Alpha-numerical sequences extraction in handwritten documents

Simon Thomas, Clément Chatelain, Laurent Heutte, Thierry Paquet
LITIS EA 4108
Université de Rouen
76801 Saint-Etienne du Rouvray, FRANCE
{*FirstName.LastName*}*@univ-rouen.fr*

Abstract

In this paper, we introduce an alpha-numerical sequences extraction system (keywords, numerical fields or alpha-numerical sequences) in unconstrained handwritten documents. Contrary to most of the approaches presented in the literature, our system relies on a global handwriting line model describing two kinds of information : i) the relevant information and ii) the irrelevant information represented by a shallow parsing model. The shallow parsing of isolated text lines allows quick information extraction in any document while rejecting at the same time irrelevant information. Results on a public french incoming mails database show the efficiency of the approach.

1. Introduction

Despite recent success of the electronic communications (emails, *SMS*, *MMS* etc.), the amount of documents exchanged between compagnies and administrations is still growing. Many specific handwriting recognition applications are already industrialized but concern very constrained documents in their layout and their syntax. We can mention the automatic reading of french checks [8], postal addresses on envelopes [6] or forms [13]. Apart from these well-defined applications, handwriting recognition is still a challenging problem when no a priori knowledge about documents to be processed is available. Full recognition methods in weakly constrained documents as free texts processing have been developed [18, 10] but, because of the complexity of the task (free layout, open vocabulary, very large language model, inter-writer variability etc.), such systems aren't reliable enough to give reasonable performances.

Instead of trying to recognize the whole content of an unconstrained handwritten document, a trend seems to be the information extraction in these documents [15, 2, 5, 16, 11, 18]. Contrary to complete and automatic reading of handwritten documents, information extraction aims at seeking only a specific kind of relevant information. It can be either predefined by an user (image or text queries) or constrained by an application (specific lexicon). In the particular field of handwritten mails processing, information extraction can be a powerful strategie to categorize these mails object [17] or identify a writer thanks to specific numerical fields [3]. The RIMES public database on which the 2009 ICDAR handwriting recognition competitions took place [7] illustrates this new trend (see figure 1 for 2 examples).

Considering information extraction, differents approaches may be used. Main literature approaches concern the keyword spotting [15, 2, 5, 16] and information extraction based on full recognition [11, 18]. On the one hand, the keyword spotting approach consists of isolating relevant information corresponding to the system inputs. These inputs consists either in word images [15, 2] or in textual queries using a lexicon [5, 16]. The document is first segmented into lines and generally in words. In the case of an input image, an edition distance between the input query and every segmented word image in the mail is computed. Then, a predefined function helps to decide whether an image corresponds to the input or not [15, 2]. The main drawbacks of this kind of approach is that they need a lot of samples for different queries. Moreover, they don't support writing variabilities and are thus often mono-writer. In the case of textual queries input, conventional recognition methods are used on word images segmented in the whole document. The acceptation or rejection is once again made thanks to a decision rule based on normalized recognition scores [5, 16]. Generally, the main drawbacks of keyword spotting systems is the word segmentation stage because segmentation errors are often irreversible and useful information surrondings the word is lost. Also, the decision rule to accept or rejet recognition

hypothesis may be difficult to design as well as data-dependent. On the other hand, full recognition based on information extraction systems aim at fully recognize a mail and then filter relevant information modeled by a keyword lexicon. In order to achieve this goal efficiently, a lot of a priori knowledge during recognition is needing(large vocabulary and large language model) and thus are greedy in computational time.

A good alternative to these approaches seems to take a whole segmentation/recognition decision at the line level in order to cancel the word segmentation problem. In this configuration, a large part of the focus has to concern the information modeling within a text line.

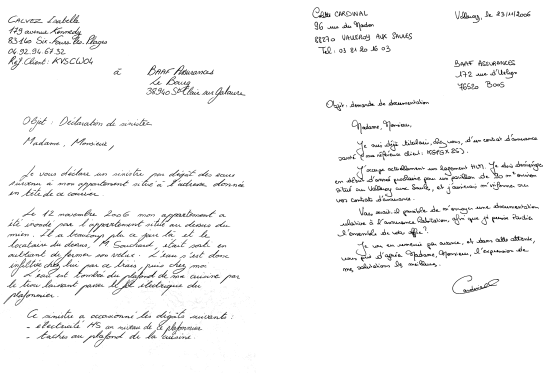

Figure 1. Incoming mails from RIMES database [7].

In this paper, an information extraction system is introduced, based on a text line model able to handle relevant (words of a lexicon) and irrelevant (everything else) information. This paper is organized as follows. We introduce our text line model in section 2. In section 3, implementation issues are described. We present our experiments on a french incoming mail database [7] in section 4. Conclusion and future works are given in section 5.

2. Our generic text line model

Designing an information extraction system requires two kinds of content have to be modeled: the relevant information (keywords, numerical sequences and alpha-numerical sequences) and the irrelevant information (everything else: Out-Of-Vocabulary (OOV) sequences, noise for example). In order to provide a good probabilistic handwriting model, all the *a priori* knowledge have to be gathered in the model. This knowledge consists of character models, language model, proportion of relevant information etc. This section contains the main details of our global text line model for an application of sequence extraction.

In this context, the handwritten modeling is text line oriented and HMMs (*Hidden Markov Models*) have been chosen to model each characters. Indeed, HMMs are well known to be one of the most interesting probabilistic tools in sequence modeling [14]. They have been widely used for the recognition of handwritten words [9] or sentences [18]. Given the sequence extraction problem, two types of information are to be considered within a line of text:

- Relevant information made of any sequences belonging to a lexicon specific to a given application. They are modeled by the concatenation of their HMM characters.
- Irrelevant information made of OOV words, other numerical information, punctuation, spaces and noise, all represented by a shallow language model whose distributions are learned on a training set.

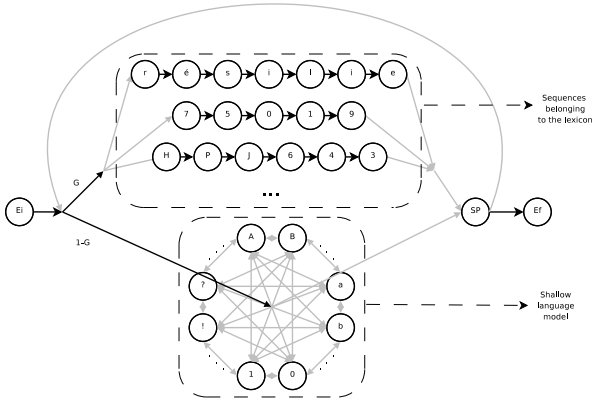

Figure 2. Global line model

Figure 2 represents the handwriting line model. It allows to simulate a competition between the two types of information modeled within a line: on one hand keywords represented in the upper part of the scheme figure 2, on the other hand our shallow parsing model regrouping uppercase and lowercase letters, digits, punctuation and spaces represented in the lower part of the scheme figure 2). In order to switch between relevant and irrelevant information, we have introduced the hyperparameter G. It represents the proportion of relevant information that could appear in a line and is therefore dependent on the lexicon size. As our text line is probabilistic, G is a probability and therefore $\in [0, 1]$. Finally, a line may be seen as a succession of keywords and irrelevant information surrounded by spaces. Before describing the global system behaviour during the

recognition phase, we first present details about how the model parameters are learnt.

2.1 Learning the model parameters

Taking a look at the global line model represented by figure 2, two kinds of information must be learned: the character models (black circles in figure 2) and transitions between them (black and grey arrows in figure 2). This is done on the part of the database devoted to learning in this way:

i) The elementary entities of a line are the character models and the space model. In order to learn all of them in an efficient way, the well known Baum-Welch algorithm is used [14] on annotated lines of text including spaces. The main advantage of this embedded learning is that every instance of a given character is considered to learn its HMM model. The learning is done on manually annotated lines at the word level, the HMM space model is also learned in an embedded way. This training method allows to learn at the same time our character models and the way they are connected to each other.

ii) The second kind of parameters to be learned is the transitions between characters in the generic line model. These transitions are equal to 1 between characters in a word belonging to the lexicon (black arrows in the upper part of the scheme figure 2) and equal to the probabilities of the language model representing the shallow parsing model (grey arrows in the scheme figure 2) learned on the training database.

The recognition constrained by this model is described in the next section.

3. The information extraction system

In this section we describe the main stages of the recognition process i.e. preprocessing, feature extraction and HMM decoding constrained by our model.

Preprocessing steps

During these steps, classical state of the art preprocessing methods are used. They are carried out in a sequential way : first, document images are binarized. Lines are segmented thanks to a convex hulls grouping method (refer [4] for more details). Secondly, the variability between writers is reduced by correcting the skew (deviation of a text line from the horizontal direction) and the slant (vertical deviation of the writing) of text lines. The skew angle is basically corrected by rotation while the slant angle is corrected by a shear transformation. In a third step, writing baselines useful for the feature extraction stage are detected. For further details on these steps, please refer to [4].

Feature extraction step

HMMs used during the recognition step take a sequence of feature vectors as input for each unknown preprocessed line to be recognized. To extract such a sequence of feature vectors, a sliding window is used. A window of d pixels width with o pixels overlap between two consecutive windows is moved from left to right over the current line. For every position of the window, a feature vector based on black pixels densities and concavity features inspired by [1] is computed. Each vector contains $20 + d$ features :

- $3 + d$ are baselines independent (for instance : black pixels density in the window and in every column of this window, position of the gravity center of writing pixels in the window)
- 17 are baselines dependent (for instance : black pixel density upon and under baselines, local pixels configurations regarding baselines,)

It has to be noticed that this feature vector had been used efficiently in the 2009 ICDAR word recognition competition in [9]. Furthermore, it has been designed especially for its use with HMMs.

Recognition stage

During the decoding stage, text lines represented by a sequence of feature vectors O, are decoded using the global line model. The recognition problem can be solved thanks to equation 1 where L_{opt} representing the best sequence of words for a given observation (feature vectors) sequence O:

$$L_{opt} = \arg\max_{L}\{P(L|O)\} \quad (1)$$

$$= \arg\max_{L}\left\{\frac{P(O|L)P(L)}{P(O)}\right\} \quad (2)$$

$$= \arg\max_{L}\{P(O|L)P(L)\} \quad (3)$$

where $P(O|L)$ the probability of a sequence of observations O given the sequence of words L. Even if $P(O)$ is difficult to compute, it is constant and thus does not affect the search of the best state sequence and is deleted from our computations. In equation 3, $P(L)$ is the prior probability of a word sequence. It allows to reject uncommon word sequences. These probabilities are part of the global line model and are learned on the training dataset and integrated in our model.

Given a text line L containing N sequences K_n belonging to the lexicon, M OOV sequences W_m and P

spaces S_p, the term $P(O|L)$ is estimated using the *Time Synchronous Beam Search* algorithm introduced in [12] as follows :

$$P(O|L) = \prod_{n}^{N} P(O_n|K_n) \times \prod_{j}^{M} P(O_m|W_m)$$

$$\times \prod_{p}^{P} P(O_p|S_p) \quad (4)$$

$$P(O|L_{opt}) = \max_{n,m,p} \{P(O_n^*|K_n) \times P(O_m^*|W_m)$$

$$\times P(O_p^*|S_p)\} \quad (5)$$

Now that our model has been defined and the recognition step explained, we present our results on a french document database [7].

4. Experiments and results

In this section, the RIMES database used for experiments is presented as well as the experimental protocol and the results obtained on this database with this protocol.

Database

The RIMES database includes 1150 french incoming mails from different writers [7]. 950 of them containing approximatively 36000 words are used for learning both the character models and the global line model transitions. 200 documents can be used as the test database.

Experimental protocol

In order to evaluate information extraction in a database of D documents, recall and precision measures must be computed. Varying the hyperparameter G allows to obtain different operating points of the system and thus enables to better describe a Recall/Precision curve: a value of G close to 1 gives an advantage to the relevant information at the expense of the shallow parsing model and thus will favor recall against precision. And vice versa, values of G close to 0 will result in improving precision of the system. In case of deployment of such a system, the value of G can be chosen depending on application needs.

Given a document d, let $N_{ok}(d)$ be the number of well detected sequences in this document, $N_{fa}(d)$ the number of false alarms and $N(d)$ the number of sequences to extract, Recall and Precision means (respectively R_{mean} and P_{mean}) for a whole database containing D documents are computed as follows (equations 6 and 7):

$$R_{mean} = \frac{1}{D} \sum_{d} \frac{N_{ok}(d)}{N(d)} \quad (6)$$

$$P_{mean} = \frac{1}{D} \sum_{d} \frac{N_{ok}(d)}{N_{ok}(d) + N_{fa}(d)} \quad (7)$$

Computing $N_{ok}(d)$, $N_{fa}(d)$ and $N(d)$ for every document allow to compute recall and precision variances over the whole database ($R_{variance}$ and $P_{variance}$, respectively). This is done regarding equations 8 and 9:

$$R_{variance} = \frac{1}{D} \sum_{d} \left(R_{mean} - \frac{N_{ok}(d)}{N(d)} \right)^2 \quad (8)$$

$$P_{variance} = \frac{1}{D} \sum_{d} (P_{mean} - \frac{N_{ok}(d)}{N_{ok}(d) + N_{fa}(d)})^2 \quad (9)$$

In order to compute relevant results in terms of recall and precision means and variances, the same amount of relevant information should be searched in every document for a given experiment. To do so, we have decided to pick up randomly 10 sequences in each document vocabularies, constituing one lexicon per document.

Results

Before describing the experimental results, we expose the way hyperparameters have been chosen. The pair (d, o) corresponding to the window size and overlap has been chosen on a word recognition problem. The pair giving the best results in this field is $(8, 5)$. Given this configuration, the optimal number of states for each character model is 4 and the number of gaussians for each state has been fixed to 5, a common value in the field of handwriting recognition with HMMs. As

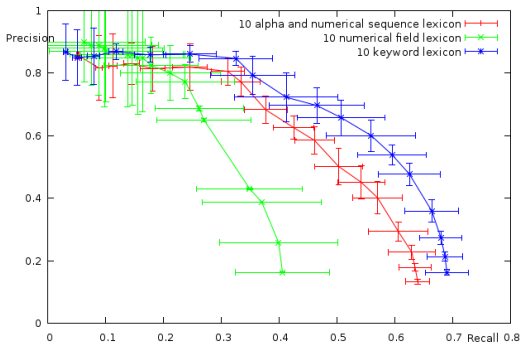

Figure 3. Recall/Precision for different kinds of lexicon

a first experiment, the protocol described in the previous section has been followed. We have tested different kind of lexicon :

- lexicon containing only keywords
- lexicon containing only numerical fields
- lexicon containing a mix of keywords and numerical fields

Results in recall and precision means are represented by the lines in figure 3 as well as their variances represented by the errorbars. At first sight, the system provides best results for keyword extraction (Break Even Point equals 0.6). Results for numerical field extraction give a break even point at 0.4. Our explanation of this result is that our baseline detection algorithm is not well suitable for numerical information as baselines are more difficult to detect on numbers. This strongly influences the feature extraction process, especially the ones depending on the baselines and thus influences the learning of HMM models of numbers. In our case, it is particularly disadvantageous as the number of sample used to train the numbers are relatively low (200 samples per class). As a second experiment, the size

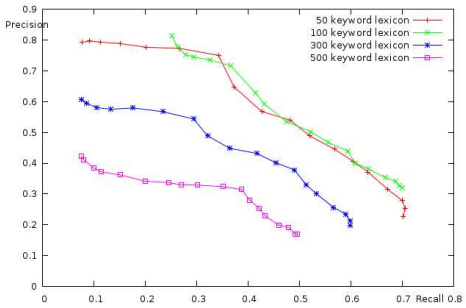

Figure 4. Recall/Precision for keyword lexicons

of the lexicon is increased. The experimental protocol stays almost the same : a L sequence lexicon is generated by picking up randomly 10 sequences in each document vocabularies. These 10 sequences are completed with $L - 10$ sequences that appear in the other documents vocabularies but not in the one of the processed document. Once again, we have decided to test different kind of lexicon made of either keywords or a mix of keywords, numerical fields and alpha-numerical informations. Results are given in figure 4 and 5 respectively in terms of recall and precision means. The variances have been deleted as they are more important when the size of the lexicon is increased. An interesting point is that the system performances decrease but are not falling down. In the case of 50 and 100 word lexicon, performances are almost as good as in the case of a ten word lexicon. Results decrease faster past 200

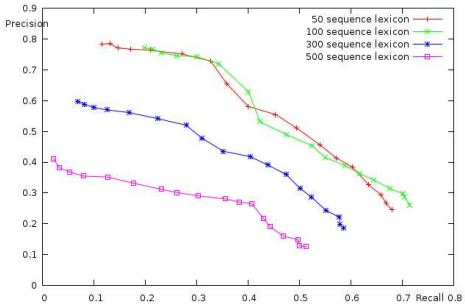

Figure 5. Recall/Precision for alpha-numerical lexicons

words in the lexicon. In the figure 5, results in recall and precision are given for lexicons made of any kind of sequences. Results are sightly less good than in the case of a keyword lexicon basically.

Figures 6 and 7 show the behaviour of our system in two different experiments. In the first experiment (refer to figure 6), 10 words are searched in a handwritten document but only one belongs to its vocabulary. In the

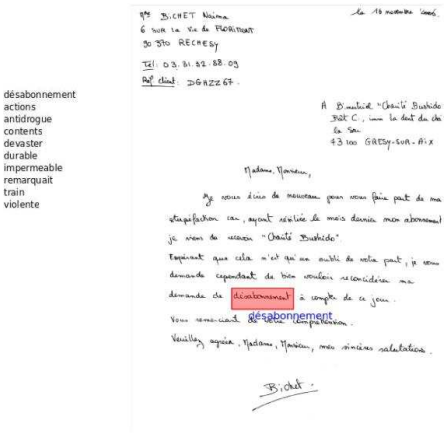

Figure 6. Illustration 1

second experiment (refer to figure 6), 8 words out of 10 belong to the document vocabulary. It illustrates the accuracy of the system as well as a good compromise between recall and the precision. Finally, we give the computational time for different lexicon size. Computational times in second are given for an entire text line or a document. On average, a document belonging to our database is made of 22 lines composed of a mean of 6 words containing an average of 5 characters. *Datas* refers to all the preprocessing steps, feature extraction and model learning. *Decoding* refers to all the HMM decoding process. The larger the lexicon is, the slower

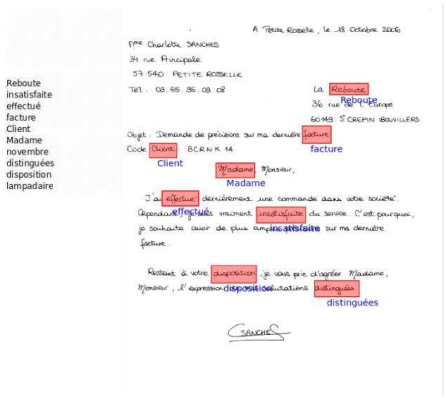

Figure 7. Illustration 2

the system will be. The time to decode a text line is less than a second when the lexicon is smaller than 300 words which is quite fast.

Lexicon size	Datas per doc.	Decoding per doc.	Decoding per line
10 word lexicon	1.31	2.23	0.101
50 word lexicon	1.38	3.46	0.157
100 word lexicon	1.53	10.23	0.46
500 word lexicon	5.15	30.46	1.38

5. Conclusion and future works

An alpha-numerical information extraction system in handwritten uncontrained documents has been introduced. It relies on a global line modelisation based on HMMs allowing a dual representation of the relevant and the irrelevant information. The shallow parsing of text lines allows the fast extraction of any kind of information. Results on the Rimes database illustrate the power of our approach.

As a short term objective, it will be useful to compare our approach to an information extraction system based on full recognition. It would be useful to see how our system compare in recognition and time performances. Then, our works will consist in improving the recognition process and enhance HMMs decoding with a discriminative classifier like neural networks.

References

[1] R. Al-Hajj, C. Mokbel, and L. Likforman-Sulem. Combination of hmm-based classifiers for the recognition of arabic handwritten words. *Proc. ICDAR*, 1:959–963, 2007.

[2] H. Cao and V. Govindaraju. Template-free word spotting in low-quality manuscripts. *Proc. ICDAR*, pages 392–396, February 2007.

[3] C. Chatelain. Extraction de séquences numériques dans les documents manuscrits quelconques. *Thèse de Doctorat, Université de Rouen*, 2006.

[4] C. Chatelain, L. Heutte, and T. Paquet. A syntax-directed method for numerical field extraction using classifier combination. *IWFHR*, pages 93–98, 2004.

[5] C. Choisy. Dynamic handwritten keyword spotting based on the nshp-hmm. *Proc. ICDAR*, 1:242–246, 2007.

[6] M. A. El-Yacoubi, M. Gilloux, and J.-M. Bertille. A statistical approach for phrase location and recognition within a text line: An application to street name recognition. *IEEE Trans. on PAMI*, 24(2):172–188, 2002.

[7] E. Grosicki and H. El-Abed. Icdar 2009 handrwriting recognition competition. *Proc. ICDAR*, 1:1398–1402, 2009.

[8] L. Heutte, P. Barbosa-Pereira, O. Bougeois, J.-V. Moreau, B. Plessis, P. Courtellemont, and Y. Lecourtier. Multi-bank check recognition system: Consideration on the numeral amount recognition module. *IJPRAI*, 11(4):595–618, 1997.

[9] Y. Kessentini, T. Paquet, and A. Benhamadou. Off-line handwritten word recognition using multi-stream hidden markov models. *Pattern Recognition Letters*, 31:60–70, 2010.

[10] U. Marti and H. Bunke. Handwritten sentence recognition. *Proc. ICDAR*, 3:467–470, 2000.

[11] U. Marti and H. Bunke. Text line segmentation and word recognition in a system for general writer independent handwriting recognition. *In Sixth International Conference on Document Analysis and Recognition*, 1:159–163, 2001.

[12] D. Moore. TODE: A Decoder for Continuous Speech Recognition. *IDIAP Research Report 02-09*, 2002.

[13] D. Niyogi, S. Srihari, and V. Govindaraju. Analysis of printed forms. *Handbook on Optical Character Recognition and Document Image Analysis*, pages 330–342, 1996.

[14] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Readings in speech recognition*, pages 267–296, 1990.

[15] T. Rath and R. Manmatha. Features for word spotting in historical manuscripts. *Proc. ICDAR*, pages 218–222, 2003.

[16] J. A. Rodríguez-Serrano, F. Perronnin, and J. Llados. A similarity measure between vector sequences with application to handwritten word image retrieval. *CVPR 09*, August 2009.

[17] A. Vinciarelli. Noisy text categorization. *IEEE Trans. on PAMI*, 27(12):1882–1895, 2005.

[18] A. Vinciarelli, S. Bengio, and H. Bunke. Offline recognition of unconstrained handwritten texts using hmms and statistical langage models. *IEEE Trans. on PAMI*, 26(6):709–720, 2004.