A Scenario Model Advocating User-Driven Adaptive Document Recognition Systems

F. Bapst A. Zramdini R. Ingold
Institute of Informatics, University of Fribourg
Chemin du Musée 3, CH-1700 Fribourg (Switzerland)

Abstract

Assisted document recognition systems have to integrate automatic recognition, manual edition and incremental learning in a single interactive environment. This paper raises the question of the organization of these three kinds of operations. When an analyzer has the ability to improve with use, there is a tradeoff between the benefits of enhancing the accuracy of automatic analysis, and the additional time spent in interacting for feedback communication. The global cost depends then on the sequence of processed entities, and on the relevance of the learning transactions. Notations are introduced to describe the evolution of a recognition session, and possible organization strategies are discussed. Then a cost model is presented to allow the comparison between different organization schemes. We describe some concrete experiments of cost measures with the $ApOFIS$ font identification tool and the ScanWorX OCR; the first results show that a user-driven approach can potentially save substantial effort in the recognition process, in comparison with machine-driven systems.

1 Introduction

Current document recognition systems have shown their limitation: except for extremely confined applications, fully automatic recognition is an illusion, and the post-correction effort is often underestimated. The *CIDRE* [1] project[2] [1] consists of a general revision of assisted document recognition. One of its leitmotivs is to aim at a *widely* usable system, i.e. without being dedicated to a particular application, and able to *improve with use*. What makes current systems tedious to use is the prominent apparition of *repetitive* errors, especially when applied on documents not explicitly foreseen by the designers. Two remedies have been proposed to reduce this phenomenon: (i) forcing the user to start with an initial training phase, where the tool is in some sense re-tuned for the new application, or (ii) offering some incremental learning facilities for feedback exploitation. *CIDRE* advocates the second solution and insists on cooperative relationships between the recognition system and the human operator. We no more believe in systems where the role of the user is restricted to the post-correction of mistakes.

In such an adaptive environment, an average recognition rate is no more sufficient to assess the quality of one analyzer, because it suggests that only correction interventions are influencing the final costs, whereas the user can also intervene for learning purposes. The time spent trying to improve the tool has to be reckoned with, and this leads to a tradeoff between the efforts serving directly the recognition process (correction, edition), and those only indirectly useful through the enhancement of the automatic analysis. This paper is devoted to this dilemma under the term *session organization*, and has the following goals: (i) to bring a piece of model to structure the discussion about the vague notions of tasks organizations, intervention costs, organization strategy or learning power; (ii) to illustrate, with a few concrete experiments and thank to our notations, that a user-driven approach is more efficient than other session organizations.

2 Tasks Organization Modeling

2.1 Document Analysis – Basic concepts

The application we deal with for this paper is the physical structure recognition of document images. In this context of document analysis, an entity e is a tree node (block, line, word) which supports three different interpretations corresponding to the basic tasks: (i) *character interpretation*, which attaches a text string to e; (ii) *font interpretation*, which attaches a font to e; (iii) *segmentation interpretation*, which either rearranges the subtree for e, or attaches it an envelope. Each of these interpretations is qualified by a *status* used to decide if that result is considered as explicitly certified by the user, to what extend the system trusts it (confidence degree), or whether it is still unknown.

The scenario followed during one evolution of the system is called a *session* and is composed of a sequence of basic operations (transactions). Our simple model defines nine symbols as the basic alphabet on which we build the notation for sessions, described in table 1. There are three transactions types, applied to every interpretation fields:

- *automatic analysis* – $A \in \{A^c, A^f, A^s\}$;
  the system owns three analyzers, one for each interpretation to compute (character, font, segmentation);

[1] For Cooperative & Interactive Document Reverse Engineering.

[2] This project is supported by the Swiss National Fund for Scientific Research, code 21-42'355.94.

their input parameters are the entity to be analyzed, the knowledge base to be used, and maybe some other configuration options;

- *manual edition* – $M \in \{M^c, M^f, M^s\}$;
  the human operator can directly modify any part of the current solution (editing a string, choosing a font, segmenting by hand). Let's point out that the detailed commands are GUI dependent;
- *incremental learning* – $L \in \{L^c, L^f, L^s\}$;
  the analyzers offer a facility to update their knowledge base in order to integrate a known (e.g. corrected) solution.

Expression	Comments
$(P_1 P_2), (P_1 \cup P_2)$	sequence and alternative
$(P_1)^*$	iterative closure
$p[A^c]$	OCR on the *specific* entity p
$Page[A^c]$	*generic* pattern for OCR on a page
A^c	abbr. for $Any[A^c]$
$e_1[A^c]e_2[M^f]$	sequence (first OCR on e_1)
$(e_1, e_2)[A^c M^f]$	abbr. for $e_1[A^c]e_1[M^f]e_2[A^c]e_2[M^f]$

Table 1: *Pattern notations for the session concept.*

2.2 Organization strategies

Batch: Among the three extreme approaches summarized in table 2, the most used is the batch one. The system does not offer incremental learning. Thus all the computation is grouped as a first step in the session, then we enter an edition mode where the user can correct the solution. In the state of the art, this is the default approach, found in most commercial systems. More precisely, these recognition systems are yet embedded in interactive environments, but no incremental learning is really integrated. Interactivity is typically used to correct the segmentation before applying OCR. Batch patterns belong to $\mathcal{P}_{batch} \subset Volume[AM]$.

Assisted, Machine-Driven: The system chooses the sequence of entities (typically read order) and the analyzers to apply. The user is solicited after each analysis to correct the solution, and optionally feeds incremental learning every time. At least one project adopted this approach: Stabler [7] describes a system specifically developed for high-volume data capture enabling 100% accurate recognition of patent documents. AMD patterns belong to $\mathcal{P}_{amd} \subset (A \cup (AM) \cup (AML))^*$.

Assisted, User-Driven: The system is completely under the control of the user, who applies recognition, learning or manual edition whenever he wants, on whatever entity. Our *CIDRE* project seems to be the first attempt to deliberately advocate the user-driven approach. Note that it is a generalization of the two others, because nothing prevents the user to operate in a systematic way. AUD patterns belong to $\mathcal{P}_{aud} \subset (A \cup M \cup L)^*$.

		BATCH	AMD	AUD
User...	role	passive	reactive	active
	feedback	no	yes	yes
	experience	useless	useless	useful
Pattern...	granularity	fixed	fixed	variable
	length	2	n (known)	m (unknown)
	traversal	atomic	imposed	free

Table 2: *Three extreme session organizations.*

2.3 Cost model for user interventions

Manual interventions and costs: The definition of a good cost model for document recognition is extremely hard, because of (i) the dependence from a concrete front-end, (ii) the influence of the context of each punctual correction, and (iii) the gap between theory and pragmatic costs. The previous endeavors of cost models addressed only OCR [2, 4, 8]; they focused essentially on string edition and not on the global process, but some practical results were found, like the notion of *optimal rejection rate* [3]. Here we restrict the discussion to the costs of human interventions. Calling one analyzer for recognition or learning purposes is supposed to cost a constant amount. What needs to be modelled is the cost of editing operations.

Editing cost model: The cost of updating manually the current solution will depend on the nature of the edited result (typing characters, selecting font attributes, defining boundings or rearranging), and on the result status, because it is cheaper to detect an error if that result was flagged as uncertain. Our proposition of cost model is summarized in table 3. Let's define W_e^t as the set of errors made by analyzer $t \in \{c, f, s\}$ on entity e, or more precisely the set of descendants needing to be edited. We need to make a distinction between the cost of an *atomic* correction δ_e^t, and the cost of correcting a whole subtree Δ_e^t.

Expression and Comments	
W_e^t	set of $e_i \in tree(e)$ having a wrong value for task $t \in \{c, f, s\}$
δ_e^t	edition cost for correcting the *atomic* error on e for task t
Δ_e^t	$= \sum_{w \in W_e^t} \delta_w^t$ — edition cost for correcting *all* errors in $tree(e)$ for task t
Δ_e	$= \sum_{t \in \{c,f,s\}} \Delta_e^t$ — total editing cost for correcting all errors in $tree(e)$ for all tasks
λ	error detection cost, function of status of wrong result
φ_e	nb of necessary insertions, deletions, and substitutions for correcting the segmentation of e
δ_e^s	$= \varphi_e \gamma + \lambda$ — segmentation edition cost
δ_e^f	$= \beta + \lambda$ — font edition cost
δ_w^c	$= \alpha + \lambda$ — for OCR, atomic correction concerns words

Table 3: *Notation for edition cost model.*

Session cost model: The definition of an edition cost model gives only a static view of user interventions. The dynamic aspects are very important because we want to compare various organizations of tasks in whole sessions.

The crucial phenomenon is that the use of incremental learning will hopefully reduce the amount of manual editing (there will be less errors), at a price proportional to the number of requested learning transactions. Another factor is the choice of entity granularity. The first step is to refine the definition of the errors set W_e^t so that it is parameterized with the sequence $e_1 ... e_n$ of learned samples. This paradigm can serve to discuss the properties that the learning functionality should ideally guarantee (cf. Table 4). [3]

Property	Comments
$W^t_{e,<e>} = \emptyset$	total efficiency
$W^t_{e,<ei,ej>} = W^t_{e,<ej,ei>}$	commutativity
$W^t_{e,<ei>} = W^t_{e,<ei,ei>}$	same re-learning
$\lvert W^t_{e,<>}\rvert \geq \lvert W^t_{e,<ei>}\rvert$	always positive effect
$W^t_{e,<e_1 ... e_n>} \subset W^t_{e,<e_1 ... e_{n-1}>}$	stability

Table 4: *Learning properties* **not** *always guaranteed.*

Then, we can make the link between edition cost, learning effect, and number of analysis requests, in order to define the cost of a complete session pattern, as stated in table 5. We define the cost $C(o \S P)$ of one transaction o so that it depends on the session history P. This is not the case for analysis or learning calls, which are supposed to have a constant cost u, regardless the processed entity. But it is of highest importance for manual edition, because the correction effort $C(e[M^t] \S P)$ depends on the errors $W^t_{e,<e_1 ... e_n>}$ made by the analyzer, which in turn depend on the knowledge learned so far.

Expressions — Comments
$W^t_{e,<e_1 ... e_n>}$ — errors set when task t learned e_1 till e_n.
$C(o \S P)$ — cost of operation o when applied after pattern P.
$u = C(e[A] \S P) = C(e[L] \S P)$ — cost of requesting $\forall P, \forall e$.
$C(e[M^t] \S e_1[L^t] ... e_n[L^t] e[A^t]) = \sum_{e_i \in W^t_{e,<e1 ... en>}} \delta^t_e$ — manual edition cost depends on the preceding learning transactions, which influence the number of mistakes to correct.
$C(M^t \S P)$ — is defined by extension for any P.
$C^M(P)$,$C^A(P)$, $C^L(P)$ — total cost of M resp. A, L transactions.
$C(P) = C(o_1 ... o_n) = \sum_{1 \leq i \leq n} C(o_i \S o_1 ... o_{i-1})$ $= C^M(P) + C^A(P) + C^L(P)$ — total cost of pattern P.

Table 5: *Notations for session cost model.*

In our model, optimizing the organization means finding a pattern P covering all specific entities that minimizes the total intervention cost represented by $C(P)$. Two antagonist factors take part in this minimum: (i) the *best* use of learning transactions so that the cost of manual correction is low; (ii) the length of P because the cost increases with the number of A and L operations. The cost of the pattern P_{batch} is the correction cost $C(v[M] \S v[A])$ on a volume v, plus an initial launching cost u. The cost of a pattern P_{amd} following the assisted, machine-driven approach, contains a fixed part xu depending on the predefined granularity of processing (e.g. lines), a variable part yu proportional to the number of learned corrections, and the edition cost $C^M(P_{amd})$ indirectly influenced by the traversal politics (e.g. reading order) via learning effect. For the assisted, user-driven approach, it is the user responsibility to aim at the ideal pattern $P_{opt} \in \mathcal{P}_{aud}$ so that $C(P_{opt})$ is minimum.

[3] One can also derive a normalized function for the learning power of e_1 on e_2, like $(\lvert W^t_{e_2,<>}\rvert - \lvert W^t_{e_2,<e_1>}\rvert) \div (\lvert W^t_{e_2,<>}\rvert - \lvert W^t_{e_2,<e_2>}\rvert)$.

3 Empirical Evaluation

3.1 Evaluation methodology

The advantages of the assisted, machine-driven organization over the batch one have already been motivated [7]. What needs still to be evaluated is the potential benefits of the "free" approach. The goal here is only to acquire some hints with two test packs, where we suppose that the cost of a manual edition is directly proportional to the number of errors in that entity ($\lambda = 0$ and $\alpha = \beta = 1$). Let's define the set $\mathcal{P}_{good} \subset \mathcal{P}_{aud} = \{P_i$ so that $C(P_i) \leq C(P_{amd})\}$. We want to verify practically the following expected assertions:

- $|\mathcal{P}_{good}| \gg 0$, which means that there is a lot of session patterns better than the assisted, machine-driven versions;
- $C(P_{opt}) \ll C(P_{amd})$, which means that the best patterns are significantly better than the assisted, machine-driven versions;
- $\mathcal{P}_{good}$ contains some "intuitive" patterns, in the sense that the user will be able to find one of $P_k \in \mathcal{P}_{good}$ with a bit of common sense and experience; it is highly important, but harder to quantify!

3.2 Experiments with $ApOFIS$

The font recognition system $ApOFIS$ [9] offers strong learning capabilities. The experiments consisted in computing the cost of a few patterns used to produce an error-free document (from the OFR point of view). The sample document used within the experiments represents a list of author references, each one composed of three entities that are distinguished by their fonts.

Table 6 lists the total costs of recognizing the document following the three main strategies: batch (P_{batch}), machine-driven (three alternatives P^i_{amd}) and user-driven (three alternatives P^i_{aud}). For P_{batch} and P^i_{amd}, the document is traversed from the machine's point of view. We supposed for P^i_{amd} that the granularity was fixed at block level, which means that the user is presented the results block by block and may then correct and possibly learn the individual words. In P^1_{amd} every correction is learned, whereas P^2_{amd} and P^3_{amd} represent different learning subsets. In the user-driven approach, the document is seen from the user's point of view with different granularities. The table shows that the intuitive P^i_{aud} scenarios are "cheaper" than the P^j_{amd} and P_{batch} ones. For instance, scenario P^3_{aud} generated a cost saving of about 27% and 39%, over P^2_{amd}, resp. P_{batch}.

Scenario and Pattern (P)		Cost			
		C^A	C^M	C^L	C
P_{batch}	$Words[A]\ Words[M]$	1	68	-	69
P^1_{amd}	$Blocks[A \cup (AML)]$	14	23	23	60
P^2_{amd}	$Blocks[A \cup (AML) \cup (AM)]$	14	29	15	58
P^3_{amd}	$Blocks[A \cup (AML) \cup (AM)]$	14	31	18	63
P^1_{aud}	$Blocks[A \cup M \cup L]$	13	27	9	49
P^2_{aud}	$Blocks[A \cup M \cup L]$	13	23	20	46
P^3_{aud}	$Blocks[A \cup M \cup L]$	13	23	16	42

Table 6: *Session costs with* $ApOFIS$.

3.3 Experiments with ScanWorX

ScanWorX [5] (developed by Xerox) is a quite representative commercial OCR system, which offers a possibility of accumulating recognition knowledge. In the experiments, we will now focus only on the accumulated number of errors made by the analyzer. Our test data set is limited to 10 pages of scientific papers, taken from the UW-III database [6], and processed at the block level. For each of our page samples, we first built the character ground-truth and the block segmentation, and then performed the following concrete experiments: (i) a whole session without any learning, which corresponds to the batch approach; (ii) a whole session following the assisted, machine-driven approach, on the sequence of blocks ordered by read-order; (iii) all couples of blocks with the local pattern $e_i[L^c]e_j[A^c]$, in order to fill a matrix showing the learning benefit of each block on each other ($|W^c_{e_i,<>} \setminus W^c_{e_i,<e_j>}|, \forall e_i, \forall e_j$).

The first impression that comes out the results is the relatively great sensitivity of the recognition process, and a few tries are necessary to understand the system reactions. Table 7 shows the total edition cost of some session patterns, for the most characteristic samples. We can see that assisted, machine-driven is always better than batch, which means that the average effect of learning is positive. The patterns $P_{opt'}$ and $P_{worst'}$ were derived from the $W^c_{e_i,<e_j>}$ matrix, so that they exploit the extreme punctual learning for each block; thus they give a lower resp. upper bound for the minimal resp. maximal cost patterns. We observe that there is really a substantial cost improvement to gain from systematic methods (21-39%). The detailed analysis

		Samples				
		V008	V00E	W0UA	W0U9	OFR1
Patterns	$P_{worst'}$	102	130	52	62	61
	P_{batch}	60	105	50	54	43
	P_{amd}	55	78	42	42	41
	$P_{opt'}$	43	54	31	30	25

Table 7: *Edition costs with ScanWorX.*

of the $W^c_{e_i,<e_j>}$ matrix confirms another intuitive statement: learning is more beneficent when applied among entities of the same font. So we find good reasons to go towards a monofont use of ScanWorX, i.e. the separation of different learning files according to the font. This gives even a motivation to prefer monofont OCR softwares.

4 Conclusion and Perspectives

This paper discussed the integration of incremental learning with automatic analysis and manual edition, in document recognition. Our contribution is twofold:

- we tried to present the problematic in a formal way: (i) notations were introduced to encode session patterns, (ii) possible organization strategies were positioned thank to the definition of three extreme schemes, and (iii) a complete cost model has been proposed;
- the direction towards an empirical evaluation of organization quality has been sketched. As a first step on that way, we conducted several tests with both an OCR and a font identification package.

The results obtained in our practical experiments encourage us to follow up the original motivations of the *CIDRE* project, because they showed that the human operator can potentially save much intervention when he can freely drive the recognition session, guided by his feeling and experience. We found that *many* patterns are *much* better than machine-driven schemes, and that *common sense* is helpful to find them.

We also are convinced that the developers should take a greater care to offer incremental facilities with existing analyzers. As a general trend, we argue that using simple recognition techniques and making most efforts in the coherent integration of components will finally be more beneficent than over-optimizing algorithmic details but leaving a hermetic interface. There is only little opportunities to increase the accuracy of each existing analyzers considered individually, but great improvements are expected from a relevant system embodiment, where all of them cooperate with the user in a coherent way.

References

[1] F. Bapst, R. Brugger, A. Zramdini, and R. Ingold. Integrated multi-agent architecture for assisted document recognition. In *DAS'96*, pages 172–188, Malvern, Pennsylvania, October 1996.

[2] R. Bradford and T. Nartker. Error correlation in contemporary OCR systems. In *ICDAR'91*, pages 516–524, 1991.

[3] C. K. Chow. Recognition error and reject trade-off. In *Symposia on Document Analysis and Information Retrieval (SDAIR'94)*. University of Nevada Las Vegas, 1994.

[4] J. Esakov, D. P. Lopresti, J. S. Sandberg, and J. Zhou. Issue in automatic OCR error classification. In *Symposia on Document Analysis and Information Retrieval (SDAIR'94)*, 1994.

[5] B. Paddock and T. J. Platt. *ScanWorX API, Programmer's Guide*. Xerox Imaging Systems, Inc., 9 Centennial Drive, Peabody, Massachusetts 01960, 1992.

[6] R. P. Rogers, I. T. Phillips, and R. M. Haralick. Semiautomatic production of highly accurate word bounding box ground truth. In *Document Analysis Systems (DAS'96)*, pages 375–386, 1996.

[7] H. R. Stabler. Experiences with high-volume, high-accuracy document capture. In *Document Analysis Systems (DAS'94)*, 1994.

[8] K. Taghva, J. Borsac, B. Bullard, and A. Condit. Post-edition through approximation and global correction. In *Annual report*, pages 57–70. UNLV Information Science Research Institute, 1993.

[9] A. Zramdini. *Study of optical font recognition based on global typographical features*. PhD thesis, IIUF-Université de Fribourg, 1995. n. 1106.