2011 International Conference on Document Analysis and Recognition

Exploiting Collection Level for Improving Assisted Handwritten Word Transcription of Historical Documents

Laurent Guichard - Joseph Chazalon - Bertrand Coüasnon
INSA de Rennes, Avenue des Buttes de Coësmes, F-35043 Rennes
UMR IRISA, Campus de Beaulieu, F-35042 Rennes
Université Européenne de Bretagne, France
{*laurent.guichard,joseph.chazalon,bertrand.couasnon*}*@irisa.fr*

***Abstract*—Transcription of handwritten words in historical documents is still a difficult task. When processing huge amount of pages, document-centered approaches are limited by the trade-off between automatic recognition errors and the tedious aspect of human user annotation work. In this article, we investigate the use of inter page dependencies to overcome those limitations. For this, we propose a new architecture that allows the exploitation of handwritten word redundancies over pages by considering documents from a higher point of view, namely the collection level. The experiments we conducted on handwritten word transcription show promising results in terms of recognition error and human user work reductions.**

***Keywords*-document analysis; document sets; handwritten word recognition; historical documents;**

I. Introduction

In the context of historical document recognition, transcription of handwritten words is still challenging. Due to the degradation or the specificity of the handwritten contents, state-of-the-art automatic recognition is not yet able to fully transcribe this kind of documents. Therefore, human help is necessary to assist document analysis systems; by correcting the detected recognition ambiguities and difficulties.

However, when processing a huge number of document pages, the user annotation work may become tedious.

Even when the system includes a re-training phase to improve over time the recognition, the ground truth for learning requires the human user intervention.

Facing the same limitations of imperfect automatic recognition and of not overloading the human user with annotation, we looked for new and complementary information sources to overcome those restrictions.

As [1] pointed out, the semantic dependencies between document pages could be exploited to improve document recognition. To benefit from this inter-page link, a higher point of view is adopted: the collection level.

From the collection level, redundancies across pages can be used. For example, some identical words written by the same person contained in two different pages can be grouped together to enforce their individual recognition hypothesis or to be annotated at the same time by a human user. Also, at this level, page contents continuity carries information. For example, the knowledge that handwritten numbers spread over pages form an increasing sequence can be used to optimally recognize them.

For the problem of historical handwritten words transcription, there exists a strong redundancy and homogeneity between document pages. We believe that grouping words at collection level will improve overall document recognition.

To evaluate this idea, we developed a document recognition architecture. In the literature, different kind of architecture are proposed. The DocMining [2] system is similar to a workflow of processing tasks. For each document type, a processing scenario is defined, made of tasks such as binarization, connected components extraction, user interaction GUI, etc. Information attached to each document is updated after the completion of each task. While this system is highly configurable, it is document centered and does not provide a common place to manipulate collections of documents. Another system, smartFIX [3], proposes an industrial framework to analyze printed medical bills. It integrates an *improving module* which checks consistency and optimizes interpretations in multi-page documents. This system was designed to process business documents, which are mostly independent. Then, its architecture cannot be configured enough to integrate collection level knowledge.

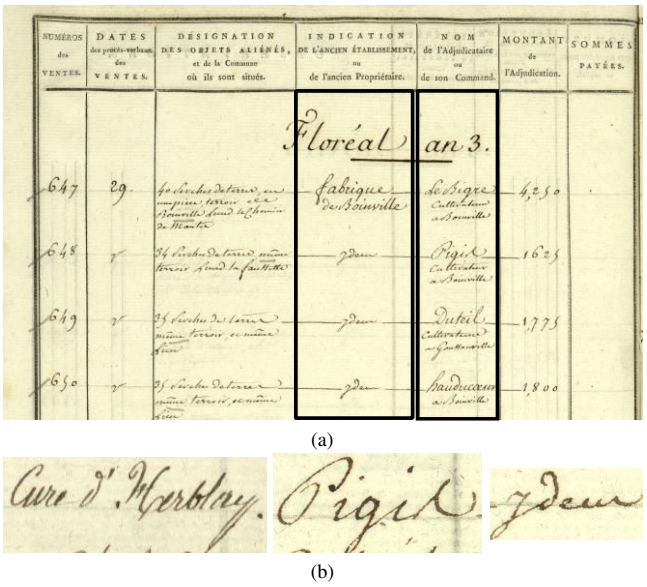
Our architecture keeps the concept of tasks cooperating through a workflow, and has the following specificities to exploit collection knowledge: it enables tasks to cooperate at collection level through its *strategy module*; and it manipulates document information at collection level, thanks to a *central storage database component*.

Furthermore, our architecture must not break the strong bond between contents extracted at document level and at collection level when grouping document data at a higher level. Therefore, our architecture employs an *iterative analysis* to satisfy this constraint.

To summarize, this article has two main contributions:

- it presents an iterative multi-level architecture to transcript handwritten words;
- it demonstrates that processing document at collection level leads to better result in terms of user intervention and recognition performances.

1520-5363/11 $26.00 © 2011 IEEE
DOI 10.1109/ICDAR.2011.179



(a)

(b)

Figure 1. (a) Sample of the *18th century French revolutionary sales* documents to process. Highlighted columns are to be extracted and recognized. (b) Extracted words: *"CURE D'HERBLAY"*, *"PIGIS"*, and *"IDEM"*.

This work is organized in the following way. Sections II and III address in detail our architecture. Experimental results and conclusion are presented in sections III and IV.

II. Exploiting Collection Level

The next two sections detail the strategy module and database component needed to work at collection level. Section II-C explains our iterative analysis which ties together the different levels. Figure 2 synthesizes the different parts of our architecture through an example strategy.

A. Cooperation at Collection Level

Considering the documents from the collection point of view offers new ways of handling their processing. In our case, we are interested in processing the *18th century French revolutionary sales* documents, shown in figure 1a, and more precisely in extracting and recognizing the handwritten words in the highlighted columns. Those documents inventory the goods that were sold during the revolutionary sales around year 1791. They are arranged in tables, each row corresponding to a sale. The first highlighted column contains the former owner name and the second contains the new owner name. Some extracted sample words are presented in figure 1b.

Those documents cannot be recognized automatically because of singular handwriting, time degradation, noise and word overlapping. Thus, human users usually have to label by hand some of the words. A simple two stages approach can be used. The first document of the collection is given to a document page analyzer which extracts and tries to recognize the handwritten words. Rejected words are annotated by a human user. Then, the second document is processed and so on.

More evolved strategies could be envisaged to address our document recognition problem. For example, in order to increase the amount of words recognized automatically, we could regroup in clusters all the words that graphically "look the same" amongst the documents and then use individual word recognition hypotheses to label the cluster with more confidence. Exploiting word redundancy at collection level by combining word spotting, word clustering and handwritten word recognition could improve automatic recognition.

Nevertheless, not matter how these processes are assembled, the human user intervention is going to be required. Some of the words would be rejected and they would need to be labeled by hand. But, again, decreasing the amount of work asked to the user could benefit from the collection abstraction level. Instead of presenting to the user one word at a time, he could label clusters: the annotation of all words in a cluster would be done with a single user action.

Those examples suggest two remarks and their corresponding consequences for our document processing architecture. First, obviously, many kind of strategies are possible. Consequently, the presented architecture allows specification of user own strategies independently from processing tasks.

Second, the strategy schedules the processing tasks. It is responsible for creating, manipulating and executing the tasks. By considering documents at collection level, cooperation of tasks like document analysis, handwritten word clustering and user interaction, is enhanced.

Intra-page regularities can easily be used as information *over all* documents can be gathered at collection level.

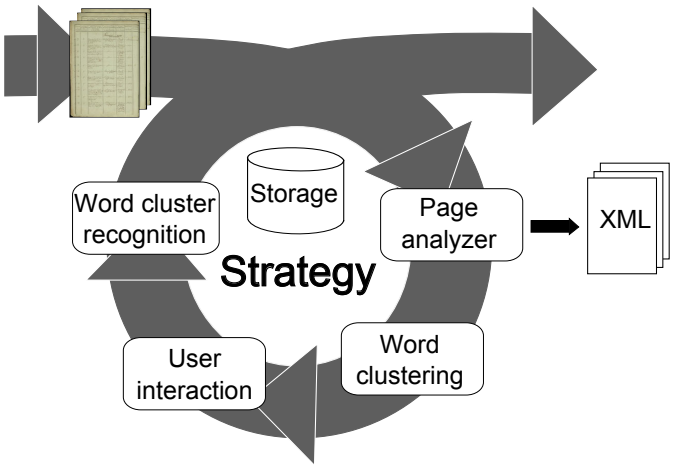


Figure 2. Example strategy to process historical documents based on our iterative architecture

B. Manipulating Document Information

To efficiently group redundant handwritten words, the clustering task must be fed with word images extracted

from several pages. Thus, the strategy is able to collect, manipulate and choose the data transiting between tasks so that they work at different levels. The collected data are stored in a central storage database.

Since we are interested in handwritten word recognition, we introduce the main data type *field* transiting between tasks and stored in the database:

- A contents type
- A 2D bounding box
- A list of top n recognition hypotheses h_i defined by a word w_i and a confidence score s_i
- A final word

In the above definition, a *word* is a transcription of a handwritten word found in a document and that belongs to a known lexicon. The ultimate value of a field is stored in its *final word* attribute.

Our architecture permits to overcome the difficulty of grouping data extracted from documents. When the strategy, for example, queries the database for all fields to feed the clustering task, document physical segmentation is abstracted to consider things at collection level. It implicitly switches from one level to another.

At this point, we have considered only the bottom-up part of our multi level architecture going from document level to collection level. Fields exchanged between tasks are simple objects, they do not embed any higher knowledge about, for example, their semantic connection to another field. It means that some constraints on a field, known only inside the page analyzer task, may not be respected by others tasks that could modify the field. To fulfill the intra-document constraints, an iterative scheme is used.

C. Iterative Analysis

As an example of intra-page bond between fields, we can consider the documents illustrated in figure 1a, where the word "IDEM", in the former owner column, means that the name in the current row is the same as the one in the previous row. This knowledge is kept, like any intra-page constraint, in a single page model embedded in the page analyzer to: i) simplify its development; and ii) ease the validation of model constraints in a single component.

When some fields are extracted by the page analyzer but not recognized because of ambiguities, they are stored in the database and another task will take those fields and affects values to their *final word* attributes that may not take into account their link. Then, the strategy calls again the page analyzer, with, as input, the page and the fields with their actual *final word* values. The attribute values are kept if no constraint is broken, otherwise they are swept and the whole process is repeated until the document is completely recognized, as symbolized in figure 2.

The top-down part of the multi level architecture consists in re-injecting fields into the page analyzer such that the constraints between them is verified, thanks to the iterative analysis method we proposed in [4]. It is not detailed in this paper as we focus on the global architecture enabling the use of collection context. The bottom-up and top-down parts form an iterative analysis in the multi level architecture, which somehow conciliates the apparent contradiction between centralizing document knowledge in one task and manipulating extracted document fields at the collection level.

III. Implementation of our Multi Level Architecture

A. Storing Information

Data to be stored is heterogeneous. Some elements are related to the collection level: the list of field clusters. Others are of document level: the fields. To mix them up easily in our database, we chose HBase part of the Hadoop framework[1]. The main advantage of this database is to ease the selection of fields according to the page they belong to, or to their content type: number, family name, city name...

B. Defining the Strategy

As the Hadoop framework would require some extensions to enable an efficient interaction, we currently use a prototype designed in Python which enables a quick chaining of the various *task* tools we use. An implemented strategy will automatically run the task processes with appropriate data, and gather their results when they are done, using the database previously presented to store them.

C. Tasks

The tasks are implemented as C/C++ shared libraries, python scripts, binary executables or remote GUI clients. They all comply with the same Python interface. We now detail the tasks used in strategies we evaluate in section IV.

1) Page Analyzer: The document recognition is performed using DMOS-P [5], a concept-driven grammatical method for structural analysis of pages, which uses page descriptions to analyze and extract contents. In order to be able to reintegrate manually annotated elements in the document structure and validate them, we use a recent extension of DMOS-P [4] which enables an *iterative analysis* of document pages. Therefore, according to a page model we defined, a page analyzer processes each page as follows.

- The analyzer is provisioned with all fields related to the current page stored in the database.
- It locates the textual fields to be transcribed.
- For each field, if a transcription is already available in external data, it is used to fill the *final word* attribute. Otherwise, the field is submitted to the handwritten word recognition system detailed in [6]. The confidence score associated to the returned word hypothesis is compared to a rejection threshold. If above, the

[1]Documentation at http://hadoop.apache.org

transcription is validated, otherwise it is rejected and marked for external correction.

- All the fields are sent back to the strategy module.

The incomplete fields may be filled elsewhere in the strategy.

2) Field Clustering: This task aims at regrouping fields containing the same word. It works on samples which are, in the present case, 2D graphic images corresponding to the field *2D bounding box* extracted from original document image.

First, the samples, transformed to a set of features, are pairwise compared using dynamic time warping to get matching scores that are stored in a cost matrix *M*. As the samples are images containing handwritten text, the extracted features are the one detailed in [7].

Then, a hierarchical agglomerative clustering algorithm is initialized by treating each sample as a cluster. Afterward, the clusters are pairwise merged up till the distance between them exceed a given threshold T_c. The distance $d_{i,j}$ between clusters C_i and C_j is computed as:

$$d_{i,j} = \max_{e_k \in C_i, e_l \in C_j} M(e_k, e_l)$$

3) Cluster Recognition: Considering that, at this stage, the clusters newly created are homogeneous, this task goal is to assign to each cluster a word label based on individual field recognition hypotheses.

Let C be a cluster composed of N samples e_i and to each sample e_i is associated a list of the top n_i recognition hypothesis $h_{i,j}$, $j \in [1, n_i]$ (coming from field recognition). We define the auxiliary functions: $L : h \mapsto w$ and $S : h \mapsto s$ where w and s are respectively the word and score associated to h. Furthermore, we introduce H_k as:

$$H_k = \{h | L(h) = w, w \in W, w \notin H_i, \forall i < k\}$$

where $W = \{w | w = L(h_{i,j}), i \in [1, N], j \in [1, n_i]\}$.

Therefore, a cluster recognition hypothesis h_k is defined as a combination of the word $w_k = L(h)$ with $h \in H_k$ and the following score s_k:

$$s_k = \frac{1}{Z} \sum_{\forall h \in H_k} S(h) \quad \text{where} \quad Z = \sum_{i=1}^{N} n_i$$

Then, the cluster recognition hypotheses are re-ordered according to their score s_k, defining a new list: $\langle \hat{h}_1, \hat{h}_2, ... \rangle$. A thresholding action is performed by assigning $w_1 = L(\hat{h}_1)$ to cluster C, according to:

if $S(\hat{h}_1) - S(\hat{h}_2) \geq T_r$ **then** accept w_1 **else** reject w_1

where T_r is a rejection threshold. The word label assigned to the cluster is spread to the fields constituting the cluster.

It is worth mentioning that the thresholding action relies on the ability of the handwritten word recognizer to efficiently reject ambiguous samples. The recognizer we used was specially developed for this purpose [6].

4) User Interaction: Human user cooperates to field recognition by annotating the clusters. He successively reviews homogeneous clusters which were not automatically recognized by the cluster recognition task. He has a view of one sample of the cluster and he must type the handwritten word he sees. The word label thereby assigned to the cluster is spread to all the fields constituting the cluster.

IV. Experiments on Assisted Handwritten Word Transcription

The experiments we conducted aim at showing that document processing at collection level can improve both automatic recognition and user annotation work. We present 2 different strategies, one exploiting the collection level. The iterative analysis is not evaluated here.

A. Test Documents

For our experiments, 70 document pages looking alike the one in figure 1a where used. It forms a set S of 1206 extracted handwritten fields that need recognition. Localization and extraction were not evaluated in the experiments. Amongst those 1206 fields, they are 502 different words.

B. Tested Strategies

We compared 2 different strategies made of the tasks detailed in III-C.

Baseline: document pages are processed with the page analyzer, rejected fields are annotated by a human user depending on threshold T_r. There is no collection level clustering and no iterative analysis.

Clustering: document pages are processed with the page analyzer. Clustering task, controlled by T_c, regroups fields. Then, the clusters are either automatically recognized or annotated by a human user depending on T_r.

The number of recognition hypotheses was set to 10. Both thresholds T_c and T_r were tuned using a grid search on a validation set.

C. Experimental setup

In those experiments, we are focusing on the amount of work for human user. We are evaluating user interaction as described in III-C4.

For the **Baseline** (resp. **Clustering**) strategy, a user interaction is to label a field (resp. cluster) by hand.

We aim at minimizing the number of user interactions, in our case, this is equivalent to minimizing the number N_M of manual annotations.

This absolute count of manual annotations has to be compared to the worst case where all the fields are labeled by hand, equal to $|S|$.

Thus, we adopt the following definitions for *Manual annotation Rate* (MR), *Error Rate* (ER) and *Automatic annotation Rate* (AR):

$$MR = \frac{N_M}{|S|} \quad ER = \frac{N_e}{|S|} \quad AR = 1 - \frac{N_M + N_e}{|S|}$$

Table I
RESULTS FOR TWO STRATEGIES. AR = AUTOMATIC ANNOTATION RATE (%), MR = MANUAL ANNOTATION RATE (%), ER = ERROR RATE (%)

Strategy	W/o reject			With reject					
	AR	MR	ER	AR	MR	ER	AR	MR	ER
Baseline	64	0	36	59	21	20	24	75	1
Clustering	67	0	33	65	15		38	61	

where N_e is the number of incorrectly annotated field.

For the **Baseline** strategy, an incorrectly annotated field is due to an error of the handwritten word recognizer. For the **Clustering** strategy, an incorrectly annotated field is either caused by a recognizer error or by a clustering mistake. As an example, suppose the clustering has incorrectly regrouped 5 fields in a cluster. 4 have the same label l and 1 has another. If label l is *automatically* assigned to the cluster, cluster MR is 0, ER is 0.2 and AR is 0.8. If label l is *manually* assigned to the cluster, cluster MR is 0.2, ER is 0.2 and AR is 0.6.

D. Results

Table I presents experimental results for two strategies.

When reject is disabled, there is no manual annotation. A field is either well (goes into AR) or incorrectly recognized (goes into ER). It has to be noted that the recognition of handwritten words in our historical documents is hard as the **Baseline** strategy only gets 64% of automatic annotation. In addition, it is worth mentioning that the top 10 AR is of 71% (+ 7%).

The AR increases from 64% to 67% between the **Baseline** and **Clustering** strategies. This moderate improvement (3%) should be to compared to the top 10 AR of 71%. Indeed, it means that **Clustering** strategy is able to "recall" 3% of those potential 7%, i.e. more than 40% of them.

Reject is used for controlling the error rate. When working in document retrieval domain, the need is to transcribe as many fields as possible with a reasonable amount of manual annotation and error. In this case, the **Clustering** strategy decreases relatively the amount of manual annotation by 28%, from 21% to 15%, with an error rate of 20%. The obtained annotation rate (AR + MR) is then 80%.

An error rate of 1% is appropriate to get automatically some reliable ground truth for recognizer retraining purpose. For such error rate, the **Clustering** strategy increases relatively the automatic annotation rate by 58% compared to the **Baseline** strategy (from 24% to 38%). More generally, the **Clustering** strategy most clearly outperforms the **Baseline** strategy for low error rate.

V. CONCLUSION

This paper introduces a document analysis architecture that allows to process documents at collection level. From this higher point of view, redundancies and homogeneities between pages can be efficiently exploited so as to improve results quality and lower human workload. The architecture is based on three elements: a strategy module; a central database; and an iterative analysis. The experiments conducted on historical documents show that clustering handwritten words according to their shape leads to an improvement in performances, for two different tasks. For document retrieval which requires the indexing as many elements as possible, a reasonable error rate is conceivable, and the use of collection context permits a relative diminution of 28% of human workload for an overall annotation rate of 80%. For the adaptation of the system through retraining, a very low error rate is necessary, and our approach enables a relative increase of 58% of automatic annotation for an overall annotation rate of 99%.

Our current perspectives are to: i) investigate the impact of the aggregation function which fusion label hypothesis in a cluster, as it could also help recognizing suspicious elements instead of suppressing them; and ii) to quantify the adaptation capability of the system over several passes, after bootstrapping using examples produced automatically.

ACKNOWLEDGMENT

This work has been done in cooperation with the *Archives départementales des Yvelines* in France, with the support of the *Conseil Général des Yvelines*.

REFERENCES

[1] E. Saund, "Scientific challenges underlying production document processing," in *Document Recognition and Retrieval XVIII*, ser. Proceedings of SPIE, vol. 7874-1, 2011.

[2] E. Clavier, G. Masini, M. Delalandre, M. Rigamonti, K. Tombre, and J. Gardes, "DocMining: A cooperative platform for heterogeneous document interpretation according to user-defined scenarios," in *Graphics Recognition*, Lladós and Kwon, Eds. Springer, 2004, vol. 3088 of LNCS.

[3] B. Klein, A. Dengel, and A. Fordan, "smartFIX: An adaptive system for document analysis and understanding," in *Reading and Learning*, Dengel, Junker, and Weisbecker, Eds. Springer, 2004, vol. 2956 of LNCS.

[4] J. Chazalon, B. Coüasnon, and A. Lemaitre, "Iterative Analysis of Pages in Document Collections for Efficient User Interaction," in *International Conference on Document Analysis and Recognition*, 2011.

[5] A. Lemaitre, J. Camillerapp, and B. Coüasnon, "Multiresolution cooperation makes easier document structure recognition," *International Journal on Document Analysis and Recognition*, vol. 11, pp. 97–109, 2008.

[6] L. Guichard, A. Toselli, and B. Coüasnon, "A novel verification system for handwritten words recognition," in *International Conference on Pattern Recognition*, 2010.

[7] T. M. Rath and R. Manmatha, "Features for word spotting in historical manuscripts," in *International Conference on Document Analysis and Recognition*, vol. 1. Los Alamitos, CA, USA: IEEE Computer Society, 2003, p. 218.