition, several segmentation hypothesis must be managed which makes in return the letter recognition module more complex since it must be therefore able to reject the bad segmentation hypothesis. However, the final decision can only be taken by the lexical verification module. This scheme of recognition is also called segmentation/recognition. The second approach, called holistic, is a top-down approach with verification. In this approach, the segmentation into letters is counteract by recognizing a word in its

whole and by selecting word candidates in a lexicon. This approach leans either on the detection of holistic features in the word [1][17] or on the verification that some letters or parts of letters are present at some positions in the word [7]. In short, the first approach is well adapted to the recognition of words belonging to a large lexicon or even without lexicon; the second one is rather well adapted to limited-lexicon applications. Note that these two approaches can be combined to improve recognition [19]. Note also that some recent studies try to cope with the problem of handwriting variability by clustering handwritings into families of handwriting styles [3]; the recognizers are then trained for each specific family but a between-style choice is needed before the recognition phase to select the fitted recognizer. In a simplified manner, we can say that these systems lean on problem-specific recognition schemes but have however no on-line learning abilities which would enable them to adapt themselves to the current handwriting. For these reasons, the conventional systems still remain recognition systems but not reading systems.

The authors have introduced in [18] the concept of writer's invariants which can be defined as the set of similar patterns automatically extracted from the segmentation of a handwriting. They have shown that this concept allows to derive new contextual graphical knowledge that can be used to adapt the recognition task to a particular handwriting and allows to make robust decisions when neither simple lexical nor syntactical rules can be used as, for example, in the case of free lexicon unconstrained handwritten text recognition. In this communication, we explain (section 2) how the recognition system can adapt itself to the current handwriting to be recognized by exploiting the graphical context defined by the writer's invariants. We show that this adaptation is guaranteed by activating interaction links over the whole text between the recognition procedures of word entities and those of letter entities. In section 3, we justify the need of an open multiple-agent architecture to support the implementation of such a principle of adaptation. The proposed platform allows to plug expert treatments dedicated to handwriting analysis. We show that this platform helps to implement specific collaboration or cooperation schemes between agents which bring out new trends in the automatic reading of handwritten texts. Finally, some conclusions and future work are drawn in section 4.

2 Adapting the Reading Task to the Writer

To adapt itself to each specific handwriting, our automatic reading machine should be able to delay the reading of some words until more contextual knowledge (either symbolic or morphological) is gathered to disambiguate doubtful interpretations. This idea of adaptation requires a non-sequential (i.e. interactive) recognition scheme able to make treatments interact at different contextual levels (graphical, symbolic, lexical) in order to take coherent decisions with respect to all of these constraints.

Based on this idea, we introduce a knowledge modelling that takes into account the structure of handwriting and allows to highlight the data and their associated

type of knowledge. Considering that the whole text of the writer has been segmented into graphemes using well-known techniques encountered in the literature [2], each grapheme (corresponding to a letter or not) is characterized by:

- Intrinsic Morphological Knowledge (IMK): any knowledge that can be extracted from the grapheme pattern alone, such as a set of features detected on the grapheme image for example [14],
- Contextual Morphological Knowledge (CMK): any knowledge about the grapheme pattern that can be extracted from its environment, such as the invariant cluster the grapheme belongs to [18].

Now the following symbolic knowledge about each grapheme can be provided by different treatments:

- Intrinsic Symbolic Knowledge (ISK): any knowledge about the possible letter (label) that can be associated to the grapheme considered alone (e.g. obtained from IMK) using classical recognition schemes that exploit inter-writer invariants [14],
- Contextual Symbolic Knowledge (CSK): any knowledge about the possible letter that can be associated to the grapheme by referring to its context, such as the invariant cluster the grapheme belongs to using the hypothesis made about its neighbors. Symbolic knowledge can also be derived thanks to the use of the lexical constraints applied to the word the grapheme belongs to.

Interaction between different levels of knowledge can then be introduced using knowledge sources to model the desired interactive architecture as shown in figure 1 for word and grapheme levels.

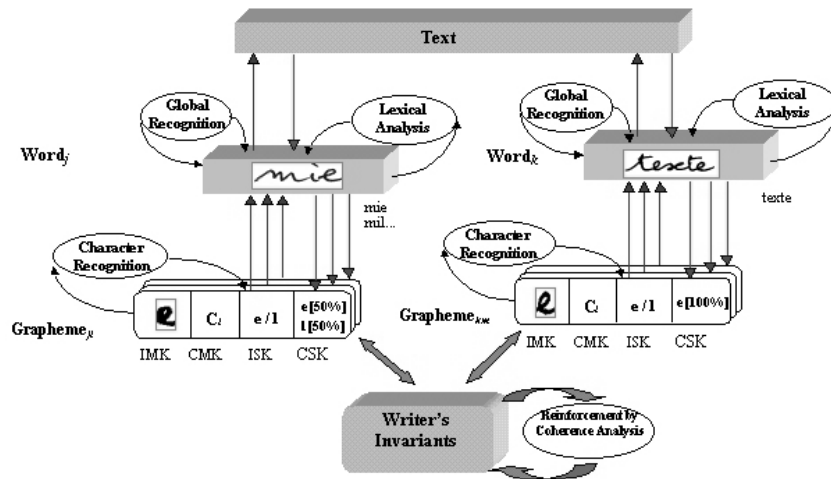


Figure 1: Illustration of the role of the writer's invariants in the interactive recognition system.

Assume that handwritten words have already been localized and that the segmentation into graphemes has been performed for each of them. Assume also that IMK and CMK have been extracted for each grapheme. Then the following links can be activated at the grapheme level. a) A character recognition procedure can provide ISK to each grapheme. b) These ISK of each grapheme can activate word level procedure. c) CSK for each grapheme can be derived from lexical constraints applied at word level. d) CSK of each grapheme can also be derived from its morphological neighbors (the invariant cluster (C_i) it belongs to) e) Global CSK of each grapheme can provide symbolic hypothesis for a writer invariant. f) A coherent analysis of each invariant cluster can reinforce the similar letter hypothesis for the similar patterns.

Assume for example that a lexical analysis cannot disambiguate the letter hypothesis e and l for the $grapheme_{jl}$. Then thanks to the writer's invariants it is possible to refer to the letter hypothesis made on $grapheme_{km}$ that belongs to the same cluster but occurs in a different lexical context $word_k$. Then since there is no ambiguity in letter hypothesis of $grapheme_{km}$ due to its lexical context, letter hypothesis on $grapheme_{jl}$ can be disambiguated by means of the writer's invariants. The activation links described above provide a general framework that can be used to implement various strategies in the reading system. Depending on the strategy used, a global coherence of the recognition hypothesis can be reached at each of the two interpretation levels (Word, Grapheme). Note that the same principles of interaction could be applied between text level and word level thanks to the use of syntactical constraints.

3 Implementation within the multiple-agent paradigm

The previous model shows the interest of a new organisation of the treatments. If, on the one hand, the data modelling can be considered relatively fixed during the whole resolution of the problem, on the other hand, the order in which the treatments are launched constitutes an important parameter of the proposed model that will directly influence the convergence of the system towards a satisfactory solution. The proposed model states that words sharing some common elementary patterns should not be recognised independently from each other. This approach is closely related to a classical problem of scene analysis [4]. In the literature, this approach is solved using constraints relaxation for object labelling. Let us recall that it can be decomposed into two phases, one dedicated to the labelling of objects, and the second dedicated to the determination of compatibility coefficients. In the present case, one can consider the problem of text recognition as one of graphemes labelling under lexical and graphical constraints. Within this framework of scene analysis, the proposed approach is rather natural. However, because of the specificity and the variability of handwriting, we think that the convergence of the relaxation process depends on a global strategy of resolution; we should say a

strategy of reading. Depending on the current objective, various strategies could be applied to drive the relaxation process.

Let us recall that the paradigm of distributed artificial intelligence has already been proposed for handwriting recognition [1] in order to proceed to the recognition of isolated hand-written words, using a blackboard architecture. Also notice that a same approach has been developed long ago, in the field of speech recognition [12]. Since the blackboard approach is dedicated to launching knowledge sources with increasing abstraction level as soon as new knowledge become available, the choice of the appropriate knowledge source becomes the central problem in real applications. Some studies have proposed therefore the use of a second blackboard to solve the control problem [11]. More recently, new distributed architectures have been proposed from the primary approach of [15] and have led to the multiple-agent paradigm [6]. Briefly speaking agents are entities that have the ability to communicate with other agents in the same environment, have an autonomous behaviour that allows them to act according to their own goal and knowledge about their environment. This model allows distributed control, and therefore is much more adapted to the implementation and the test of various strategies of control, e.g. various strategies of reading, when replaced in our context of constraints relaxation for hand-written text recognition. However, since it is necessary, for agents to cooperate, to share the same common data about the problem, we have found that the major drawback of the multiple-agents architecture is the necessity for agents to incorporate the data into control messages.

With this perspective in mind, we have built an open platform called EMAC [13] that allows to plug expert treatments dedicated to handwriting analysis with the ability to share a common distributed workspace. This platform also gives general tools for experts to communicate either with each other thanks to the use of agent-based communication language KQML or to broadcast messages among a group of experts thanks to the presence of a broker. The broker also allows to inform agents as soon as new information become available in the workspace.

3.1 The EMAC model of organisation

The EMAC model of the organisation of agents corresponds to groups of experts that have the ability to be notified of the occurrence of particular events thanks to the presence of a broker within each group. This allows each agent to broadcast messages over the whole group. Note that each agent can belong to several groups if necessary, or on the contrary, it can remain alone. The number of agents per group is unrestricted and the number of groups as well. Each agent can become member of a group as soon as it has declared itself to the broker.

The communication can also take place between two agents (that belong either to the same group or to different groups) thanks to simple message passing. This is an efficient way for agents to collaborate when one of them knows the existence of the other.

At last, one of the most remarkable aspects of the EMAC model of organisation and resource is the presence of common workspaces that allows agents to reach a particular information about the problem without resorting to the use of communication links. Figure 2 gives the global overview of the EMAC model of organisation.

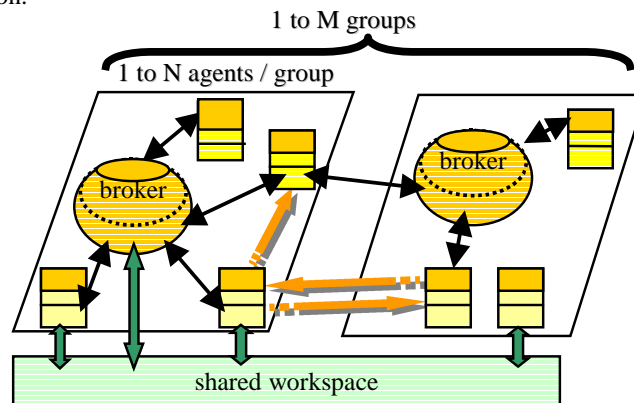


Figure 2: The EMAC model of organisation.

3.2 Internal organisation of the EMAC agent

An EMAC agent has a static description that is constituted by a set of constant characteristics such as its name and a list of its abilities to communicate, analyse messages, access to common workspace, and finally to make a particular expert treatment depending on the kind of application.

The dynamic behaviour of the EMAC agent is due either to external solicitations or to internal goals fixed by the programmer of the application. Furthermore, the dynamic behaviour depends on the organisation of the capacities and knowledge of the agents. These are organised as follows (figure 3):

- **Communication abilities:** Messages are temporary queued before being analysed by the agent by resorting to one of his abilities. Messages to be sent are also queued before being sent.
- **Control abilities:** These capacities are the motor of the dynamic behaviour of the agent which is constituted by deliberation-action infinite loop. This loop includes the analysis of the current received message as well as the format of new messages to be sent.
- **State of the agent:** It is constituted by the set of current knowledge either local to the agent or shared with others as well as the knowledge of its environment, in our case the knowledge of peer addresses.

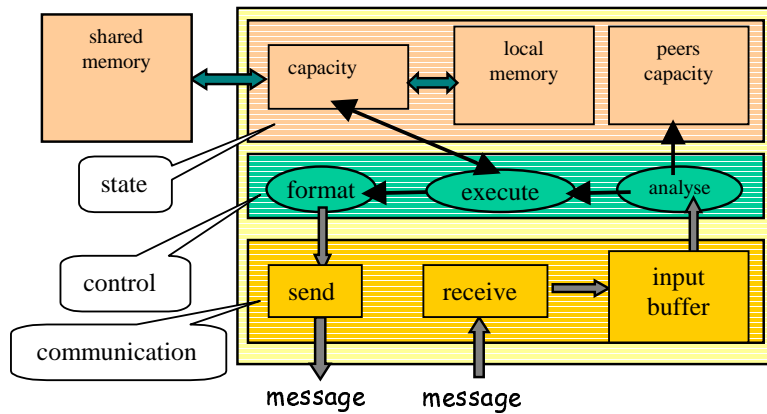


Figure 3: Internal model of the EMAC agent.

3.3 Implementation of the EMAC multiple-agent model

The EMAC model has been implemented in C++ so as to provide the user with an EMAC agent class that integrates the set of the basic capacities described previously. When used in a specific application, a particular agent class will then inherit the EMAC agent class as well as particular classes of expert treatments, for example particular treatments dedicated to handwriting analysis.

Since the EMAC model resort to a dynamic behaviour, there was a need to choose a suitable environment able to manage communication links, to execute methods of each agent, ensure the sharing of a common workspace and at last provide agents with a standard communication language. All of these are well-known problems in the multiple-agent community and have been the object of numerous propositions [8].

The actual platform for EMAC is based on the PVM powerful architecture [10] dedicated to parallel computing in a virtual environment made by the connection of multiple machines, and is briefly described in figure 4. Each EMAC agent is then implemented by a PVM task and can take benefit of the communication tools provided by this environment for sending and receiving ascii messages. The communication standard language between agents is KQML [9], and therefore each agent has the capacity to analyse a KQML message and to act according to the predefined performatives of this language. The last tool used in EMAC is the distributed and shared memory tool implemented using DREAM [5]. This tool ensures the sharing of the same virtual addressing space among a set of UNIX like systems. Furthermore it provides a programmable time refreshment of shared regions of memory between all the systems.

We think that the choice we have made yet for a particular implementation of our model could easily evolve towards the Java environment.

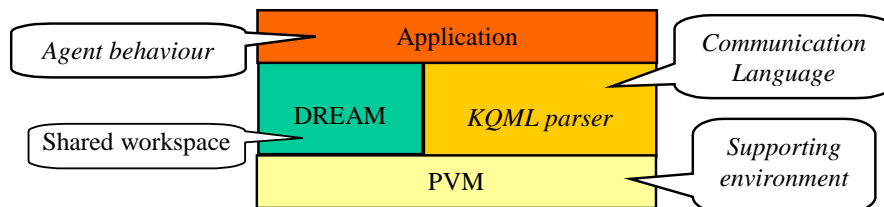


Figure 4: The EMAC current architecture.

3.4 Using EMAC

At present, three groups of agents are implemented within the EMAC platform. These are experts in grapheme analysis and recognition, handwritten word recognition and text segmentation. Within each group, multiple agents are dedicated to expert treatments while one of them is attached to control. The *text group* includes the following agents: segmentation into lines of text, segmentation into graphemes, segmentation into words. The *grapheme group* includes writer's invariant determination, feature extraction and letter recognition agents. The *word group* includes agents for Viterbi-based word recognition, word verification in a dictionary, deriving new grapheme scores from a list of word candidates.

Notice that among the various approaches devoted to hand-written word recognition presented earlier, we have chosen a bottom-up approach since in the context of text recognition we are faced to a large vocabulary.

Recognition modules are at present under individual evaluation. Within the EMAC platform, control strategies will be easily implemented thanks to the use of a high level language for communication and the presence of an agent dedicated to control within each group. Various strategies of interaction i.e. reading strategies, will be evaluated. The following gives an overview of the various parameters on which a particular interaction scheme is based.

Let us recall that the proposed adaptation scheme relies on the interaction of lexical and graphical constraints. Interaction can take place according to the following scheme:

1. Select words for recognition
2. Update current interpretation of each grapheme
3. Evaluate best word candidates for the selected words
4. Repeat step 1 to 3 until all words have been processed

Step one, devoted to the selection of words, is the central point of a particular strategy. A good rule would consist in the selection of words for which either lexical or graphical constraints are a priori known to bring the larger amount of information to disambiguate between word candidates.

We now give some examples of selection rules:

- The default rule consists in the selection of every words in the text. This corresponds to a classical relaxation scheme and is probably the worst interaction rule.
- A second rule consists in the selection of words of increasing length during the repeating loop. Indeed, since short words are known to be less numerous for a particular dictionary, we expect that lexical constraints will disambiguate word hypothesis in an effective way in this case. Similarly we could select long words as well.
- A third rule can be implemented based on graphical constraints. Indeed we can expect that words with graphemes that belong to big groups of invariants will be easier to recognise.

This gives rise indeed to a wide range of interaction strategies to evaluate. Furthermore, the proposed architecture could be extended in a similar way in order to integrate higher level of interpretation such as syntactical or semantic rules.

4 Conclusion and future work

In this paper we have proposed a new approach to the difficult task of reading hand-written texts. Thanks to the concept of the writer invariants, we have introduced new graphical constraints that will help the recognition of letters in addition to common lexical constraints which enable to disambiguate recognition. These new constraints give new perspectives in the field of hand-written analysis since similar patterns cannot be recognised independently from each other. This paradigm is very similar to the well known approach of scene analysis and provide a reading system with the ability to adapt itself to the current handwriting.

This paradigm also gives rise to the question of how to make interact both lexical and graphical constraints. We believe that reading strategies will bring part of the answer to this problem. For this purpose of testing reading strategies, we have designed a multiple-agent platform able to make groups of agents collaborate. This tool will help to implement specific collaboration schemes between agents using the high-level communication language KQML.

5 References

1. P.E. Bramal, C. A. Higgins. A cursive recognition system based on human reading models. *Machine Vision Application*, Vol. 8, pp. 224-231, 1995.
2. G. Casey, E. Lecolinet. A survey of methods of segmentation. *IEEE Trans. on PAMI*, Vol.18, No.7 pp. 690-706, 1996.
3. JP Crettez. A set of handwriting families: style recognition. *Proc. ICDAR'95*, Montreal, Canada, pp.489-494, 1995.
4. R.O. Duda, P.E. Hart, *Pattern Classification and Scene Analysis*. Wiley, 1972.

5. C. Dumoulin. DREAM: Une mémoire partagée répartie à cohérence programmable. Thèse de doctorat de l'USTL, Lille, France, 1997.
6. E.H. Durfee, T.R. Lesser, D.D. Corkill. Coherent cooperation among communicating problem solvers. *IEEE T. on Computers*, pp. 1275-1291, 1987.
7. C. Farouz, M. Gilloux and J.M. Bertille. Handwritten word recognition with contextual hidden markov models. *Proc. 6th IWFHR*, Korea, pp. 133-142, 1998.
8. J. Ferber. Les systèmes multi-agents, vers une intelligence collective, InterEditions, Paris, 1995.
9. T. Finin, D. McKay, R. Fritzson, R. McEntire. KQML : an information and knowledge exchange protocol, in *Knowledge Building and Knowledge Sharing*, Ohmsha and IOS Press, 1994.
10. A. Geist, A. Geguelin, J. Dongarra, W. Jiang, R. Manchek, V. Sunderman. PVM: Parallel Virtual Machine, A user's guide and Tutorial for Networked Parallel Computing. The MIT Press Cambridge, London, England, 1994.
11. B. Hayes-Roth. Blackboard architecture for control. *Artificial Intelligence*, 26, pp. 251-321, 1985.
12. L. D. Herman, F. Hayes-Roth, V. R. Lesser, R. D. Reddy. The HEARSAY-II Speech Understanding System: Integrating Knowledge to Resolve Uncertainty, *ACM Computing Surveys*, 12, pp. 213-253, 1980.
13. C. Hernoux. EMAC, Un environnement Multi-Agents à mémoire Collective. Mémoire d'ingénieur, CNAM, CRA de Rouen, Juin 1999.
14. L. Heutte, T. Paquet, J.V. Moreau, Y. Lecourtier, C. Olivier. A structural/statistical feature based vector for handwritten character recognition. *Pattern Recognition Letters*, vol. 19, no. 7, pp. 629-641, 1998.
15. C. E. Hewitt. Viewing Control Structures as Pattern of passing Messages. *Artificial Intelligence*, vol. 8, pp. 323-364, 1977.
16. G. Kim, V. Govindaraju. A lexicon driven approach to handwriting word recognition for real-time application. *IEEE PAMI* 18, no. 4, pp. 366-379, 1997.
17. A. Leroy. Correlation between handwriting characteristics. In *Handwriting and Drawing Research: Basic Applied Issues*, M. L. Simner and C. G. Leedham, pp. 403-417, 1996.
18. A. Nosary, L. Heutte, T. Paquet and Y. Lecourtier. Defining writer's invariants to adapt the recognition task. *Proc. ICDAR'99*, India, pp. 765-768, 1999.
19. B. Plessis, A. Sicsu, L. Heutte, E. Menu, E. Lecolinet, O. Debon and J.V. Moreau. A multi-classifier combination strategy for the recognition of handwritten cursive words. *Proc. ICDAR'93*, Japan, pp. 642-645, 1993.
20. S.N. Srihari. Recognition of handwritten and machine-printed text for postal address interpretation. *Pattern Recognition Letters* 14, no. 4, pp. 291-302, 1993.
21. M. Shridhar, G. Houle, F. Kimura. Handwritten word recognition using lexicon free and lexicon directed word recognition algorithms. *Proc. ICDAR'97*, Germany, pp.861-865, 1997.
22. Y.Y. Tang, S.W. Lee, C.Y. Suen. Automatic document processing: a survey. *Pattern Recognition* 29, 12, pp. 1931-1952, 1996.